1.5.1

ROS Actions – action messages and goal processing

Mukunda Bharatheesha

ROS Actions – generated ROS messages

```
donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_msgs/msg$ tree -L 1
.
├── CounterWithDelayActionFeedback.msg
├── CounterWithDelayActionGoal.msg
├── CounterWithDelayAction.msg
├── CounterWithDelayActionResult.msg
├── CounterWithDelayFeedback.msg
├── CounterWithDelayGoal.msg
└── CounterWithDelayResult.msg
```

Used internally by ROS (actionlib package)

User application

Messages generated for CounterWithDelay action type.

ROS Actions – generated ROS messages

```
donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_msgs/msg$ rosmsg show hrwros_msgs/CounterWithDelayGoal
uint32 num_counts

donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_msgs/msg$ rosmsg show hrwros_msgs/CounterWithDelayResult
string result_message

donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_msgs/msg$ rosmsg show hrwros_msgs/CounterWithDelayFeedback
uint32 counts_elapsed

donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_msgs/msg$
```

Messages for use on the application side

ROS Actions – generated ROS messages

```
donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_msgs/msg$ tree -L 1
.
├── CounterWithDelayActionFeedback.msg
├── CounterWithDelayActionGoal.msg
├── CounterWithDelayAction.msg
├── CounterWithDelayActionResult.msg
├── CounterWithDelayFeedback.msg
├── CounterWithDelayGoal.msg
└── CounterWithDelayResult.msg
```

Messages generated for CounterWithDelay action type.

ROS Actions – action client

hrwros_week1/scripts/counter_with_delay_ac.py

```
client = actionlib.SimpleActionClient(counter_with_delay,
CounterWithDelayAction)

client.wait_for_server()

goal = CounterWithDelayGoal(num_counts = 10)
client.send_goal(goal)
```

ROS Actions – action message

```
donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_msgs/msg$ rosmsg show hrwros_msgs/CounterWithDelayAction
hrwros_msgs/CounterWithDelayActionGoal action_goal
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  actionlib_msgs/GoalID goal_id
    time stamp
    string id
  hrwros_msgs/CounterWithDelayGoal goal
    uint32 num_counts
hrwros_msgs/CounterWithDelayActionResult action_result
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  actionlib_msgs/GoalStatus status
    uint8 PENDING=0
    uint8 ACTIVE=1
    uint8 PREEMPTED=2
    uint8 SUCCEEDED=3
    uint8 ABORTED=4
    uint8 REJECTED=5
    uint8 PREEMPTING=6
    uint8 RECALLING=7
    uint8 RECALLED=8
    uint8 LOST=9
    actionlib_msgs/GoalID goal_id
      time stamp
      string id
    uint8 status
    string text
  hrwros_msgs/CounterWithDelayResult result
```

Time stamp and header information

Unique goal identifier

Goal message

Time stamp and header information

Different states the goal can be in

ROS Actions – action message

```
hrwros_msgs/CounterWithDelayActionFeedback action_feedback
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  actionlib_msgs/GoalStatus status
    uint8 PENDING=0
    uint8 ACTIVE=1
    uint8 PREEMPTED=2
    uint8 SUCCEEDED=3
    uint8 ABORTED=4
    uint8 REJECTED=5
    uint8 PREEMPTING=6
    uint8 RECALLING=7
    uint8 RECALLED=8
    uint8 LOST=9
    actionlib_msgs/GoalID goal_id
      time stamp
      string id
    uint8 status
    string text
  hrwros_msgs/CounterWithDelayFeedback feedback
    uint32 counts_elapsed
```

Time stamp and header information

Different states the goal can be in

Feedback message

ROS Actions – action server goal callback

hrwros_week1/scripts/counter_with_delay_as.py

```
self._as = actionlib.SimpleActionServer(”counter_with_delay”,
CounterWithDelayAction, self.execute_cb=execute_cb,
auto_start=False)


def execute_cb(self, goal):
        #Process any preemption request first.
        for counter_idx in range(0, goal.num_counts):
                self._feedback.counts_elapsed=counter_idx
                rospy.sleep(1.0)
```

ROS Actions – action server topics

```
donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_week1/scripts$ rostopic list | grep counter
/counter_with_delay/cancel
/counter_with_delay/feedback
/counter_with_delay/goal
/counter_with_delay/result
/counter_with_delay/status
```

```
donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_week1/scripts$ rostopic info /counter_with_delay/feedback
Type: hrwros_msgs/CounterWithDelayActionFeedback

Publishers:
 * /counter_with_delay (http://tudelft:39495/)

Subscribers: None
```

```
donnie@tudelft:~/ros/hrwros_ws/src/hrwros/hrwros_week1/scripts$ rostopic info /counter_with_delay/cancel
Type: actionlib_msgs/GoalID

Publishers: None

Subscribers:
 * /counter_with_delay (http://tudelft:39495/)
```

ROS Actions - Remember!!

Attention!

- Simple Action Server/Client: Instance of the generic action server from actionlib ROS package.
- Non-blocking execution of **ONE** goal at a time.
- New goal to the same action server will pre-empt an active goal.
- Always possible to have your own action server implementation that can process multiple goals.