**FP6-004381-MACS**

**MACS**

Multi-sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances

Instrument: Specifically Targeted Research Project (STReP)

Thematic Priority: 2.3.2.4 Cognitive Systems

# D6.4.1 Report on Experiment Design

Due date of deliverable: March 31, 2005
Actual submission date of version 1: May 2, 2005
Actual submission date of version 2: October 10, 2005

Start date of project: September 1, 2004 Duration: 36 months

**Fraunhofer Institute for Autonomous Intelligent Systems (FhG/AIS)**

Revision: Version 2

| **Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)** | | |
|---|---|---|
| **Dissemination Level** | | |
| **PU** | Public | **X** |
| **PP** | Restricted to other programme participants (including the Commission Services) | |
| **RE** | Restricted to a group specified by the consortium (including the Commission Services) | |
| **CO** | Confidential, only for members of the consortium (including the Commission Services) | |

**Contract:** MACS
**Project no.:** FP6-004381
**Document no.:** D6.4.1 Report on experiment design

Project co-funded by the European Commission
Sixth Framework Programme (2002-2006)
Call_ID: FP6-2003-IST-2

# D6.4.1 Report on experiment design

**Key Issues:** mission matrix, mission description, experiments

## Version History

| Version | Date | Author | Reason for modification | Status |
|---|---|---|---|---|
| 001 | 04.02.2005 | R.Breithaupt FHG/AIS | creation under consideration of the previous work of METU-KOVAN and OeFAI | preliminary |
| 002 | 30.03.2005 | R.Breithaupt FHG/AIS | Further revisions | 1st release |
| 003 | 28.09.2005 | R.Breithaupt FHG/AIS | Further revisions + Appendix by E. Rome FHG/AIS | final draft |
| | | | | |
| | | | | |
| | | | | |

## Abstract

In this document, we introduce a systematic method to define the necessary experiments that should lead us step by step towards the final demonstrator scenario as defined in D6.1.1. Therefore, the experiments are designed as complete missions, ordered in different affordance categories, each with levels of increasing complexity. As the complexity level increases, the categories will be combined in joined missions until the complete demonstrator scenario setting is reached. In this manner we present a mission matrix where each mission can be identified by its key affordances and grade of complexity. In order to support a standardization and exchangeability of mission settings we introduce a mission description form where all necessary information about the setting, the used affordances, the expected process sequence, etc. can be listed.

INDEX

# 1 Introduction

To reach the goal of the MACS project and develop a new affordance-based control architecture for mobile robots, a considerable amount of research and development in the areas of perception, representation, learning, behaviour engineering and planning has to be invested.
So, although the purpose of this deliverable is the specification of the experimental design, it is impossible to list all the small but important experiments that have to be done for every architectural component before a first working architecture can be completed.
Hence, we start our experimental specification with the premise that our agent is fully operational and focus on complete missions. Seeing the demonstrator scenario (as described in D6.1.1) as the final goal, those missions should reflect the different aspects of its complex setting and enable us to sequentially increase the level of complexity in small controllable steps. In order to ensure a certain standard and exchangeability we introduce a methodology to order the missions in an affordance category/complexity matrix, described in a standardized form as seen later.
It will be an outcome of our research efforts in this project to describe all the details of all the important missions. Therefore, this is a living document and will be completed as our understanding of the nature of affordance based control progresses.

# 2 Mission matrix

## 2.1 The concept

The demonstrator scenario (as described in D6.1.1) will be the final proof of concept in the MACS project. But due to the fact that it is very difficult to measure the performance of an explicit affordance-based architecture, we have to give this proof on every step of our research towards that goal. In order to guarantee comparable and comprehensible results on every step, we divide the final scenario into its three key affordance categories and define levels of increasing complexity for each category. Thus we have a matrix that enables us to specify missions with certain affordance tasks and certain complexities in order to analyse and develop the performance of our approach in an isolated and well defined way, starting with simple tasks and increase the complexity step by step towards the final scenario.
According to the three basic affordances our agent KURT2 can work with, the affordance categories that at first should be examined separately are navigation, simple object handling (e.g. relocating cans), and complex object handling (e.g. using switches to open doors) (see Tab. 2-1). At each complexity level, several missions can be defined to completely investigate the nature of the given problem with respect to behavioural, perceptual and learning aspects, under controlled conditions. If a sufficient understanding is reached we can proceed to the next level. As we gain the respective level of the final demonstrator scenario, we can start combining the categories in joined missions until all the aspects of the complete demonstrator scenario are complete.

| Level of complexity | Navigation | Handling Simple Objects | Handling Complex Objects |
|---|---|---|---|
| Level 1 | Nav_Mission_1.1,.. | SO_Mission_1.1,.. | CO_Mission_1.1,.. |
| Level 2 | Nav_Mission_2.1,.. | SO_Mission_2.1,.. | CO_Mission_2.1,.. |
| Level 3 | Nav_&_SO_Mission_3.1,… | | |
| Level … | | … | … |
| End Level | Final Demo Scenario (Navigating by using cans to trigger switches) | | |

**Tab. 2-1:** An example of a MACS mission matrix: Divided into the three affordance categories of the demonstrator scenario different missions are defined to reach higher levels of complexity step by step.

Right now we already can specify a set of missions that are essential for our development. As our understanding of the nature of affordance based control grows, the missions will be adapted accordingly and new missions will be added to complete this matrix.
The mission matrix itself as well as all the mission descriptions are meant to be common base of work for the MACS project and must therefore be approved by every partner.

