some research topics in software engineering

Wardah Mahmood, Mukelabai Mukelabai, Patrick Franz, Daniel Strüber, Thorsten Berger

RUHR
UNIVERSITÄT
BOCHUM
RUB

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

EASElab (Empirical and Automated Software Eng. Lab)

We automate **software engineering** for the next generation of **intelligent, autonomous, and variant-rich** software systems. We explore new ways of **software creation, analysis, and evolution**. Our application domains are **automotive systems, systems software (e.g., Linux kernel), software ecosystems (e.g. Android apps), and mobile robots**.

http://www.easelab.org

Focus areas

- **Model-Driven Software Engineering (MDSE)**
- Software Analytics (SWA)
- AI Engineering (SE4AI/AI4SE)
- Software Product Line Engineering (SPLE)

RUHR UNIVERSITÄT BOCHUM RUB

CHALMERS UNIVERSITY OF TECHNOLOGY

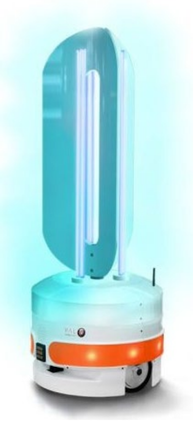
UNIVERSITY OF GOTHENBURG

ANDROID

ABB

TIAGO
TIAGO@HOME

TIAGo++



SCANIA
SCANIA
SCANIA
HIGHLINE

agenda

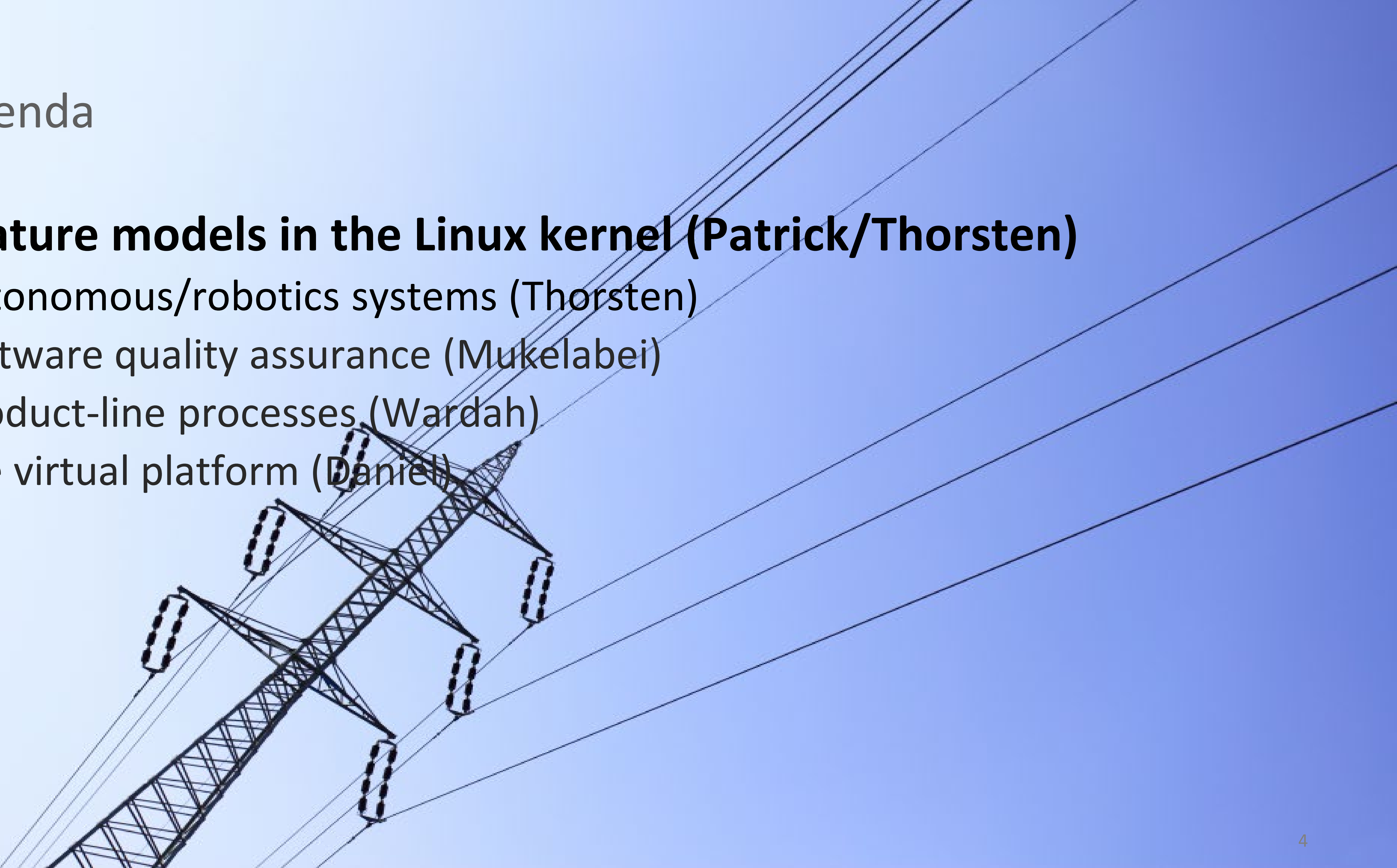
feature models in the Linux kernel (Patrick/Thorsten)

autonomous/robotics systems (Thorsten)

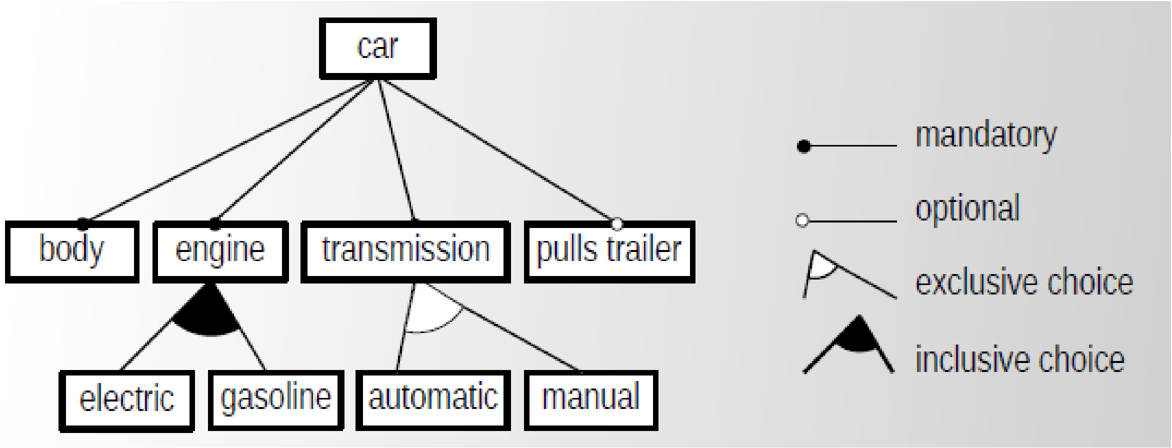
software quality assurance (Mukelabei)

product-line processes (Wardah)

the virtual platform (Daniel)



