Course of

Robot Programming with ROS 2

Day 3

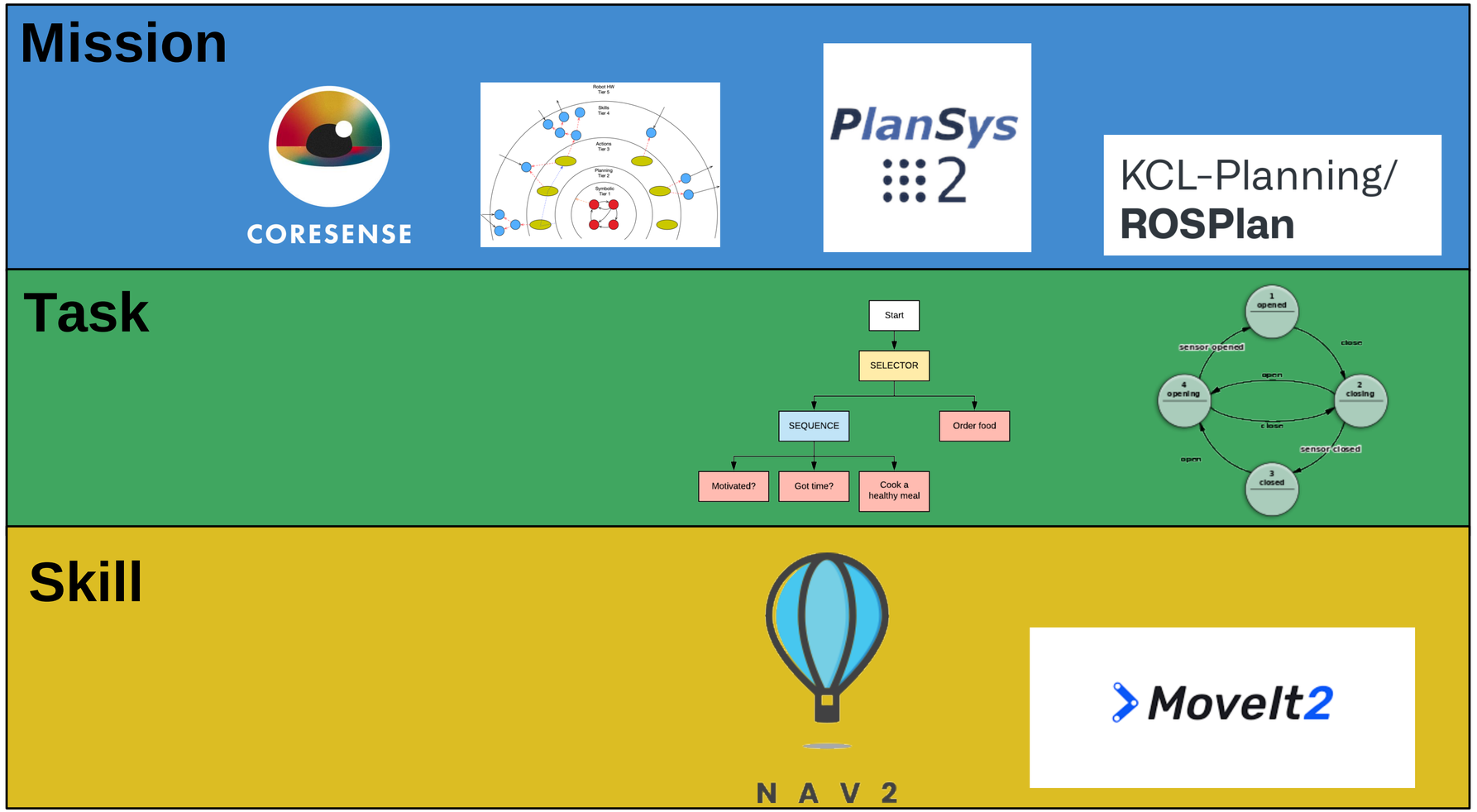
5. PlanSys2

ikerlan

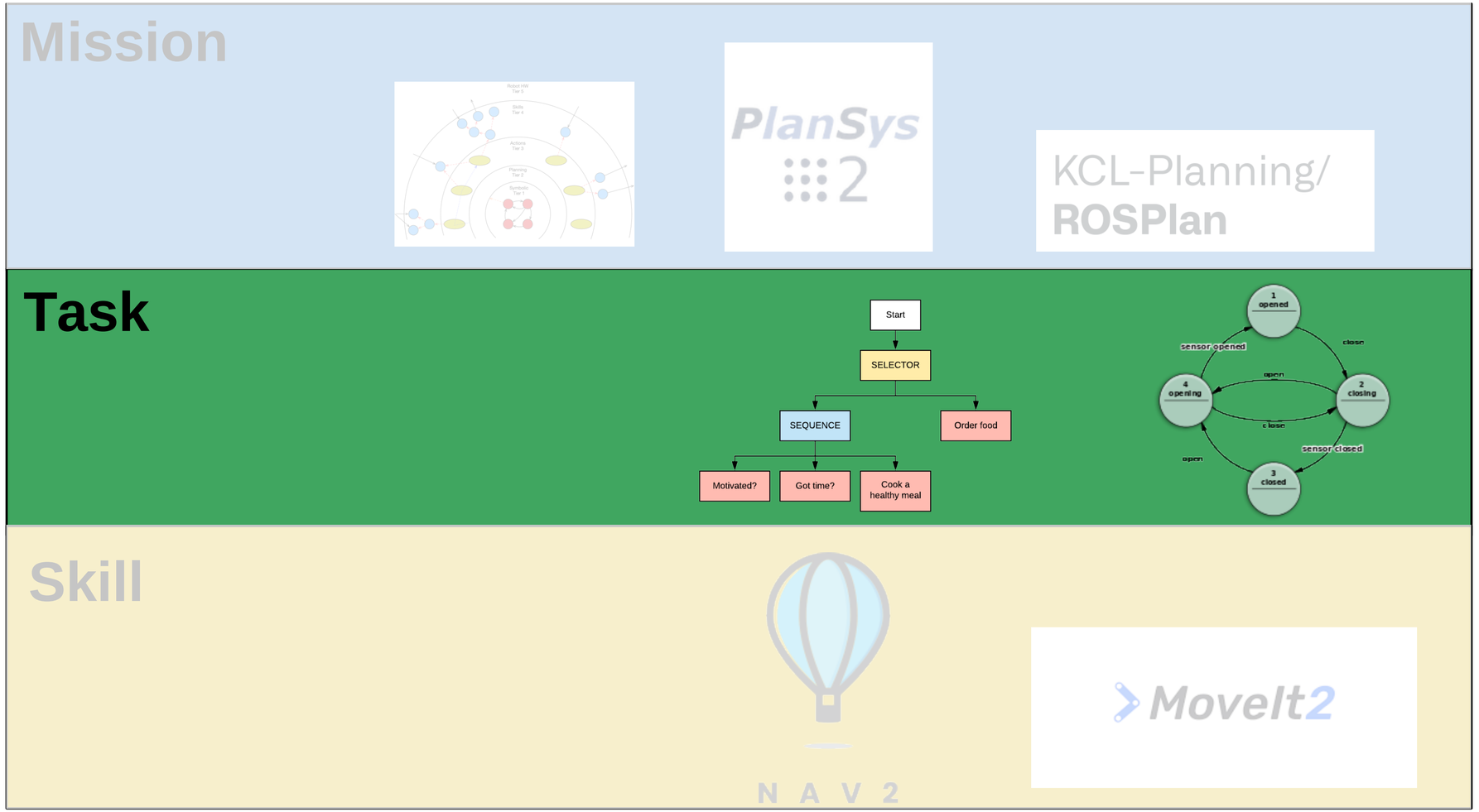
Universidad Rey Juan Carlos

Intelligent Robotics Lab

Preamble

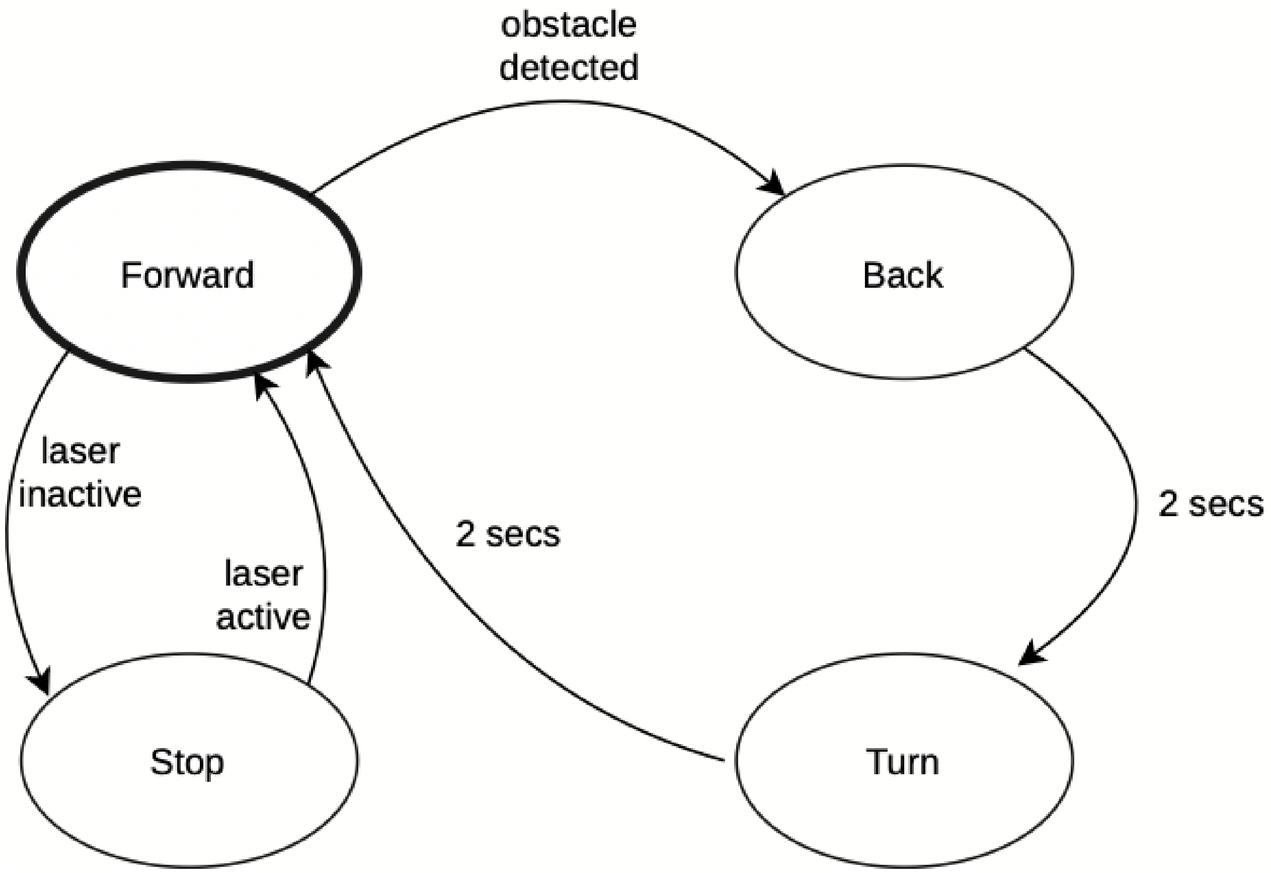

Mission
CORESENSE
Robot HW
Tier 5
Skills
Tier 4
Actions
Tier 3
Planning
Tier 2
Symbolic
Tier 1
PlanSys 2
KCL-Planning/
ROSPlan
Task
Start
SELECTOR
SEQUENCE
Order food
Motivated?
Got time?
Cook a healthy meal
1 opened
2 closing
3 closed
4 opening
sensor_opened
close
open
close
sensor_closed
open
Skill
NAV2
MoveIt2

Part I


Mission
PlanSys2
KCL-Planning/
ROSPlan
Task
Start
SELECTOR
SEQUENCE
Order food
Motivated?
Got time?
Cook a healthy meal
1 opened
2 closing
3 closed
4 opening
sensor_opened
close
open
close
sensor_closed
open
Skill
NAV2
MoveIt2