## 2.2 Mission matrix

Every mission that is designed and used in the MACS project should be registered in this mission matrix.

| Level of complexity | Navigation | Simple Objects | Complex Objects |
|---|---|---|---|
| Level 1 | Nav_Mission_1.1 | | |
| Level 2 | | | |
| Level 3 | | | |
| Level … | | … | … |
| End Level | Final Demo Scenario (Navigating by using cans to trigger switches) | | |

This document will be stored to be frequently updated on the MACS-DITO-server:
MACS-ext > Experiments / Results

# 3 Mission descriptions

## 3.1 Mission description form

In the MACS project five research partners have to coordinate their efforts. In order to avoid misunderstandings and parallel developments we have to be very accurate in describing the missions. Whenever a new mission is elaborated it should be categorized, specified in a standardized way and registered in the mission matrix. We suggest the following form to support this objective.

| **Name:** | *name of the mission* | **Category:** | *single or combined* | **Level:** | *1,2,.* |
|---|---|---|---|---|---|
| **Scene Description:** | - *list of differences of the standard environment*<br>- *list of all objects and their properties*<br>- *list of connections between objects (e.g. switches and doors, …)*<br>- *list of changes between mission phases* | | | | |
| **Essential Affordances:** | - *description of affordances of all elements and objects in that scene that are additional to or different from the standard scene.* | | | | |
| **Task Description:** | - *report on the background of that task*<br>- *mission schedule (list of phases etc.)*<br>- *detailed task description* | | | | |
| **Expected Performance:** | - *A founded story board of the expected behaviour and events* | | | | |

**Tab. 3-1:** The mission description form to support exchangeability and a common base of work.

## 3.2 Mission report form

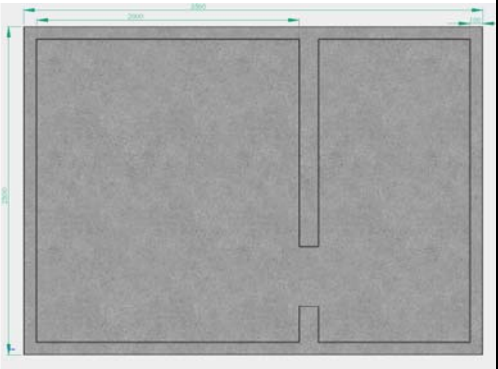
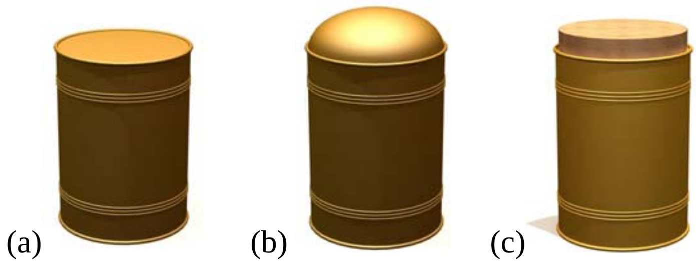
For the same reasons we should report the results of each mission trail and evaluate the observed performance of the agent in an evenly standardized way. The idea is to have many reports from different partners and/or with different software versions for each mission and to learn from comparing and analysing them. The following mission report form should help:

| **Name:** | *name of the mission* | **TrailNo:** | *1,...* | **Date:** | *dd,mm,yy* | **Sign:** | *name* |
|---|---|---|---|---|---|---|---|
| **Software Versions:** | - *if simulated: state the version of simulator software*<br>- *name differences from the standard software (firmware of KURT2 & Arm, development environment, etc…)*<br>- *version numbers of all software components of the architecture* | | | | | | |
| **Hardware configuration** | - *list of differences in KURT2 object & hardware configuration to the standard setting* | | | | | | |
| **Result log:** | - *chronological description of the mission trail*<br>- *qualitative description of the mission trail* | | | | | | |
| **Performance Evaluation:** | - *analyze the performance in this mission*<br>- *name errors and analyze differences from the expected behaviour*<br>- *compare with previous trails or missions*<br>- *formulate a ToDo list for further development if possible* | | | | | | |

**Tab. 3-2:** The standardized mission report form.

## 3.3 Mission setup standards

Another way to support comparable and reproducible results is to define some standards according to the environmental parameters of all missions. It is expectable that in the beginning an affordance based architecture will be very dependent on environmental properties. Therefore it will sometimes be very difficult to see why an experiment failed while others report success in a very similar scenario. In order to reduce misunderstandings, frustrating detail analysis and variable influences to a minimum we should agree to a standard scenario and define details of the mission field, standard objects and maybe also light settings for both simulation and the real world.
If a scenario differs from that common standard we should make it a habit to report that accurately in the "Scene Description"-Field of the mission description form.
Here are the actual suggestions as a base for discussions:

| **Name:** | Standard mission setting | **Category:** | - | **Level:** | -. |
|---|---|---|---|---|---|
| **Scene Description:** | *mission field:*<br>The mission field is a rectangle completely surrounded by walls.<br>The overall size is 3.5 x 2.5 meters with a wall width of 10 cm and wall height of ~30 cm. The floor colour is a light grey – the walls have a similar but darker color.<br>The optional separating wall is a matter of the individual mission settings.<br><br>*standard objects*<br>Standard objects for the first experiments should be as simple as possible and of a certain size that make them easily perceivable for Kurt2's sensors. Hence we use cylinder and brick shaped objects that have simple color and differ in size, weight and in the appearances of their tops to modify their liftability properties (see below: a:flat top [liftable], b:spherical top [hardly liftable], c:isolated flat top [not liftable]).<br><br><br>The objects should have a minimal height of 18 cm and a minimum width of 10cm. The weights should differ between 100-500g for liftable objects. | | | | |
| **Essential Affordances:** | - the walls are not navigable, not movable and not liftable<br>- the ground generally is navigable in all areas<br>- the corners and areas near walls are navigable, but only under special circumstances: certain orientation of KURT2 & its arm | | | | |
| **Task Description:** | - in general there are two separate phases: an exploration phase with the simple task to gain as much information as possible and a second phase with a specific task based on that previous experience<br>- It may serve as a common behavioural bias to use a beacon to indicate goal positions | | | | |
| **Expected Performance:** | - | | | | |

**Tab. 3-3:** Description of the standard mission settings

As the experiments/missions become more complex and additional objects are introduced the scene description should have a complete list of them along with their properties.