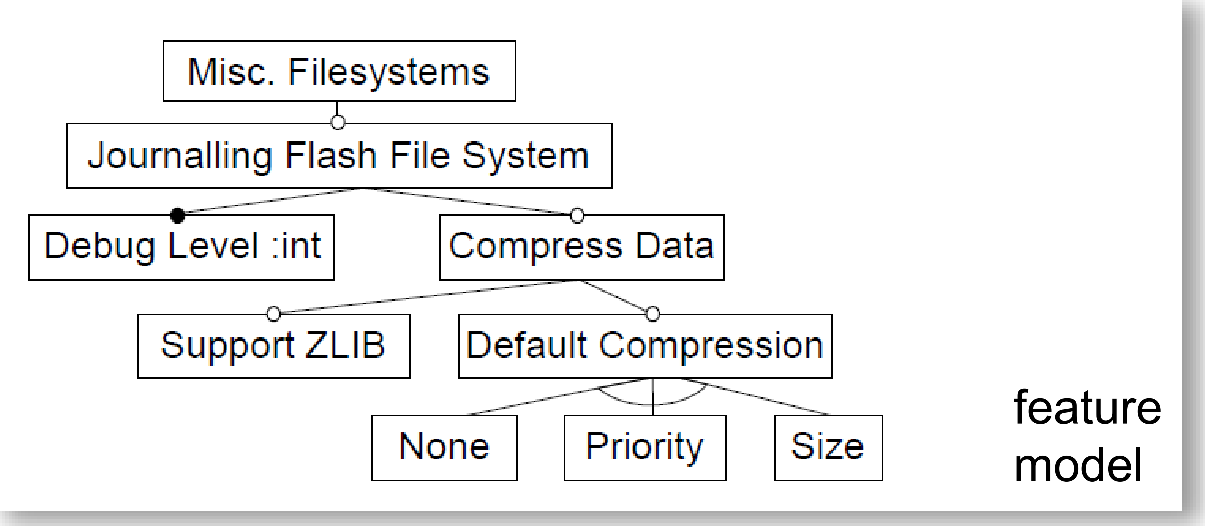
feature modeling

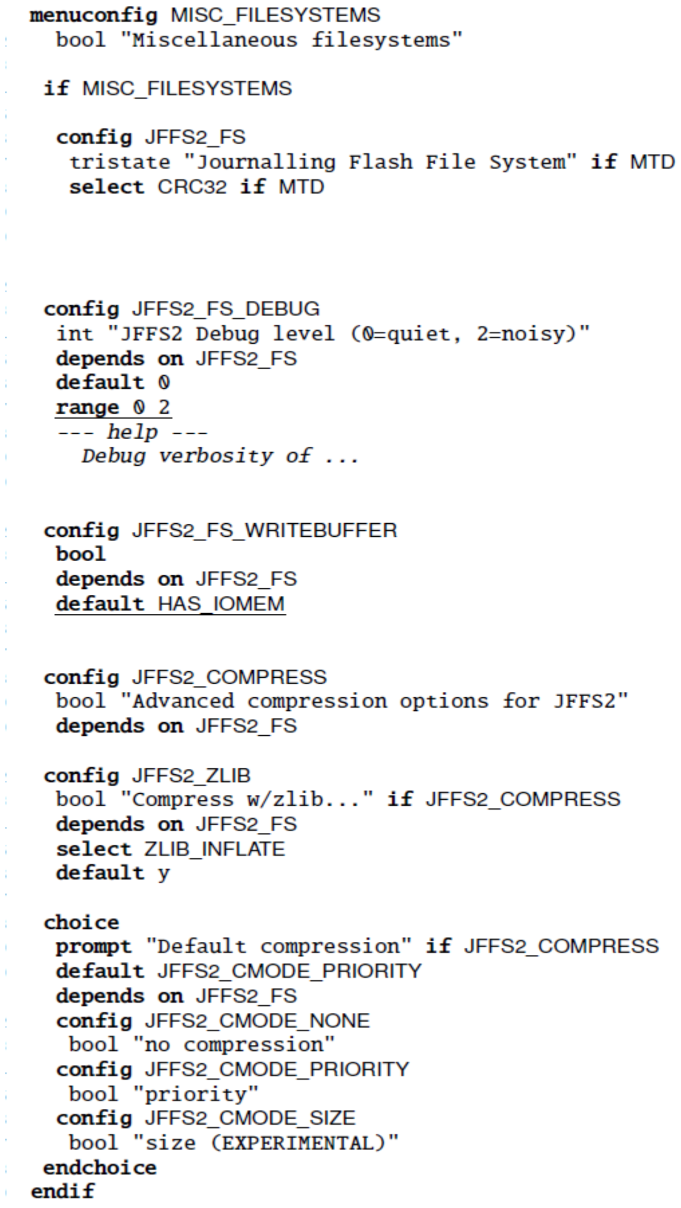


electric *requires* automatic

avoid illegal variants

feature models in the Linux kernel



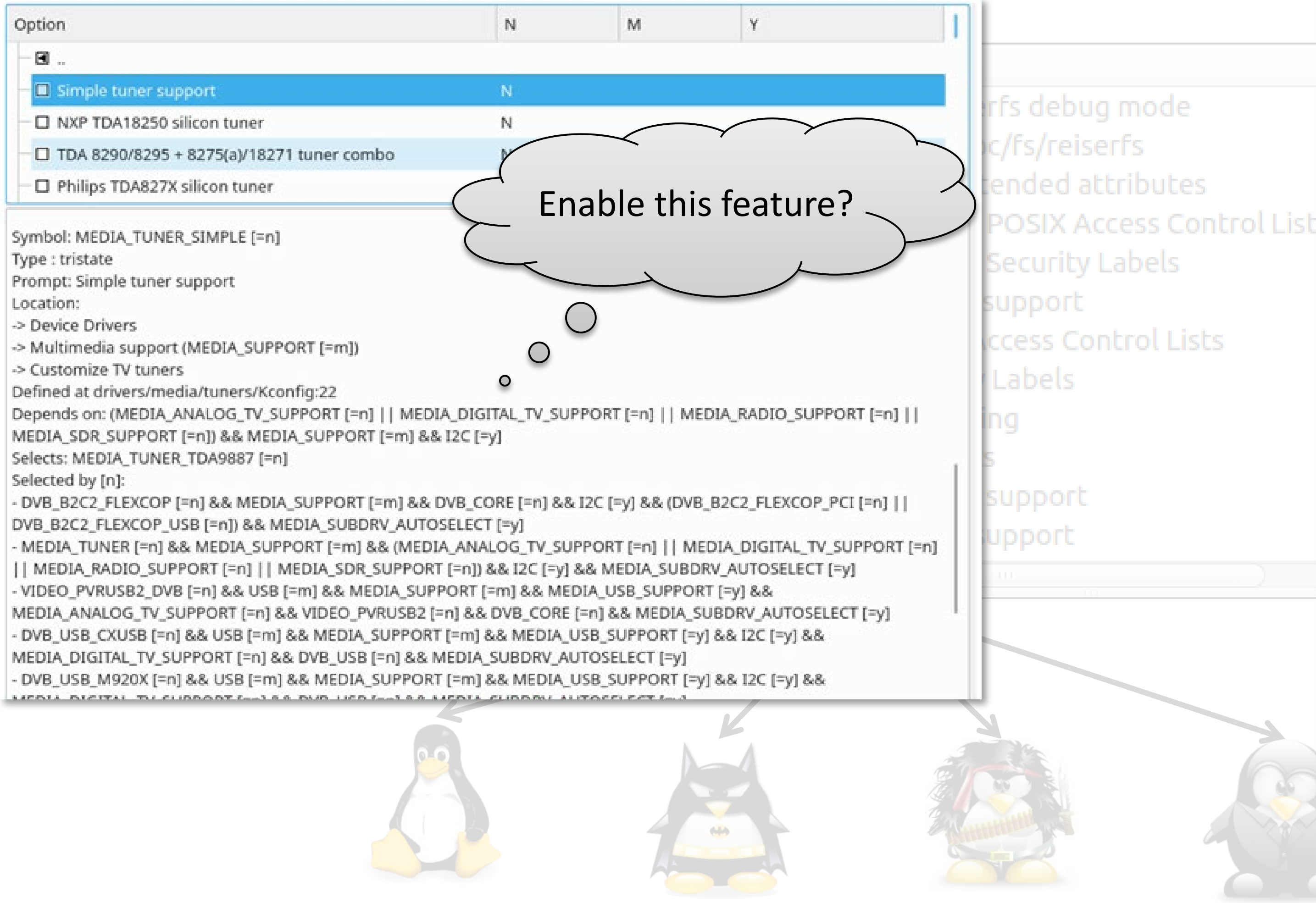


```
menuconfig MISC_FILESYSTEMS
  bool "Miscellaneous filesystems"

 if MISC_FILESYSTEMS

  config JFFS2_FS
   tristate "Journalling Flash File System" if MTD
   select CRC32 if MTD



 config JFFS2_FS_DEBUG
  int "JFFS2 Debug level (0=quiet, 2=noisy)"
  depends on JFFS2_FS
  default 0
  range 0 2
  --- help ---
    Debug verbosity of ...


 config JFFS2_FS_WRITEBUFFER
  bool
  depends on JFFS2_FS
  default HAS_IOMEM


 config JFFS2_COMPRESS
  bool "Advanced compression options for JFFS2"
  depends on JFFS2_FS

 config JFFS2_ZLIB
  bool "Compress w/zlib..." if JFFS2_COMPRESS
  depends on JFFS2_FS
  select ZLIB_INFLATE
  default y

 choice
  prompt "Default compression" if JFFS2_COMPRESS
  default JFFS2_CMODE_PRIORITY
  depends on JFFS2_FS
  config JFFS2_CMODE_NONE
   bool "no compression"
  config JFFS2_CMODE_PRIORITY
   bool "priority"
  config JFFS2_CMODE_SIZE
   bool "size (EXPERIMENTAL)"
 endchoice
endif
```

Berger, She, Lotufo, Wasowski, Czarnecki. A study of variability models and languages in the systems software domain. *IEEE Transactions on Software Engineering*, 2013

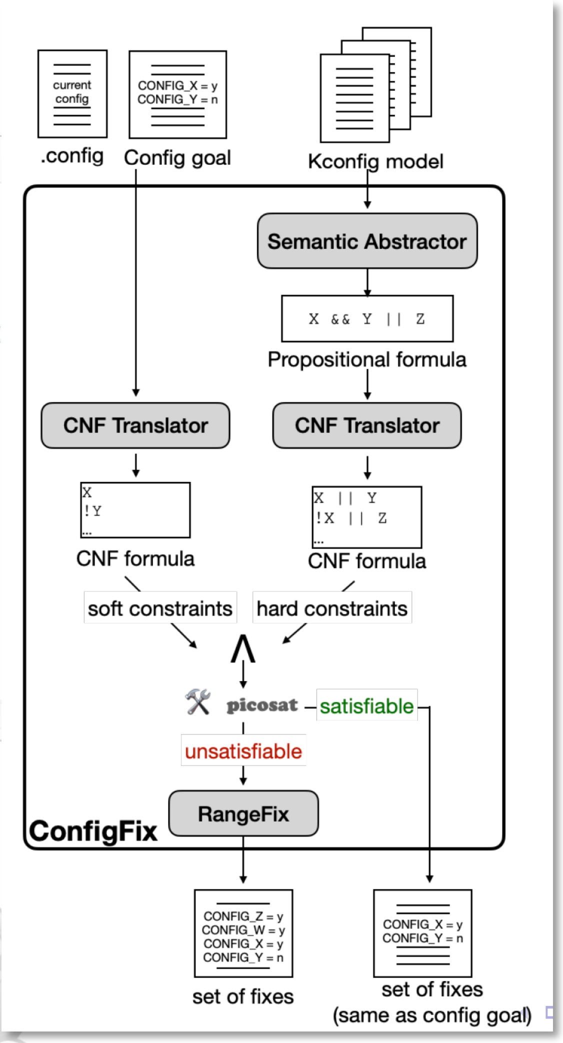
Franz, Berger, Fayaz, Nadi, Groshev. ConfigFix: Interactive Configuration Conflict Resolution for the Linux Kernel,” *43rd International Conference on Software Engineering.* 2021.

Linux kernel configurator (xconfig)



Option	N	M	Y
..			
□ Simple tuner support	N		
□ NXP TDA18250 silicon tuner	N		
□ TDA 8290/8295 + 8275(a)/18271 tuner combo	N		
□ Philips TDA827X silicon tuner			

```
Symbol: MEDIA_TUNER_SIMPLE [=n]
Type : tristate
Prompt: Simple tuner support
Location:
-> Device Drivers
-> Multimedia support (MEDIA_SUPPORT [=m])
-> Customize TV tuners
Defined at drivers/media/tuners/Kconfig:22
Depends on: (MEDIA_ANALOG_TV_SUPPORT [=n] || MEDIA_DIGITAL_TV_SUPPORT [=n] || MEDIA_RADIO_SUPPORT [=n] ||
MEDIA_SDR_SUPPORT [=n]) && MEDIA_SUPPORT [=m] && I2C [=y]
Selects: MEDIA_TUNER_TDA9887 [=n]
Selected by [n]:
- DVB_B2C2_FLEXCOP [=n] && MEDIA_SUPPORT [=m] && DVB_CORE [=n] && I2C [=y] && (DVB_B2C2_FLEXCOP_PCI [=n] ||
DVB_B2C2_FLEXCOP_USB [=n]) && MEDIA_SUBDRV_AUTOSELECT [=y]
- MEDIA_TUNER [=n] && MEDIA_SUPPORT [=m] && (MEDIA_ANALOG_TV_SUPPORT [=n] || MEDIA_DIGITAL_TV_SUPPORT [=n]
|| MEDIA_RADIO_SUPPORT [=n] || MEDIA_SDR_SUPPORT [=n]) && I2C [=y] && MEDIA_SUBDRV_AUTOSELECT [=y]
- VIDEO_PVRUSB2_DVB [=n] && USB [=m] && MEDIA_SUPPORT [=m] && MEDIA_USB_SUPPORT [=y] &&
MEDIA_ANALOG_TV_SUPPORT [=n] && VIDEO_PVRUSB2 [=n] && DVB_CORE [=n] && MEDIA_SUBDRV_AUTOSELECT [=y]
- DVB_USB_CXUSB [=n] && USB [=m] && MEDIA_SUPPORT [=m] && MEDIA_USB_SUPPORT [=y] && I2C [=y] &&
MEDIA_DIGITAL_TV_SUPPORT [=n] && DVB_USB [=n] && MEDIA_SUBDRV_AUTOSELECT [=y]
- DVB_USB_M920X [=n] && USB [=m] && MEDIA_SUPPORT [=m] && MEDIA_USB_SUPPORT [=y] && I2C [=y] &&
```



Berger, She, Lotufo, Wasowski, Czarnecki. A study of variability models and languages in the systems software domain. *IEEE Transactions on Software Engineering*, 2013

agenda

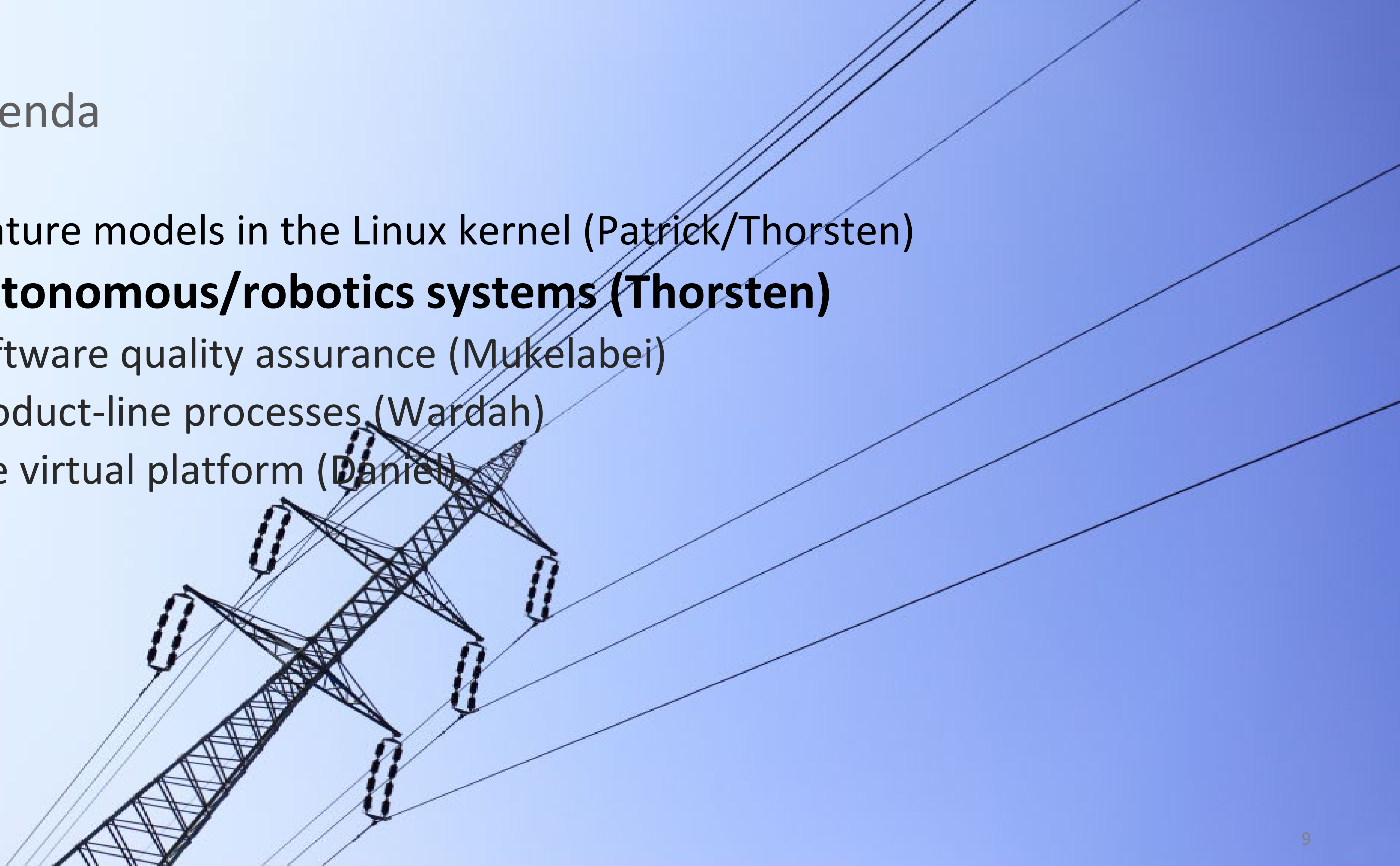
feature models in the Linux kernel (Patrick/Thorsten)

autonomous/robotics systems (Thorsten)

software quality assurance (Mukelabei)

