Technische Universität München

Faculty of Informatics

Robotics and Embedded Systems

http://www6.in.tum.de	May 5, 2010
Lab Course: Human Robot Interaction	**Sheet 3**
Manuel Giuliani, Thomas Müller, Prof. Alois Knoll	giuliani@in.tum.de, muelleth@in.tum.de

Submission Deadline: 12 May 2010

Exercise 3

In this exercise, your dummy robobt server from the last exercise will be extended. The interface

```
bool isReachable(Hand h, ::jast::common::TableCoordinates coords);
```

now has to return `true` for coordinates in certain ranges. The robot is able to grasp objects with coordinates located inside a rectangle on the table, which is described by these ranges. We will measure these coordinates together in the JAST room and write them down on the course website.

Exercise 4

Now, that your interaction system can tell whether a coordinate is reachable by the robot, we have to actually integrate the object recognition. Again, for testing your code, in this exercise you will implement an object recognition dummy server. The `slice` interfaces you need for that can be found at http://www6.in.tum.de/Main/TeachingSs2010LabCourseHRI.

Your object recognition dummy needs to instantiate objects, as the real object recognizer behaves very similar. To get an impression of the information, that the dummy needs to send, have a look at `jast/common/OrecObject.ice`:

- Objects have a **unique id** as long as they are visible on the table.
- Objects furthermore have coordinates on the table (x, y, orientation).
- The object description holds the type of the object (see `jast/common/Object.ice` for all known types), the color (possible values are `Blue, Red, Green, White, Yellow, Orange, Black`, and `Undefined`) and for `Slat` objects also the size (`Small, Medium`, or `Large`).

Implementing the object recognition dummy, you need to take the following into account:

1. Your dummy holds a list of objects currently "visible" on the table at any time. This list is empty initially.

2. The module adds a new random object to the list every 0 to 30 seconds (both reachable and non-reachable). Of course, objects may also disappear from the table (imagine the user or the robot has picked one up), so your module also removes random objects from time to time.

3. Publishing the list of current objects is done at 15 cycles per second and the module always publishes a complete list of objects. To publish the objects you have to use the interface `recognizedObjects(OrecObjectList objects, Timestamp time)` that is defined in `jast/listener/ObjectRecognitionListener.ice`.

Exercise 5

The most interesting part comes next: we want you to implement an **object inventory** on the cognition side of the interaction system. The goal of this exercise is to keep track of objects appearing on the table or disappearing from it and reason whether the object can be manipulated with one of the robot's arms. If so, the robot could for example hand an object over to the user or put it next to another one on the table.

1. For the purpose of implementing an inventory of objects, you first need to setup a listener that subscribes to the object recognition module (the dummy you implemented before, or the real one in the JAST room). The inventory needs to maintain a list of objects, equal to the one the object recognition.

2. Now implement the connection to the actuator module (robot and gripper interface).

3. Extend your own object representation with an additional attribute `isReachable`. In the list of objects this additional information about every object shall be maintained. You can retrieve the value of `isReachable` by quering the robot interface.

4. Implement a terminal interface. The terminal offers an easy way of interaction with the robot. The interface shows the list of objects maintained by the inventory and offers one of the actions "`hand over`", "`put next to`" for objects reachable by the robot.

Notice

- From this lesson on we will require you to show a demonstration of your code at the lab and try it at the real robot.
- Nevertheless, please send your solution files by 12 May 2010 to `giuliani@in.tum.de` and `muelleth@in.tum.de` with subject `HRI Lab Course` as usual.