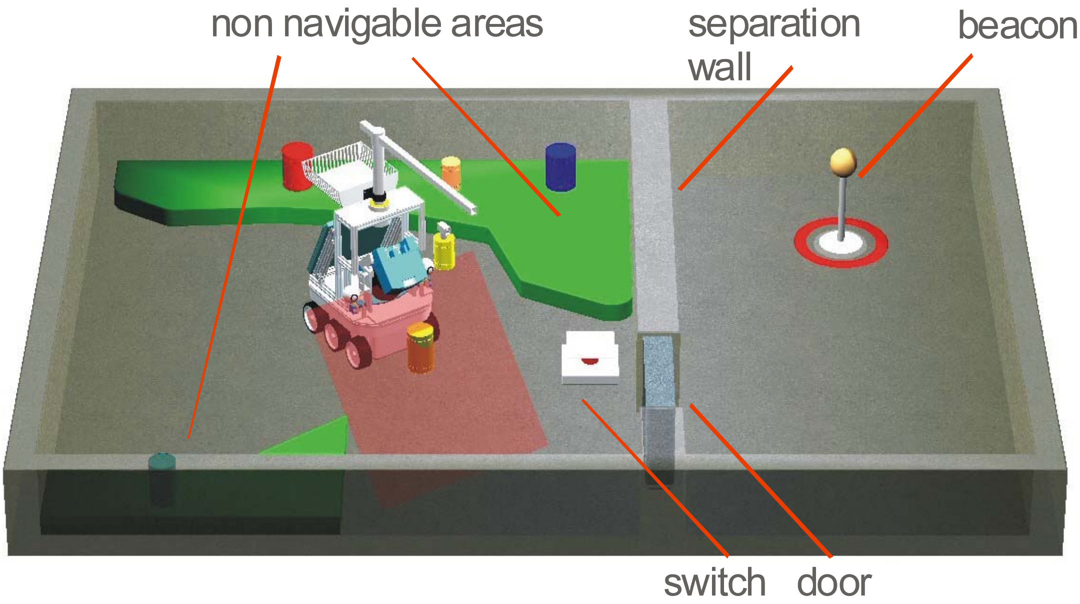


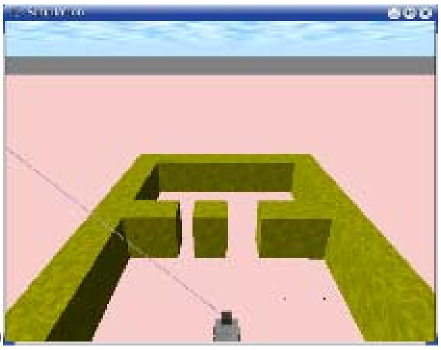
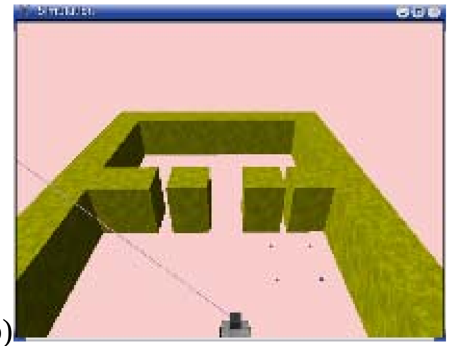

**Fig. 3-1:** Example for a more complex scenario with additional objects.

Fig.3-1 shows the final demonstrator scenario as an example for such a complex scenario. Here the object list in the scene description consists of :

1. a light beacon, height of light bulb: 50 cm, power: 100W
2. non navigable elevated areas, elevation 7cm
3. a separation wall, separating the room in two parts with lengths of 2m and 1.40m
4. a sliding door in the separation wall, light blue color, width 50 cm, the door can be opened manually
5. a switch with a plate size of 20x20cm, activation load: 1kg
6. 2 big cans, height: 20cm, width: 15cm, weight: 300g, 1x red, 1x blue, flat tops
7. 4 small cans, height 18cm, width: 10cm, weight: 100g, 2x yellow, 1x orange, 1x green, flat tops

The exact specifications of more complex and frequently used objects (e.g. the switch) should be published in separate documents.

### 3.4 Example: Category NAVIGATION 1.1

| **Name:** | NAV_Mission 1.1 | **Category:** | Navigation | **Level:** | 1 |
|---|---|---|---|---|---|
| **Scene Description:** | *Phase 1: exploration*<br>The scenario has no objects. The mission field is separated by a wall with two different openings. One opening is wide enough to allow KURT2 to pass through; the other is too small for that (a).<br><br>*Phase 2:*<br>The separating wall has a third opening that is again too small for KURT2 to pass (b).<br><br>(a)  (b)  | | | | |
| **Essential Affordances:** | Navigability of the separate areas of the mission field. Only one opening is wide enough to be passed. | | | | |
| **Task Description:** | *Background:*<br>Initially the robot has only feature detectors that categorize the width of an opening into two categories: wide or narrow. The robot learns that it can pass through all openings that are classified as wide and cannot pass the ones that are classified as narrow. Later, the environment is changed such that some of the openings classified as wide, no longer affords passing through. The robot should be able to develop new categories to resolve these conflicts.<br><br>*Phase 1:*<br>no task – only exploration<br><br>*Phase2:*<br>KURT2 has to find the way to the second room as quick as possible under consideration of the new configuration of the environment. | | | | |
| **Expected Performance:** | *Phase1:*<br>Only exploration behaviour occurs. KURT2 has to examine both openings and has to be at least once in the other room.<br><br>*Phase2:*<br>KURT2 examines the new openings only by distant sensor reading and chooses the only usable opening at once to traverse to the other room. | | | | |