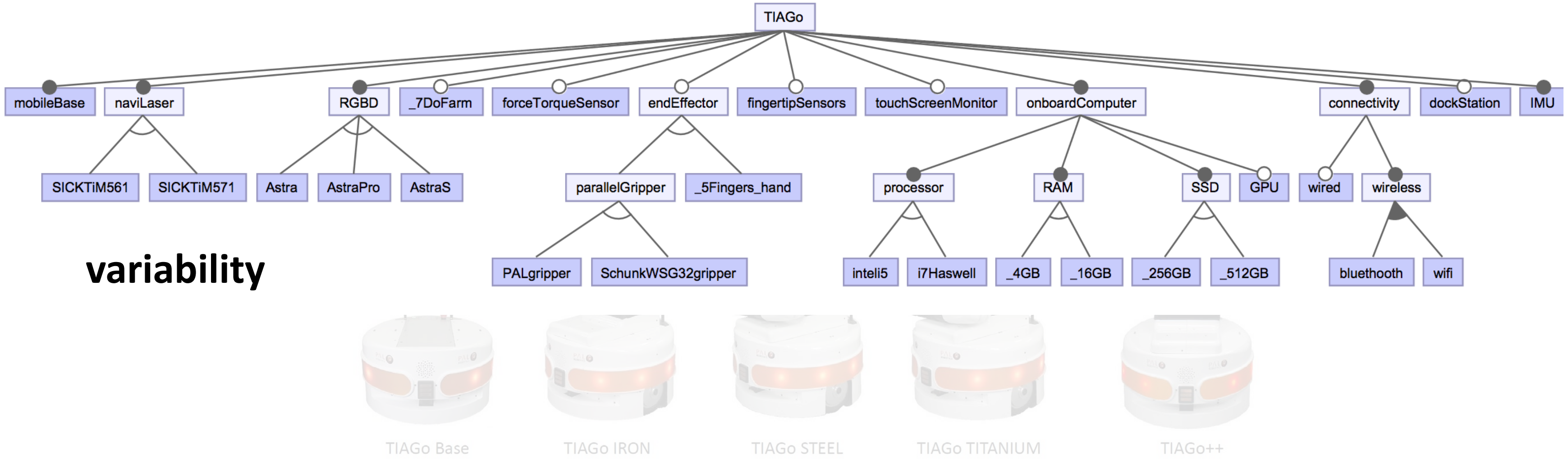
product-line processes (Wardah)

the virtual platform (Daniel)

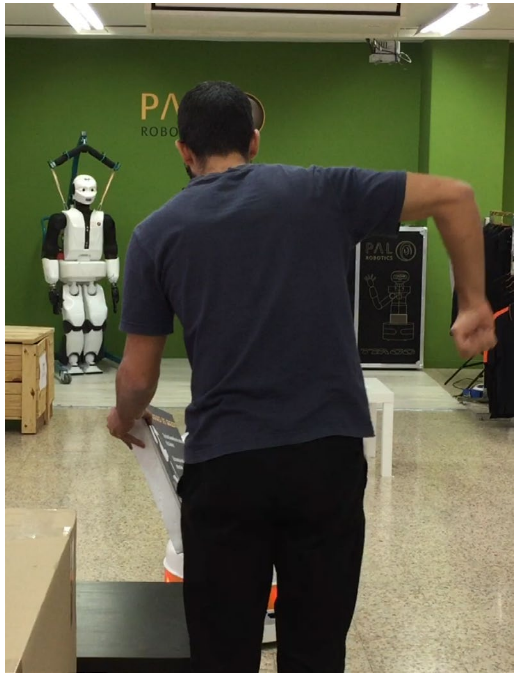


autonomous robots

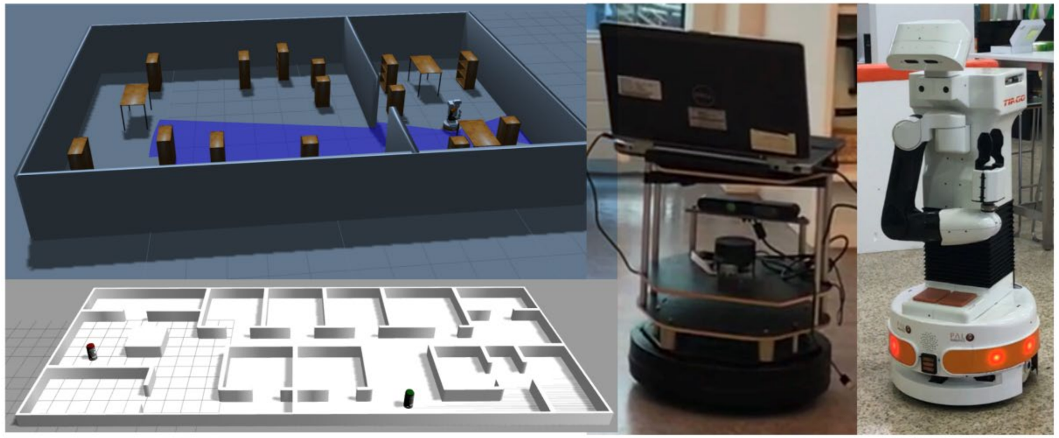
variability



gesture recognition



RoboCup tasks



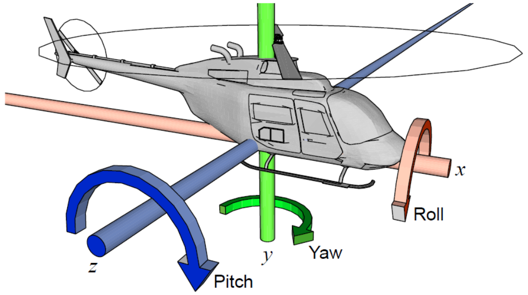
perception

human-robot interaction

behavior of autonomous systems

continuous dynamics

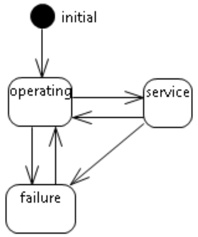
- Newton‘s mechanics
- ordinary differential equations
- actor models
- control properties (e.g., stability)



control theory

discrete dynamics

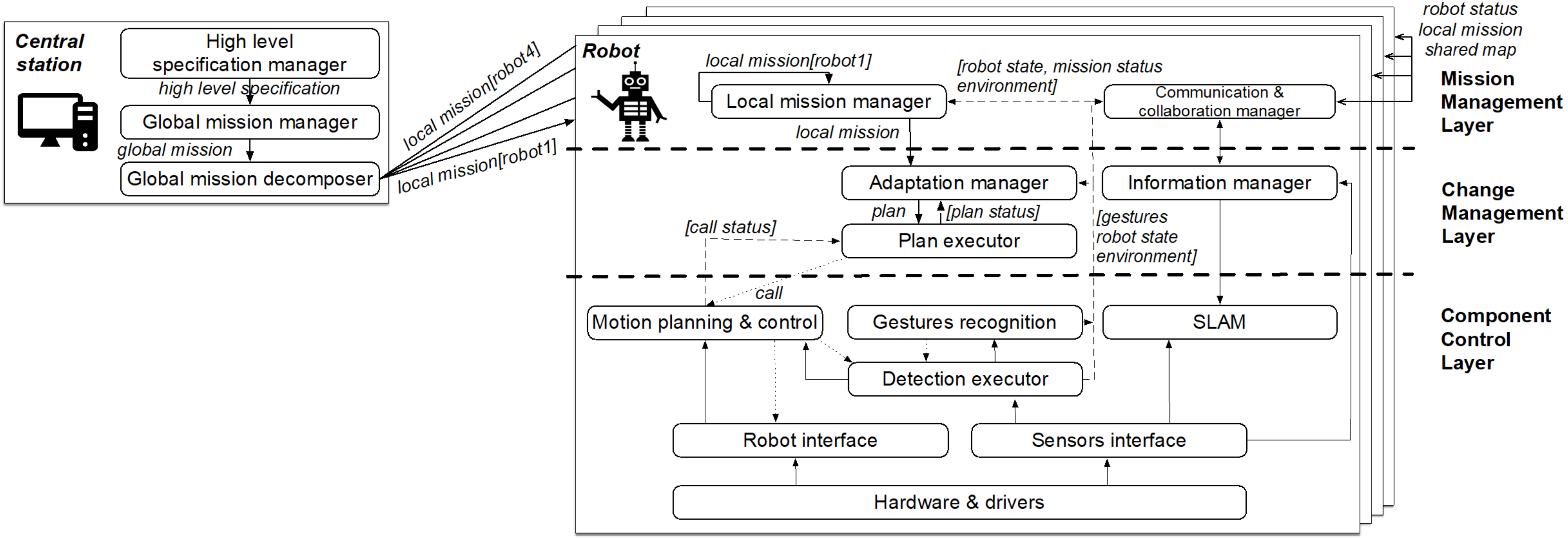
- states and state transitions
  - abrupt changes
  - operation modes
  - fail-safe states
- temporal properties



software engineering

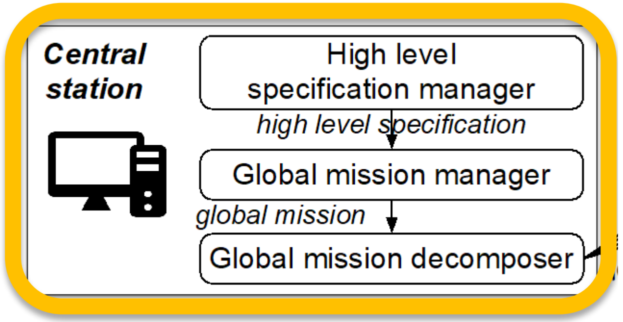
ROBOTICS ARCHITECTURE

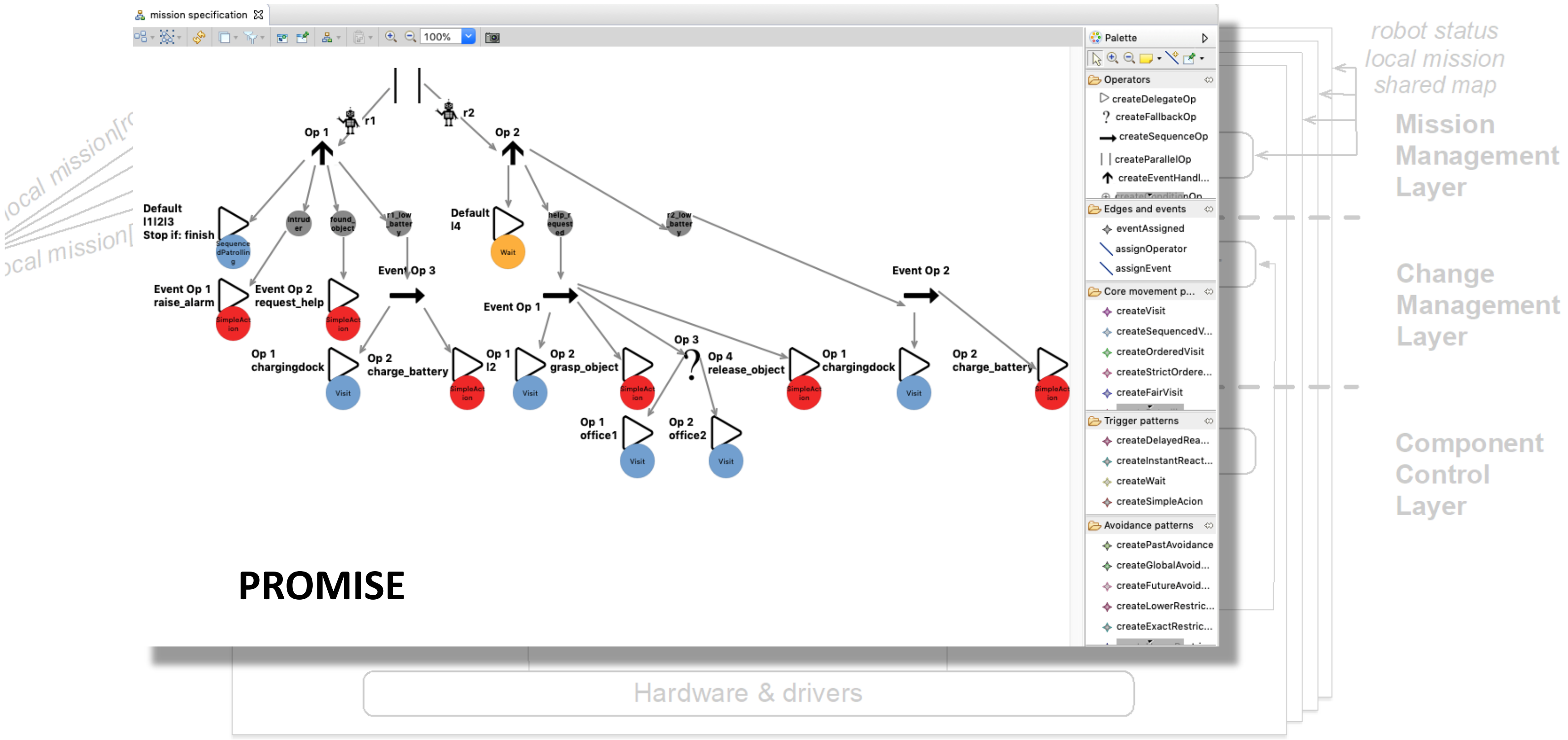
SERA (Self- adaptive dEcentralized Robotic Architecture)



