Desarrollo de robots sociales con RoboComp

Pablo Bustos
RoboLab
Universidad de Extremadura

UCM - 2016

A quick introduction to

robolab

ROBOTICS LABORATORY

UEX

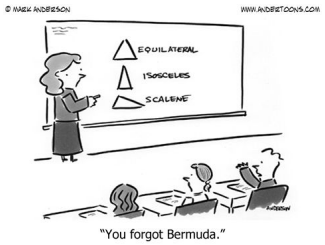
2000-∞

Social and Service Robotics

- Robots that help humans in everyday activities

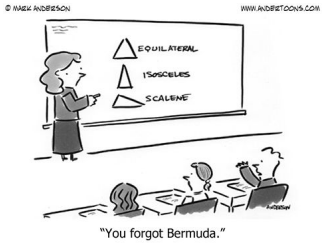
Social and Service Robotics

- Robots that help humans in everyday activities

 Educational				

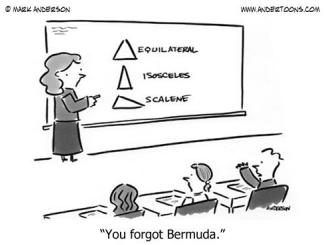
Social and Service Robotics

- Robots that help humans in everyday activities

 Educational	 Rehabilitation			

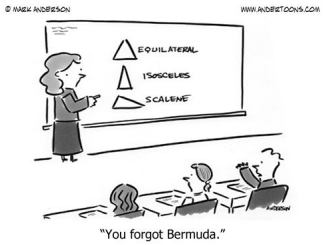
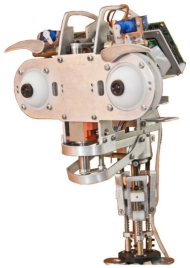
Social and Service Robotics

- Robots that help humans in everyday activities

 Educational	 Rehabilitation	 Advertisement		

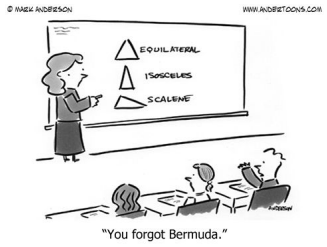
Social and Service Robotics

- Robots that help humans in everyday activities

				
Educational	Rehabilitation	Advertisement	Affective	

Social and Service Robotics

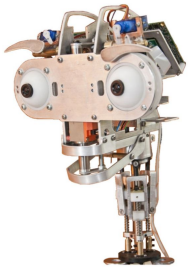
- Robots that help humans in everyday activities