# 4 List of Mission Reports

To provide an overview on the current state of investigated experiments we should register all existing mission reports in the following list.

| Mission Name | Location | Date | TrailNo. |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Those reports will be available separately on the MACS-DITO-server:
MACS-ext > Experiments / Results

# ANNEX Architectural Development Iteration 1: Defining Affordance Use Cases for Perception

… see the attached document.

EU Project **MACS**

Draft report on architectural development procedures

# Architectural Development Iteration 1: Defining Affordance Use Cases for Perception

*Erich Rome, Ralph Breithaupt, Lucas Paletta, Hartmut Surmann, Gerald Fritz, Martin Hülse*



| | |
|---|---|
| Number: | **MACS/3/0.1** |
| WP: | 3 (plus 2, 5 & 6) |
| Status: | draft, version 1 |
| Created at: | August 12, 2005 |
| Revised at: | October 11, 2005 |





| | |
|---|---|
| **FhG/AIS** | Fraunhofer Institut für Autonome Intelligente Systeme, Sankt Augustin, D |
| **JR_DIB** | Joanneum Research Graz, A |
| **LiU-IDA** | Linköpings Universitet, Linköping, S |
| **METU-KOVAN** | Middle East Technical University, Ankara, T |
| **OFAI** | Österreichische Studiengesellschaft für Kybernetik, Vienna, A |

This research was partly funded by the European Commission's 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission's support is gratefully acknowledged.



**Author addresses:**

Dr.-Ing. Erich Rome Fraunhofer Institut für
Autonome Intelligente Systeme
Schloß Birlinghoven
D-53754 Sankt Augustin, Germany

| | |
|---|---|
| Fraunhofer Institut für<br>Autonome Intelligente Systeme<br>Schloss Birlinghoven<br>D-53754 Sankt Augustin<br>Germany | Tel.: +49 (0) 2241 14-2683<br>(Co-ordinator)<br><br>**Contact:**<br>Dr.-Ing. Erich Rome |
| Joanneum Research<br>Institute of Digital Image Processing<br>Computational Perception (CAPE)<br>Steyrergasse 9<br>A-8010 Graz<br>Austria | Tel.: +43 (0) 316 876-1769<br><br>**Contact:**<br>Dr. Lucas Paletta |
| Linköpings Universitet<br>Dept. of Computer and Info. Science<br>Linköping 581 83<br>Sweden | Tel.: +46 13 24 26 28<br><br>**Contact:**<br>Prof. Dr. Patrick Doherty |
| Middle East Technical University<br>Dept. of Computer Engineering<br>Inonu Bulvari<br>TR-06531 Ankara<br>Turkey | Tel.: +90 312 210 5539<br><br>**Contact:**<br>Prof. Dr. Erol Şahin |
| Österreichische Studiengesellschaft<br>für Kybernetik (ÖSGK)<br>Freyung 6<br>A-1010 Vienna<br>Austria | Tel.: +43 1 5336112 0<br><br>**Contact:**<br>Prof. Dr. Georg Dorffner |

# Contents

# 1 Introduction

This technical report is based on discussions and agreements made during the bilateral meeting between Fraunhofer AIS and Joanneum Research in Sankt Augustin, August 10–11, 2005. Here, we describe a method for defining so-called “use cases” of particular affordances at iteration stage 1, the most basic architectural iteration. These use cases are the means to break up the “chicken-and-egg” problem of defining a complete affordance-based control architecture.

To start with, we assume that we have a working robot system with certain *basic skills* at iteration stage 1 which ensure a controlled motion sequence to execute abstract actions (e.g. lift object, push object, etc.). These basic skills are predefined for each of the stage 1 use cases separately. The next step is to describe an affordance that we want to investigate. This includes a number of simple *test objects* and their physical properties as well as assumptions on the environment. For each such affordance, we describe the sensors to be used for perceiving this affordance, and the basic features that would probably best describe the affordance at hand.

The next step is to define basic actions to be performed by the robot for learning a description for the affordance. These basic actions shall be performed by using the basic skills. For testing the results of the learning stage, we define a test mission. Here, the robot has to make use of the learned affordance.

In order to get a real grip on the affordance notion, the test objects, learning scenario and test missions shall be kept as simple as possible in iteration stage 1. Only standard cases are considered. Exceptions will be investigated at later stages.

Later stages of this development process will include:

- Combinations of the affordances of iteration 1
- More test objects
- More affordances
- More features to distinguish affordances
- Exception cases
- Make scenarios increasingly complex, combine scenarios
- Varying environmental settings (static/dynamic, unknown test objects etc.)
- More complex affordances
- Generalization of objects
- Categorization of objects
- Learning by observation
- Non-permanent affordances
- Making basic skills affordance-based

In the main part of this report we will describe use cases for three affordances. The next sections states briefly our assumptions on perception of affordances.