Finite State Machines

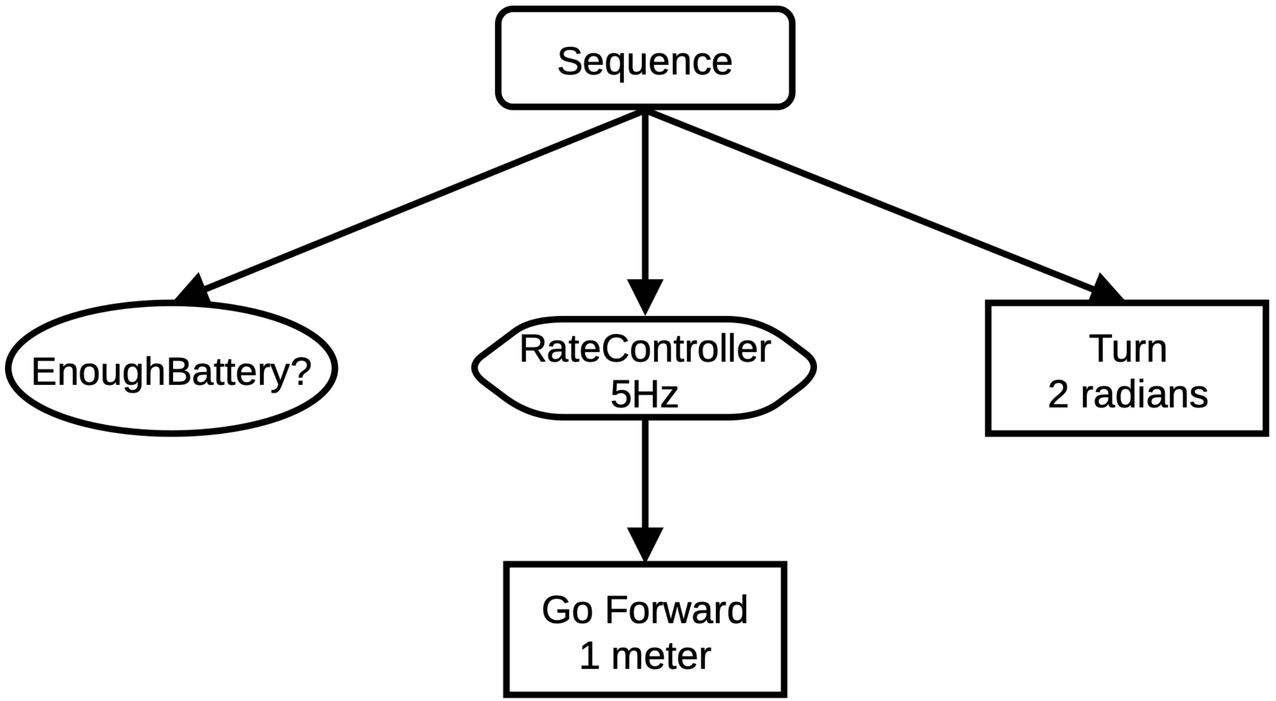
Bump&Go Behavior

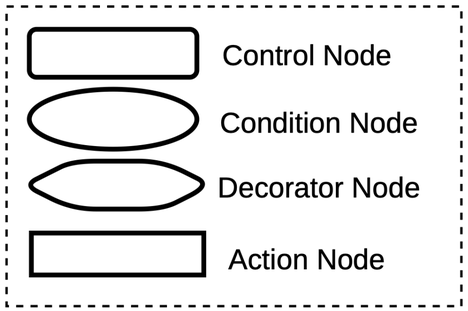


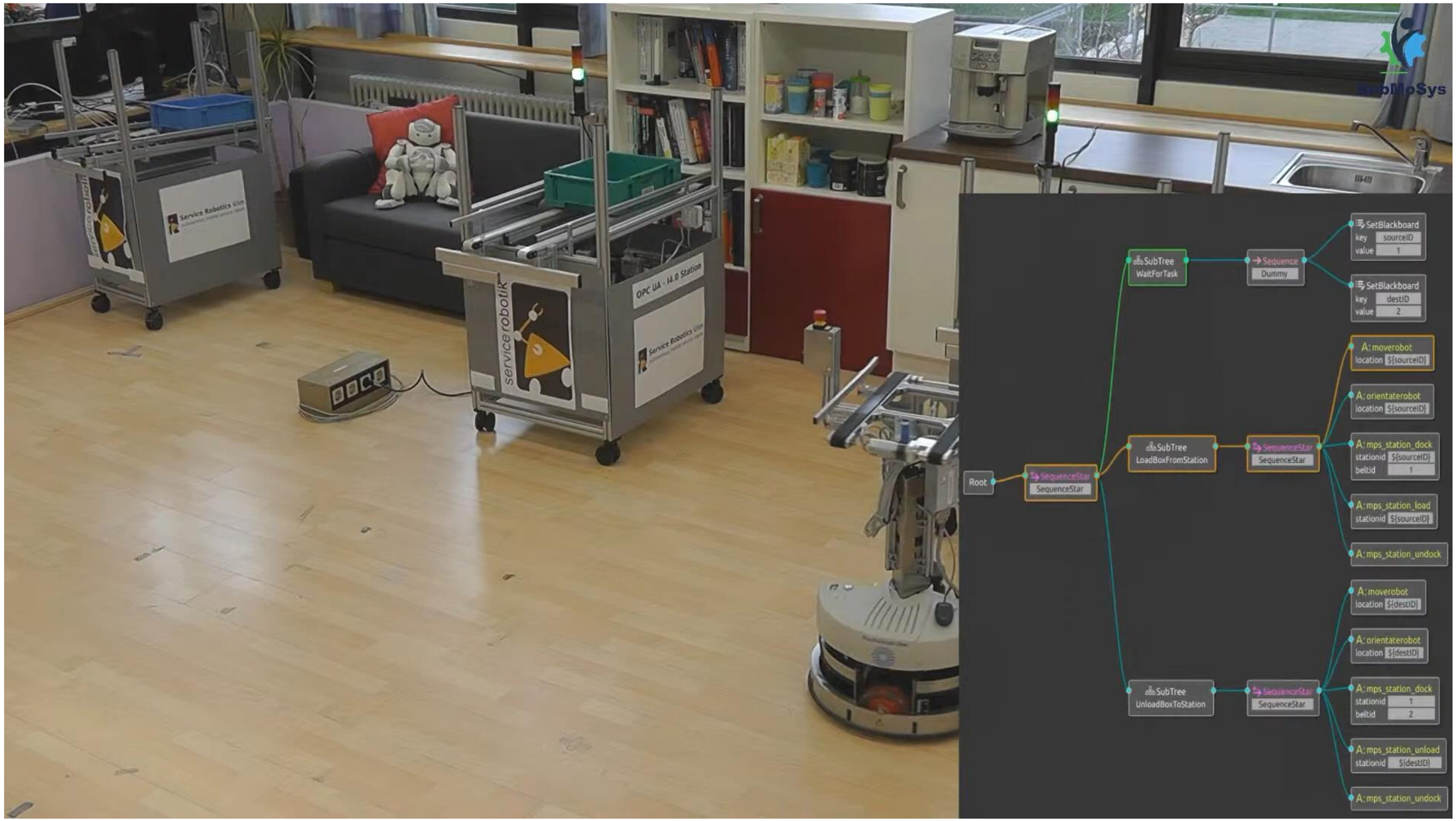
Finite State Machines

Bump&Go Behavior

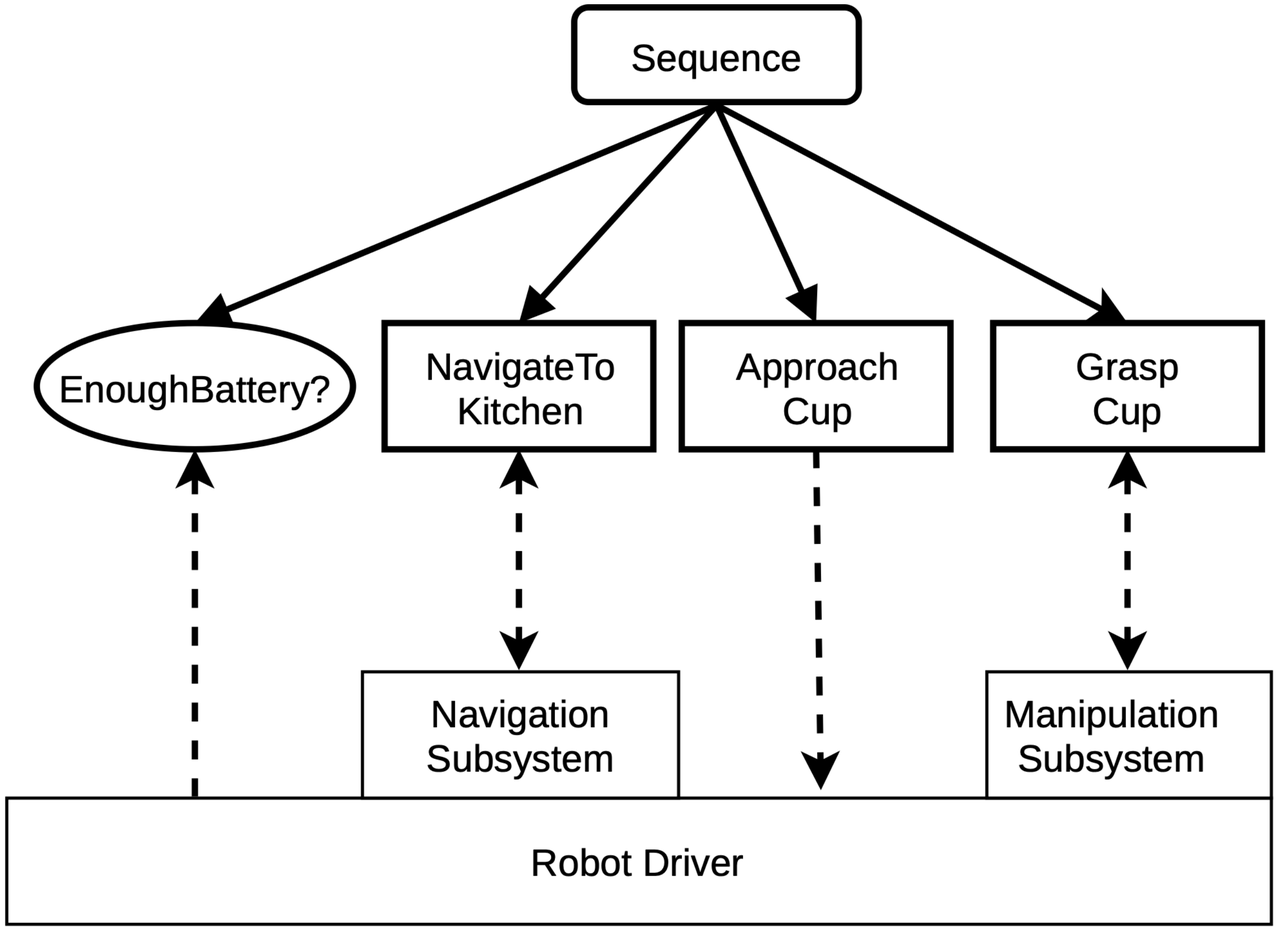
Behavior Trees



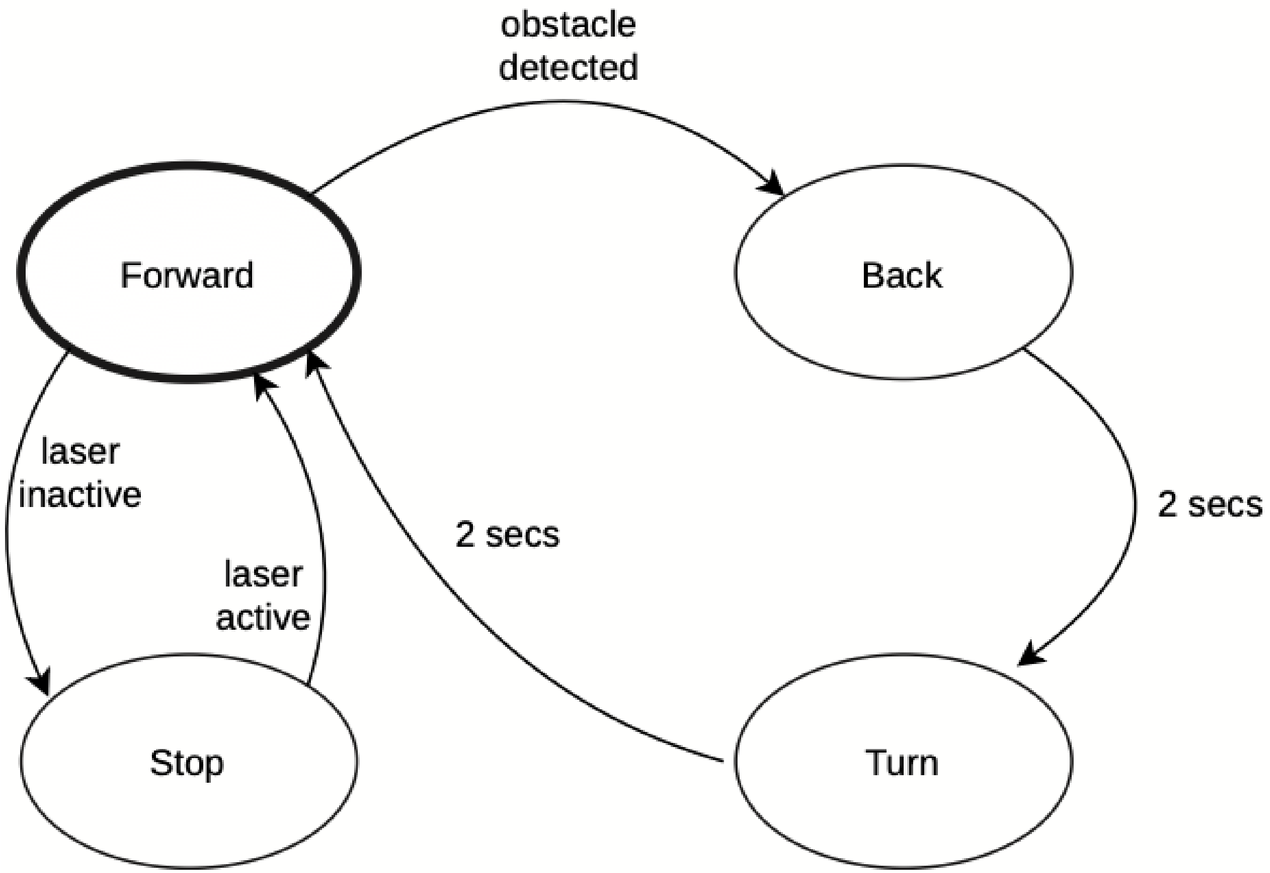



PlanSys
Root
SequenceStar
SequenceStar
SubTree
WaitForTask
Sequence
Dummy
SetBlackboard
key sourceID
value 1
SetBlackboard
key destID
value 2
SubTree
LoadBoxFromStation
SequenceStar
SequenceStar
A: moverobot
location ${sourceID}
A: orientaterobot
location ${sourceID}
A: mps_station_dock
stationid ${sourceID}
beltid 1
A: mps_station_load
stationid ${sourceID}
A: mps_station_undock
SubTree
UnloadBoxToStation
SequenceStar
SequenceStar
