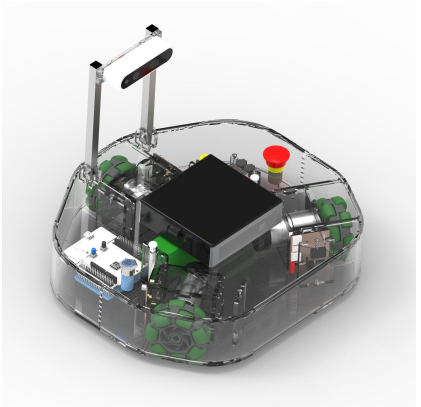
ROS Tools and Programming Basics

ROS Training for Industry: Day 2



Veiko Vunder

17.09.2019

Tartu, Estonia

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Agenda: Day 2 (17.09)

- 09:15 ROS Build/Debug/Visualization Tools
- 10:15 Coffee Break
- 10:30 Workshop
  - Catkin workspace, ROS package, Creating a node
  - Publisher & Subscriber
  - Rqt & RViz Visualization
- 12:00 Lunch Break
- 13:00 ROS Programming: Messages, Services, Actions, Launch files
- 14:30 Coffee Break
- 14:45 Workshop:
  - Parameters & Launch files
  - Messages & Services
- 17:00 End of Day 2

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Catkin and its Workspace

Workspace Structure, Package Building Flow, Usage

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Catkin workspace: Overview

- To build ROS packages from source, one needs a catkin workspace.
- Catkin workspace is a folder (e.g. **`catkin_ws/`**) that contains **`src/`**
- Running **`catkin build`** anywhere in the workspace will build all the packages in **`catkin_ws/src/`**

```
$ source devel/setup.bash
```

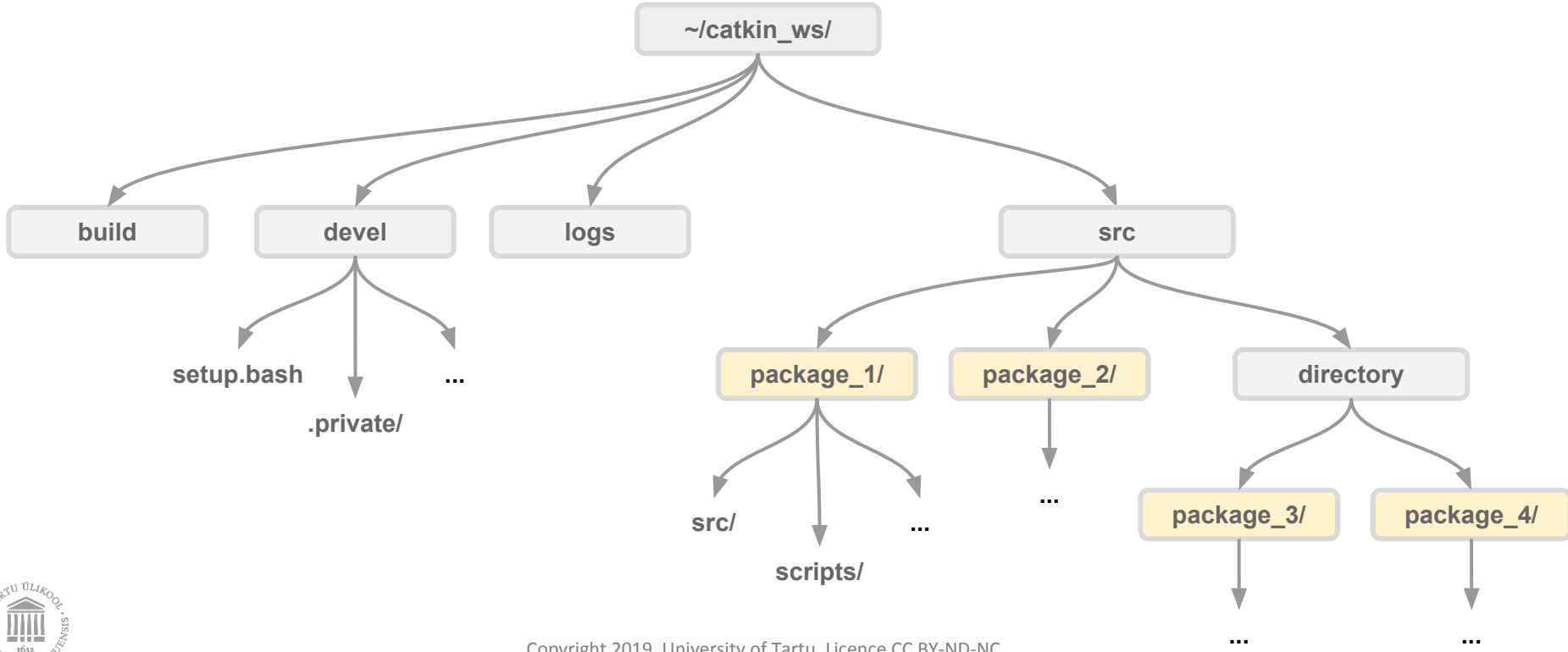
Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Catkin Build Process: Setup

- Create a catkin workspace
  - catkin_ws
- src subdirectory must be created manually
- build, devel directories are created automatically
- Run catkin init from workspace root
- Download / create packages in src subdirectory

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Structure of Catkin workspace