Educational

Rehabilitation

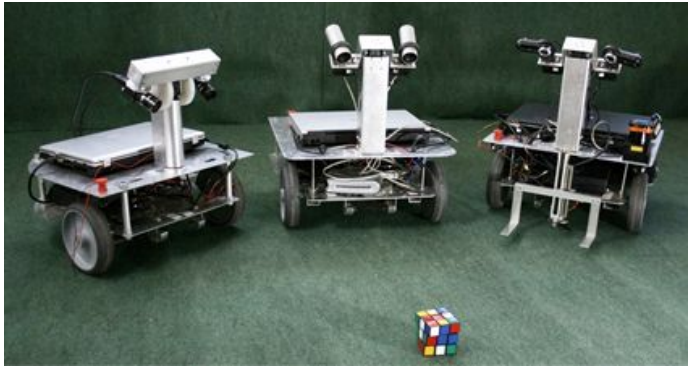
Advertisement

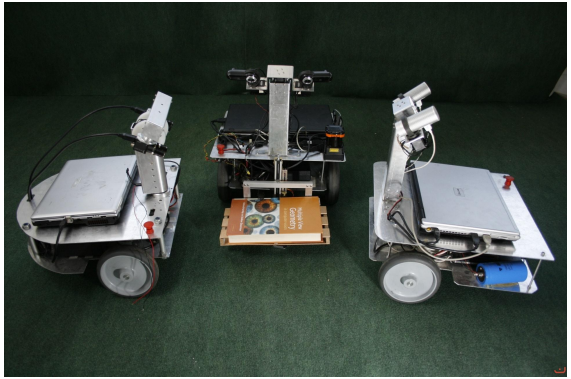


Affective

Butler

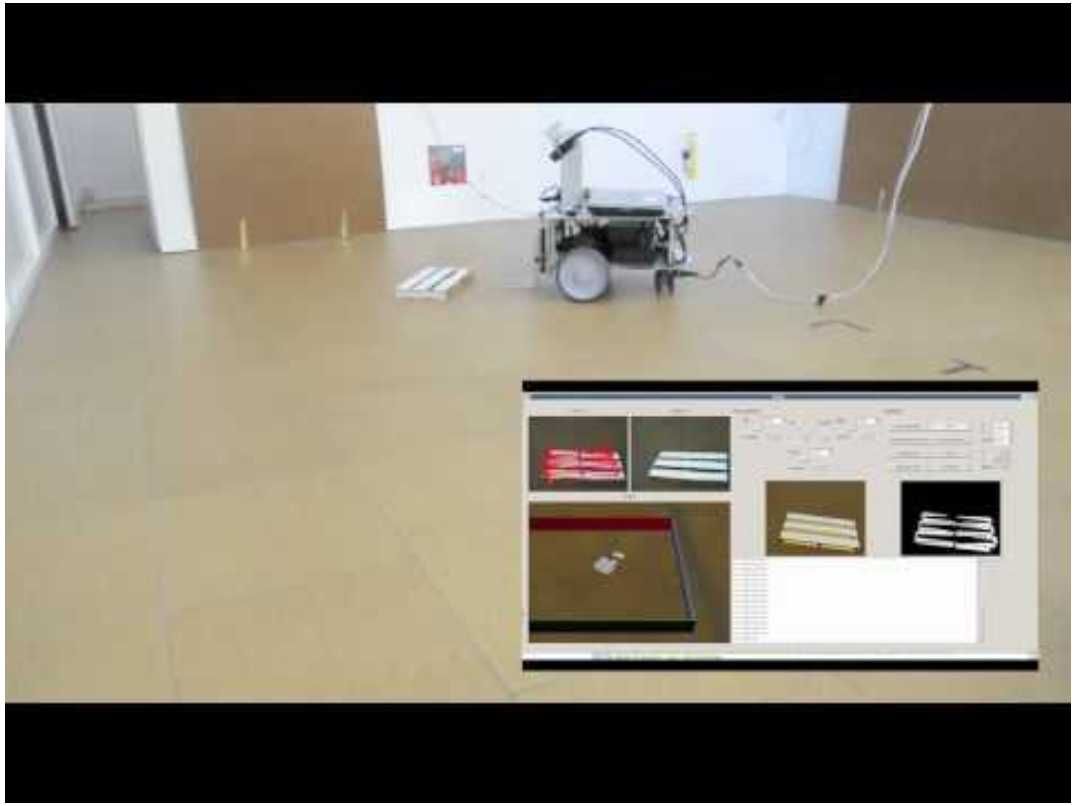
Educational robots - Robex



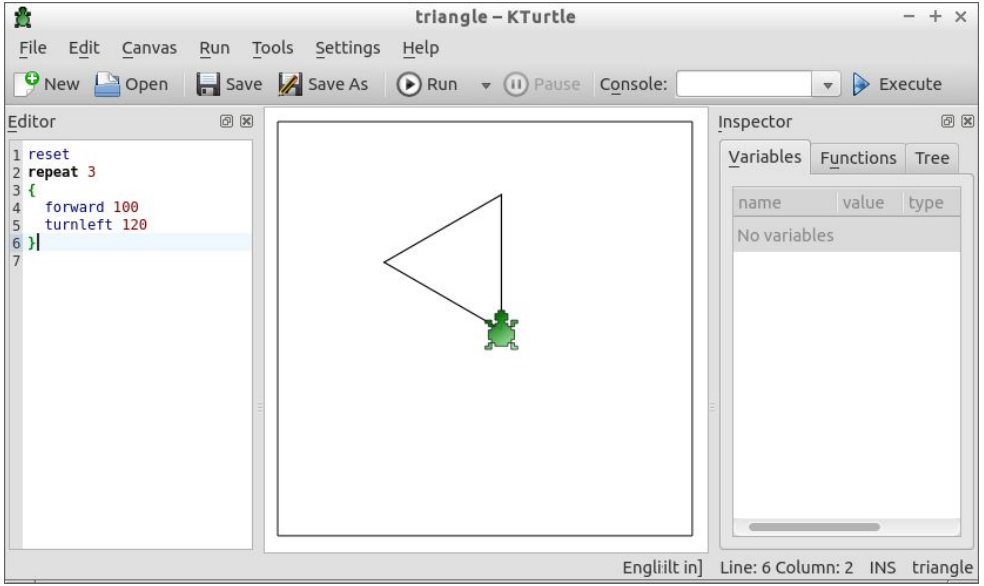


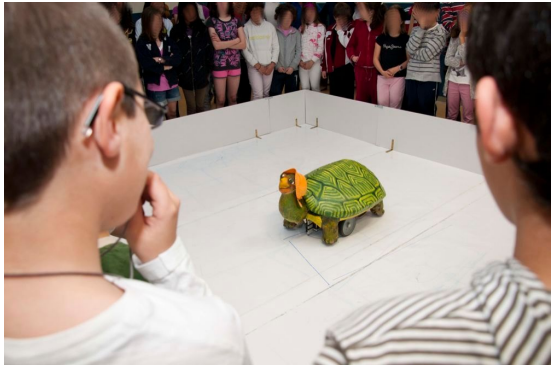
- 2005 - now
- college, comp. sci.
- low-cost
- versatile
- robust
- classroom
- adaptable

Educational robots - Robex



Educational robots in school - Dulce





- 2011
- 6th grade primary school
- 1 semester experiment

Educational robots in school - Dulce

Educational robots in school - Learnbot 1

- 2012
- Initiation to Python programming for kids in 5th to 12th grades
- Build with RaspberryPi and Arduino
- WiFi connection to laptop running Python

Educational robots in school - LearnBot 2

- Odroid only
- WiFi, battery, USs and USB camera
- under 150€ for one prototype

Educational robots in school - LearnBot 2

```
def code(self):
        # 0 back 3 left 2 right 1 front
        actions = ((150,-3), (450, 0), (150, 3))
        while(True):
            frame = self.getImage()
            rois = self.getROIS(frame)
            maxIndex = np.argmax(rois)
            self.setRobotSpeed(actions[maxIndex])
```

sum of black pixels in three ROIs

Rehabilitation Robots

Rehabilitation robots - Ursus

- 2010
- Dynamixel servos
- Visual mark detection

Rehabilitation robots - Ursus

Rehabilitation robots - Ursus

- Technology validation experiment
- 5 patients in 7-10 age range
- mild lesions affecting the shoulder (brachial or cerebral paralysis)