A: moverobot
location ${destID}
A: orientaterobot
location ${destID}
A: mps_station_dock
stationid 1
beltid 2
A: mps_station_unload
stationid ${destID}
A: mps_station_undock

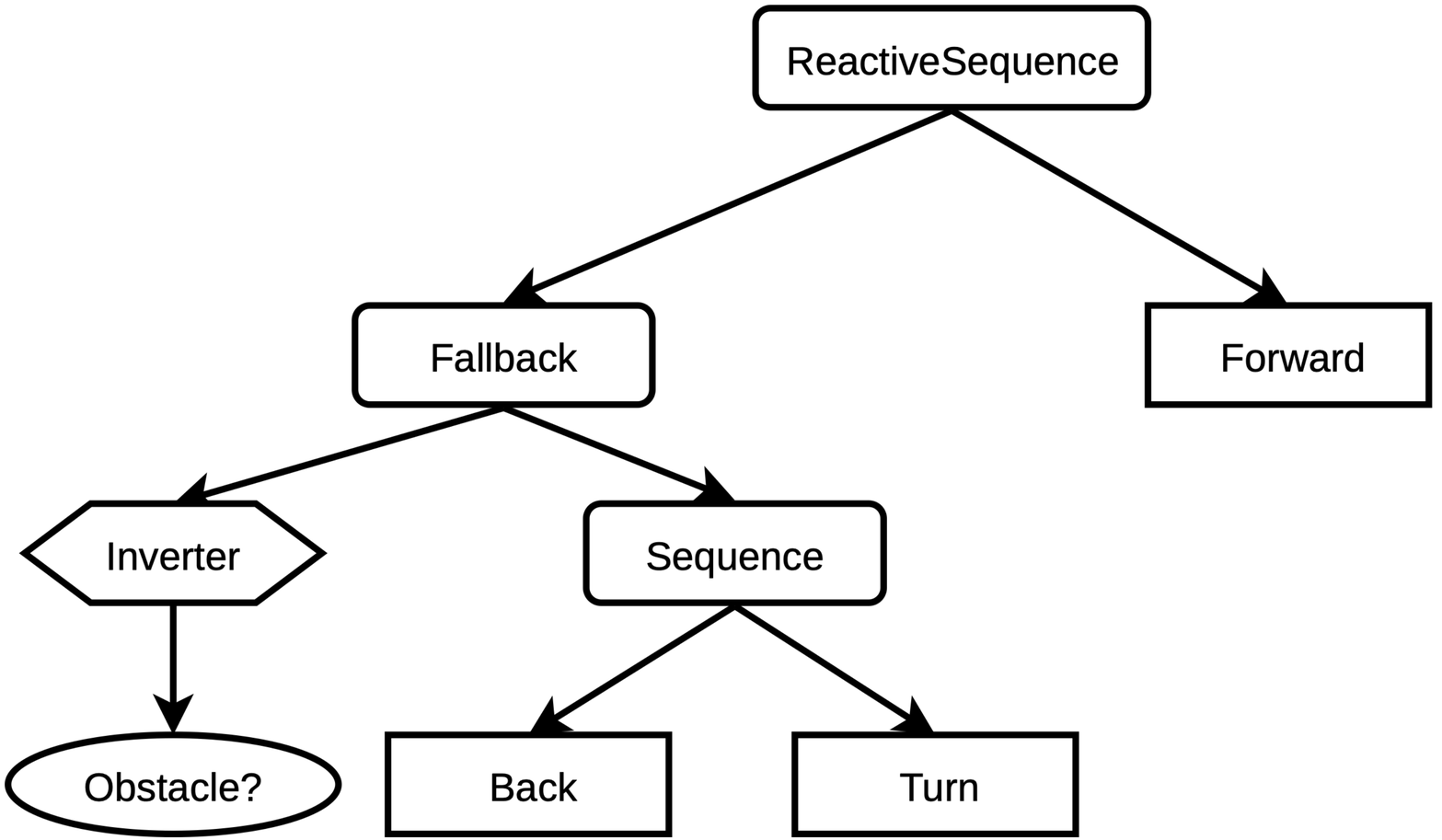
Behavior Trees



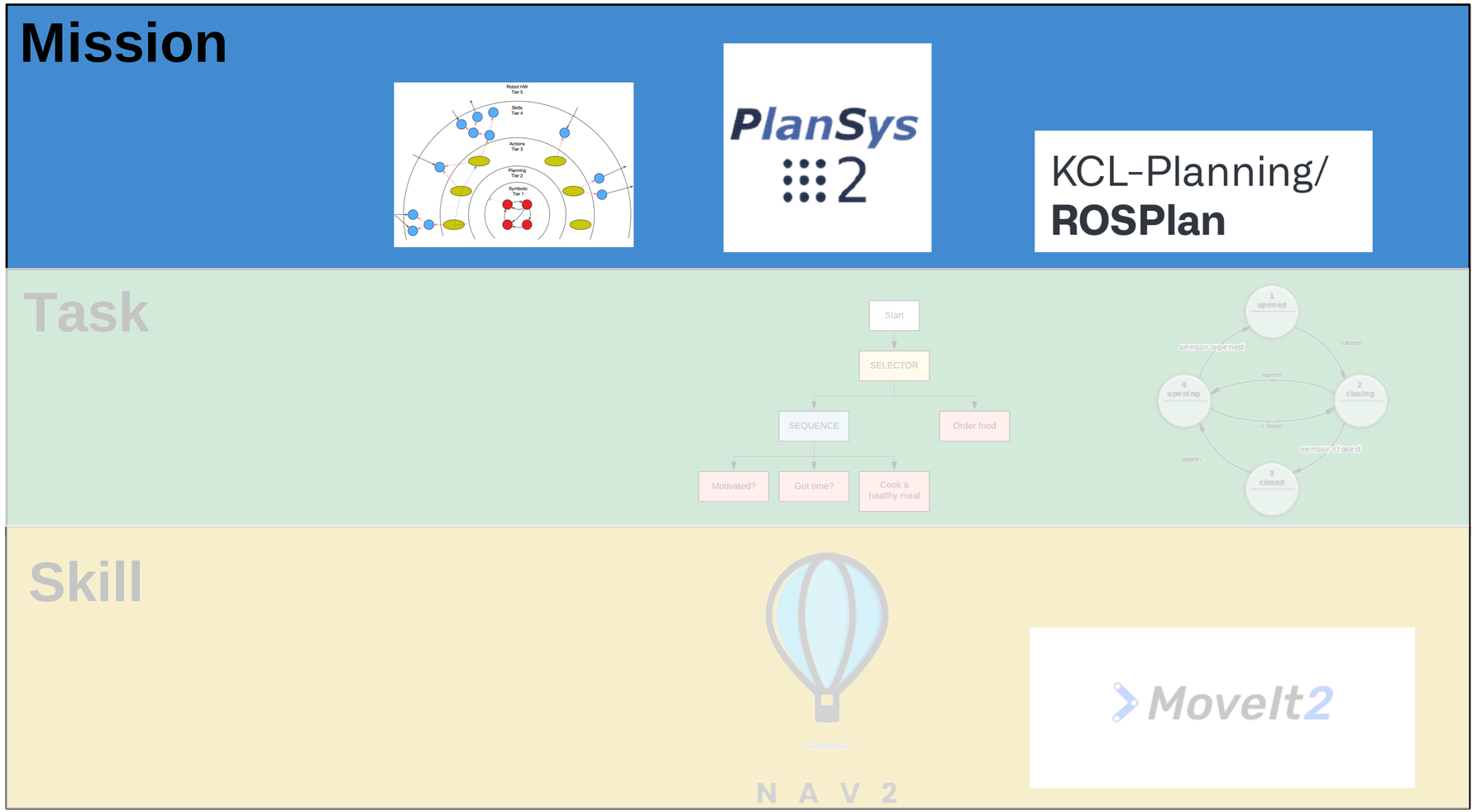
Behavior Trees



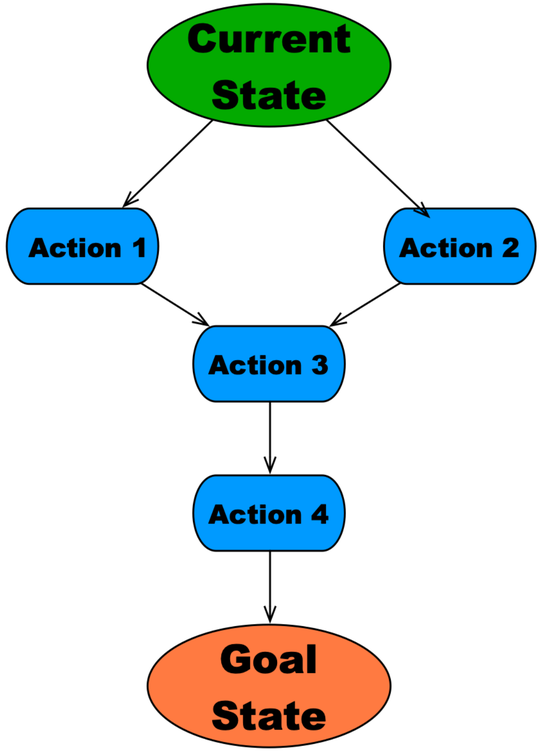
Behavior Trees



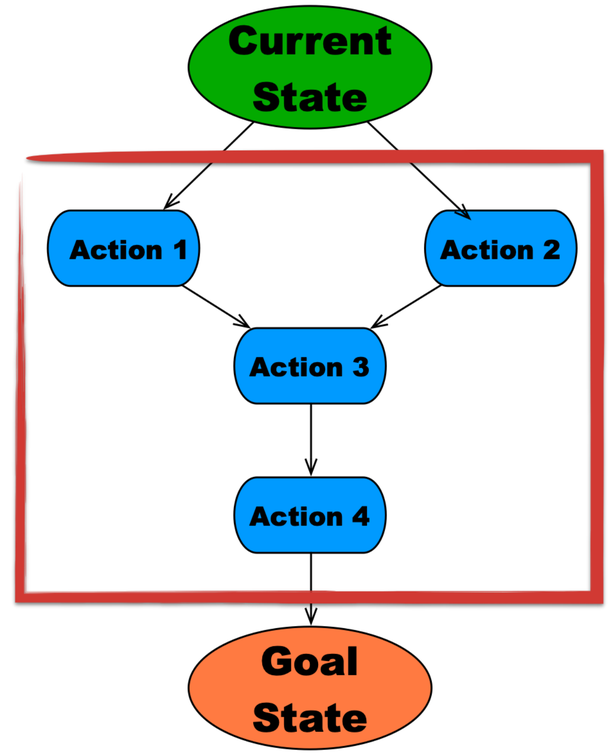
Part II


Mission
Robot HW
Tier 5
Skills
Tier 4
Actions
Tier 3
Planning
Tier 2
Symbolic
Tier 1
PlanSys2
KCL-Planning/
ROSPlan
Task
Start
SELECTOR
SEQUENCE
Order food
Motivated?
Got time?
Cook a healthy meal
1 opened
2 closing
3 closed
4 opening
sensor_opened
sensor_closed
Skill
NAV2
MoveIt2

AI Planning



AI Planning



This is automatically calculated!!!

