

Cognitive Systems

Practical Exercises – Session 4

Prof. Dr.-Ing. habil. Alois Knoll

Alexander Perzylo, Nikhil Somani, Ahmed Ghazi, Hendrik Walzel

Robotics and Embedded Systems (Informatik VI)

TUM

Outline

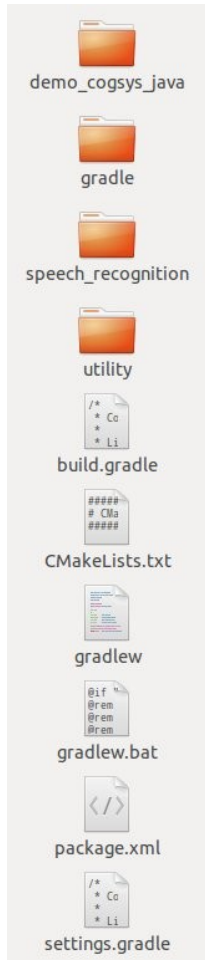
- Brief rosjava introduction
- Links to rosjava related information
- Description of today's exercise
- Commands to run the demo

Brief rosjava Introduction

Java-based client library for ROS

- Enables ROS communication in Java
- Allows for running ROS on Android devices
- Not all core tools available yet (e.g. tf)
- Uses its own build system called *gradle*
- Encapsulates gradle repositories in catkin packages
- One gradle repository may contain multiple (sub)projects

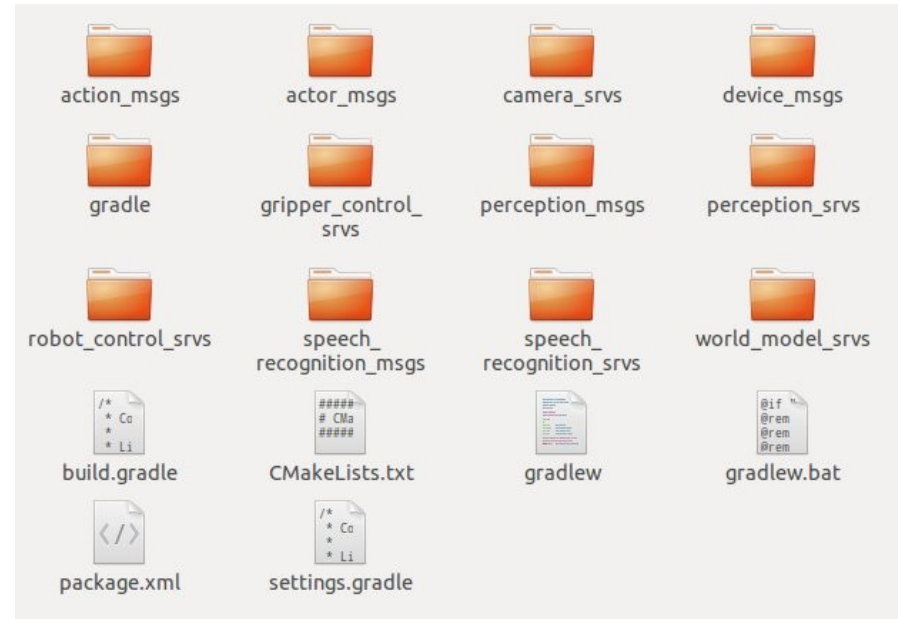
Content of ROS package `rosjava_projects`



- rosjava project *demo_cogsys_java*
- build system related files
- rosjava project *speech_recognition*
- rosjava project *utility*
- Gradle's build file
- Catkin's cmake file
- Gradle wrapper for building rosjava projects directly
- Gradle wrapper for building rosjava projects on windows
- Catkin's package description
- Gradle settings for specifying rosjava projects which shall be built

Content of ROS package rosjava_msgs_srvs

- rosjava requires Java versions of messages and services to be generated
- Use utility scripts to create rosjava
 - catkin packages
 - projects
 - msg/srv projects



```

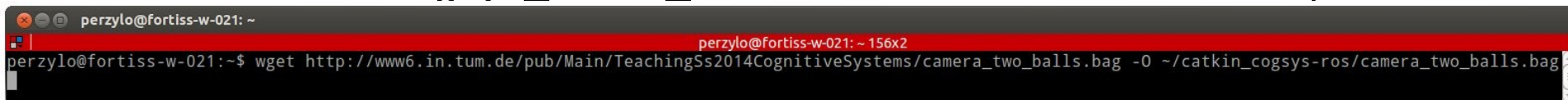
perzylo@fortiss-w-021: ~
perzylo@fortiss-w-021: ~ 72x1
perzylo@fortiss-w-021:~$ catkin_create_rosjava_pkg
perzylo@fortiss-w-021: ~ 72x1
perzylo@fortiss-w-021:~$ catkin_create_rosjava_project
perzylo@fortiss-w-021: ~ 72x1
perzylo@fortiss-w-021:~$ catkin_create_rosjava_msg_project
  
```

Links to rosjava related information

- ROS Wiki <http://wiki.ros.org/rosjava>
- Documentation of rosjava for ROS Hydro
http://rosjava.github.io/rosjava_core/hydro/index.html
- rosjava source code on GitHub <https://github.com/rosjava>
- The Gradle build system <http://www.gradle.org/>