Garcia, Pelliccione, Menghi, Berger, Wohlrab, “An Architecture for Decentralized, Collaborative, and Autonomous Robots,” in *International Conference on Software Architecture (ICSA)*, 2018

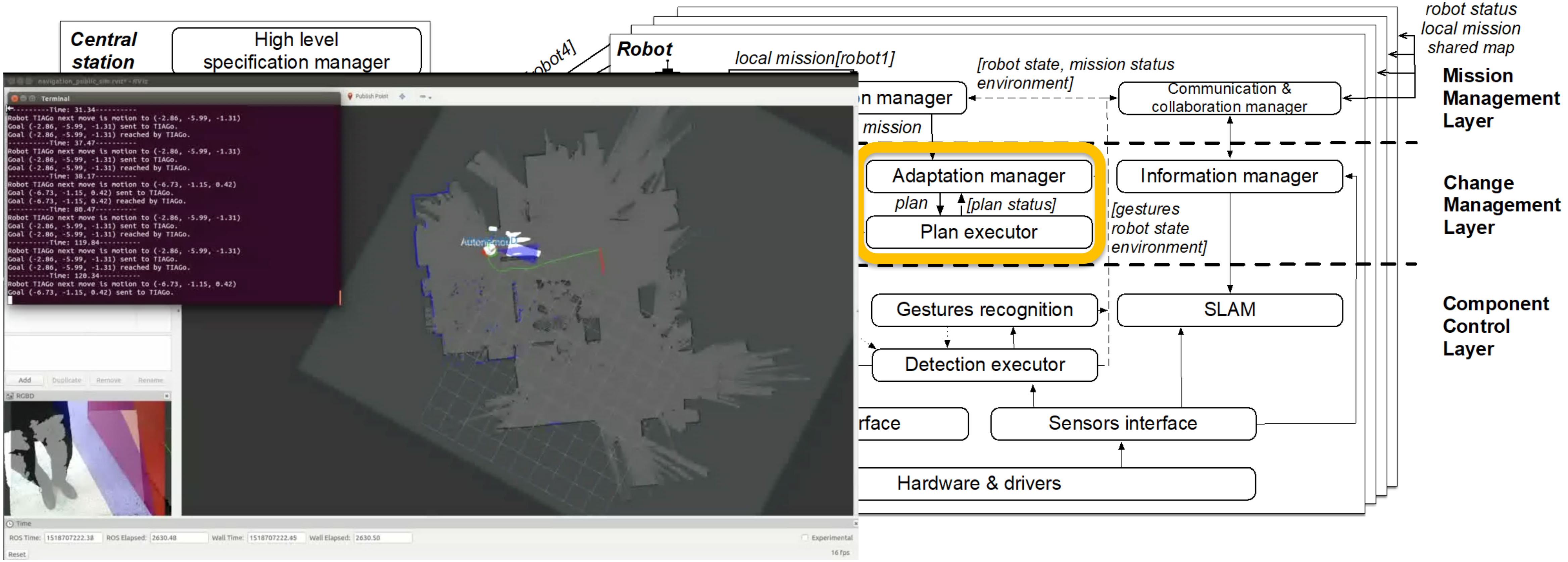
SERA: multi-robot mission control





PROMISE

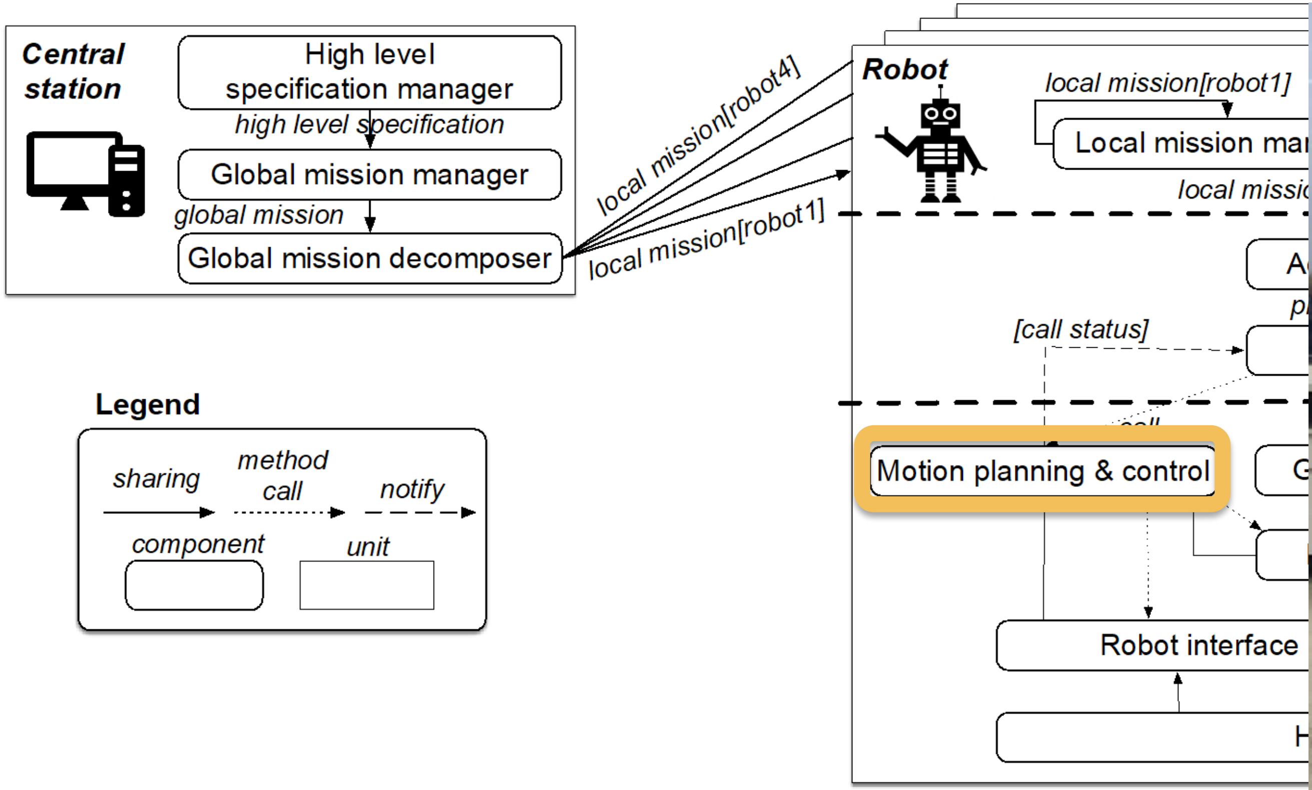
SERA: planning and adaptation



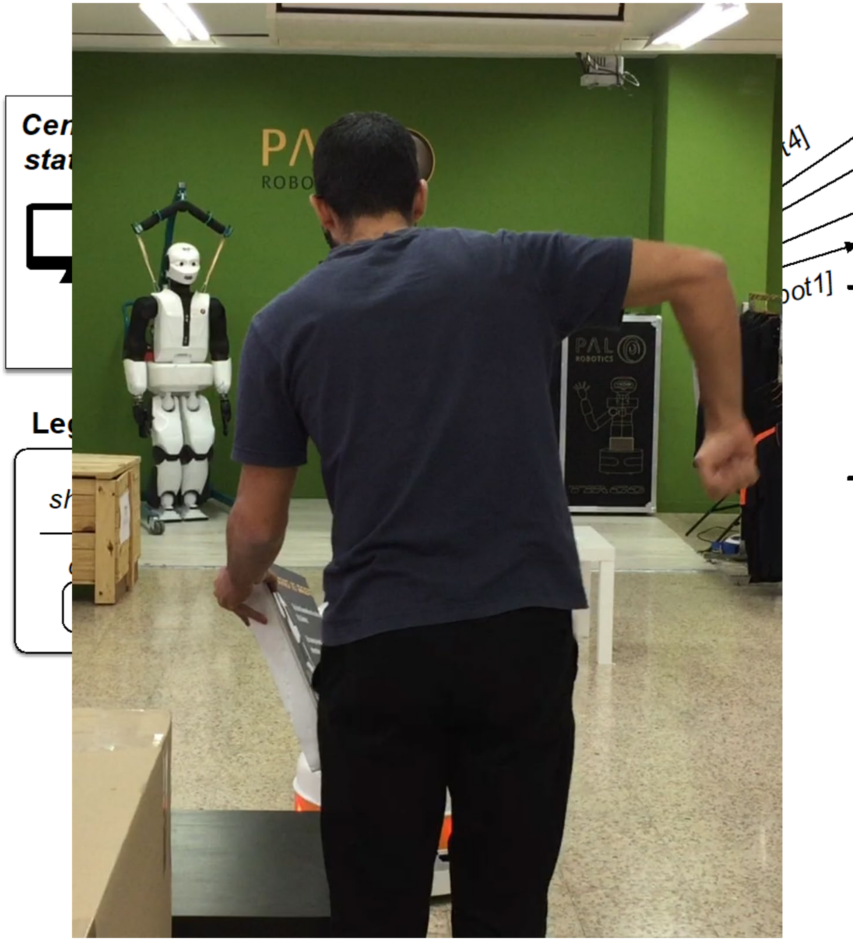
```
----------Time: 31.34----------
Robot TIAGo next move is motion to (-2.86, -5.99, -1.31)
Goal (-2.86, -5.99, -1.31) sent to TIAGo.
Goal (-2.86, -5.99, -1.31) reached by TIAGo.
----------Time: 37.47----------
Robot TIAGo next move is motion to (-2.86, -5.99, -1.31)
Goal (-2.86, -5.99, -1.31) sent to TIAGo.
Goal (-2.86, -5.99, -1.31) reached by TIAGo.
----------Time: 38.17----------
Robot TIAGo next move is motion to (-6.73, -1.15, 0.42)
Goal (-6.73, -1.15, 0.42) sent to TIAGo.
Goal (-6.73, -1.15, 0.42) reached by TIAGo.
----------Time: 80.47----------
Robot TIAGo next move is motion to (-2.86, -5.99, -1.31)
Goal (-2.86, -5.99, -1.31) sent to TIAGo.
Goal (-2.86, -5.99, -1.31) reached by TIAGo.
----------Time: 119.84---------
Robot TIAGo next move is motion to (-2.86, -5.99, -1.31)
Goal (-2.86, -5.99, -1.31) sent to TIAGo.
Goal (-2.86, -5.99, -1.31) reached by TIAGo.
----------Time: 120.34---------
Robot TIAGo next move is motion to (-6.73, -1.15, 0.42)
Goal (-6.73, -1.15, 0.42) sent to TIAGo.
```