AI Planning

- Planning Domain Definition Language (PDDL)
- Establish the rules (Domain)
- Present a situation and a goal (Problem)
- Use a plan solver
- Get a plan

```
(define (domain simple)
(:types robot room)
(:predicates
  (robot_at ?r - robot ?ro - room)
  (connected ?ro1 ?ro2 - room))
(:durative-action move
  :parameters (?r - robot ?r1 ?r2 - room)
  :duration ( = ?duration 5)
  :condition (and
    (at start(connected ?r1 ?r2))
    (at start(robot_at ?r ?r1)))
  :effect (and
    (at start(not(robot_at ?r ?r1)))
    (at end(robot_at ?r ?r2))))
)
```

Domain.pddl

Intelligent Robotics Lab

AI Planning

- Planning Domain Definition Language (PDDL)
- Establish the rules (Domain)
- Present a situation and a goal (Problem)
- Use a plan solver
- Get a plan

```
(define (domain simple)
(:types robot room)
(:predicates
  (robot_at ?r - robot ?ro - room)
  (connected ?ro1 ?ro2 - room))
(:durative-action move
  :parameters (?r - robot ?r1 ?r2 - room)
  :duration ( = ?duration 5)
  :condition (and
    (at start(connected ?r1 ?r2))
    (at start(robot_at ?r ?r1)))
  :effect (and
    (at start(not(robot_at ?r ?r1)))
    (at end(robot_at ?r ?r2))))
)
```

Domain.pddl

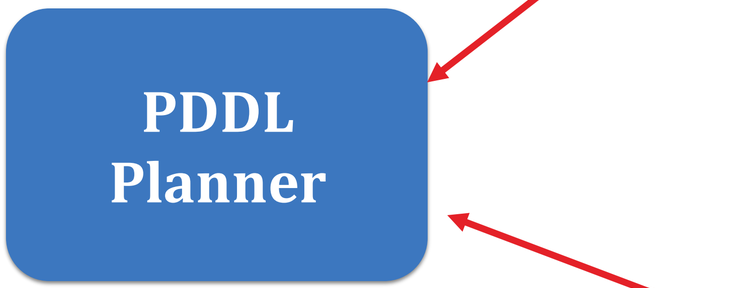
```
(define (problem problem_1)
(:domain simple)
(:objects
  r2d2 - robot
  bedroom living kitchen - room
)
(:init
  (robot_at r2d2 bedroom)
  (connected living bedroom)
  (connected bedroom living)
  (connected living kitchen)
  (connected kitchen living))
(:goal (and(robot_at r2d2 kitchen)))
)
```

Problem1.pddl

Intelligent Robotics Lab

AI Planning

- Planning Domain Definition Language (PDDL)
- Establish the rules (Domain)
- Present a situation and a goal (Problem)
- Use a plan solver
- Get a plan




```
(define (domain simple)
(:types robot room)
(:predicates
  (robot_at ?r - robot ?ro - room)
  (connected ?ro1 ?ro2 - room))
(:durative-action move
  :parameters (?r - robot ?r1 ?r2 - room)
  :duration ( = ?duration 5)
  :condition (and
    (at start(connected ?r1 ?r2))
    (at start(robot_at ?r ?r1)))
  :effect (and
    (at start(not(robot_at ?r ?r1)))
    (at end(robot_at ?r ?r2))))
)
```

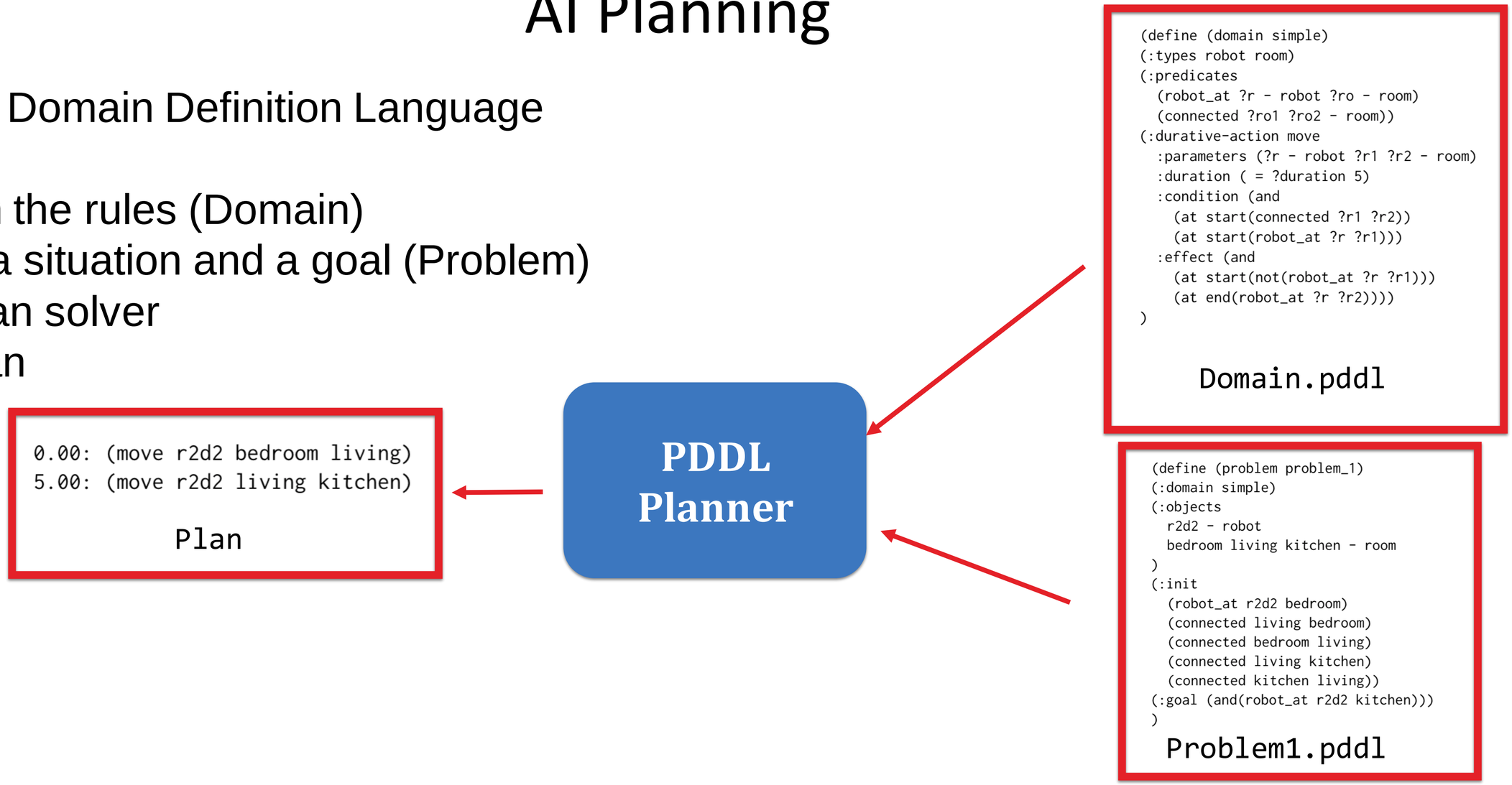
Domain.pddl

```
(define (problem problem_1)
(:domain simple)
(:objects
  r2d2 - robot
  bedroom living kitchen - room
)
(:init
  (robot_at r2d2 bedroom)
  (connected living bedroom)
  (connected bedroom living)
  (connected living kitchen)
  (connected kitchen living))
(:goal (and(robot_at r2d2 kitchen)))
)
```

Problem1.pddl

AI Planning

- Planning Domain Definition Language (PDDL)
- Establish the rules (Domain)
- Present a situation and a goal (Problem)
- Use a plan solver
- Get a plan



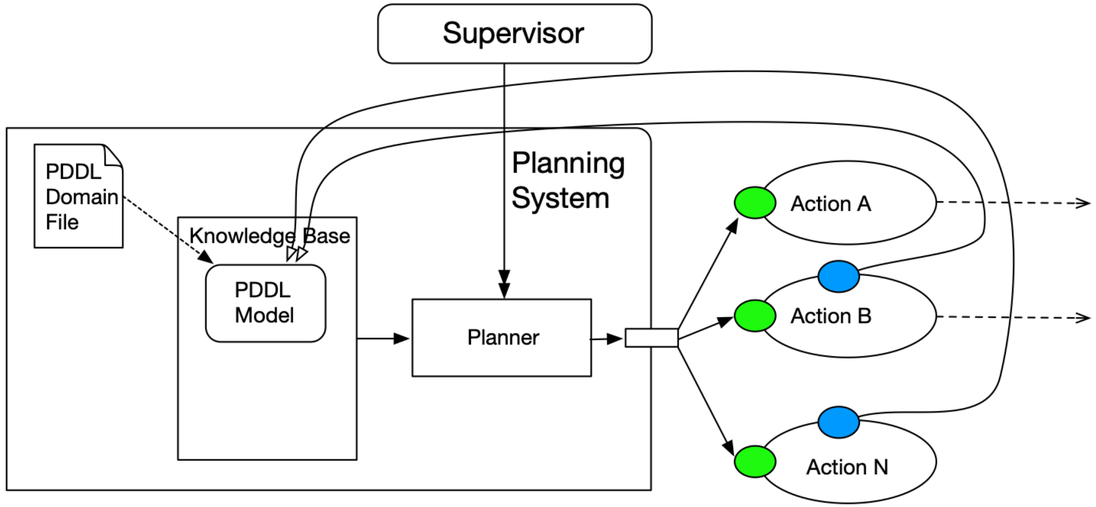
Why a Planning System?

- **To bring AI planning to real world!!**

1. Read a model
2. Add/remove/update/query the knowledge
3. Create a plan using a PDDL planner
4. Execute the plan
5. Implement the actions with real effect
6. Tools for monitor and control

ROSPlan

- ROSPlan is the inspiration of PlanSys2
- PDDL Planning reference package in ROS
- No plans for migrating to ROS2



KCL-Planning/
ROSPlan

The ROSPlan framework provides a generic method for task planning in a ROS system.

30 Contributors · 49 Issues · 239 Stars · 132 Forks

ROSPlan

Home | Documentation & Tutorials | Virtual Machine | Demos and Conferences | Publications | View on GitHub | Contact

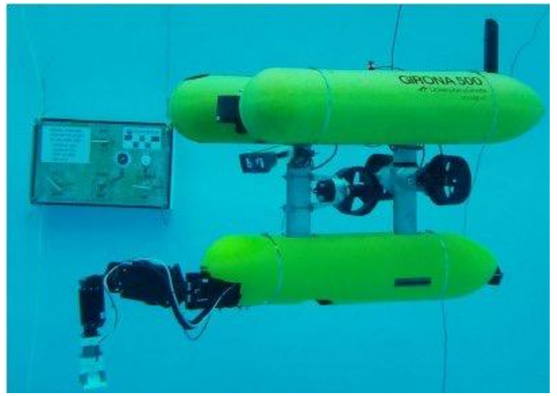
Please contact us if you have a link to add.