Copyright 2019, University of Tartu, Licence CC BY-ND-NC

catkin

- official build system of ROS
- successor to rosbuild
- CMake macros, Python scripts
- provide extra functionality:
  - automatic 'find package' infrastructure
  - build multiple, dependent projects at the same time
- Builds are out-of-source

http://wiki.ros.org/catkin/conceptual_overview

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Isolated vs non isolated build

catkin_make

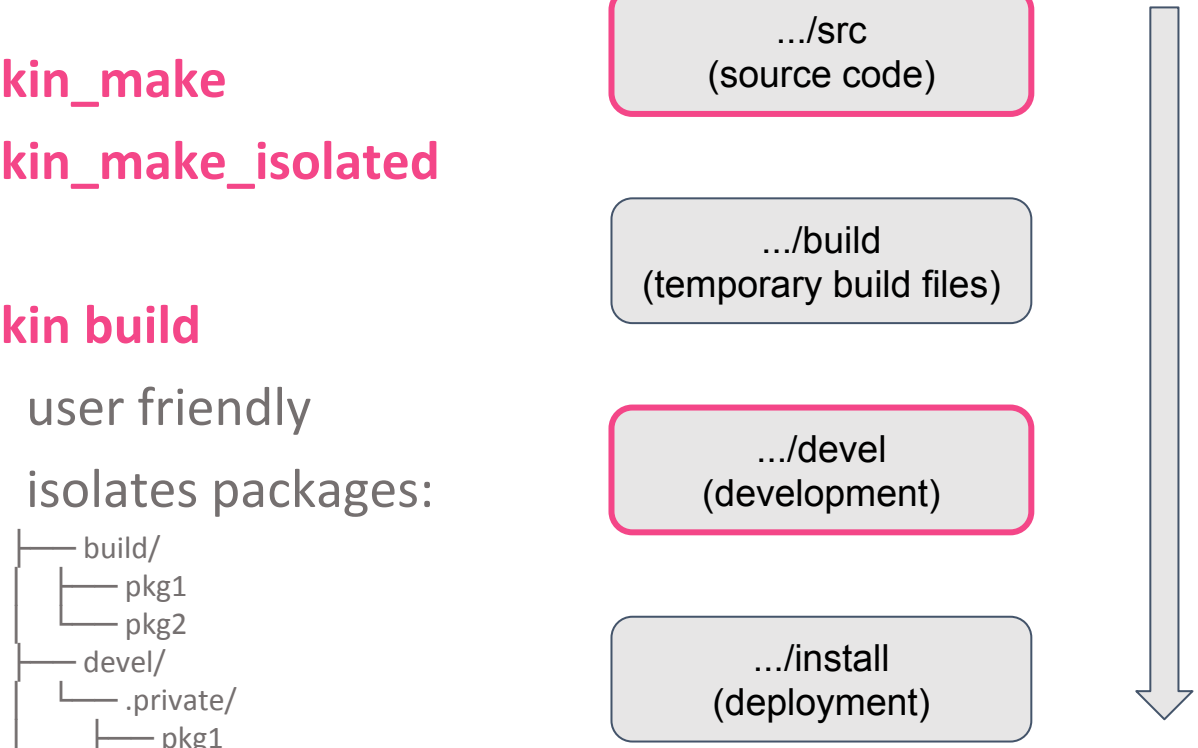
catkin_make_isolated

catkin build

user friendly

isolates packages:

```
build/
  pkg1
  pkg2
devel/
  .private/
    pkg1
    pkg1
```



Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Catkin Build Process: After building

Run source devel/setup.bash to make workspace visible to ROS

- Re-execute in each new terminal window
- Add to ~/.bashrc to automate this process

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow

Getting From “0” to Executable Project

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow

1. **Using just other packages**
   - Precompiled Binaries (from package manager, e.g., *apt*)
   - From source files
2. **Making your own custom & independent package**
3. **Both 1+2**

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

1.

Using Other Packages

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Using Other Packages

Debian Packages

- Nearly "automatic"
- Recommended for end-users
- Stable
- Easy to use

Source Repositories

- Access "latest" code
- Most at Github.com
- More effort to set up
- Potentially unstable

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Using Other Packages

How to find the right package?

ROS Website (http://ros.org/browse)

- Browse for known packages

ROS Answers (http://answers.ros.org)

- When not sure, ask someone!

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Using Other Packages

Install via package manager

```
sudo apt install ros-kinetic-package
```

- Fully automatic install:
- Download .deb package from central ROS repository
- Copies files to standard locations (`/opt/ros/kinetic/...`)
- Install any other required dependencies

```
sudo apt remove ros-kinetic-package
```

- Removes software, but not dependencies!

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Using Other Packages

Catkin build + source-your-workspace

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Using Other Packages

Install from Source

- Find a GitHub repo
- Clone repo into your workspace src directory

```
cd catkin_ws/src
```

```
git clone http://github.com/user/repo.git
```

- Build your catkin workspace

```
cd catkin_ws
```

```
catkin build
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Using Other Packages

Catkin build + source-your-workspace

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

2.

Make Your Own Package

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Make Your Own Pkg

```
catkin_create_pkg pkg_name roscpp rospy
```

Example:

```
catkin_create_pkg sonar_driver roscpp
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Make Your Own Pkg

Modify CMakeLists.txt

1. Leave the stone age (C++ 98)
   - `add_compile_options(-std=c++11)`
2. Create new executable
   - `add_executable( test_node src/test.cpp)`
3. for lazy people … consider ALL imported ***header files***
   - `include_directories( ${catkin_INCLUDE_DIRS} )`
4. for lazy people … consider ALL imported ***libraries***
   - `target_link_libraries( test_node ${catkin_LIBRARIES} )`

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Make Your Own Pkg

Catkin build + source-your-workspace

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

3.