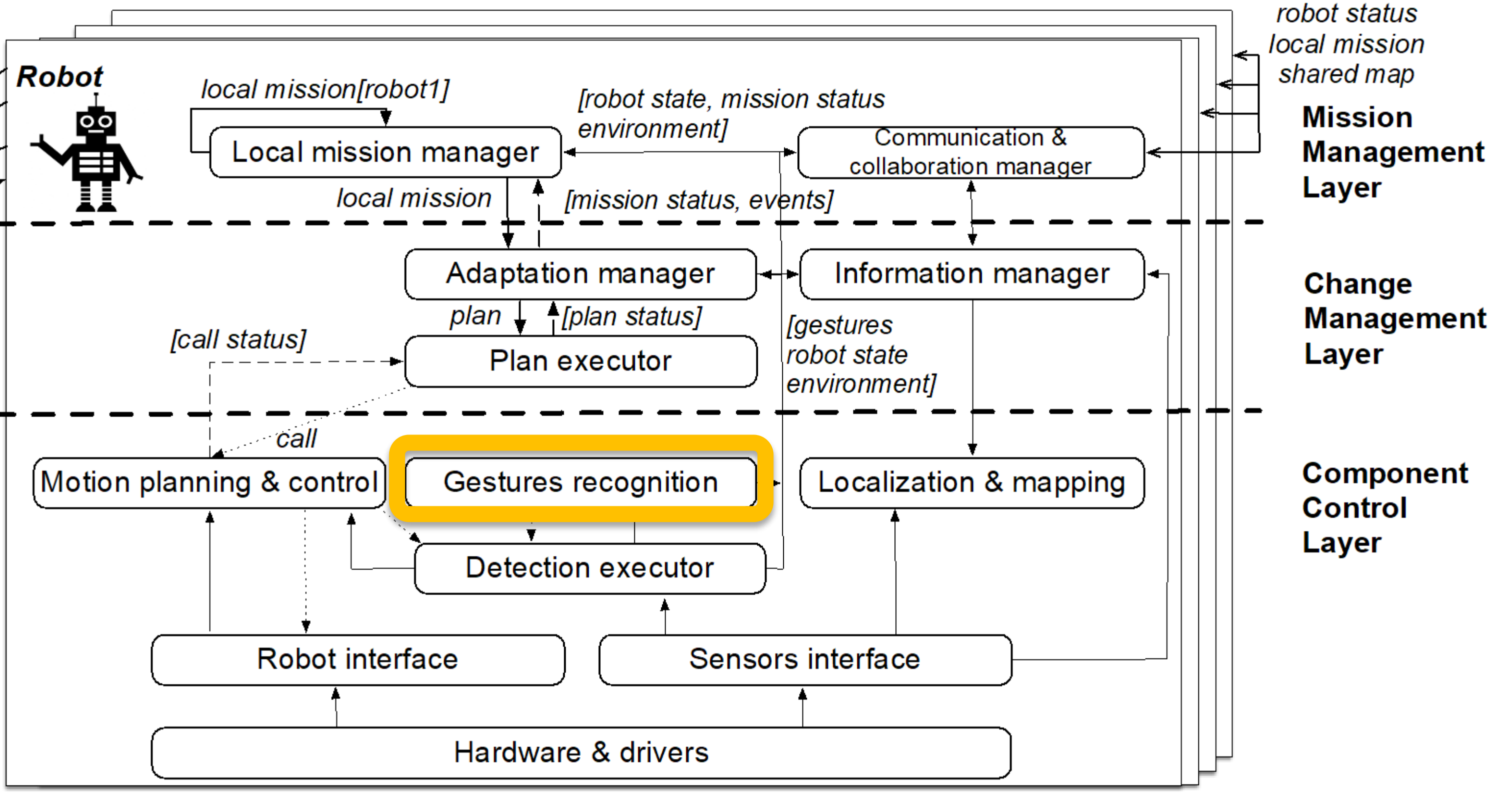
ROS Time: 1518707222.38 ROS Elapsed: 2630.48 Wall Time: 1518707222.45 Wall Elapsed: 2630.50

SERA: motion planning and control



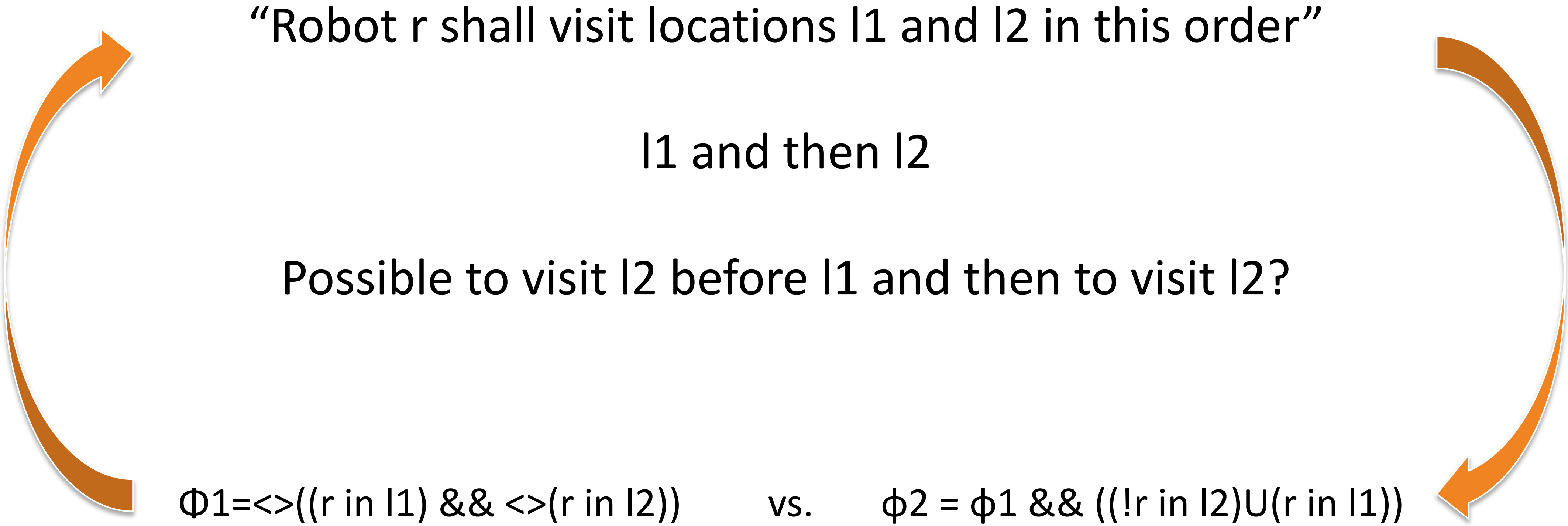
SERA: gesture recognition





ROBOTICS MISSION SPECIFICATION

user-friendly mission specification



“Robot r shall visit locations l1 and l2 in this order”

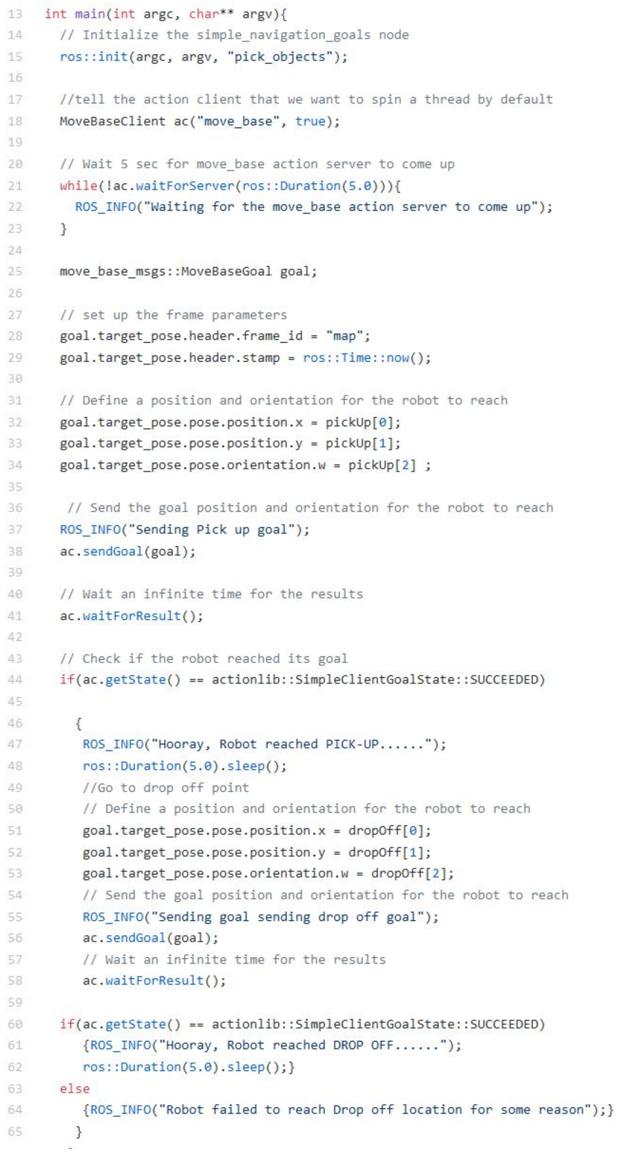
l1 and then l2

Possible to visit l2 before l1 and then to visit l2?

Φ1=<>((r in l1) && <>(r in l2)) vs. ϕ2 = ϕ1 && ((!r in l2)U(r in l1))

discrete dynamics: code and models

plain code



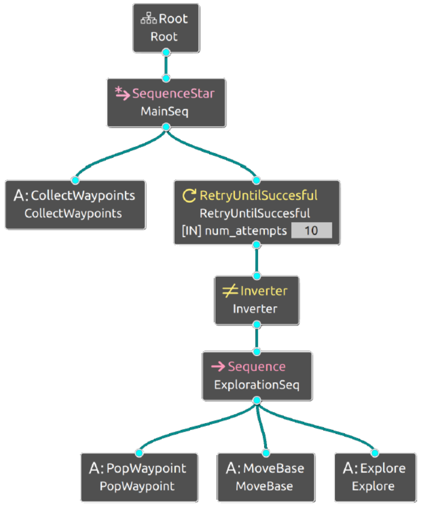
```
int main(int argc, char** argv){
  // Initialize the simple_navigation_goals node
  ros::init(argc, argv, "pick_objects");

  //tell the action client that we want to spin a thread by default
  MoveBaseClient ac("move_base", true);

  // Wait 5 sec for move_base action server to come up
  while(!ac.waitForServer(ros::Duration(5.0))){
    ROS_INFO("Waiting for the move_base action server to come up");
  }

  move_base_msgs::MoveBaseGoal goal;

  // set up the frame parameters
  goal.target_pose.header.frame_id = "map";
  goal.target_pose.header.stamp = ros::Time::now();

  // Define a position and orientation for the robot to reach
  goal.target_pose.pose.position.x = pickUp[0];
  goal.target_pose.pose.position.y = pickUp[1];
  goal.target_pose.pose.orientation.w = pickUp[2] ;

   // Send the goal position and orientation for the robot to reach
  ROS_INFO("Sending Pick up goal");
  ac.sendGoal(goal);

  // Wait an infinite time for the results
  ac.waitForResult();

  // Check if the robot reached its goal
  if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)

    {
     ROS_INFO("Hooray, Robot reached PICK-UP......");
     ros::Duration(5.0).sleep();
     //Go to drop off point
     // Define a position and orientation for the robot to reach
     goal.target_pose.pose.position.x = dropOff[0];
     goal.target_pose.pose.position.y = dropOff[1];
     goal.target_pose.pose.orientation.w = dropOff[2];
     // Send the goal position and orientation for the robot to reach
     ROS_INFO("Sending goal sending drop off goal");
     ac.sendGoal(goal);
     // Wait an infinite time for the results
     ac.waitForResult();

  if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
     {ROS_INFO("Hooray, Robot reached DROP OFF......");
     ros::Duration(5.0).sleep();}
  else
     {ROS_INFO("Robot failed to reach Drop off location for some reason");}
    }
```