Conference Tutorials
2017 AAAI Tutorial on AI Planning for Robotics
2020 AAAI Tutorial on AI Planning for Robotics with ROSPlan
2017 ICAPS Tutorial on AI Planning for Robotics and Human-Robot Interaction
2017 ICRA Tutorial on Planning and Robotics
ICAPS 2018 Tutorial on Integrating Classical Planning and Mobile Service Robots using ROSPlan

Demonstrations
Demonstration with Blocks
Nessie AUV Inspection task
Girona 500 AUV Valve Turning
SQUIRREL Demo

Other Links
Interface for Turtlebot2
Interface for COLA2
Interface for hector_quadrotor

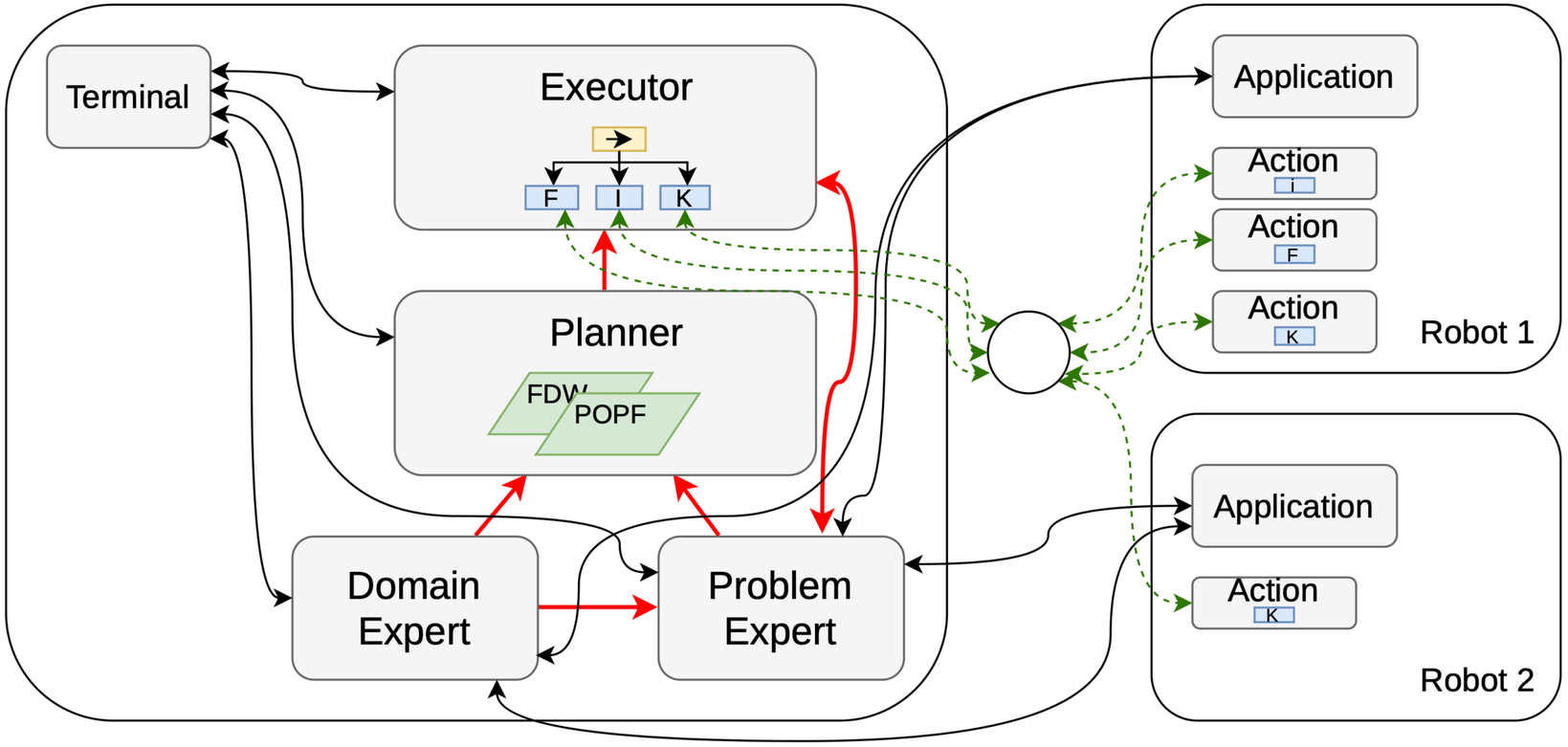
2020 AAAI Tutorial on AI Planning for Robotics with ROSPlan



Intelligent Robotics Lab

2 Planning System

- Framework for AI Planning
- Inspired en ROSPlan
  - Easier interface
  - More reactive
  - Controlled execution
- Modular design
- For ROS 2
- New Features
  - PlanSys2 Terminal
  - Composed domain
  - Multirobot
  - Parallel execution
  - Load balancing



2 Planning System

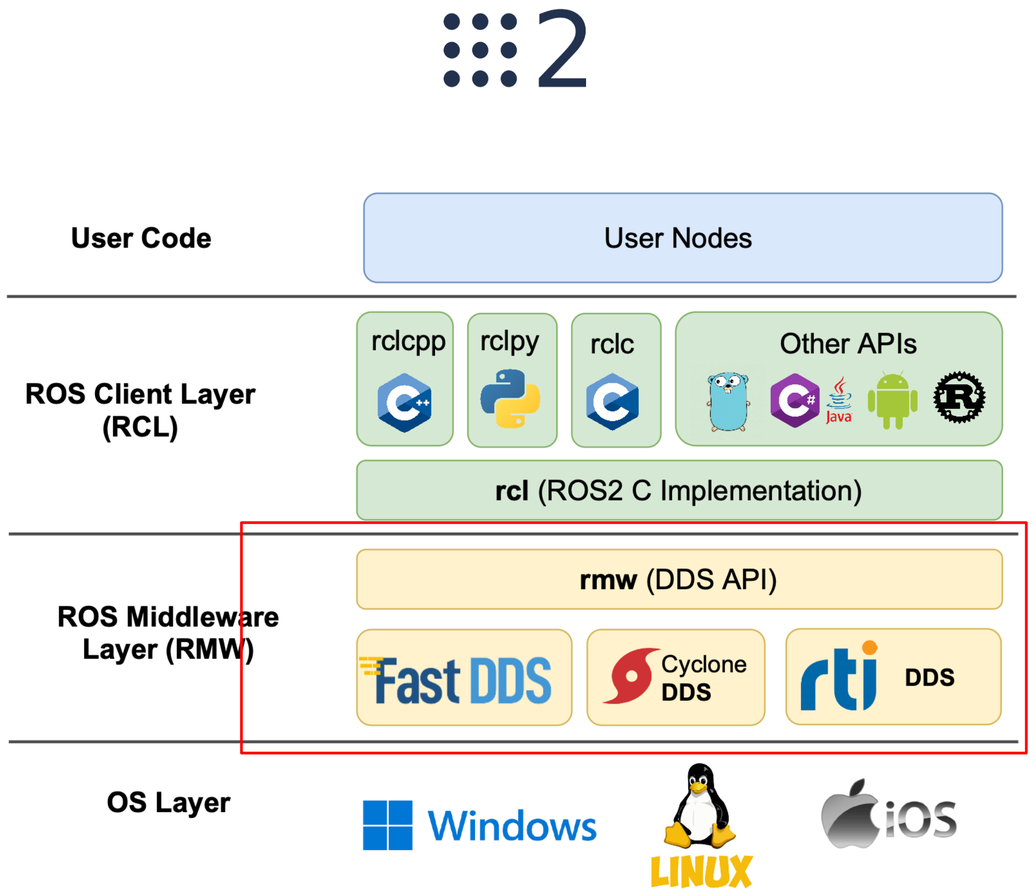
- Framework for AI Planning
- Inspired en ROSPlan
  - Easier interface
  - More reactive
  - Controlled execution
- Modular design
- For ROS 2
- New Features
  - PlanSys2 Terminal
  - Composed domain
  - Multirobot
  - Parallel execution
  - Load balancing

```
fmrico@localhost ~$ ros2 run plansys2_terminal terminal
PlanSys2 Terminal
> get domain
 …
> get problem
 …
> get problem predicates
 …
> set instance r2d2 robot
> set instance fmrico person
> set predicate (robot_at r2d2 kitchen)
> set goal (robot_at r2d2 bedroom)
> get plan
 …
> run
  [(move r2d2 kitchen corridor) succeded]
  [(move r2d2 corridor bedroom) 57 %]
```

Intelligent Robotics Lab

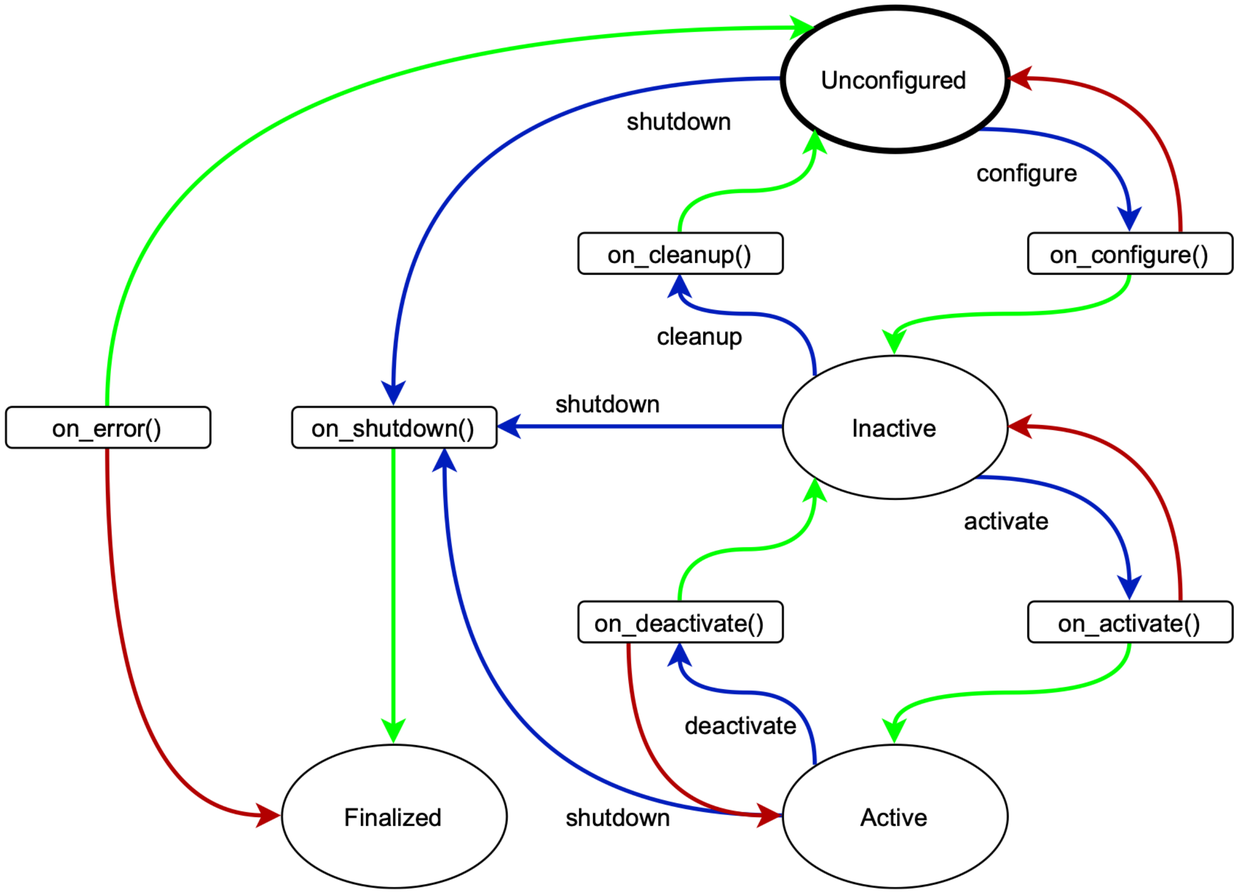
2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info