Combine Your Work With Other Packages

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Combine

```
catkin_create_pkg pkg_name dep_1 dep_2 dep_n
```

Example:

```
catkin_create_pkg sonar_driver roscpp sensor_msgs
```

```
Created file sonar_driver/CMakeLists.txt
Created file sonar_driver/package.xml
Created folder sonar_driver/include/sonar_driver
Created folder sonar_driver/src
Successfully created files in /home/veix/test/sonar_driver. Please adjust the
values in package.xml.
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Combine

Modify CMakeLists.txt - *just like in case "2" but ...*

1. Point out where to get package resources
   - `find_package( catkin REQUIRED COMPONENTS` **`<dep_1> <dep_2> <dep_n>`** `…`
2. Declare that you depend on this package **!**
   - `CATKIN_DEPENDS` **`<dep_1> <dep_2> <dep_n>`** `…`

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Combine

Modify package.xml

- If the ***dependency*** is used during **build time** (*headers, libraries*)
  - `<build_depend>dep_1</build_depend>`
  - `<build_depend>dep_2</build_depend>`
  - `<build_depend>dep_n</build_depend>`
- If the ***dependency*** is used during **run time** (*nodes in launch files*)
  - `<exec_depend>dep_1</exec_depend>`
  - `<exec_depend>dep_2</exec_depend>`
  - `<exec_depend>dep_n</exec_depend>`

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS workflow - Combine

Catkin build + source-your-workspace

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Closer look at ROS Nodes

Node setup, Publishers, Subscribers

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Coding example: publisher

```
#include "ros/ros.h"
#include "sensor_msgs/Image.h"
#include "camera.h"

int main(int argc, char* argv[]){
  ros::init(argc, argv, „camera_driver");     // ROS node initialisation
  ros::NodeHandle nh;                         // ROS node handle
  ros::Rate frequency(10);                    // Rate 10 Hz

  // Let's create a ROS publisher on topic called „front camera"
  ros::Publisher pub_cam = nh.advertise<sensor_msgs::Image>(„front_camera", 10);

  while( ros::ok() )
  {
    pub_cam.publish( getCameraImage() );      // Publish single image
    ros::spinOnce();                          // Let other nodes work ;)
    frequency.sleep();                        // Sleep to meet the frequency
  }
  return 0;
}
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Coding example: subscriber

```
#include "ros/ros.h"
#include "sensor_msgs/Image.h"
#include "std_msgs/Point.h"

ros::Publisher pub_position;

void findCircle(sensor_msgs::Image input_image) {
  std_msgs::Point circle_position;
  ...                                          // here be algorithm
  pub_position.publish( circle_position );     // publish circle position
}

int main(int argc, char *argv[]) {
  ros::init(argc, argv, „detect_circles");     // ROS node initialisation
  ros::NodeHandle nh;                          // ROS node handle
  // Let's create a ROS subscriber to „front camera"
  ros::Subscriber subscriber_cam = nh.subscribe(„front_camera", 1, findCircle);
  // Let's create a ROS publisher on „circle location"
  pub_position = nh.advertise<std_msgs::Point>(„circle_location", 1);
  ros::spin();
  return 0;
}
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Modify CMakeLists.txt

1. Create new executable
   - `add_executable( test_node src/test.cpp)`
2. for lazy people … consider ALL imported ***header files***
   - `include_directories( ${catkin_INCLUDE_DIRS} )`
3. for lazy people … consider ALL imported ***libraries***
   - `target_link_libraries( test_node ${catkin_LIBRARIES} )`

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Done with coding?

Catkin build + source-your-workspace

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Debugging & Management Tools

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Debugging Tools

rospack

- … **list**
- … **find**
- … **depends**
- … **depends1**
- … **depends-on**

Example:

```
rospack find laser_filters
:~$ /opt/ros/kinetic/share/laser_filters
```

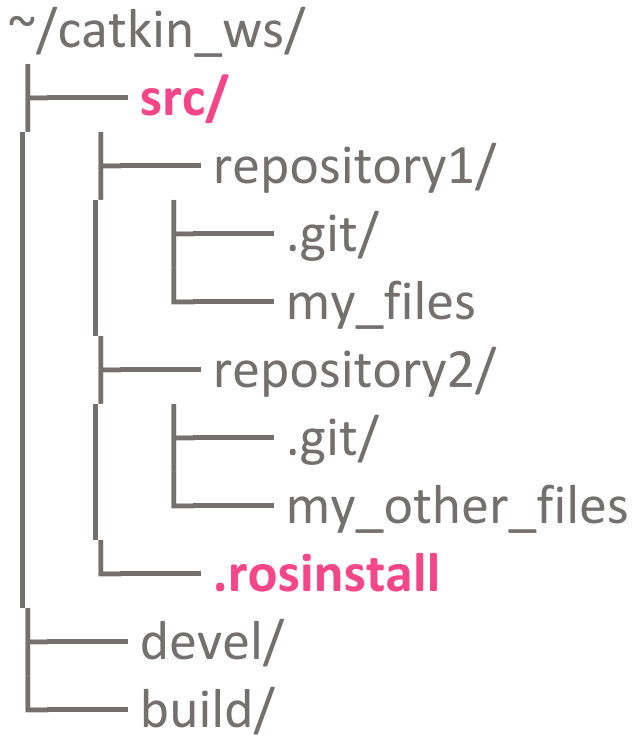
Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Debugging Tools

- **rosnode** [list info kill]
- **rostopic** [list info echo type]
- **rosmsg** [list info/show]
- **roswtf** - examines your ROS setup, searches for configuration issues
- **rosdep check** - check the dependencies of package(s)

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Management Tools - wstool




```
wstool init src
wstool update -t src
wstool merge -t src PATH_TO_ROSINSTALL.rosinstall
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

catkin_ws/src/.rosinstall

YAML file, combines different sources:

