UHH Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Technical Aspects of Multimodal Systems
Department of Informatics

TAMS

Project Intelligent Robotics
Assignment #3

On this Assignment, you will learn to use the communication possibilities of ROS. You already used existing messages and worked with them. Now you will create your own messages.

Task 3.1 Write a simple server and a client node:

3.1.1: Write a service with a request and a response. You will find useful information in this tutorials:

```
http://wiki.ros.org/ROS/Tutorials/WritingServiceClient(python)
http://wiki.ros.org/ROS/Tutorials/WritingServiceClient(c++)
```

Define a common message type with all project members!

3.1.2: Write a simple server that gets an integer and answers with $integer + 1$.

3.1.3: Commit your Sever to the git. Write a client that uses a service from one of the other groups.

Task 3.2 Write a simple action server and a client node:

3.2.1: Write an action sever that does this:

```
i = 0
while i < 100:
        i =- 1
        print "True"
```

```
http://wiki.ros.org/actionlib_tutorials/Tutorials/SimpleActionServer(ExecuteCallbackMethod)
```

3.2.2: Write a client that uses this action server and test it.

```
http://wiki.ros.org/actionlib_tutorials/Tutorials/SimpleActionClient
```

3.2.3: Find out what you can do if the action takes to long and modify your client to exit the loop.

3.2.4: Fix the code to get a successful action.

Task 3.3 Count collaboratively: Now write several nodes (one per group) that perform a collaborative count from 1 to 20. Every node has to do at least one counting step. Run the nodes on different computers. Hint: you can use `rqt_graph` to visualize your node topology.

For this task the whole group has to work together. Make sure that everybody has accomplished Assignment #1 and #2. Discuss Assignment #3 in a group meeting before you start.

3.3.1: Gather as a group and work on a concept to reach this goal. Plan your communication.

3.3.2: Write the nodes in small groups

3.3.3: Start the nodes on different computers. Make sure that each node registers at the same ROS master.

Task 3.4 Distributed architecture: Adapt your code from task 3.3 to use multiple ROS masters. Use the `multimaster_fkie` package for this purpose.