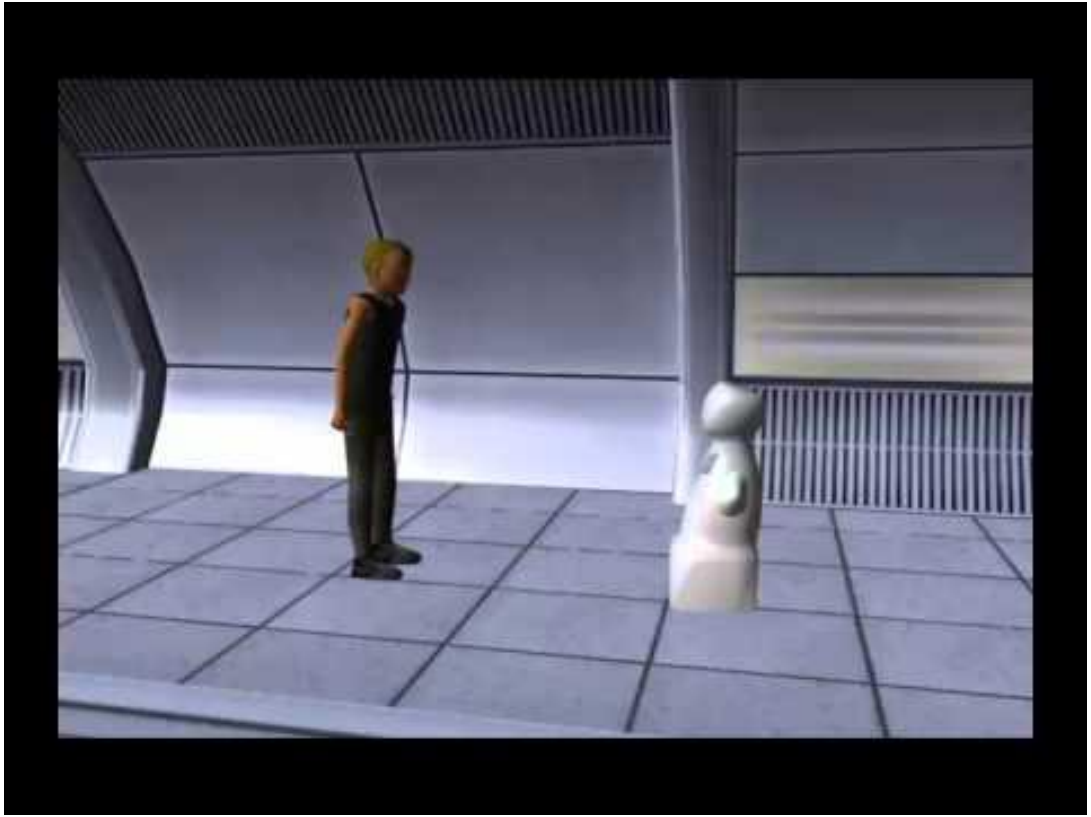
Social robots - Nao

Social robots - Nao

Advertisement robots - Gualzru

- a product for INDRA Spanish tech company.
- completely designed and built our group of research teams.
- differential base, batteries, tactile panel, leds for the eyes, micro, speaker, laser, RGBD.
- runs RoboCog on top of RoboComp framework.
- dozens of hours of operation.

Advertisement robots - Gualzru



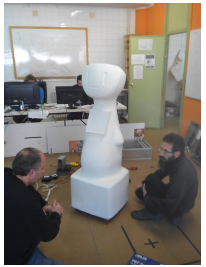
Gualzru building process









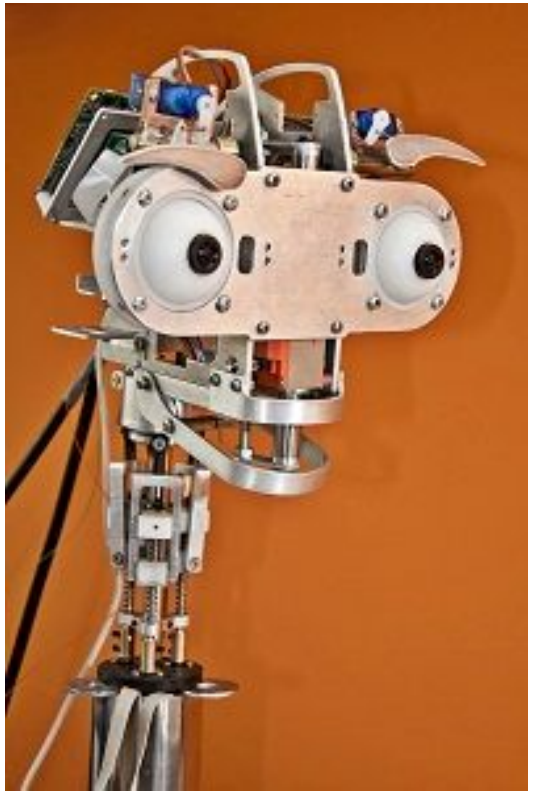
Gualzru

Conversational skills

Gualzru

multi-modal communication

Affective robots - Muecas



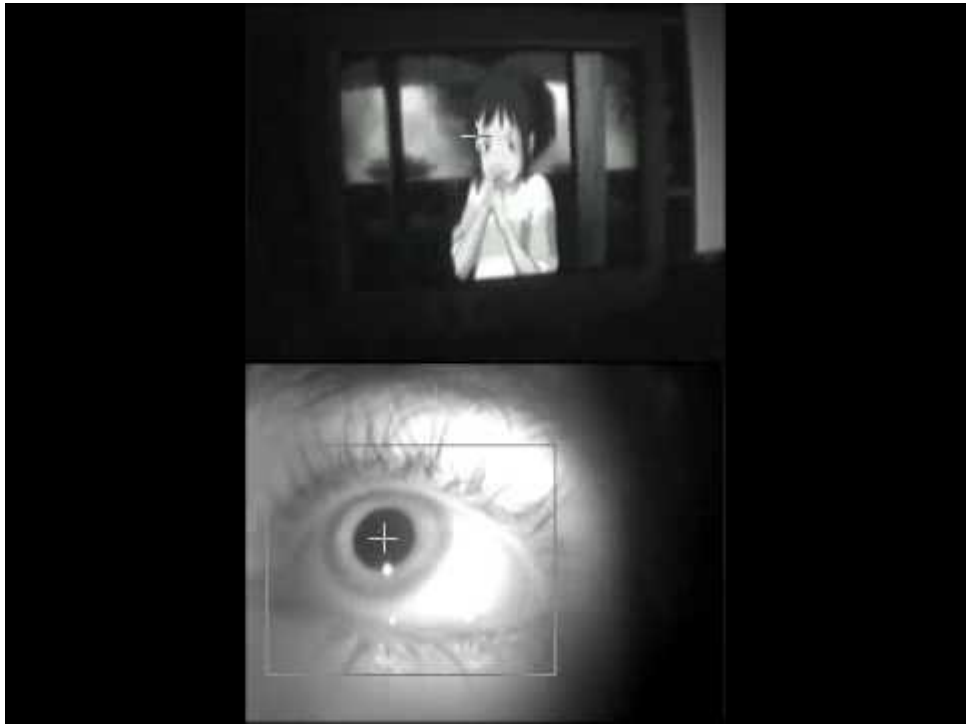
- 13 motors
- Stereo vision
- IMU
- Great for HRI
  - affective robotics
  - visual attention
  - mouth synchronization
  - generation of facial gestures
  - speech augmented with facial expressions

