Project: welcome robot

Olivier Aycard

Associate Professor

University of Grenoble

Laboratoire d'Informatique de Grenoble

http://membres-liglab.imag.fr/aycard

olivier.aycard@imag.fr

Outline

1. What you have to do in your project ?
2. Suggestions of project

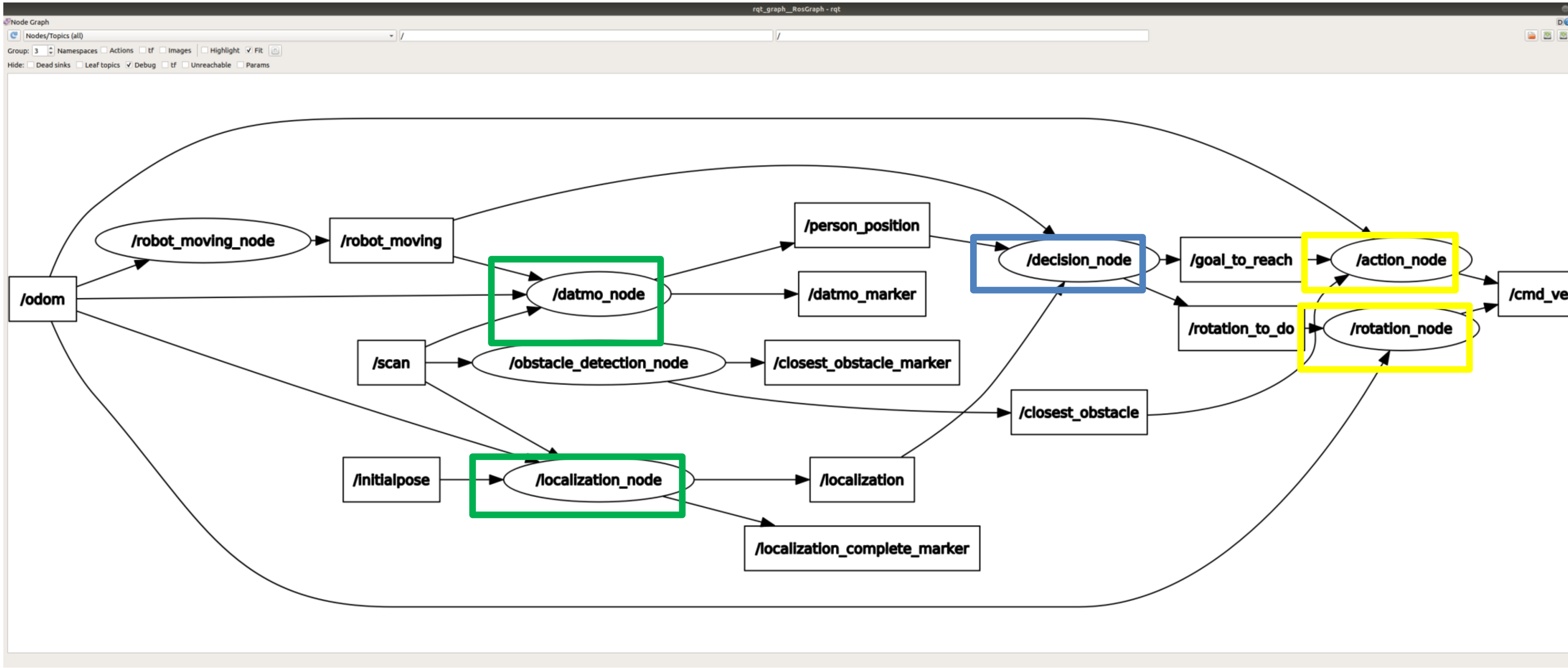
What you have to do in your project ? (1/2)

- **Apply and « get in depth » these basic concepts on a practical project:**
  1. **Improve existing nodes/functions**
  2. **Implement new nodes:**
     - **A decision/supervisor node**

What you have to do in your project ? (2/2)

- **Design your software architecture in terms of nodes and messages/topics**
- **For each node, identify:**
  - **Data that it needs;**
  - **Data that it produces;**
  - **Which kind of nodes is it ? (perception, decision, action)**