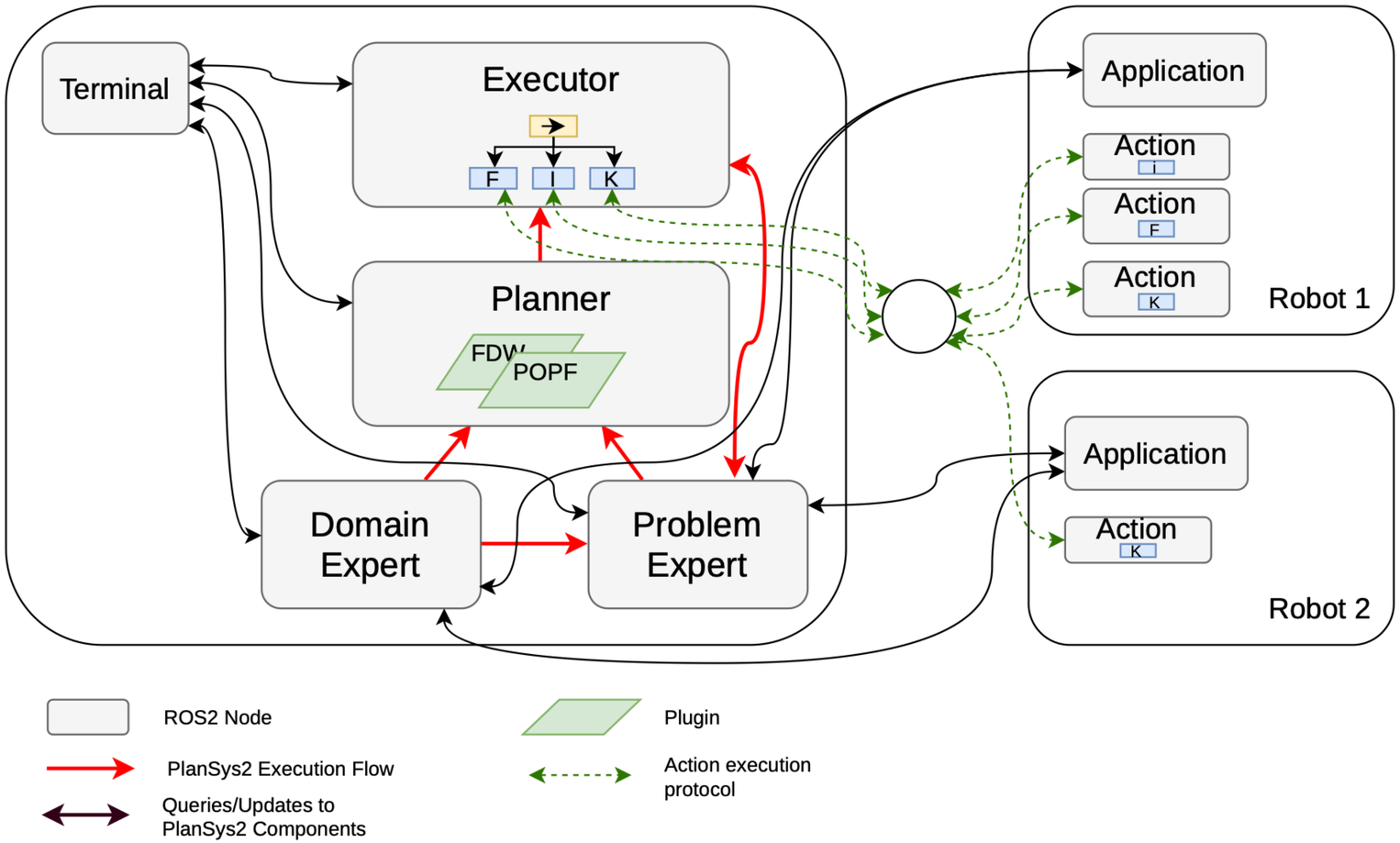
2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info



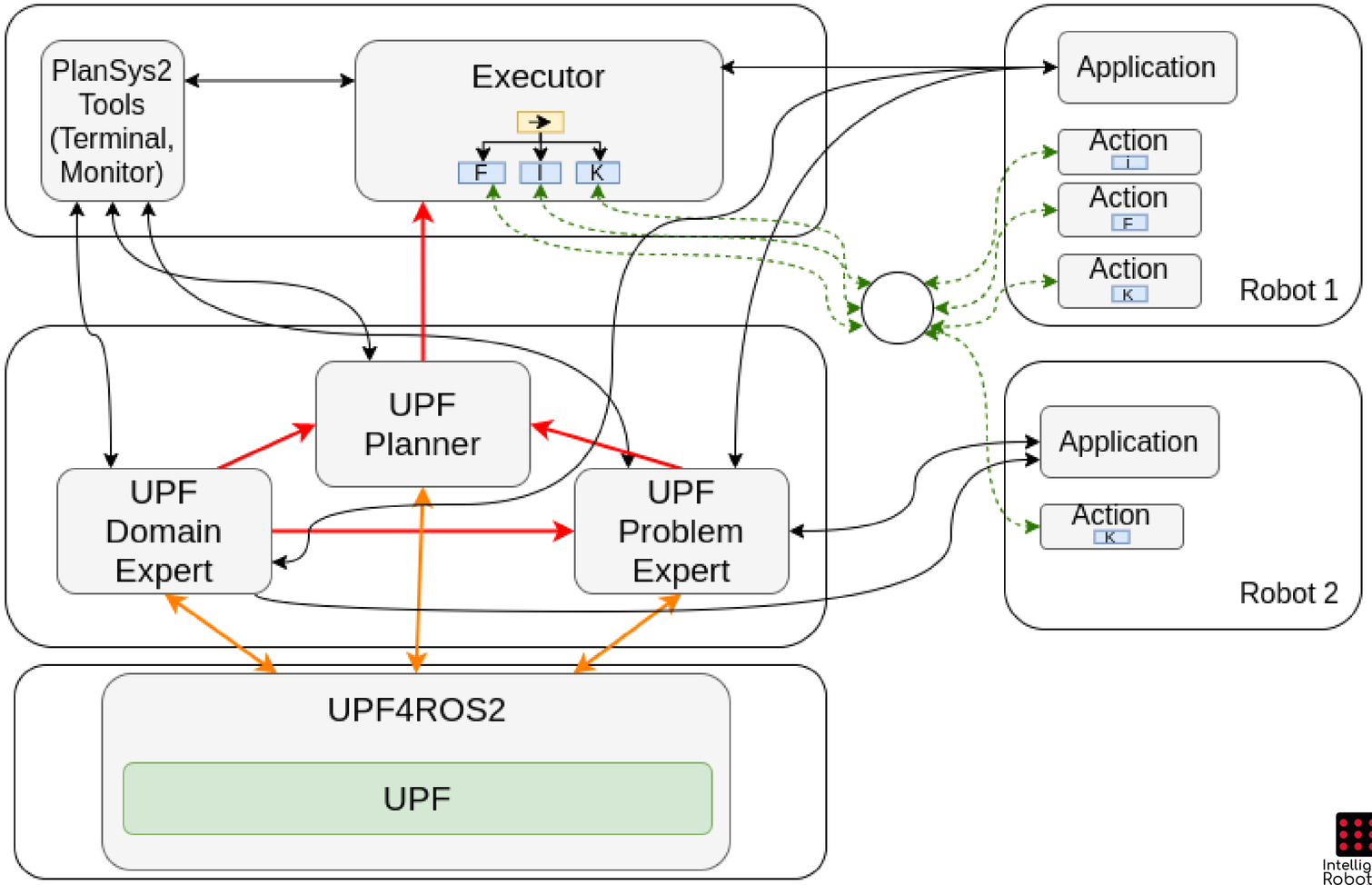
2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info



2 Planning System

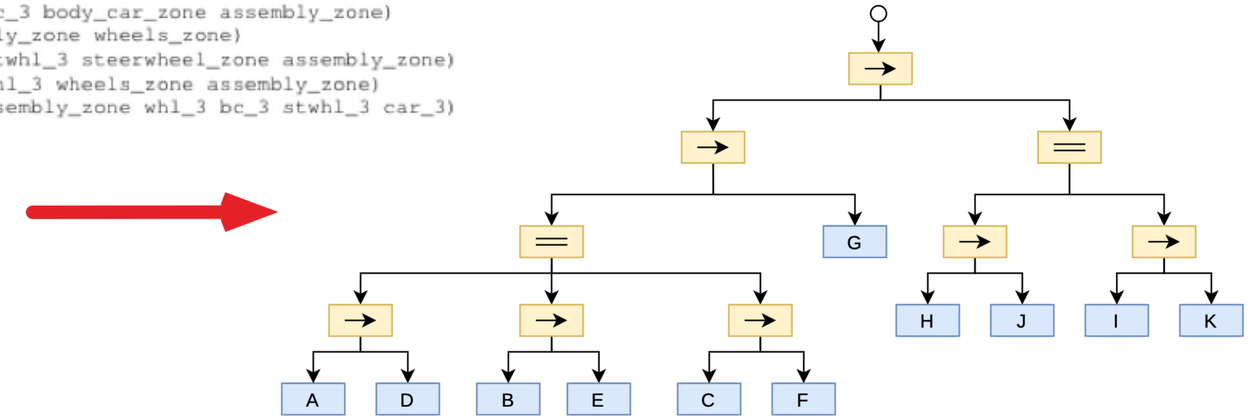
- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info



2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info

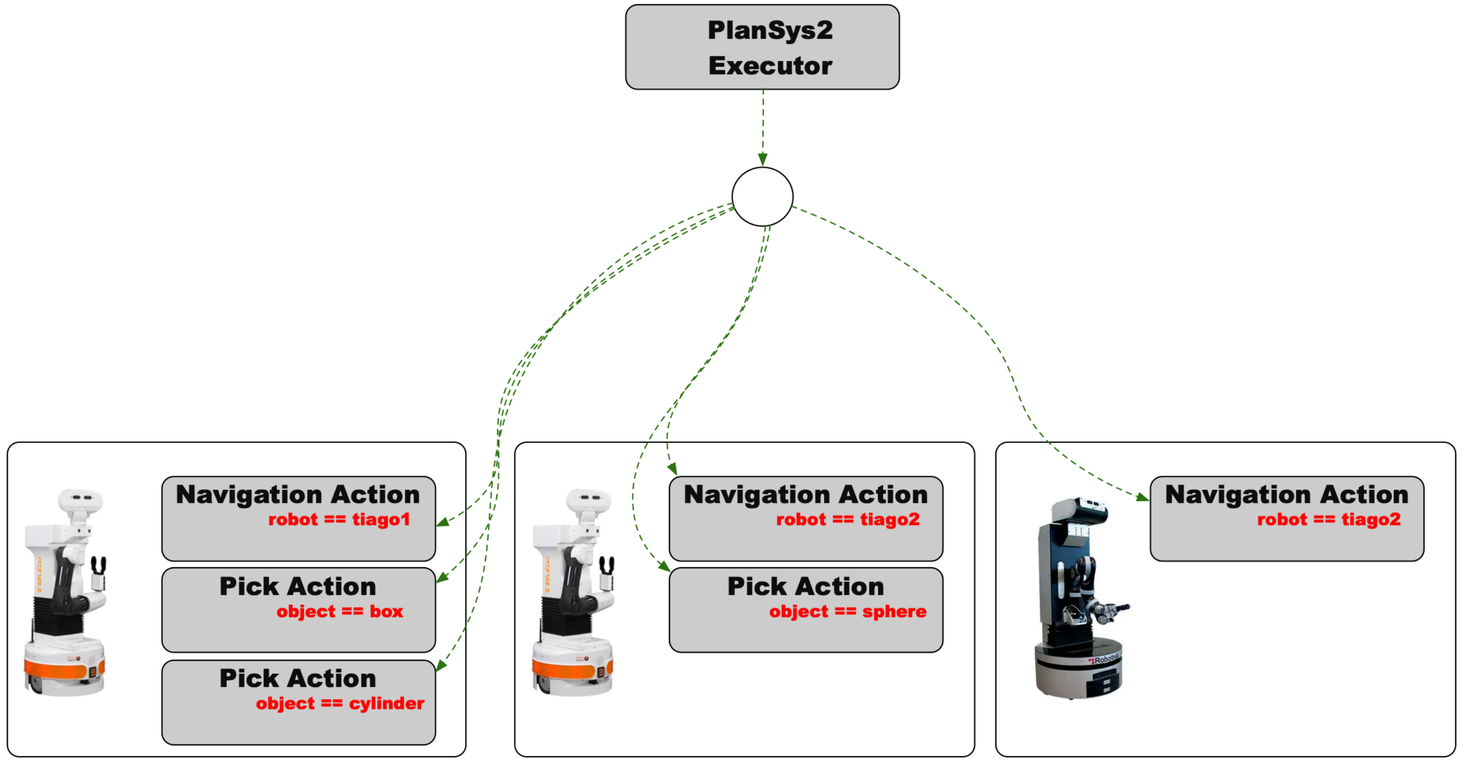
```
0       (move rb1 assembly_zone body_car_zone)
0       (move rb2 assembly_zone steerwheel_zone)
0       (move rb3 assembly_zone wheels_zone)
5.001   (transport rb1 bc_1 body_car_zone assembly_zone)
5.001   (transport rb2 stwhl_1 steerwheel_zone assembly_zone)
5.001   (transport rb3 whl_1 wheels_zone assembly_zone)
10.002  (assemble rb1 assembly_zone whl_1 bc_1 stwhl_1 car_1)
10.002  (move rb2 assembly_zone body_car_zone)
10.002  (move rb3 assembly_zone steerwheel_zone)
15.003  (move rb1 assembly_zone wheels_zone)
15.003  (transport rb2 bc_2 body_car_zone assembly_zone)
15.003  (transport rb3 stwhl_2 steerwheel_zone assembly_zone)
20.004  (transport rb1 whl_2 wheels_zone assembly_zone)
20.004  (move rb3 assembly_zone body_car_zone)
25.005  (assemble rb2 assembly_zone whl_2 bc_2 stwhl_2 car_2)
25.005  (move rb1 assembly_zone steerwheel_zone)
25.005  (transport rb3 bc_3 body_car_zone assembly_zone)
30.006  (move rb2 assembly_zone wheels_zone)
30.006  (transport rb1 stwhl_3 steerwheel_zone assembly_zone)
35.007  (transport rb2 whl_3 wheels_zone assembly_zone)
40.008  (assemble rb1 assembly_zone whl_3 bc_3 stwhl_3 car_3)
```