Affective robots - Muecas

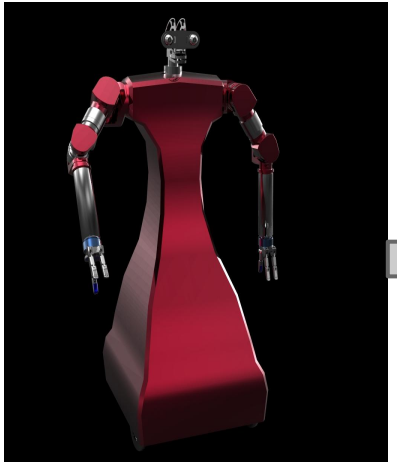
Affective robots - Muecas

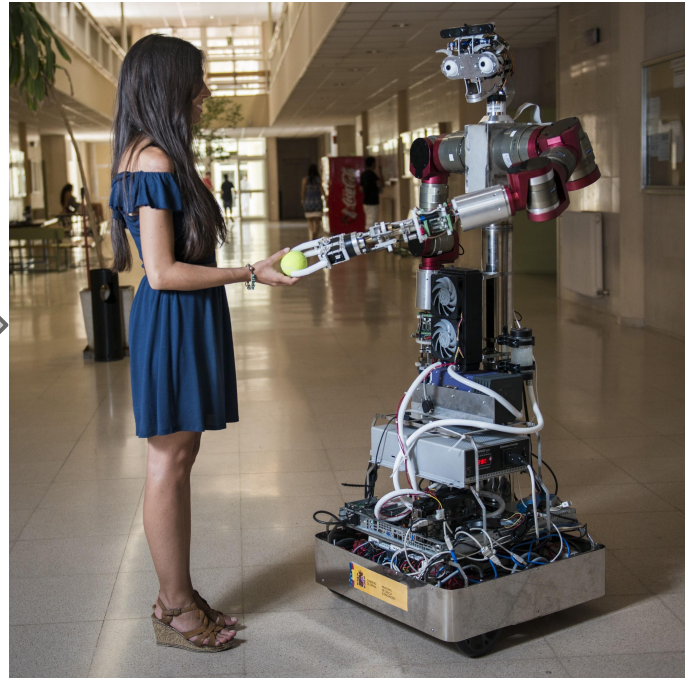
Affective robots - Muecas

Eye tracking (2006)



Social robots - Loki

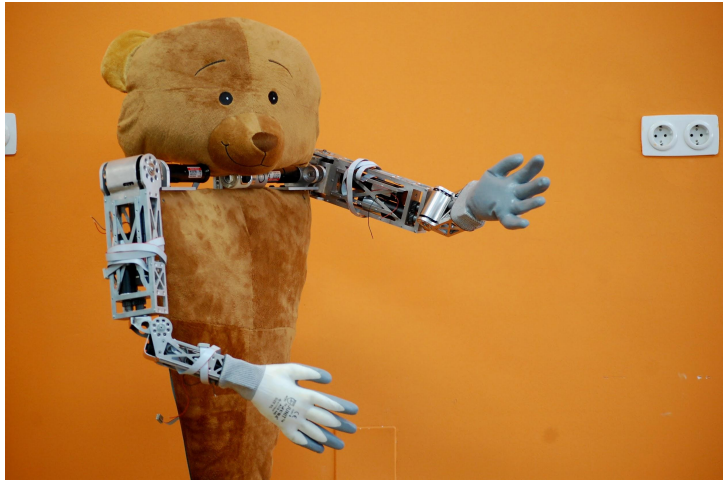




- too expensive
- Schunk motors and Barrett hand, hard to integrate in open source designs