- git
- svn
- hg

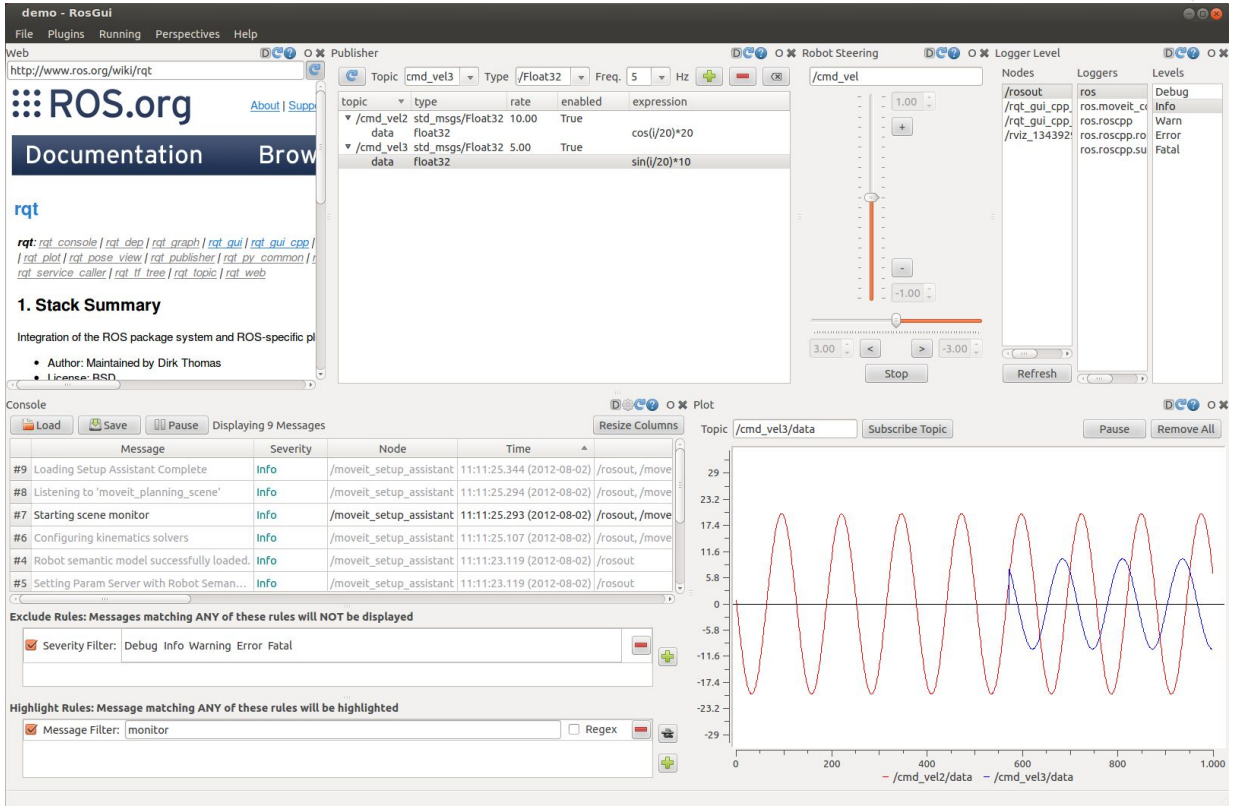
```
- git: {local-name: cartographer, uri:
'https://github.com/googlecartographer/cartographer.git', version: '1.0.0'}

- git: {local-name: cartographer_ros, uri:
'https://github.com/googlecartographer/cartographer_ros.git', version: '1.0.0'}

- git: {local-name: ceres-solver, uri:
'https://ceres-solver.googlesource.com/ceres-solver.git', version: '1.13.0'}
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Graphical Debugging Tools:

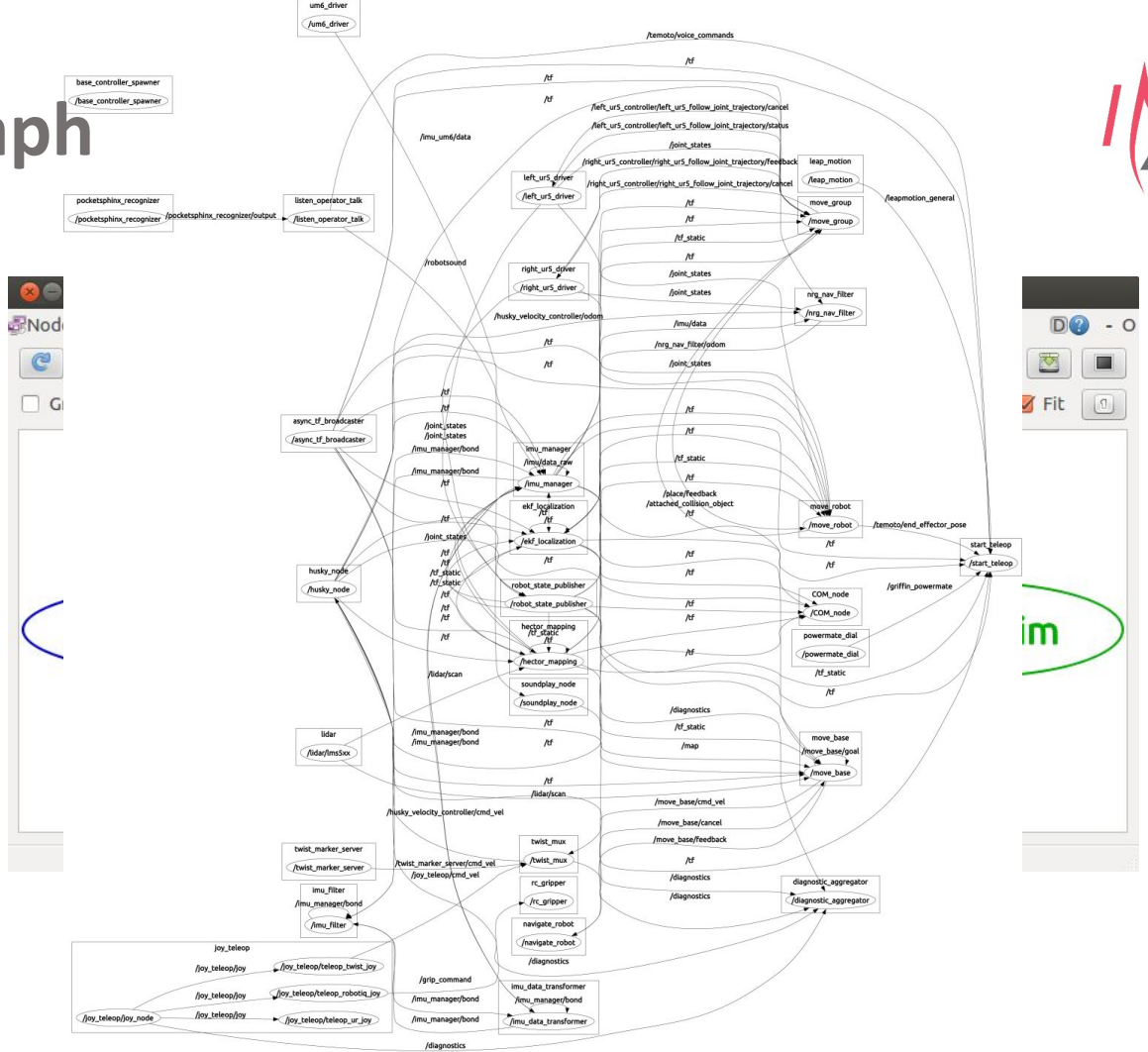
Copyright 2019, University of Tartu, Licence CC BY-ND-NC

rqt

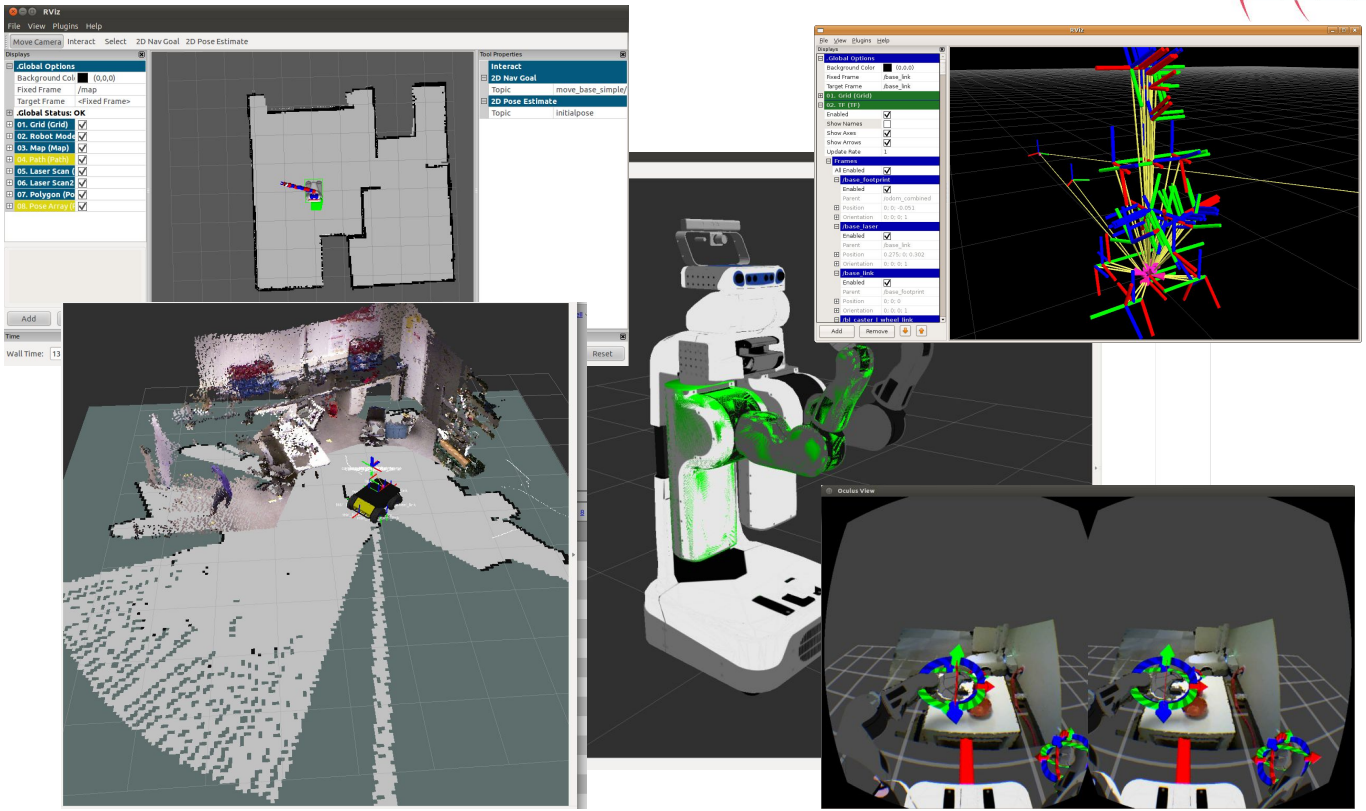


Copyright 2019, University of Tartu, Licence CC BY-ND-NC

rqt_graph


um6_driver
/um6_driver
/temoto/voice_commands
/tf
base_controller_spawner
/base_controller_spawner
/left_ur5_controller/left_ur5_follow_joint_trajectory/cancel
/left_ur5_controller/left_ur5_follow_joint_trajectory/status
/imu_um6/data
/joint_states