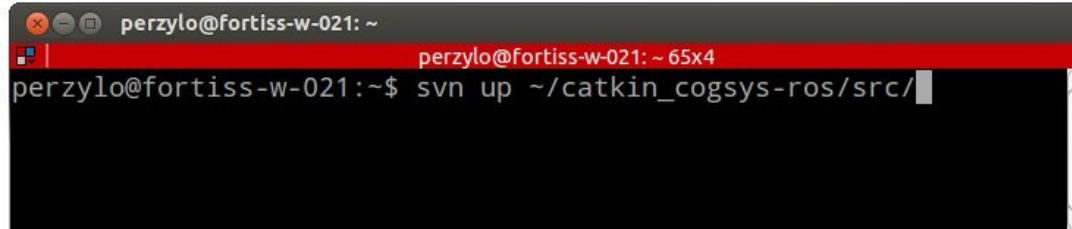
Today's exercise

- Download new ROS bag file (you may copy & paste line from the bottom of the *cogsys_howto_v4.txt* from the lecture's website)



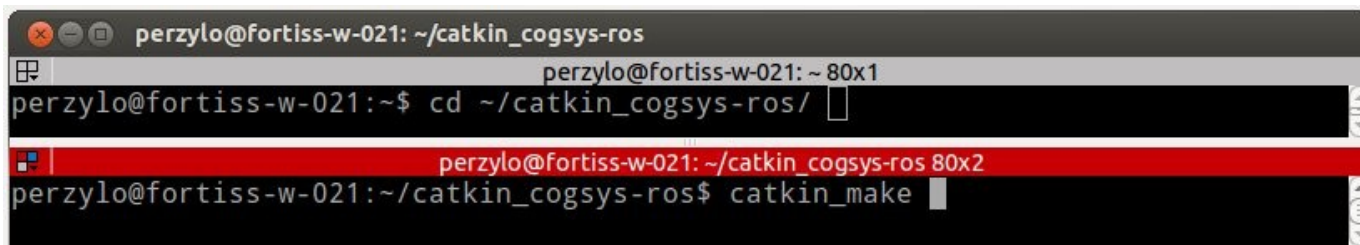
```
perzylo@fortiss-w-021: ~  
perzylo@fortiss-w-021: ~ 156x2  
perzylo@fortiss-w-021:~$ wget http://www6.in.tum.de/pub/Main/TeachingSs2014CognitiveSystems/camera_two_balls.bag -O ~/catkin_cogsys-ros/camera_two_balls.bag
```

- Important:** Update your local ROS packages to the newest version



```
perzylo@fortiss-w-021: ~  
perzylo@fortiss-w-021: ~ 65x4  
perzylo@fortiss-w-021:~$ svn up ~/catkin_cogsys-ros/src/
```

- Run *catkin_make* in *~/catkin_cogsys-ros/*



```
perzylo@fortiss-w-021: ~/catkin_cogsys-ros  
perzylo@fortiss-w-021: ~ 80x1  
perzylo@fortiss-w-021:~$ cd ~/catkin_cogsys-ros/  
perzylo@fortiss-w-021:~/catkin_cogsys-ros 80x2  
perzylo@fortiss-w-021:~/catkin_cogsys-ros$ catkin_make
```

Today's exercise - Overview

- Pick-and-place tasks:
 - Functionality to be able to grasp objects of a user-specified color and put it into containers of a user-specified color.
 - Utility service in *world_model* which can be queried for a transform between different frames.
- Speech based instruction
 - Define grammar (in JSGF format) which enables recognition of certain utterances. Extend file `~/catkin_cogsys-ros/src/rosjava_projects/speech_recognition/grammar/grammar.gram`
 - Map analyzed speech to pick-and-place tasks

Today's exercise – Implementation details

Finish implementation of world model and then **either choose the Java based or the C++ based demo package to fill in the gaps**

- World Model
 - *world_model/src/main_session4.cpp* : Implement two additional static transform publishers
 - *world_model/src/tflistener_helper.cpp* : Implement the callback function for the *world_model/get_transform* service.
- Demo program (*demo_cogsys_cpp_viz/src/main_session4.cpp*)
 - *pick(string object_name)* and *place(string container_name)* functions for 'pick-and-place' tasks
 - service clients for the *world_model/get_transform* service.
 - speech-based instruction

Today's exercise – Implementation details

- Demo program
(`rosjava_projects/demo_cogsys_java/src/main/java/de/tum/i6/cogsys/Demo.java`)
 - Mapping of recognized speech to pick and place task (in *run* method)
 - Pick and place functionality (in *run* method)

Commands to run the demo

- roscore
- cogsys_visualization
- Robot controllers
- camera_dummy
- Play ROS bag file
- object_detector
- world_model
- speech_recognition
- demo_cogsys_cpp_viz
- **OR**
- demo_cogsys_java

```
perzylo@fortiss-w-021: ~  
perzylo@fortiss-w-021:~$ roscore  
perzylo@fortiss-w-021:~$ roslaunch cogsys_visualization cogsys_demonstrator.launch  
perzylo@fortiss-w-021:~$ roslaunch robot_control_viz bothbots.launch  
perzylo@fortiss-w-021:~$ rosruncamera_dummy camera_dummy  
perzylo@fortiss-w-021:~$ rosbag play -l ~/catkin_cogsys-ros/camera_two_balls.bag  
perzylo@fortiss-w-021:~$ rosruncamera_dummy object_detector objdetect  
perzylo@fortiss-w-021:~$ rosruncamera_dummy world_model world_model_node_session4  
perzylo@fortiss-w-021:~$ rosruncamera_dummy rosjava_projects speech_recognition  
perzylo@fortiss-w-021:~$ rosruncamera_dummy demo_cogsys_cpp_viz demo_cogsys_viz_cpp_session4  
perzylo@fortiss-w-021:~$ rosruncamera_dummy rosjava_projects demo_cogsys_java
```