Social robots - Loki

Social robots - Shelly



Social robots - Shelly

Social robots - Shelly

Social robots - Shelly

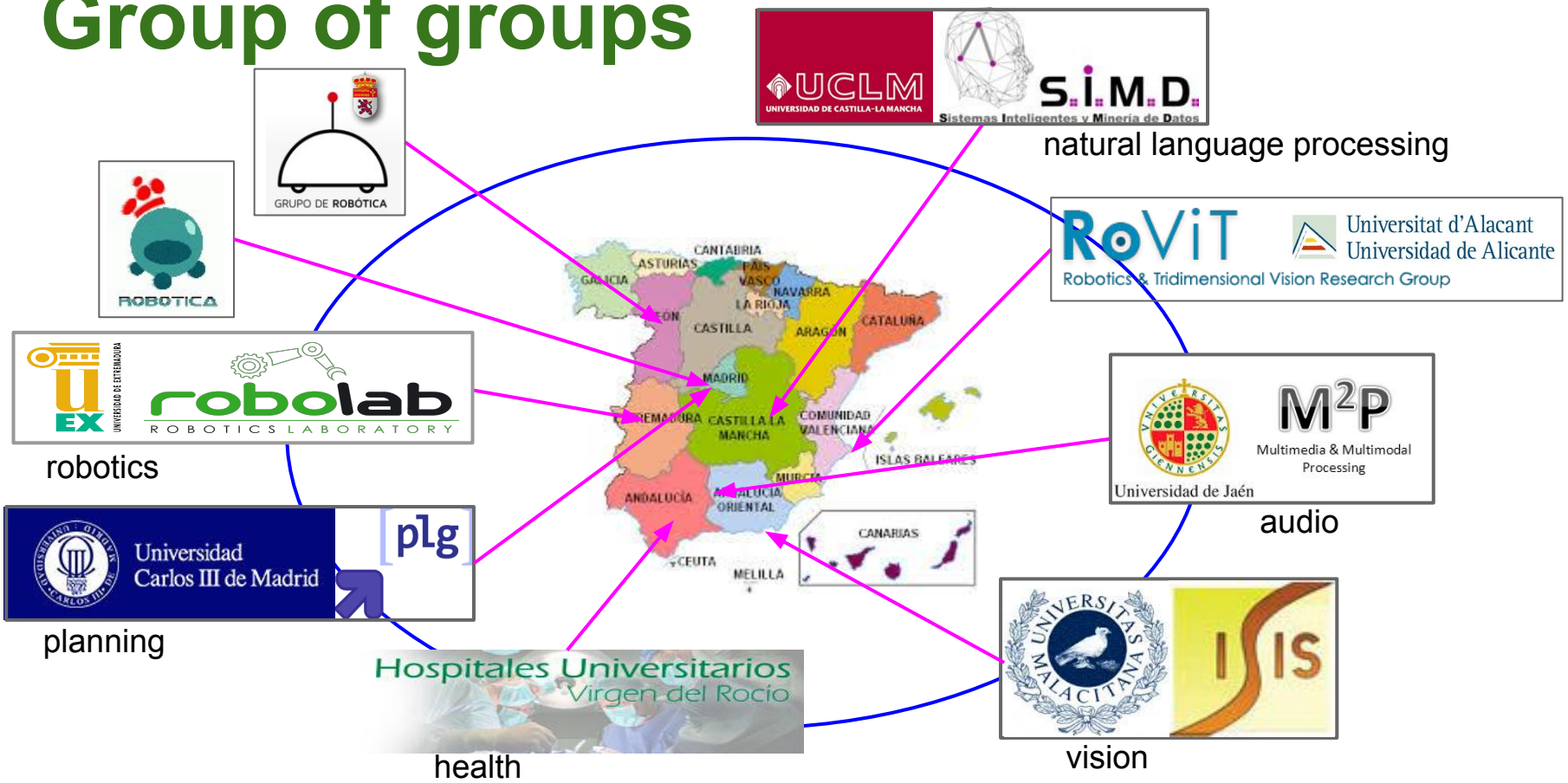
2014 Toulouse

2015 Lisboa

Social Robotics - Autonomy Lab

- 70 m²
- almost ready to live apartment

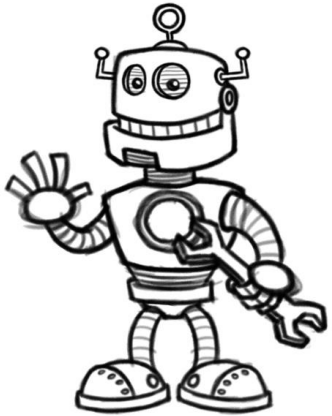
Group of groups



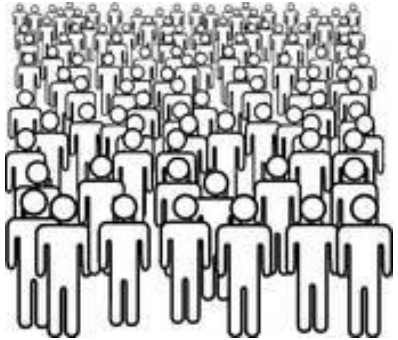
A family of robots

RoboComp

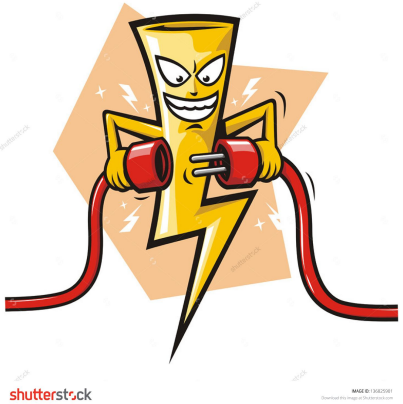
Why is robotics software hard?