/right_ur5_controller/right_ur5_follow_joint_trajectory/feedback
/right_ur5_controller/right_ur5_follow_joint_trajectory/cancel
leap_motion
/leap_motion
left_ur5_driver
/left_ur5_driver
pocketsphinx_recognizer
/pocketsphinx_recognizer
/pocketsphinx_recognizer/output
listen_operator_talk
/listen_operator_talk
move_group
/move_group
/leapmotion_general
/tf_static
/robotsound
right_ur5_driver
/right_ur5_driver
/husky_velocity_controller/odom
nrg_nav_filter
/nrg_nav_filter
/imu/data
/nrg_nav_filter/odom
async_tf_broadcaster
/async_tf_broadcaster
imu_manager
/imu/data_raw
/imu_manager
/place/feedback
/attached_collision_object
move_robot
/move_robot
/temoto/end_effector_pose
start_teleop
/start_teleop
/griffin_powermate
husky_node
/husky_node
ekf_localization
/ekf_localization
robot_state_publisher
/robot_state_publisher
COM_node
/COM_node
hector_mapping
/hector_mapping
powermate_dial
/powermate_dial
/lidar/scan
soundplay_node
/soundplay_node
/diagnostics
/map
lidar
/lidar/lms5xx
move_base
/move_base/goal