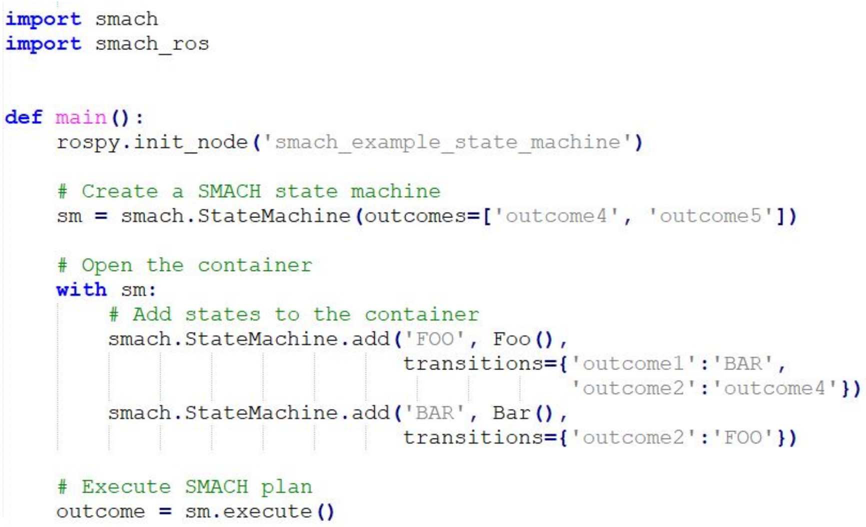
temporal logics

Φ1=<>((r in l1) && <>(r in l2))

```
<root main_tree_to_execute="BehaviorTree">

    <BehaviorTree ID="BehaviorTree">

        <Fallback name="ReachTable">

            <Sequence name="SelfFollowing">
                <Condition ID="TableKnown"/>
                <Action ID="GoToTable"/>
            </Sequence>

            <Sequence name="ShouldFollow">
                <Action ID="AskForHelp"/>
                <Action ID="FollowHuman"/>
            </Sequence>

        </Fallback>

    </BehaviorTree>

</root>
```

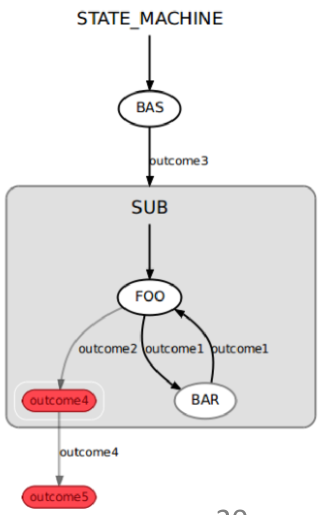
behavior trees





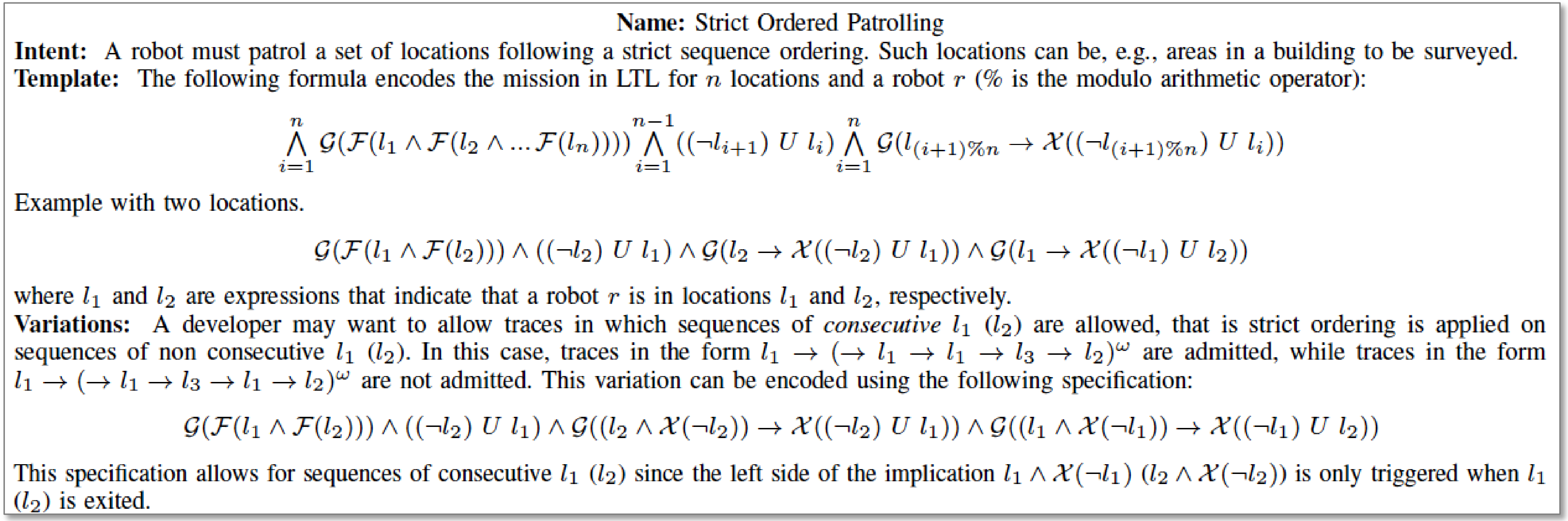
```
import smach
import smach_ros

def main():
    rospy.init_node('smach_example_state_machine')

    # Create a SMACH state machine
    sm = smach.StateMachine(outcomes=['outcome4', 'outcome5'])

    # Open the container
    with sm:
        # Add states to the container
        smach.StateMachine.add('FOO', Foo(),
                               transitions={'outcome1':'BAR',
                                            'outcome2':'outcome4'})
        smach.StateMachine.add('BAR', Bar(),
                               transitions={'outcome2':'FOO'})

    # Execute SMACH plan
    outcome = sm.execute()
```

state machines



Ghzouli, Berger, Broch Johnsen, Dragule, Wasowski, “Behavior Trees in Action: A Study of Robotics Applications,” *13th ACM SIGPLAN International Conference on Software Language Engineering (SLE)*, 2020

mission specification patterns



Name: Strict Ordered Patrolling

Intent: A robot must patrol a set of locations following a strict sequence ordering. Such locations can be, e.g., areas in a building to be surveyed.

Template: The following formula encodes the mission in LTL for n locations and a robot r (% is the modulo arithmetic operator):

$$\bigwedge_{i=1}^{n} \mathcal{G}(\mathcal{F}(l_1 \wedge \mathcal{F}(l_2 \wedge \ldots \mathcal{F}(l_n)))) \bigwedge_{i=1}^{n-1} ((\neg l_{i+1})\ U\ l_i) \bigwedge_{i=1}^{n} \mathcal{G}(l_{(i+1)\%n} \rightarrow \mathcal{X}((\neg l_{(i+1)\%n})\ U\ l_i))$$

Example with two locations.

$$\mathcal{G}(\mathcal{F}(l_1 \wedge \mathcal{F}(l_2))) \wedge ((\neg l_2)\ U\ l_1) \wedge \mathcal{G}(l_2 \rightarrow \mathcal{X}((\neg l_2)\ U\ l_1)) \wedge \mathcal{G}(l_1 \rightarrow \mathcal{X}((\neg l_1)\ U\ l_2))$$

where l_1 and l_2 are expressions that indicate that a robot r is in locations l_1 and l_2, respectively.

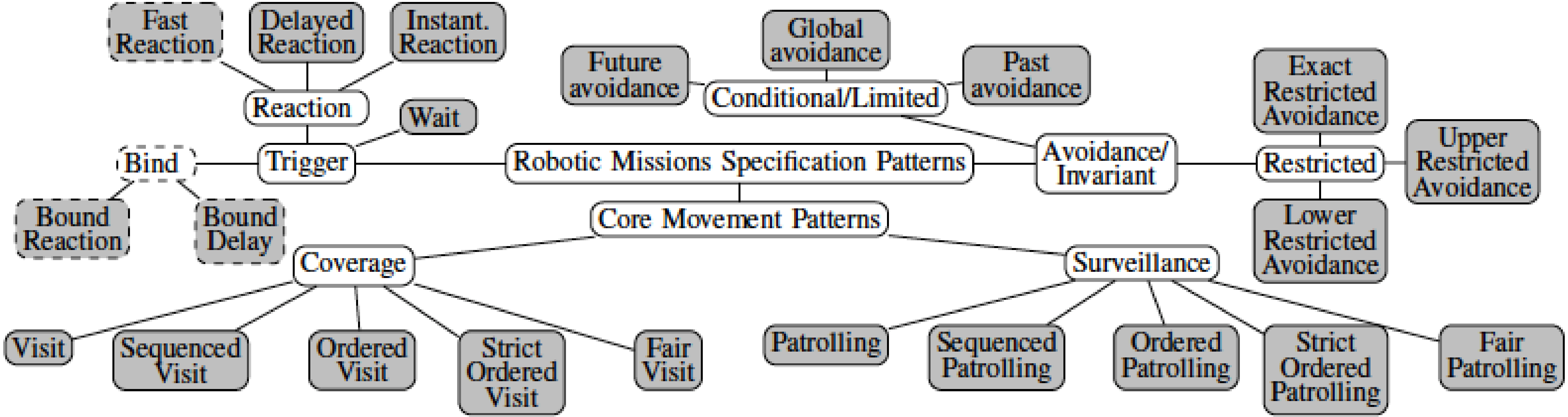
Variations: A developer may want to allow traces in which sequences of *consecutive* l_1 (l_2) are allowed, that is strict ordering is applied on sequences of non consecutive l_1 (l_2). In this case, traces in the form $l_1 \rightarrow (\rightarrow l_1 \rightarrow l_1 \rightarrow l_3 \rightarrow l_2)^{\omega}$ are admitted, while traces in the form $l_1 \rightarrow (\rightarrow l_1 \rightarrow l_3 \rightarrow l_1 \rightarrow l_2)^{\omega}$ are not admitted. This variation can be encoded using the following specification:

$$\mathcal{G}(\mathcal{F}(l_1 \wedge \mathcal{F}(l_2))) \wedge ((\neg l_2)\ U\ l_1) \wedge \mathcal{G}((l_2 \wedge \mathcal{X}(\neg l_2)) \rightarrow \mathcal{X}((\neg l_2)\ U\ l_1)) \wedge \mathcal{G}((l_1 \wedge \mathcal{X}(\neg l_1)) \rightarrow \mathcal{X}((\neg l_1)\ U\ l_2))$$

This specification allows for sequences of consecutive l_1 (l_2) since the left side of the implication $l_1 \wedge \mathcal{X}(\neg l_1)$ ($l_2 \wedge \mathcal{X}(\neg l_2)$) is only triggered when l_1 (l_2) is exited.

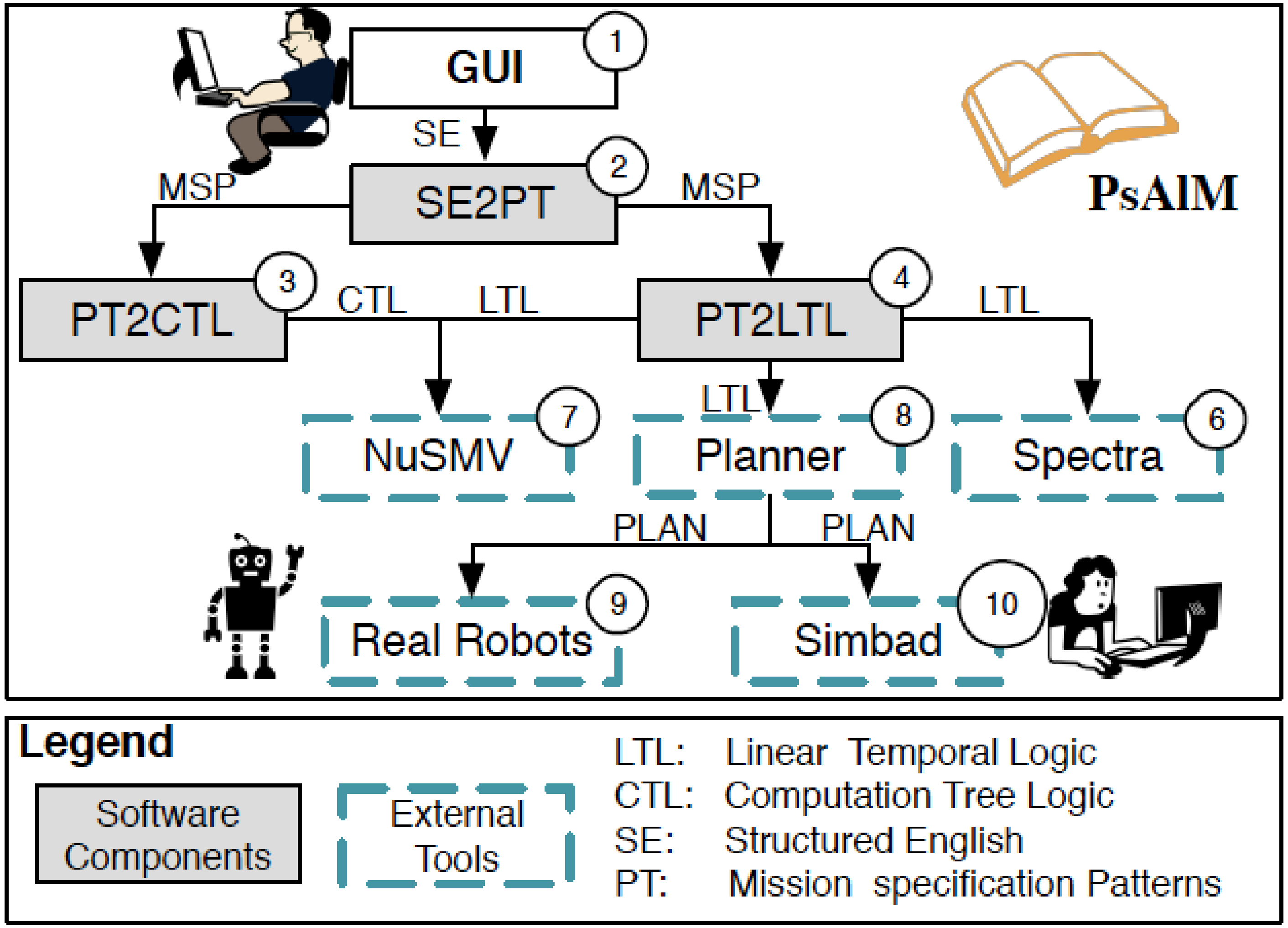
C. Menghi, C. Tsigkanos, P. Pelliccione, C. Ghezzi and T. Berger. "Specification Patterns for Robotic Missions,"
in *IEEE Transactions on Software Engineering (TSE)*, 2019