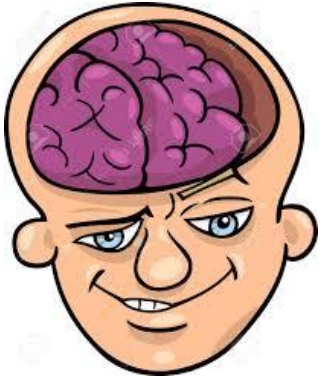
Drivers



People

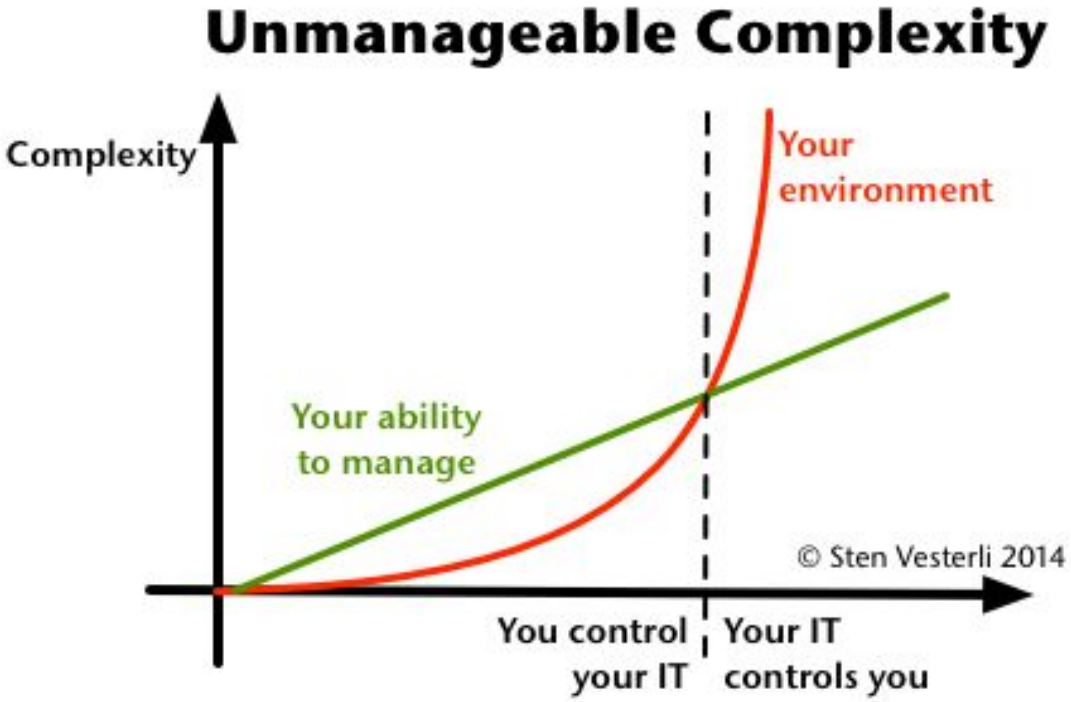


Failure

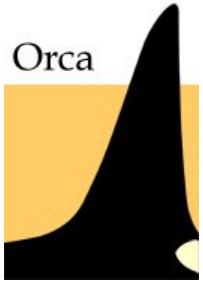


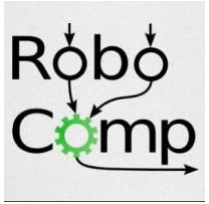
Algorithms

The complexity trap



Robotics frameworks

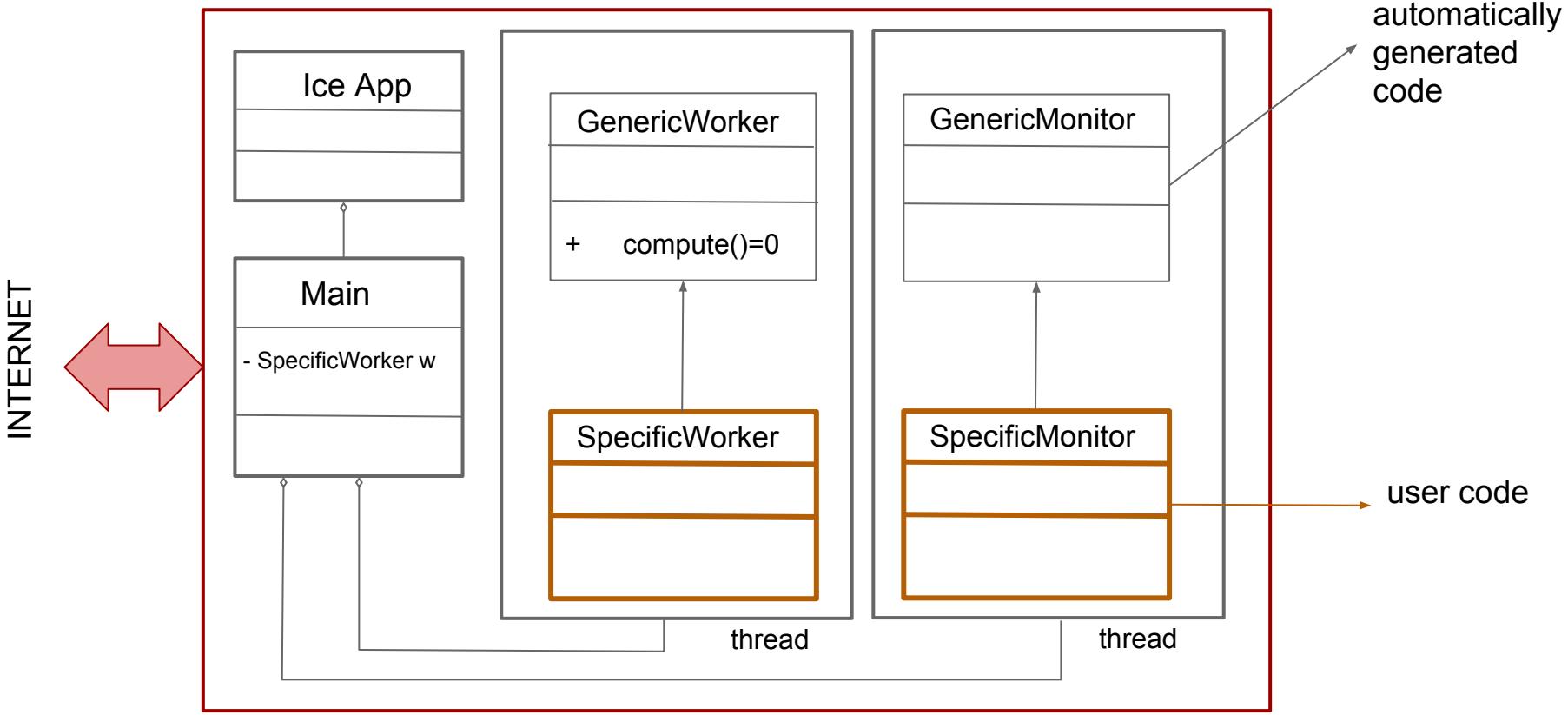




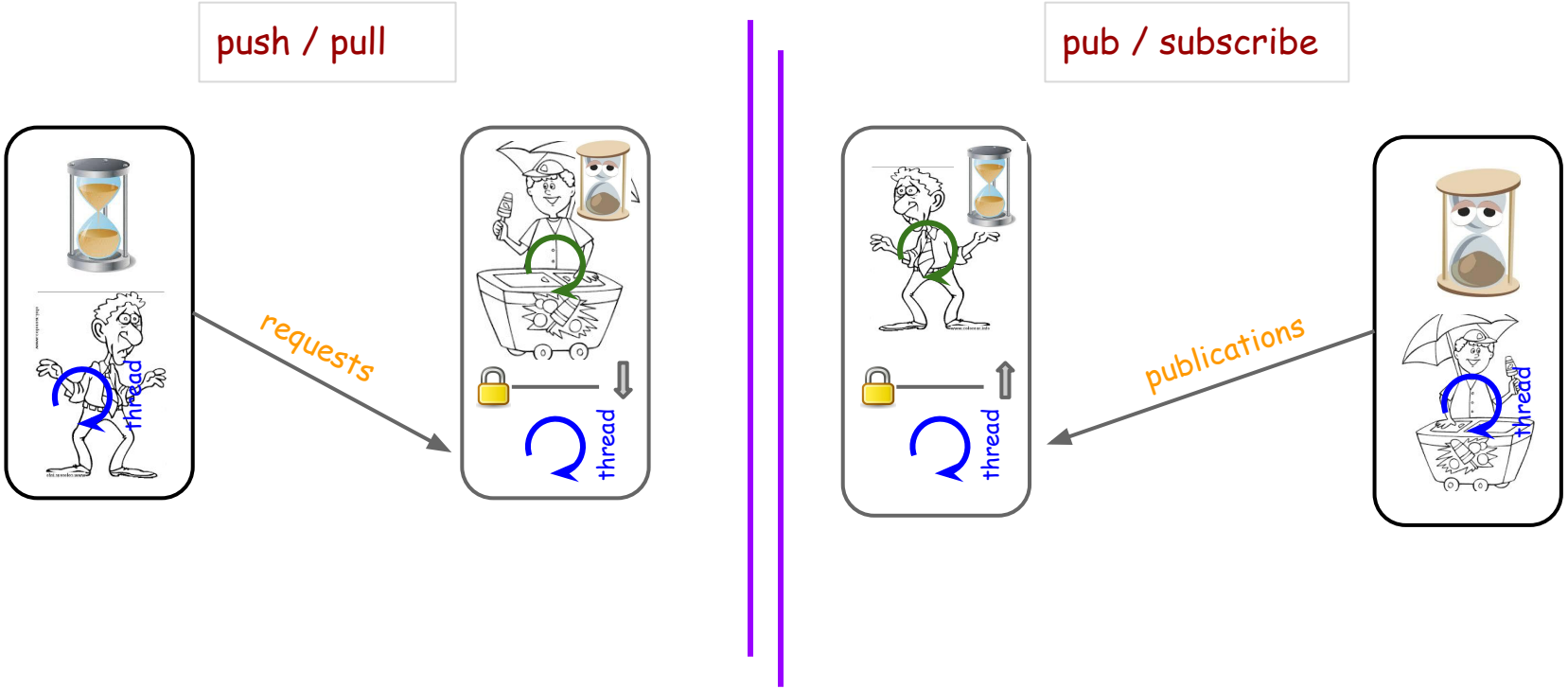
RoboComp

- Distributed, component-oriented
- Formal **component model**
- pub/sub and RPC comm models
- Industrial grade communications middleware (Ice) with **public interfaces**
- **Code generation** using DSLs technology
- Tool set: deploy, monitor, log, simulator