/move_base
/husky_velocity_controller/cmd_vel
/move_base/cmd_vel
/move_base/cancel
/move_base/feedback
twist_marker_server
/twist_marker_server
/twist_marker_server/cmd_vel
twist_mux
/twist_mux
/joy_teleop/cmd_vel
rc_gripper
/rc_gripper
diagnostic_aggregator
/diagnostic_aggregator
imu_filter
/imu_manager/bond
/imu_filter
navigate_robot
/navigate_robot
joy_teleop
/joy_teleop/joy
/joy_teleop/teleop_twist_joy
/joy_teleop/teleop_robotiq_joy
/joy_teleop/teleop_ur_joy
/joy_teleop/joy_node
/grip_command
imu_data_transformer
/imu_data_transformer

RViz

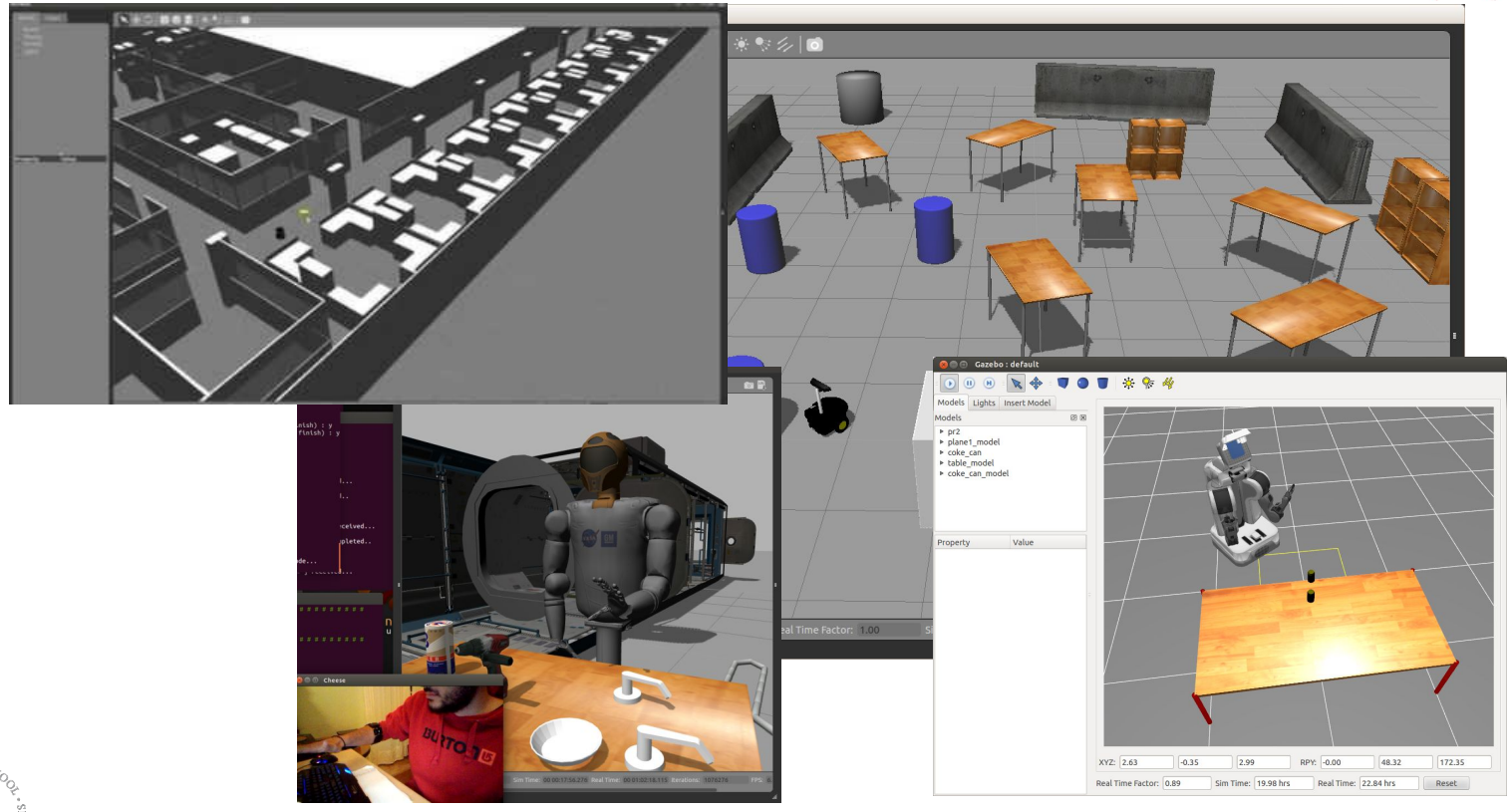


Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Simulation and Data Recording

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Gazebo



Copyright 2019, University of Tartu, Licence CC BY-ND-NC

rosbag

Allows recording and playing back any ros topic

For recording:

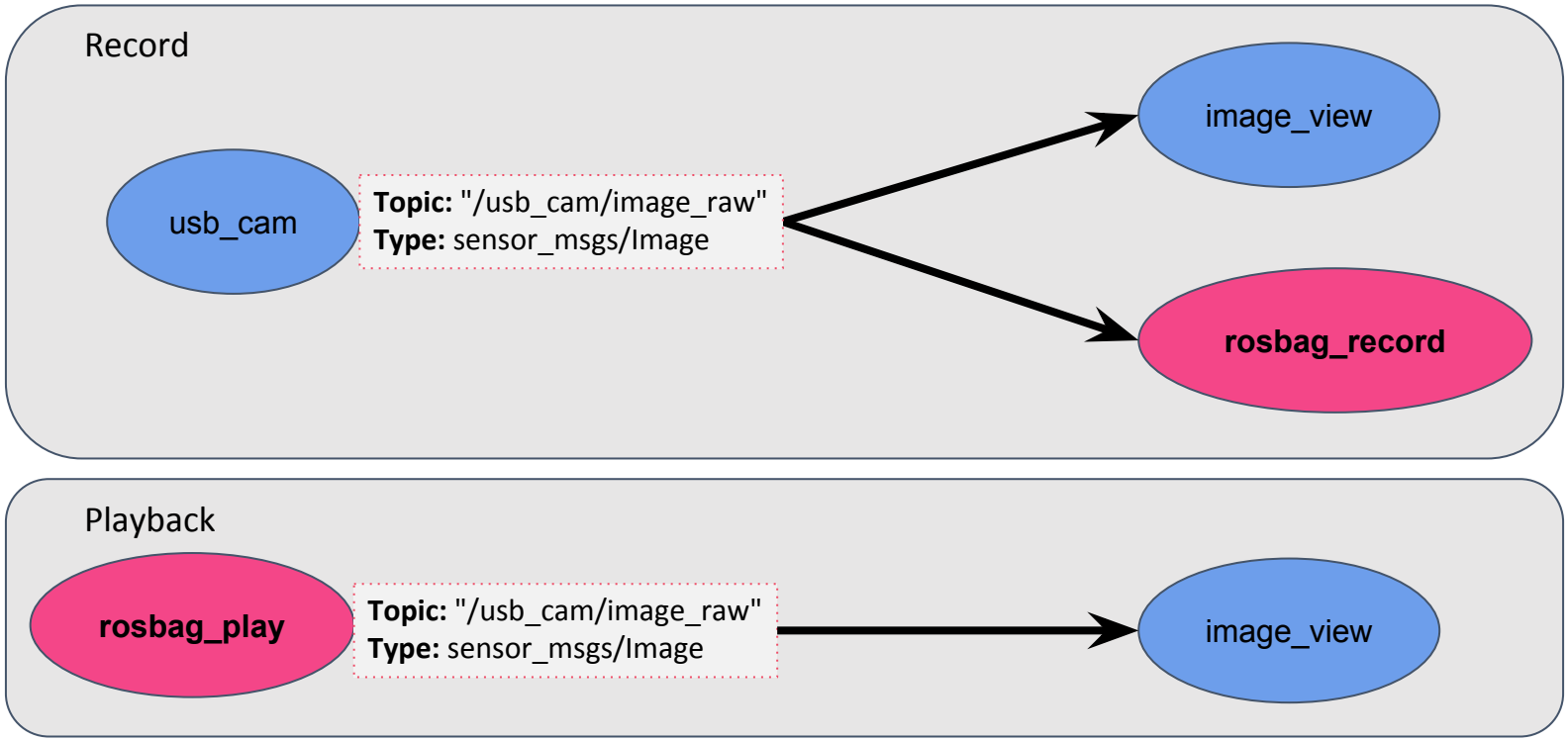
```
rosbag record -O test.bag /topic1 /topic2 /topicN
```

For looped playback