# 2 Perception of Affordances

For iteration stage 1, we use only *bottom-up perception*. This means that initially, there is no object notion. Instead, perceived features are hierarchically aggregated for describing

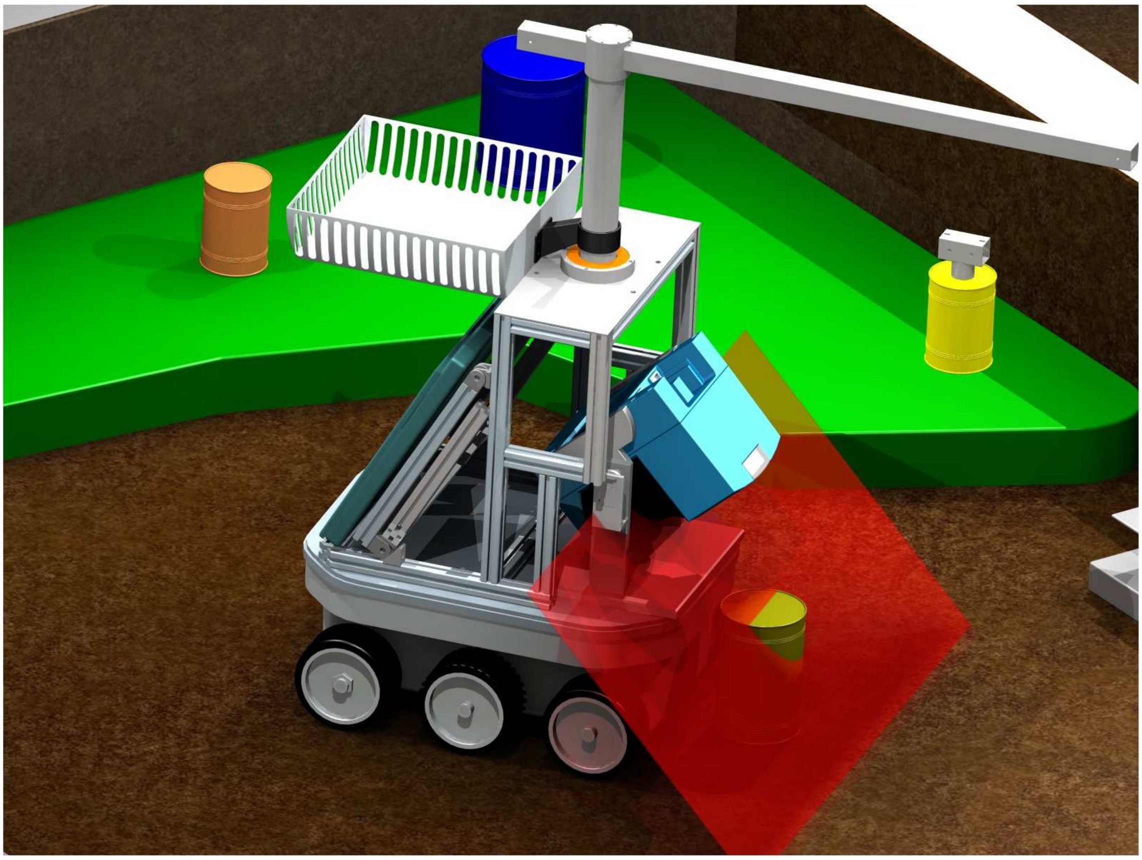


Figure 1: Rendered scene: KURT3D sensing a can.

affordances. Generalization and categorization based on such percepts takes places at later iteration stages.

For later iteration stages, we expect to use *top-down perception*, including

- filtered segmentation
- object/feature categorization
- specialization
- individual objects
- explicit representation of affordances
- change detection / before-after comparison in the environment

The available repertoire of perception methods that could be used as basic perception skills includes:

- Image processing
- Object recognition
- Visual attention methods (bottom-up and top-down)
- Feature sets (SIFT features)
- 3D sensing and recognition

- movement detection/tracking
- load detection (lift & push)

Immediate questions are:

- What do the mappings between perception of affordances and basic skill look like?
- Which feature sets do we need for the perception of affordances?
- Which perceptual methods do we need for acquiring these feature sets?

To find answers to these questions, we will exercise some use cases. For example: The robot shall grasp a can. Which affordances play a role for this task?

# 3 Schema for Specifying Use Cases

This description shall serve as a reference procedure within MACS for investigating affordances. Based on the mission description schema given in D6.4.1 each such description must specify:

- test objects and their physical properties
- environment setup
- basic skills
- sensors and features to be used for perceiving the affordance
- affordance learning procedure and learning missions
- missions for applying learned affordance
- basic assumptions (explicit and implicit ones)

# 4 Iteration 1 Affordances and Use Cases

Affordances in iteration 1:

1. Liftability: A test object affords to be lifted by the robot.
2. Pushability: A test object affords to be pushed by the robot.
3. Traversability: A part of the environment affords to be traversed by the robot.

Use cases for affordances in iteration 1:

1. Lift test object: A test object is lifted by the robot.
2. Push test object: A test object is pushed by the robot.
3. Drive: A part of the environment is traversed by the robot.