- C++ and Python
- > 100 components
- Talks to ROS using DSL tech.

Component model



Communication models



both can be synchronous or asynchronous

Communication models (code)

push / pull

```
while(True):
     c = proxyA.getData(a,b)
     d = proxyB.update(c)
     do_something_useful( c, d )
```

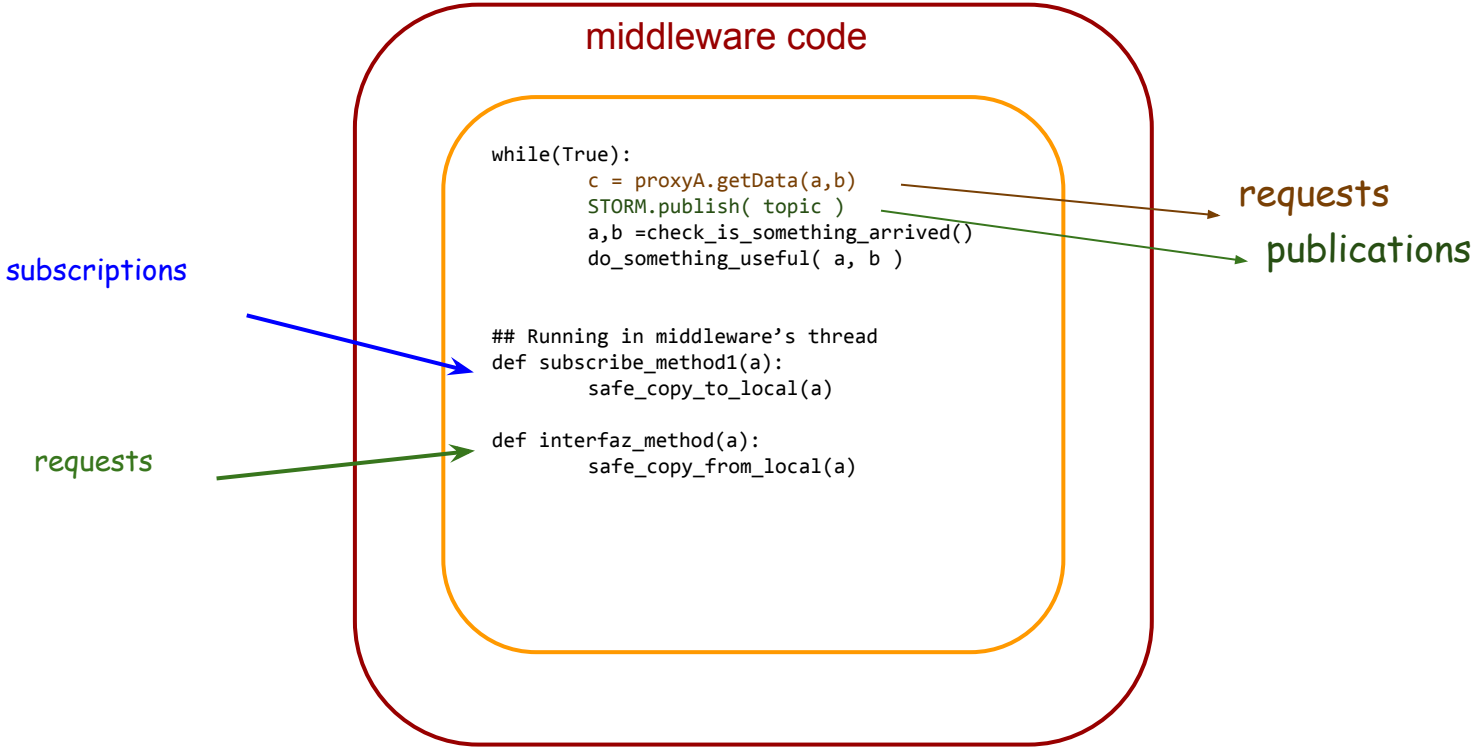
pub / subscribe

```
while(True):
     a,b =check_is_something_arrived()
     do_something_useful( a, b )


## Running in middleware's thread
def subscribe_method1(a):
     safe_copy_to_local(a)

def subscribe_method2(b):
     safe_copy_to_local(b)
```