mission specification patterns

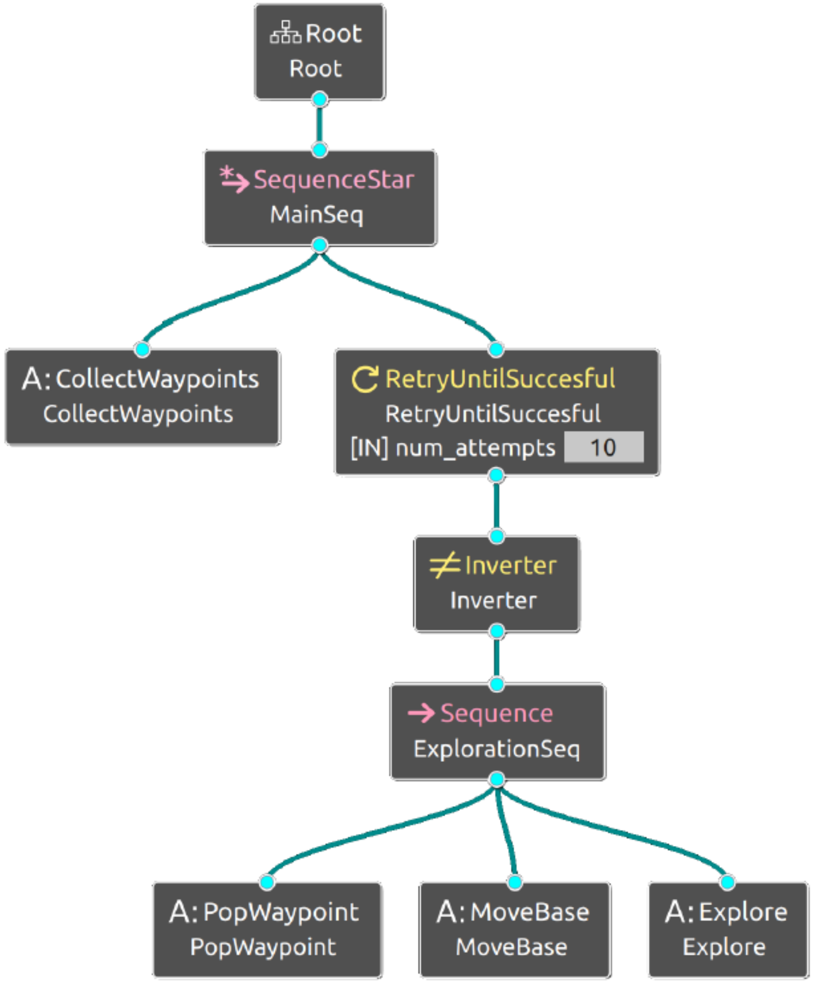


C. Menghi, C. Tsigkanos, P. Pelliccione, C. Ghezzi and T. Berger. "Specification Patterns for Robotic Missions,"
in *IEEE Transactions on Software Engineering (TSE)*, 2019

model checking, planning, and modeling

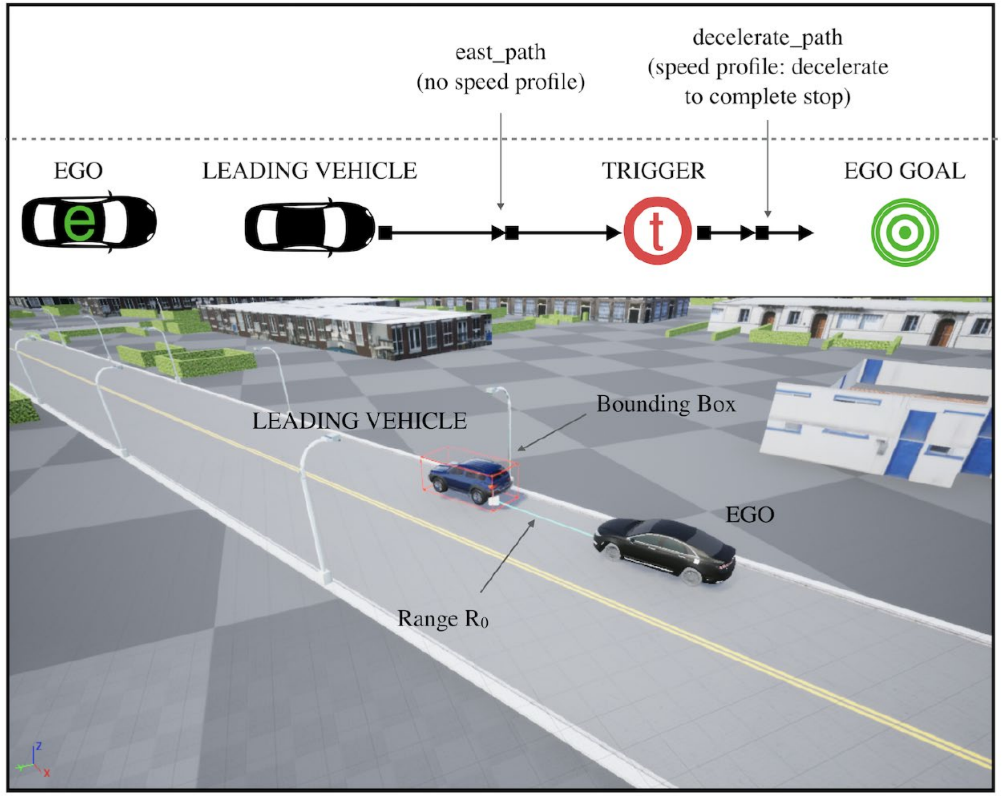


LTL: Linear Temporal Logic
CTL: Computation Tree Logic
SE: Structured English
PT: Mission specification Patterns



PROMISE
(behavior tree language)

autonomous driving



Rodrigo Queiroz, Thorsten Berger, Krzysztof Czarnecki, “Geoscenario: An Open DSL for Autonomous Driving Scenario Representation,” in *30th IEEE Intelligent Vehicles Symposium (IV)*, 2019.

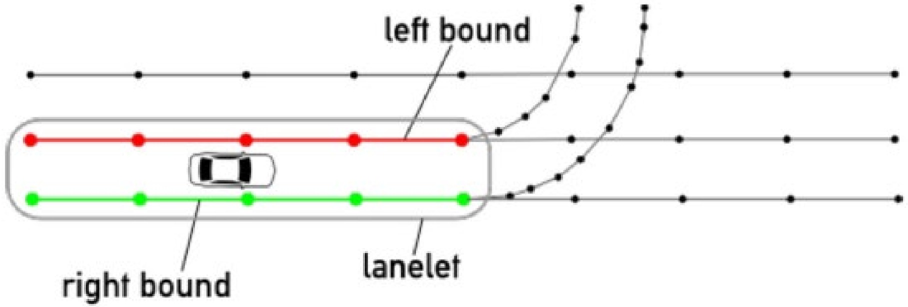
UWaterloo, CA

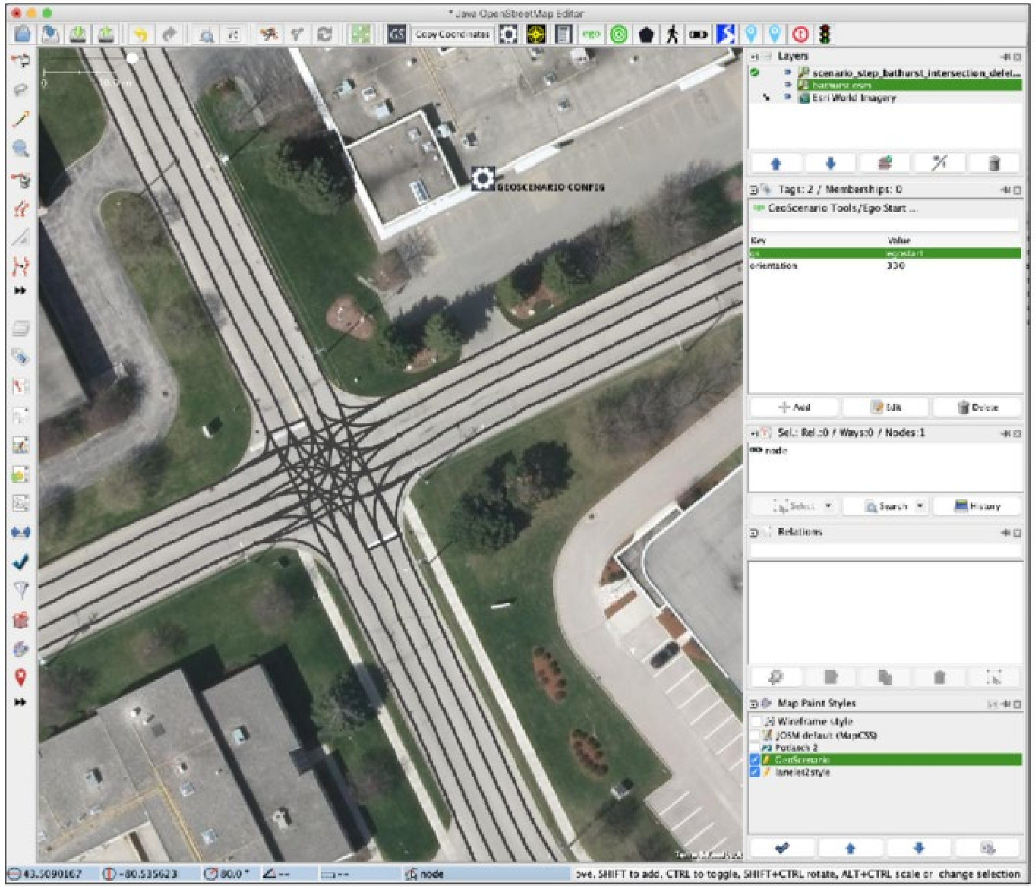
Autonomoose Self-Driving Research Platform
Lincoln MKZ Hybrid

CHALMERS, SE

REVERE Lab, Volvo XC90

road network models





* Java OpenStreetMap Editor

Layers
- scenario_step_bathurst_intersection_delet...
- bathurst.osm
- Esri World Imagery

Tags: 2 / Memberships: 0

GeoScenario Tools/Ego Start ...