## 4.1 Affordance "liftability"

### 4.1.1 Experimentation scenario

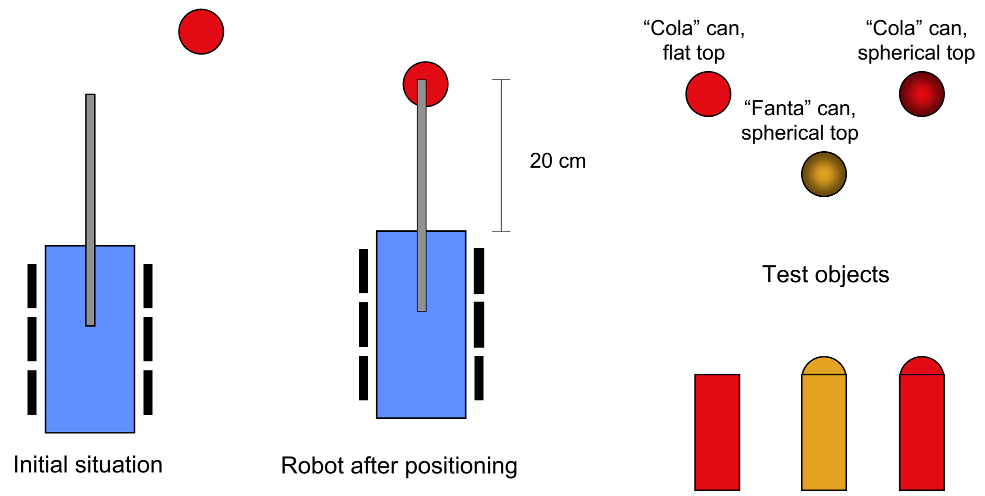



Figure 2: Simple scenario for affordance "liftability"

| **Name:** | Use Case "liftability" | **Category:** | single | **Level:** | 1 |
|---|---|---|---|---|---|
| **Scene description:** | Elements of the environment:<br>*Scenario setup:*<br>• Mostly homogeneous background (standard)<br>• One test object in robots sensor range, but out of crane arm range<br>• No other test object present<br>• Test object stands upright<br>*Test objects:*<br>• A red "Cola" can with flat top is liftable<br>• An orange "Fanta" can with spherical top is not liftable<br>• A red "Cola" can with spherical top is not liftable<br>These test objects are selected to maximize the probability that the "flat top" feature gets selected for describing the affordance "liftability". If the extension sensor is being implemented in the robot's crane arm, the use case could be extended by objects that are too heavy to be lifted. | | | | |

Table 1: Mission description for use case "liftability", part 1

| | |
|---|---|
| **Scene description:** | *To be done:*<br>• Specify exact physical properties of test objects.<br>• Specify exact physical properties of scenario.<br>• Construct test objects and scenario, both in simulation and reality |
| **Essential affordances:** | **liftability** |
| **Task description:** | **Phase1 Learning of Affordances:**<br>• Try lifting a variety of test objects<br>• Result: Description of affordance "liftability"<br>**Phase2 Goal-directed use of the affordance "liftability":**<br>• Subtask1: Search a liftable test object and lift it<br>• Subtask2: Try unknown test object (either color or size differs): Is the learned affordance description sufficient for lifting the unknown test object? Or put differently: In goal-directed mode, will the perception filtered by the goal affordance recognize the unknown test object as offering the same affordance as the original test object? |
| **Expected performance:** | specify |

Table 2: Mission description for use case "liftability", part 2

#### 4.1.2 Use Case "Lift test object": Basic Skills

Basic robot skills for attempting to lift a test object:

- Use bottom-up attention to focus color blob in both camera images (FhG/AIS attention system VOCUS)
- Determine approximate 2D position using triangulation (new)
- Compute robot target pose (test object 20 cm in front of robot)
- Assume target pose (controllers exist)
- Controlled by 3D Laser scanner
- Track lowering magnet in both camera images and control lowering (new)
- Switch on magnet (new)
- Lift magnet (new)
- Feedback on success of lifting attempt via image processing (new)

Concluding actions for experiments that are not required for lifting as such: "Letting loose", including: Lower magnet, switch off magnet, lift magnet to home position, drive to home position.
To be done: Describe implicit assumptions (perception, action)

### 4.1.3 Possible feature sets for affordance "liftability"

**Extension sensor of crane arm rope:** Higher weight indicates magnetic contact

**Optical features:** Color
Attention system VOCUS can supply: FOA region and center position (image coordinates, ¡x,y¿) Feature vector of a FOA

**Optical features:** Flat top of test object
Combination with 3D required?

**SIFT:** Rectangular and circular features could be used for discrimination

Open questions:

- Is segmentation required? If so, use attention window
- Could the depth image be used?

## 4.2 Affordance "pushability"

### 4.2.1 Experimentation scenario

Pushable test objects have a smooth object bottom to reduce friction and a low center of gravity, in order to avoid turning over whilst being pushed (Tab. 3).