Welcome robot: software architecture



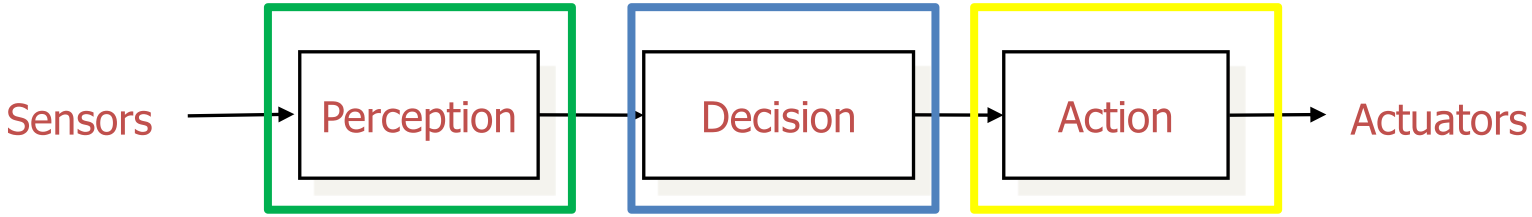
rqt_graph__RosGraph - rqt

Node Graph

Nodes/Topics (all) /

Group: 3 Namespaces □ Actions □ tf □ Images □ Highlight ☑ Fit

Hide: □ Dead sinks □ Leaf topics ☑ Debug □ tf □ Unreachable □ Params



Sensors → Perception → Decision → Action → Actuators

Welcome robot: what you have to do

1. **Implement decision/supervisor node**
2. **Improve/modify datmo_node, localization_node, rotation_done, action_node**