```
rosbag play -l test.bag
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

rosbag



Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Workshop

We **recommend** to **set up IDE** (VS Code)

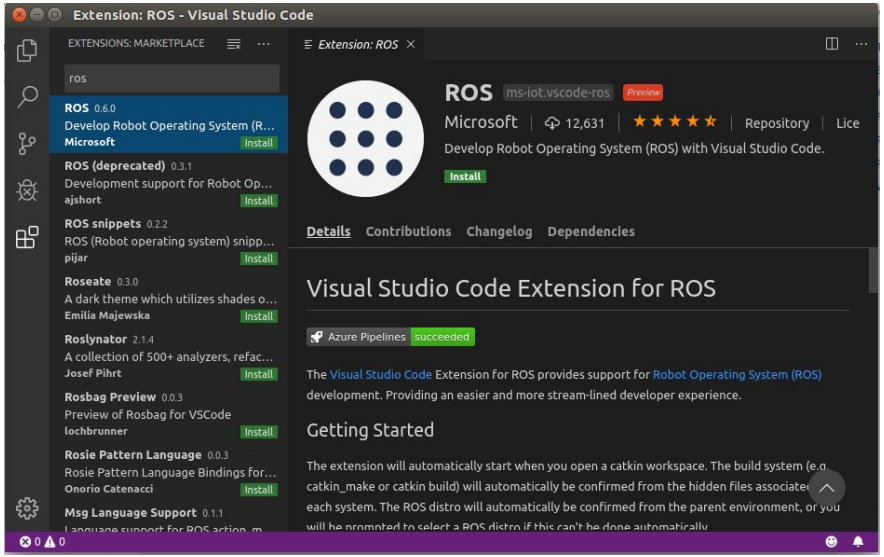
- **C++** autocompletion (Intellisense)
- **ROS** syntax highlight
- **cmake**