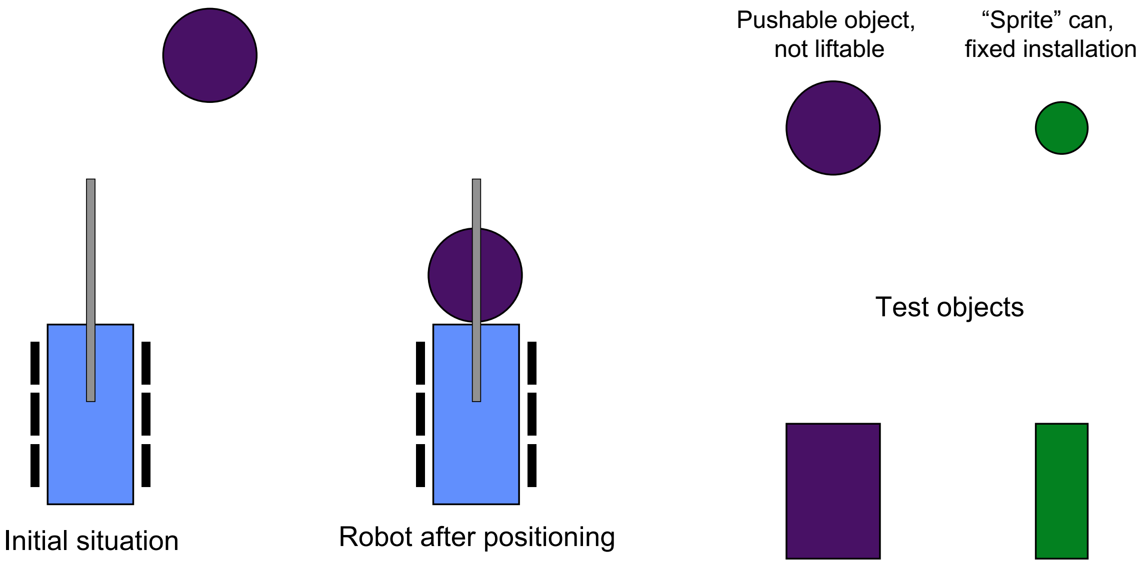



Figure 3: Simple scenario for affordance "pushability"

### 4.2.2 Use Case "Push test object": Basic Skills

Basic robot skills for attempting to push a test object:

- Use bottom-up attention to focus color blob in both camera images (VOCUS)
- Determine approximate 2D position using triangulation (new)
- Compute robot target pose (test object directly in front of robot)

<table>
<tr><td>Name:</td><td>Use Case “pushability”</td><td>Category:</td><td>single</td><td>Level:</td><td>1</td></tr>
<tr><td>Scene description:</td><td colspan="5">Elements of the environment:<br>Scenario setup:<br>• Mostly homogeneous background (standard)<br>• One test object in robot’s sensor range, but not in contact with robot<br>• No other test object present<br>• Test object stands upright<br>Test objects:<br>• A green “Sprite” can, fixed installation is not pushable<br>• A purple “Block”, smooth bottom, not fixed, is pushable</td></tr>
<tr><td>Essential affordances:</td><td colspan="5">liftability</td></tr>
<tr><td>Task description:</td><td colspan="5">Phase1 Learning of Affordances:<br>• Try pushing a variety of test objects<br>• Result: Description of affordance “pushability”<br>Phase2 Using the affordance “pushability”:<br>• Task: Search a pushable test object and push it<br>• Try unknown test object (either color or size differs): Is affordance sufficient for pushing the unknown test object?<br>Open question: Attach “pushability primarily” to feature “size”?</td></tr>
<tr><td>Expected performance:</td><td colspan="5">specify</td></tr>
</table>

Table 3: Mission description for use case “pushability”

- Assume target pose (controllers exist), ontrolled by 3D Laser scanner
- Robot initiates pushing -¿ Modify controller to avoid damages

Concluding actions for experiments that are not required for pushing as such: “Retreat”, including: drive to home position.
Action points:

- How to determine self motion?
- How to determine pushability? Answer: Pressure foil sensors for KURT3D robot (to be developed)
- Develop push test behavior
- Develop push behavior
- Describe implicit assumptions (Perception, action)

#### 4.2.3 Possible feature sets for affordance "pushability"

**Pressure sensor:** Force feedback
**Optical feature:** Color of test object
**3D Laser scanner:** Size of test object

### 4.3 Affordance "traversability"

#### 4.3.1 Experimentation scenario

<table>
<tr><td>Name:</td><td>Use Case "traversability"</td><td>Category:</td><td>single</td><td>Level:</td><td>1</td></tr>
<tr><td>Scene description:</td><td colspan="5">Elements of the environment:<br>Scenario setup:<ul><li>Mostly homogeneous background (standard)</li><li>Elevated regions</li><li>Walls</li><li>Plain, empty floor</li><li>Passages</li><li>Ramps</li></ul>Open question: Separate scenarios for each of these elements?<br>Test objects:<ul><li>A green "Sprite" can, fixed installation is not pushable</li><li>A purple "Block", smooth bottom, not fixed, is pushable</li></ul></td></tr>
<tr><td>Essential affordances:</td><td colspan="5">liftability</td></tr>
<tr><td>Task description:</td><td colspan="5">Phase1 Learning of Affordances:<ul><li>to be defined</li><li>Result: Description of affordance "traversability"</li></ul>Phase2 Using the affordance "pushability":<ul><li>Task: to be defined</li></ul></td></tr>
<tr><td>Expected performance:</td><td colspan="5">specify</td></tr>
</table>

Table 4: Mission description for use case "traversability"

#### 4.3.2 Use Case "Driving": Basic Skills

- Use 3D Laser scanner for determining minimum distances to environmental elements (traversable free space, new)
- Use attention to drive towards a salient region (VOCUS)