Intelligent Robotics Lab

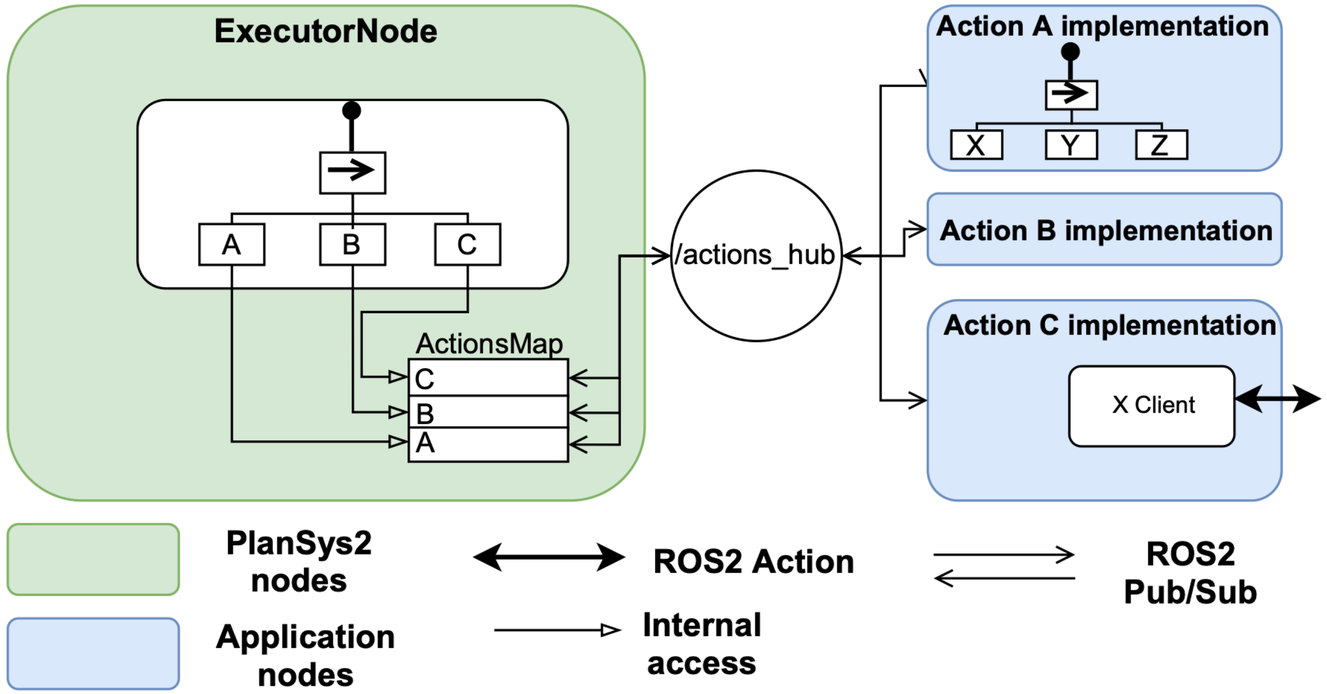
2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info



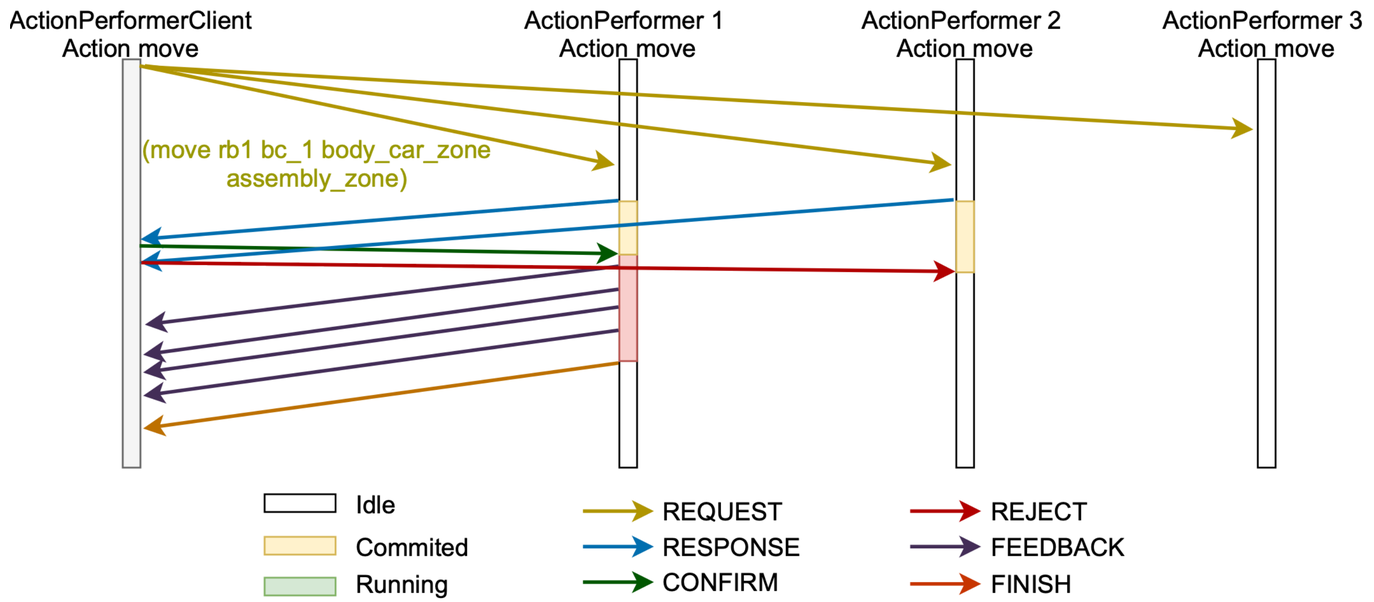
2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info



2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info



2 Planning System

- **Reliable, Secure and Predictable**
  - Real-time communications with DDS
  - Lifecycle nodes
- **Modular and Extensible**
  - PDDL Planners as plugins
  - Different modules
  - Plans as BTs
- **Multirobot**
  - Support for multi-robot
  - Specialized Action Performers
- **Efficient and Accountable**
  - Parallel execution of actions
  - Lot of debug info

`/action_execution_info` [`plansys2_msgs/ActionExecutionInfo.msg`] This topic is dedicated to get the feedback from the action execution (start, and status time, completion, messages, arguments...).

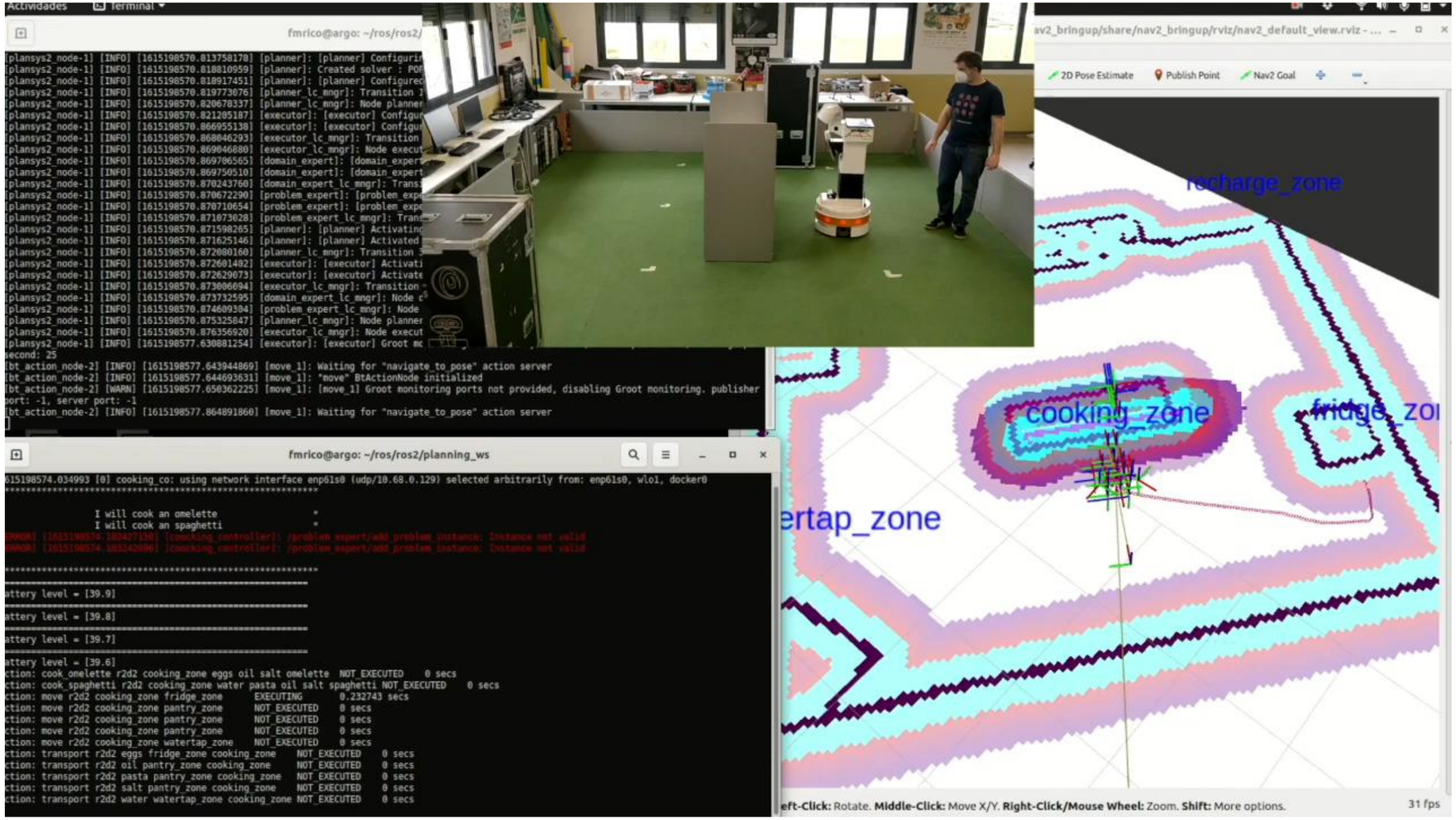
`/performers_status` [`plansys2_msgs/ActionPerformerStatus.msg`] This topic receives a heartbeat from the action performers, with their metainfo.