Key	Value
gs	egostart
orientation	330

Sel.: Rel.:0 / Ways:0 / Nodes:1

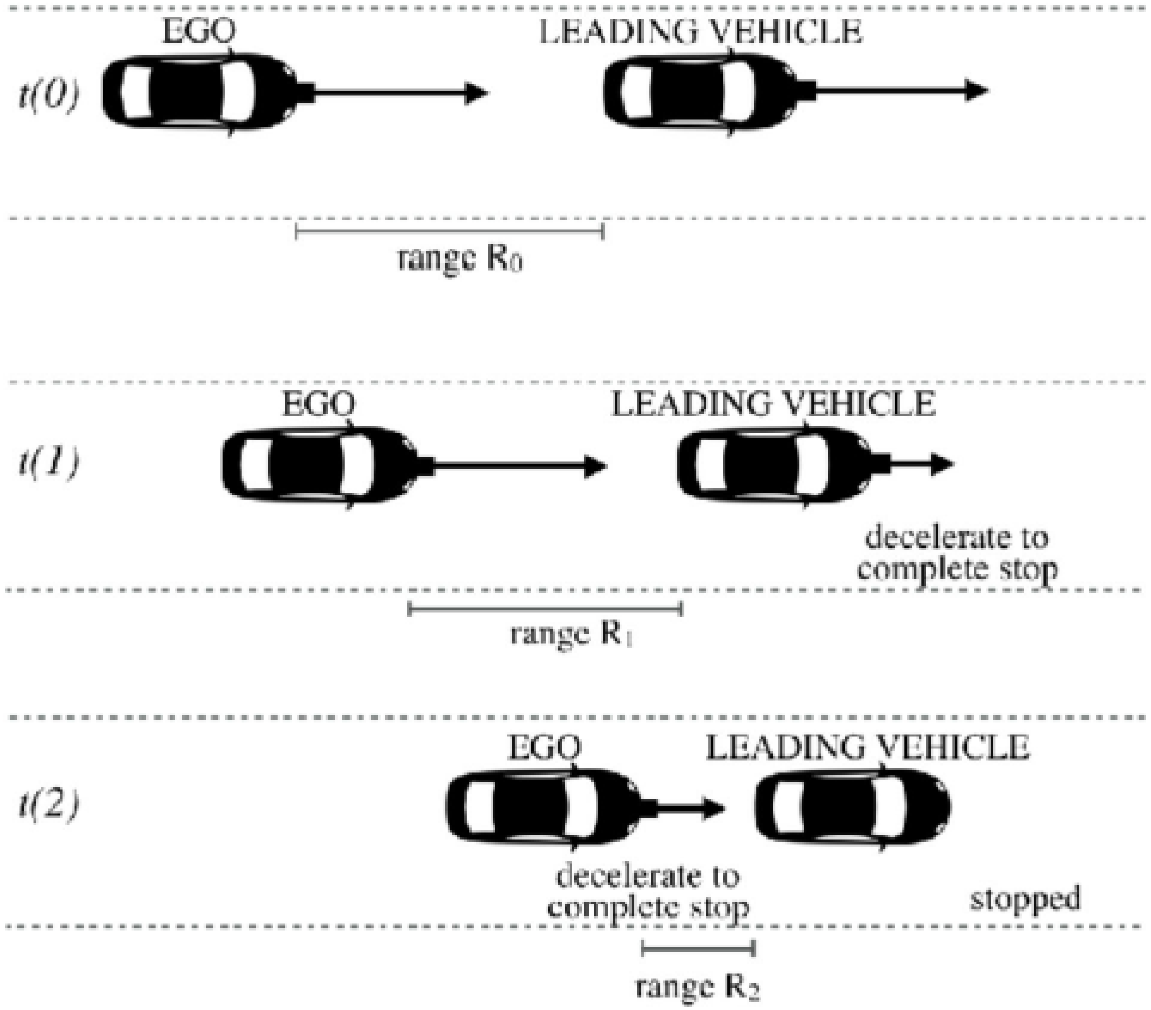
node

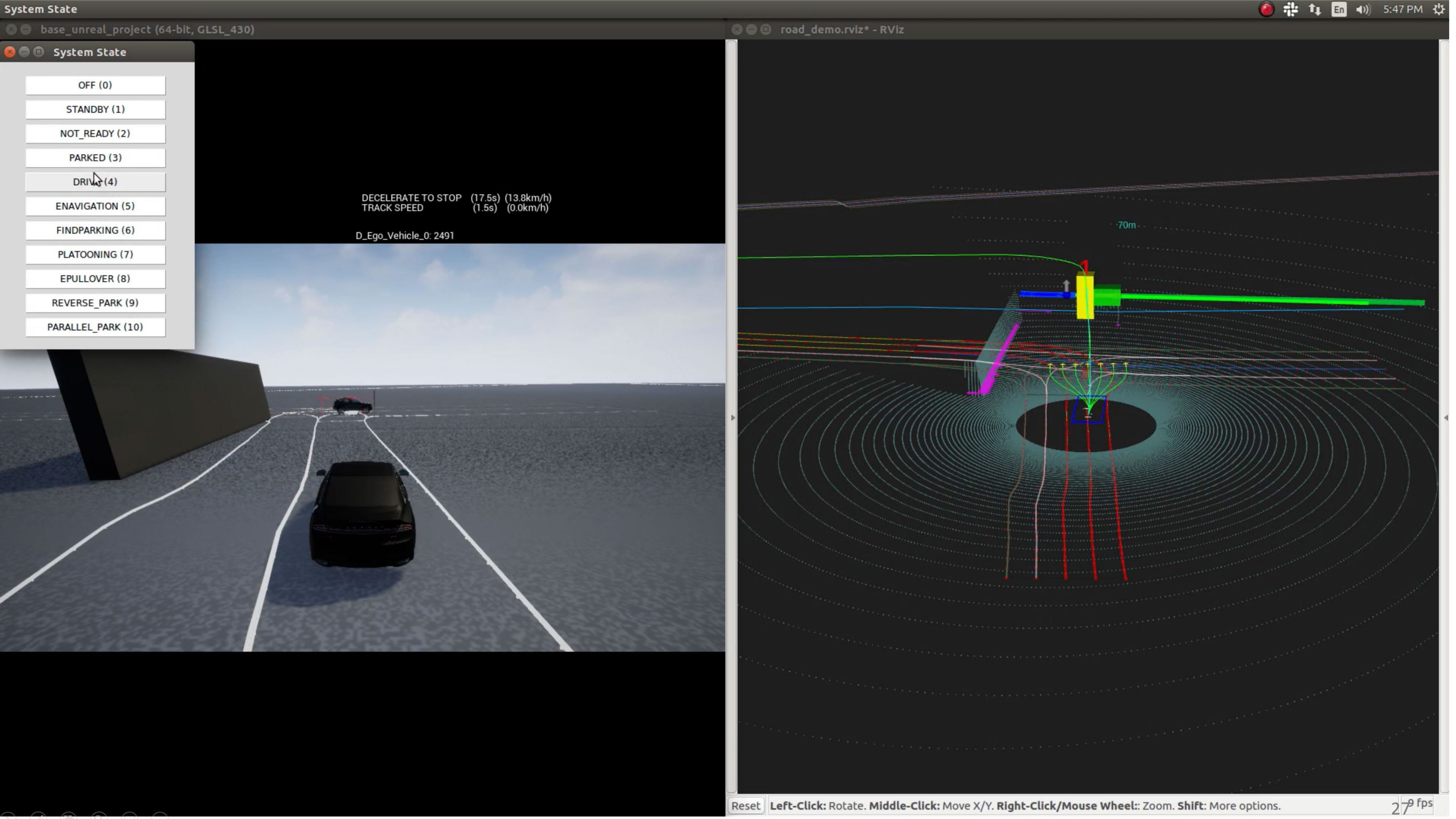
Relations

Map Paint Styles
- □ Wireframe style
- □ JOSM default (MapCSS)
- □ Potlatch 2
- ☑ GeoScenario
- ☑ lanelet2style

43.5090167 | -80.535623 | 80.0 ° | node | ove, SHIFT to add, CTRL to toggle, SHIFT+CTRL rotate, ALT+CTRL scale or change selection

behavior and maneuver models

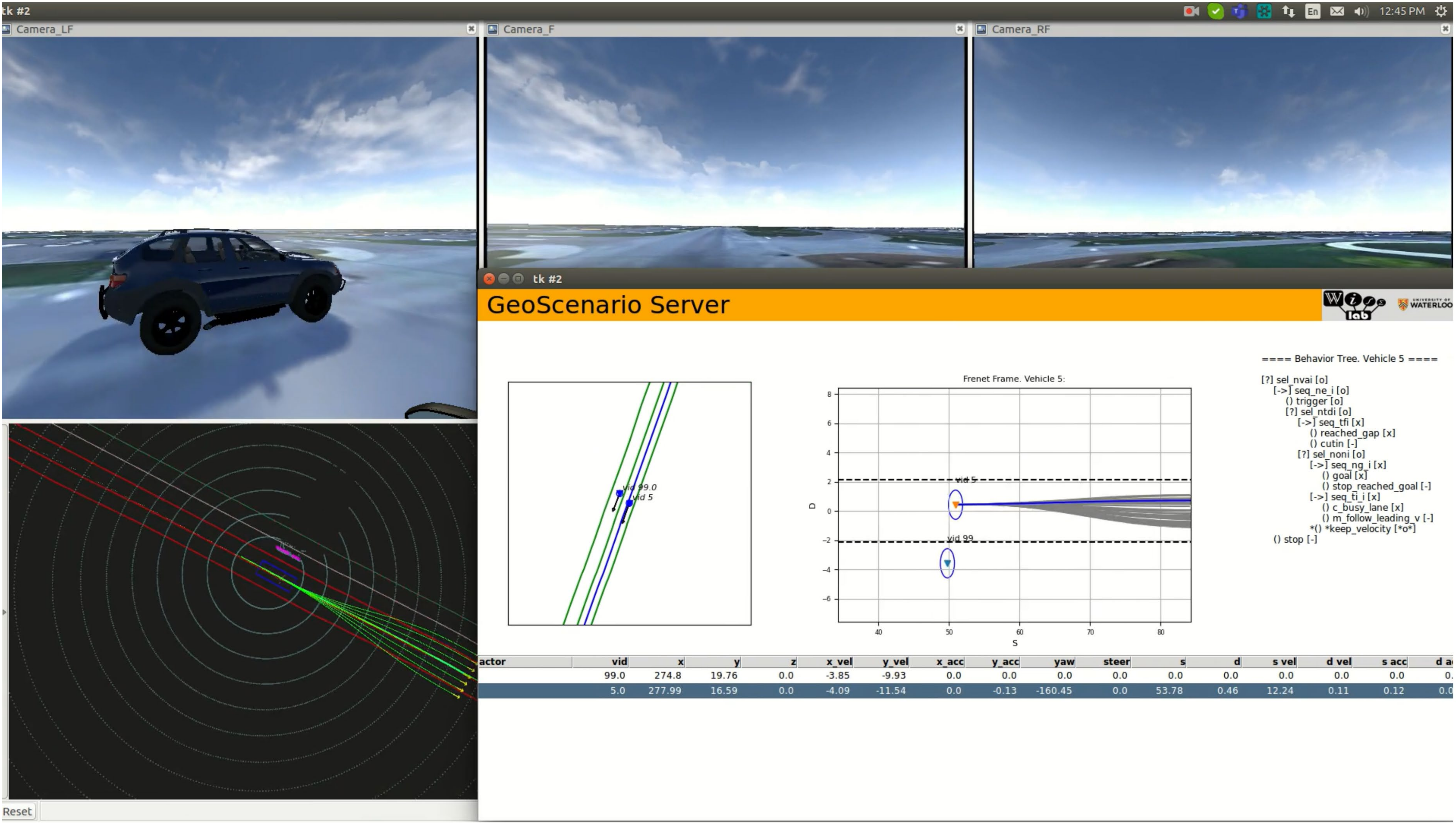




```
DECELERATE TO STOP   (17.5s)  (13.8km/h)
TRACK SPEED           (1.5s)   (0.0km/h)

D_Ego_Vehicle_0: 2491
```

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

9 fps



GeoScenario Server

```
==== Behavior Tree. Vehicle 5 ====

[?] sel_nvai [o]
    [->] seq_ne_i [o]
        () trigger [o]
        [?] sel_ntdi [o]
            [->] seq_tfi [x]
                () reached_gap [x]
                () cutin [-]
            [?] sel_noni [o]
                [->] seq_ng_i [x]
                    () goal [x]
                    () stop_reached_goal [-]
                [->] seq_ti_i [x]
                    () c_busy_lane [x]
                    () m_follow_leading_v [-]
                *() *keep_velocity [*o*]
    () stop [-]
```

actor	vid	x	y	z	x_vel	y_vel	x_acc	y_acc	yaw	steer	s	d	s vel	d vel	s acc	d a
	99.0	274.8	19.76	0.0	-3.85	-9.93	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
	5.0	277.99	16.59	0.0	-4.09	-11.54	0.0	-0.13	-160.45	0.0	53.78	0.46	12.24	0.11	0.12	0.0

research assistants and thesis topics

research assistants, deadline: Dec. 20!

https://web103.reachmee.com/ext/I005/1035/job?site=7&lang=UK&validator=9b89bead79bb7258ad55c8d75228e5b7&job_id=23034

thesis topics

- http://www.cse.chalmers.se/~bergert/teaching

agenda

feature models in the Linux kernel (Patrick/Thorsten)

autonomous/robotics systems (Thorsten)

software quality assurance (Mukelabei)

product-line processes (Wardah)

the virtual platform (Daniel)