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Development environments

Extensions in many editors & IDEs:

- VIM
- QtCreator
- Visual Studio Code
- Eclipse
- …



Benefits:

- Syntax highlighting for ROS files
- Autoformatting:
  - clang-format for cpp files
- Integrate ROS concepts into the IDE

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

ROS Programming *Continued*

Messages. Services. Actions. Launch files - *syntax, integration, usage*

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Messages - syntax

Messages can be created by using:

- **Primitive types**
  - uint64
  - float32
  - float64
  - string
  - *etc …*
- **Other messages**
  - geometry_msgs/Pose pose
  - sensor_msgs/Imu imu
  - std_msgs/String string

CaMeL CaSe!

Example - *MyMessagePrimitive.msg*

```
int64 A
string B
```

Example - *MyMessageCombo.msg*

```
int64 A
string B
geometry_msgs/PoseStamped pose
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Messages - primitive datatypes

Primitive Type	Serialization	C++	Python2	Python3
bool (1)	unsigned 8-bit int	uint8_t (2)	bool	
int8	signed 8-bit int	int8_t	int	
uint8	unsigned 8-bit int	uint8_t	int (3)	
int16	signed 16-bit int	int16_t	int	
uint16	unsigned 16-bit int	uint16_t	int	
int32	signed 32-bit int	int32_t	int	
uint32	unsigned 32-bit int	uint32_t	int	
int64	signed 64-bit int	int64_t	long	int
uint64	unsigned 64-bit int	uint64_t	long	int
float32	32-bit IEEE float	float	float	
float64	64-bit IEEE float	double	float	
string	ascii string (4)	std::string	str	bytes
time	secs/nsecs unsigned 32-bit ints	ros::Time	rospy.Time	
duration	secs/nsecs signed 32-bit ints	ros::Duration	rospy.Duration	

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Messages - build

```
<build_depend>message_generation</build_depend>
<run_depend>message_runtime</run_depend>
```

```
find_package(catkin REQUIRED COMPONENTS roscpp rospy std_msgs
  message_generation)

catkin_package(
   …
   CATKIN_DEPENDS message_runtime …
   …
)
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Messages - build

```
add_message_files(
  FILES
  MyMessageAbc.msg
)

generate_messages(
  DEPENDENCIES
  std_msgs
)
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Messages - Application

Publisher

```
ros::Publisher pub = nh.advertise<my_pkg::MyMsg>(„topic", 10);
```

Subscriber

```
ros::Subscriber sub = nh.subscribe(„topic", 1, callback);
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Services - Syntax

- **Request**

  `msg_type_1` **`a`**

  `msg_type_2` **`b`**

  `other_pkg/Msg` **`c`**

- **Response**

  `msg_type_x` **`d`**

  `msg_type_y` **`e`**

Example - *MyServiceAbc.srv*

CaMeL CaSe!

```
int64 a
int64 b
---
int64 sum
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Services - build

```
<build_depend>message_generation</build_depend>
<run_depend>message_runtime</run_depend>
```

```
find_package(catkin REQUIRED COMPONENTS roscpp rospy std_msgs
  message_generation
)

catkin_package(
  ...
  CATKIN_DEPENDS message_runtime ...
  ...
)
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Services - build

```
add_service_files(
  FILES
  MyServiceAbc.srv
)

generate_messages(
  DEPENDENCIES
  std_msgs
)
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Services - Application

Server side

```
ros::ServiceServer service = nh.advertiseService("topic", callback);
```

```
2  #include "beginner_tutorials/AddTwoInts.h"
3
4  bool add(beginner_tutorials::AddTwoInts::Request  &req,
5           beginner_tutorials::AddTwoInts::Response &res)
6  {
7    res.sum = req.a + req.b;
8    ROS_INFO("request: x=%ld, y=%ld", (long int)req.a, (long int)req.b);
9    ROS_INFO("sending back response: [%ld]", (long int)res.sum);
10   return true;
11 }
```

Client side

```
ros::ServiceClient client = nh.serviceClient<my_pkg::MySrv>("topic");
```

```
16   beginner_tutorials::AddTwoInts srv;
17   srv.request.a = atoll(argv[1]);
18   srv.request.b = atoll(argv[2]);
19   if (client.call(srv))
20   {
21     ROS_INFO("Sum: %ld", (long int)srv.response.sum);
22   }
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Quick Comparison - msg, srv, action

message

- **Fields**

```
field_type_1 a
field_type_2 b
field_type_3 c
```

service

- **Request**

```
msg_type_1 a
msg_type_2 b
msg_type_3 c
```

- **Response**

```
msg_type_x d
msg_type_y e
```

action

- **goal**

```
msg_type_1 a
msg_type_2 b
msg_type_3 c
```

- **result**

```
msg_type_x d
msg_type_y e
```

- **feedback**

```
Msg_type_u f
msg_type_v g
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

roslaunch

- Launch-files enable:
  - Running multiple nodes with a single command
  - Specifying arguments for nodes
  - Remapping
  - Loading parameters to ROS parameter server
- Uses XML

```
$ roslaunch <package> <launch-file>
$ roslaunch ur_modern_driver ur5_bringup.launch
```

Copyright 2019, University of Tartu, Licence CC BY-ND-NC

Workshop

Copyright 2019, University of Tartu, Licence CC BY-ND-NC