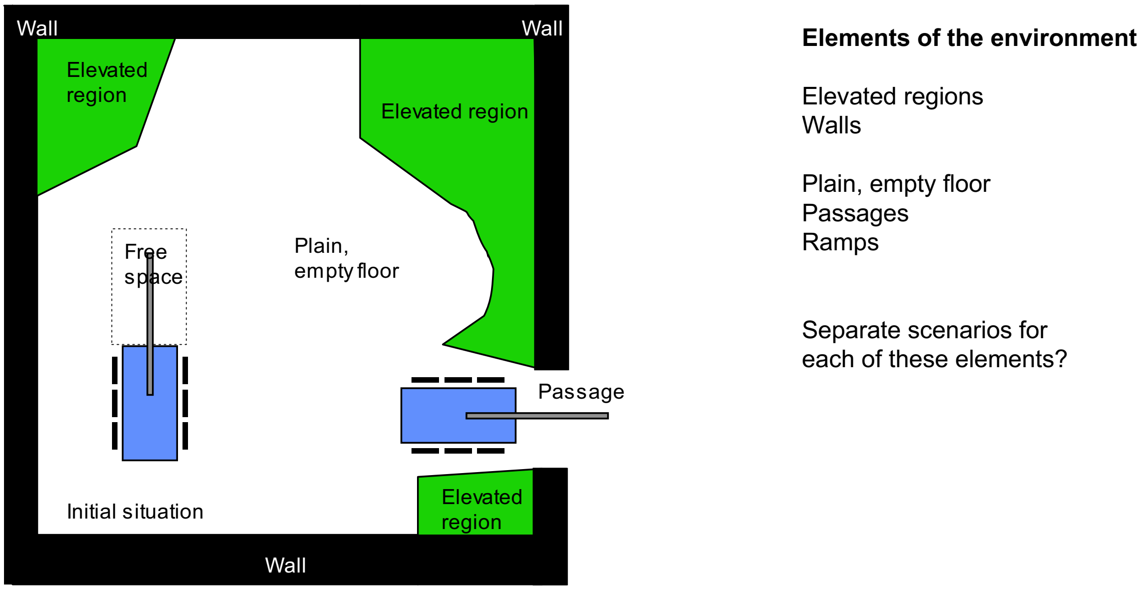



Figure 4: Scenario for affordance "traversability"

Open Questions:

- Is traversability the proper affordance? Should non-traversability be used instead?
- Do we need a new controller for driving?
- What are the assumptions on environmental elements?
- Steer or control?

Action points: Describe implicit assumptions (perception, action). Integrate use case with work on affordance-based motion planner (Deliverables D4.2.1+D4.3.1).

### 4.3.3 Possible feature sets for affordance "traversability"

**Odometry:** motion feedback
**Optical feature:** To be defined
**3D Laser scanner:** To be defined

## 5 Discussion on other affordances

So far, the three basic affordances described in this documents are relations between the robot and its environment. The affordance "stackability" means that one test object is placable on top of another test object. It has been suggested as a basis for manipulation of more than one test object. This is an example of an *inter-object affordance*. It is not yet clear whether this type of affordance should be investigated as well. It might be sufficient to use other descriptive means, e.g. the affordance of a planar surface spot in the environment to be used as support for "placables" (placable test objects).

Possible affordances for implementing stacking of test objects:

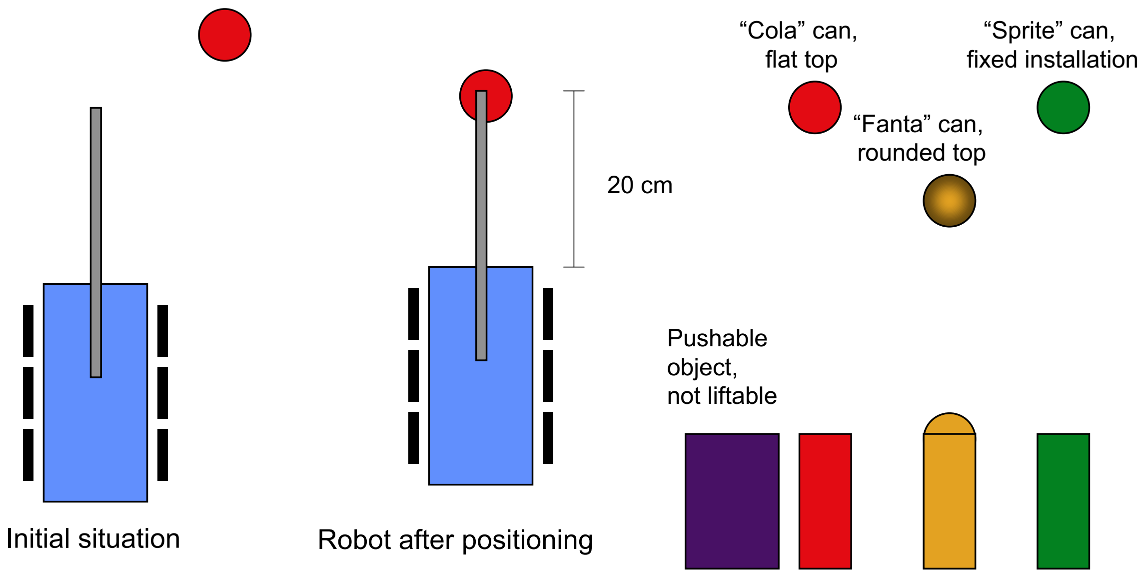



Figure 5: Combined scenario for affordances “liftability” and “pushability”.

Supportability: A rigid horizontal planar surface in the environment to be used as support. It can be a patch on the floor or the top of a test object.

Placability: A test object that can be placed by the robot (a lifted test object by lowering or a pushable by directed pushing).

## 6 Outlook and future work

Define affordances for iteration stage 2:

- Reachability: A test object can be reached by the robot. That is, the robot can drive through free space close enough to the test object so that it gets into the range of the robot's crane arm.
- Stackability: Confer discussions section above.

Method for defining other affordances:

- Describe use cases analogously
- Describe exploration phase analogously
- Describe mission phase analogously

Next stages of this development process will include:

- Combinations of the affordances of iteration 1
- More test objects
- More affordances
- More features to distinguish affordances
- Exception cases

- Make scenarios increasingly complex, combine scenarios
- Varying environmental settings (static/dynamic, unknown test objects etc.)
- More complex affordances
- Generalization
- Categorization
- Learning by observation
- Non-permanent affordances
- Making basic skills affordance-based

Architectural issues:

- Guarantee compatibility between low levels and higher levelsff
- Aggregate simple affordances to more complex ones

Open questions:

- How to combine affordances?
- Relation between pushability and traversability? Can it be learned?
- Where to use multi sensor information fusion? Liftability?