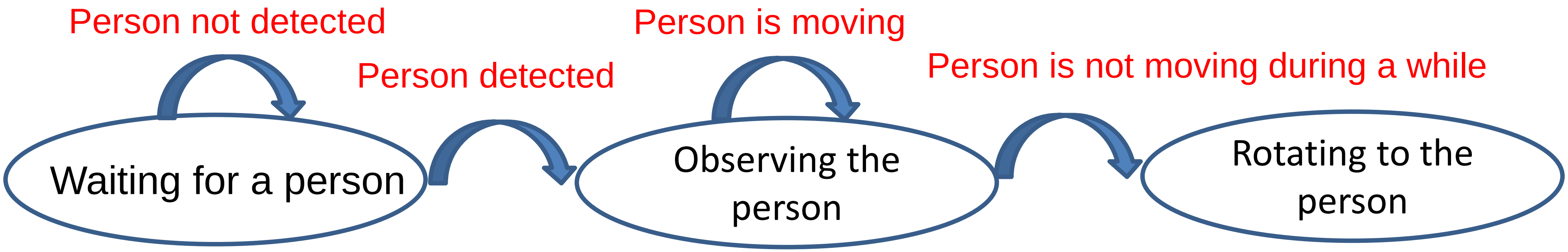
Welcome robot: scenario

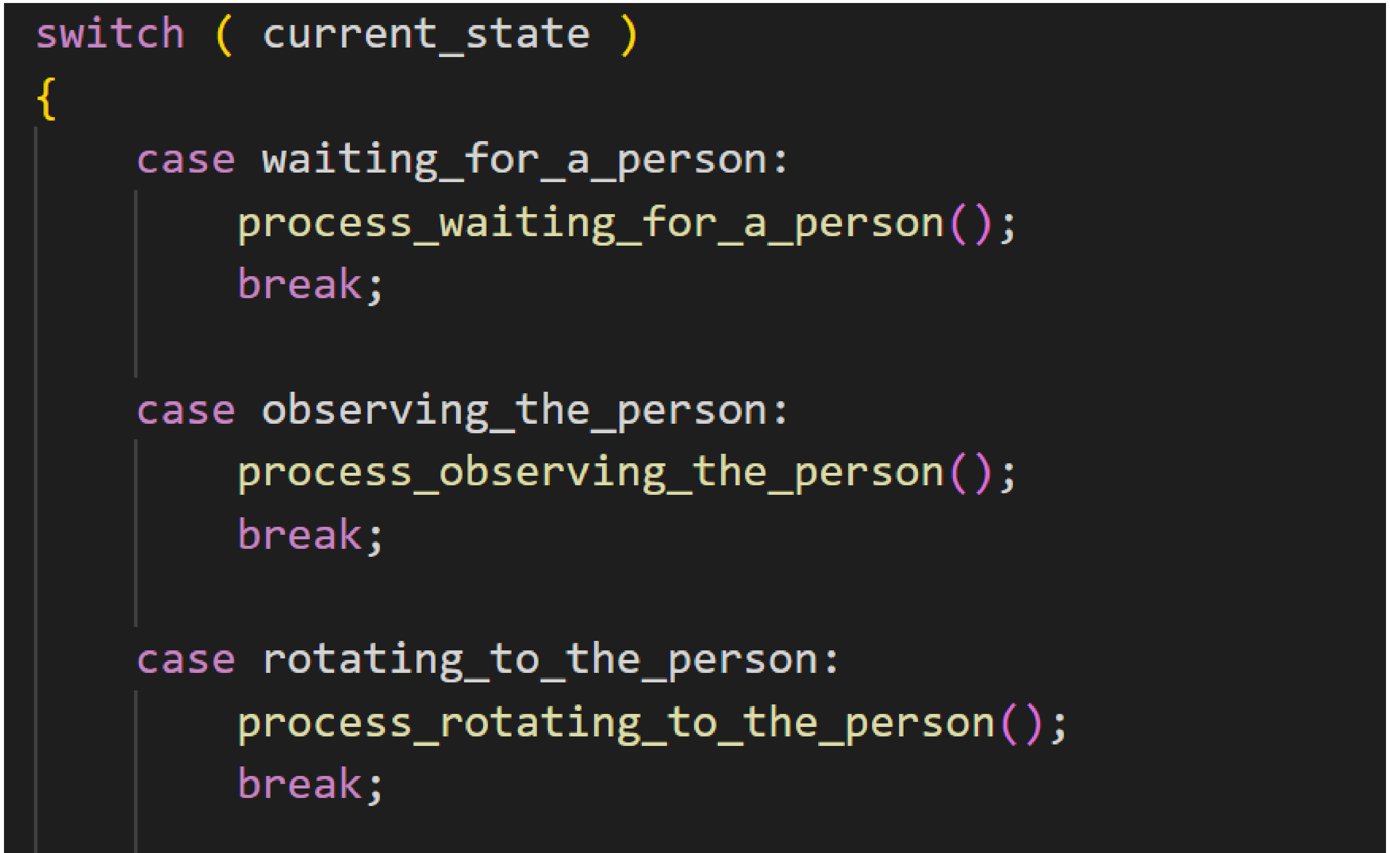
1. **Robair waits for a moving person;**
2. **Robair detects and tracks the moving person in its field view;**
3. **Robair rotates to be face to the moving person;**
4. **Robair moves to the moving person;**
5. **Robair interacts with this moving person;**
6. **Robair rotates to be face to its initial position (ie, its base)**
7. **Robair goes back to its initial position (ie, its base)**
8. **Robair rotates to go back to its initial orientation**