`/actions_hub` [`plansys2_msgs/ActionExecution.msg`] This topic is used to implement the protocol to deliver actions to the action performers. It includes feedback on the completion percentage and the status.

Intelligent Robotics Lab


Actividades
Terminal
jue 31 de mar 09:32
95 %
fmrico@argo: ~/ros/ros2/planning_ws
fmrico@argo:~/ros/ros2/planning_ws$ ros2 launch plansys2_patrol_navigation_example patrol_example_launch.py
fmrico@argo: ~/ros/ros2/planning_ws
fmrico@argo:~/ros/ros2/planning_ws$ ros2 run plansys2_patrol_navigation_example patrolling_controller_node
fmrico@argo: ~/ros/ros2/planning_ws
fmrico@argo:~/ros/ros2/planning_ws$ rqt_gui
fmrico@argo: ~/ros/ros2/planning_ws
fmrico@argo:~/ros/ros2/planning_ws$ ros2 run plansys2_terminal plansys2_terminal
Do{
live_life(♥);
}while(1==1);
book_ros2-source-code.zip

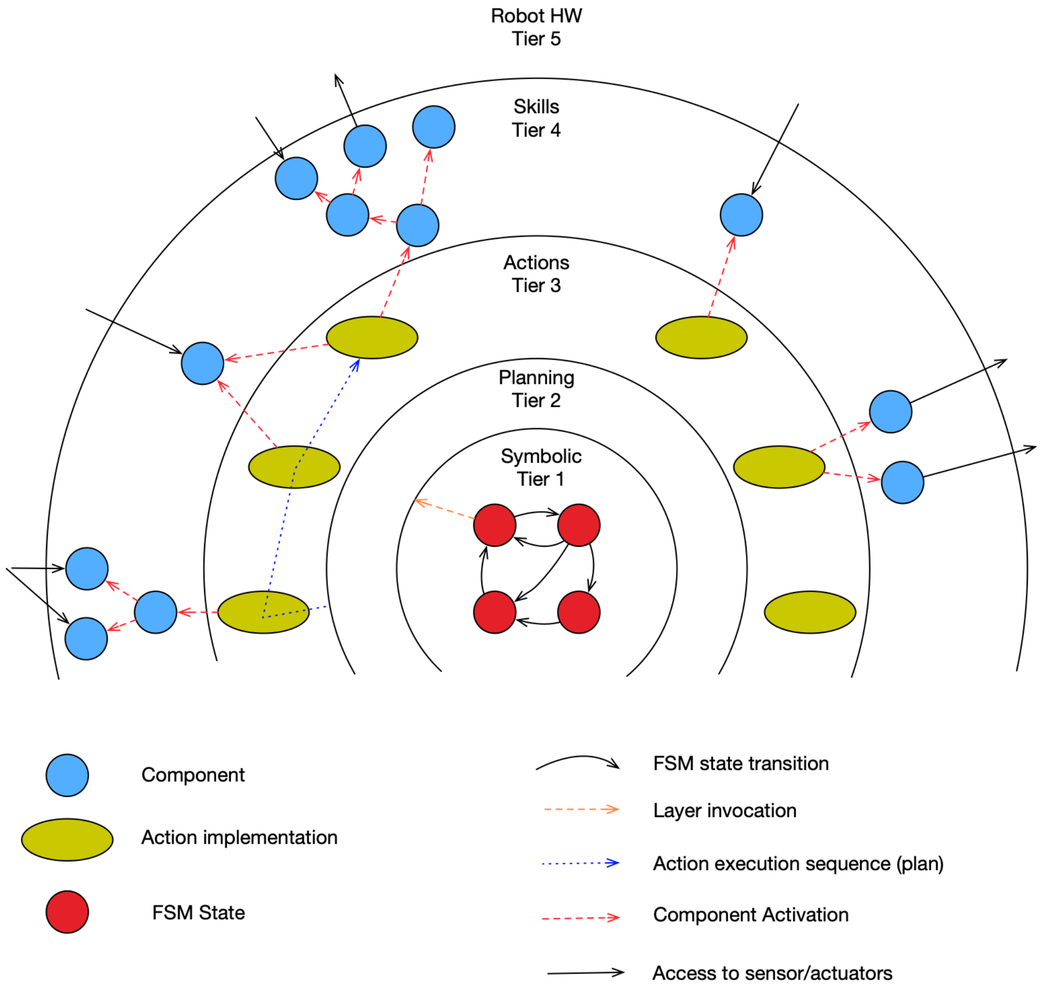
2 Planning System



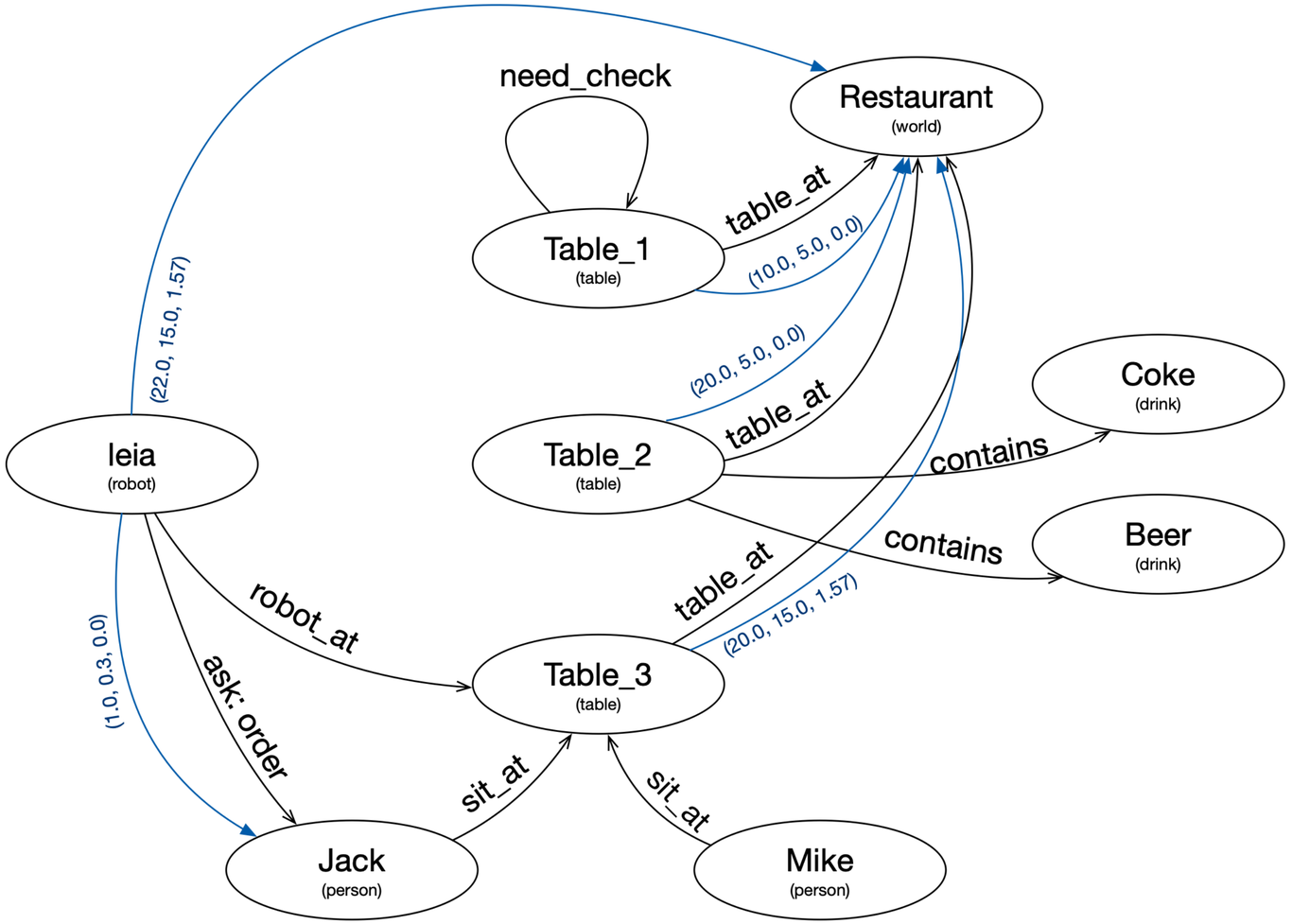
https://github.com/Docencia-fmrico/plansys2-gpsr-nocompila

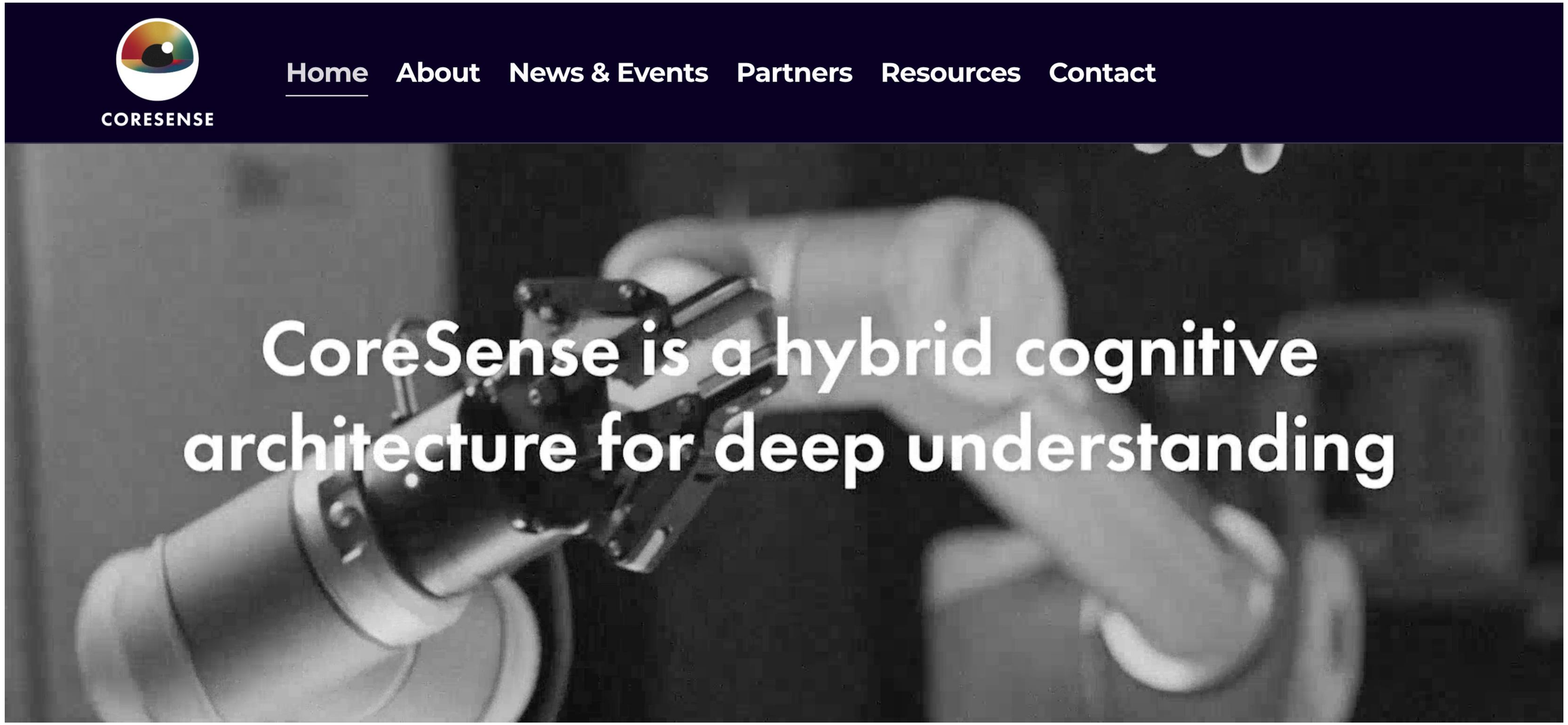
Cognitive Architectures

Cognitive Architectures



Cognitive Architectures





The **CORESENSE project** has received funding from the European Union's **Horizon Europe** Research and innovation programme under gran agreement No. **101070254**.

Funded by the European Union

https://coresense.eu/