See video (in French) at: https://youtu.be/4xJ6G4p6ITM

Welcome robot: finite state automata

- **Implement a finite state automata/machine**

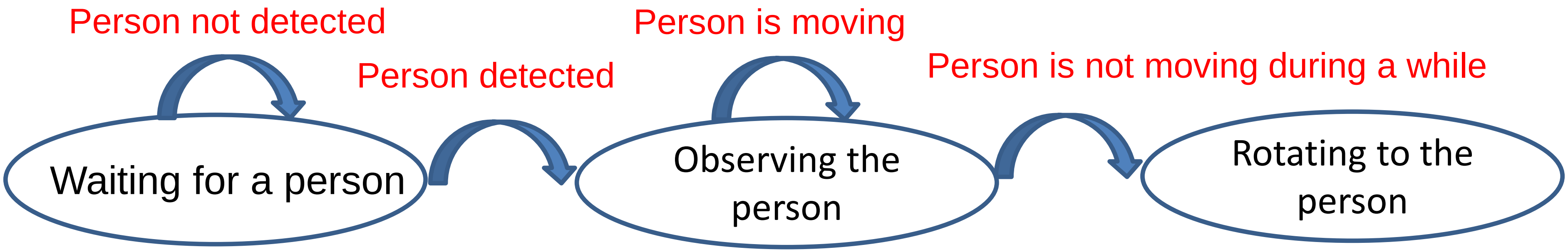


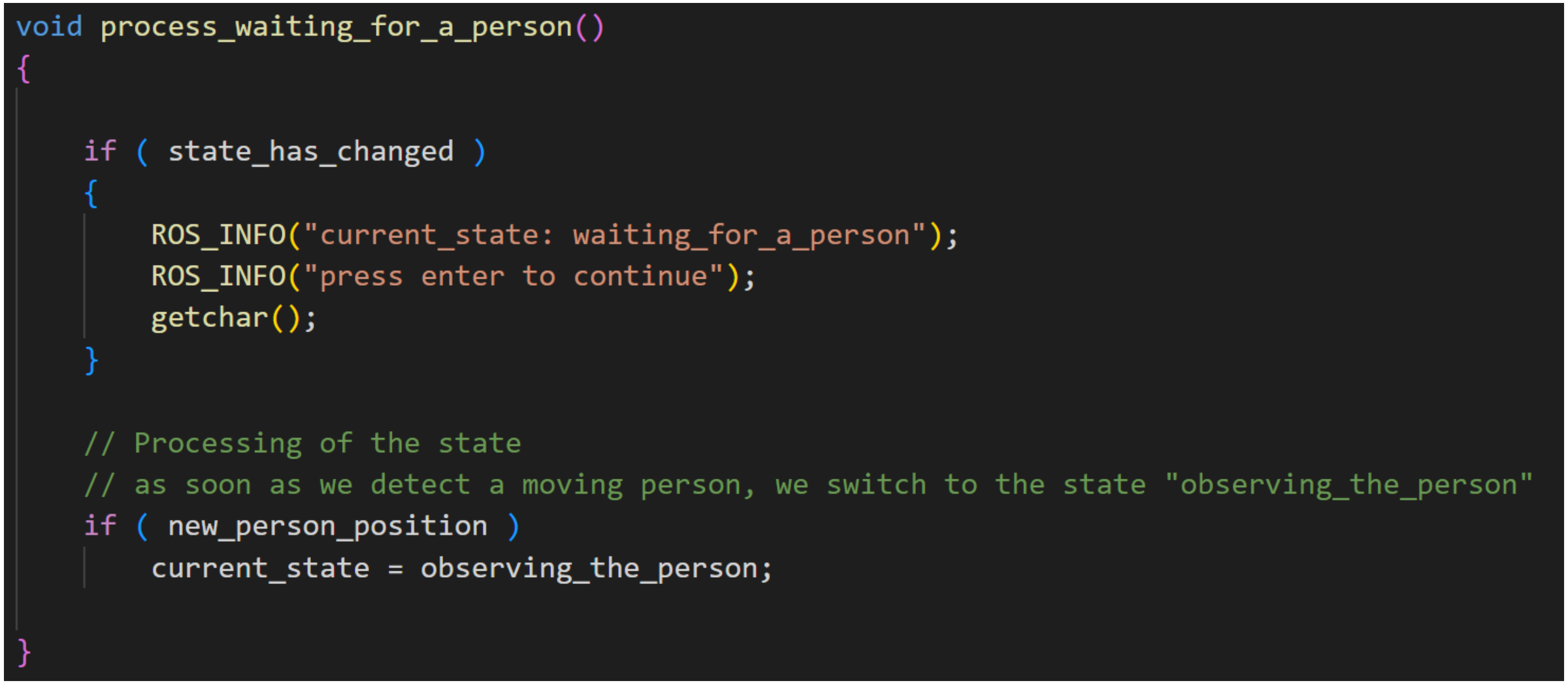


```
switch ( current_state )
{
    case waiting_for_a_person:
        process_waiting_for_a_person();
        break;

    case observing_the_person:
        process_observing_the_person();
        break;

    case rotating_to_the_person:
        process_rotating_to_the_person();
        break;
```

Welcome robot: finite state automata

- **Implement a finite state automata/machine**





```
void process_waiting_for_a_person()
{

    if ( state_has_changed )
    {
        ROS_INFO("current_state: waiting_for_a_person");
        ROS_INFO("press enter to continue");
        getchar();
    }

    // Processing of the state
    // as soon as we detect a moving person, we switch to the state "observing_the_person"
    if ( new_person_position )
        current_state = observing_the_person;

}
```

Welcome robot: evaluation

- **Defense + demo during our examen week (beginning of may)**
    - **Defense: 3 points**
    - **Demo: 9 points**
        - **8-9 points: excellent;**
        - **6-7 points: very good;**
        - **4-5 points: good;**
        - **2-3 points: average;**
        - **0-1 point: insufficient.**