Communication models (comp view)



Component model: ComponentDSL

```
import "/robocomp/interfaces/IDSLs/interface1.idsl";
import "/robocomp/interfaces/IDSLs/interface2.idsl";

Component mycomponent
{
    Communications
    {
        implements interface1;
        requires interface2;
        subscribesTo topicToSubscribeTo;
        publishes topicToPublish;
    };
    language Cpp;
    gui Qt(QWidget);
    statemachine statemachine.smdsl;
};
```

Component model: InterfazDSL

```
module RoboCompDifferentialRobot {
 exception HardwareFailedException { string what; };

 struct TBaseState {
  float x;
  float correctedX;
  float z;
  float correctedZ;
  float alpha;
  float correctedAlpha;
  float advV;
  float rotV;
  float adv;
  float rot;
  bool isMoving;
  float voltage;
 };
```

```
interface DifferentialRobot
 {
  void getBaseState(out TBaseState state) throws HardwareFailedException;
  void getBasePose(out int x, out int z, out float alpha) throws  HardwareFailedException;
  void setSpeedBase(float adv, float rot) throws HardwareFailedException;
  void stopBase() throws HardwareFailedException;
  void resetOdometer() throws HardwareFailedException;
  void setOdometer(TBaseState state) throws HardwareFailedException;
  void setOdometerPose(int x, int z, float alpha) throws HardwareFailedException;
  void correctOdometer(int x, int z, float alpha) throws HardwareFailedException;
 };
};
```

.idsl file

Ice provides binaries to generate proxies for different languages

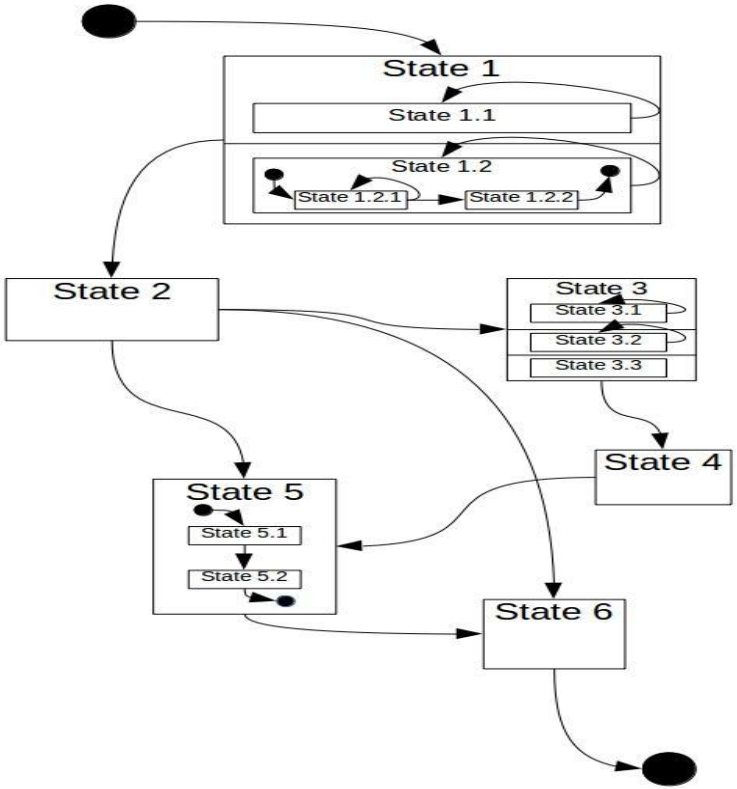
Component model: StateMachineDSL

```
Name_machine
{
    [ states name_state *[, name_state];]
    [ initial_state name_state;]
    [ end_state name_state;]
    [ transition
    {
        name_state => name_state *[, name_state];
         *[name_state => name_state *[, name_state];]
    };]
};
```

```
[:parent_state [parallel]
{
    states name_state *[, name_state];
    [ initial_state name_state;]
    [ end_state name_state;]
    [ transition
     {
        name_state => name_state *[, name_state];
        *[name_state => name_state *[, name_state];]
     };
   ]
};]
```

Component model: StateMachineDSL

```
MyMachine {
    states State2, State3, State4, State5;
    initial_state State1;
    end_state State6;
    Transition  {
        State1 => State1, State2;
        State2 => State3, State5, State6;
        State3 => State3, State4;
        State4 => State5;
        State5 => State6;
    };
};
:State1 parallel {
    states State11, State12;
    Transition {
    State11 => State11;
    State12 => State12;
    };
};
……..
```



DSL Code Generator

user@machine: **robocompcdsl** mycomponent.cdsl

.idsl files

.cdsl file

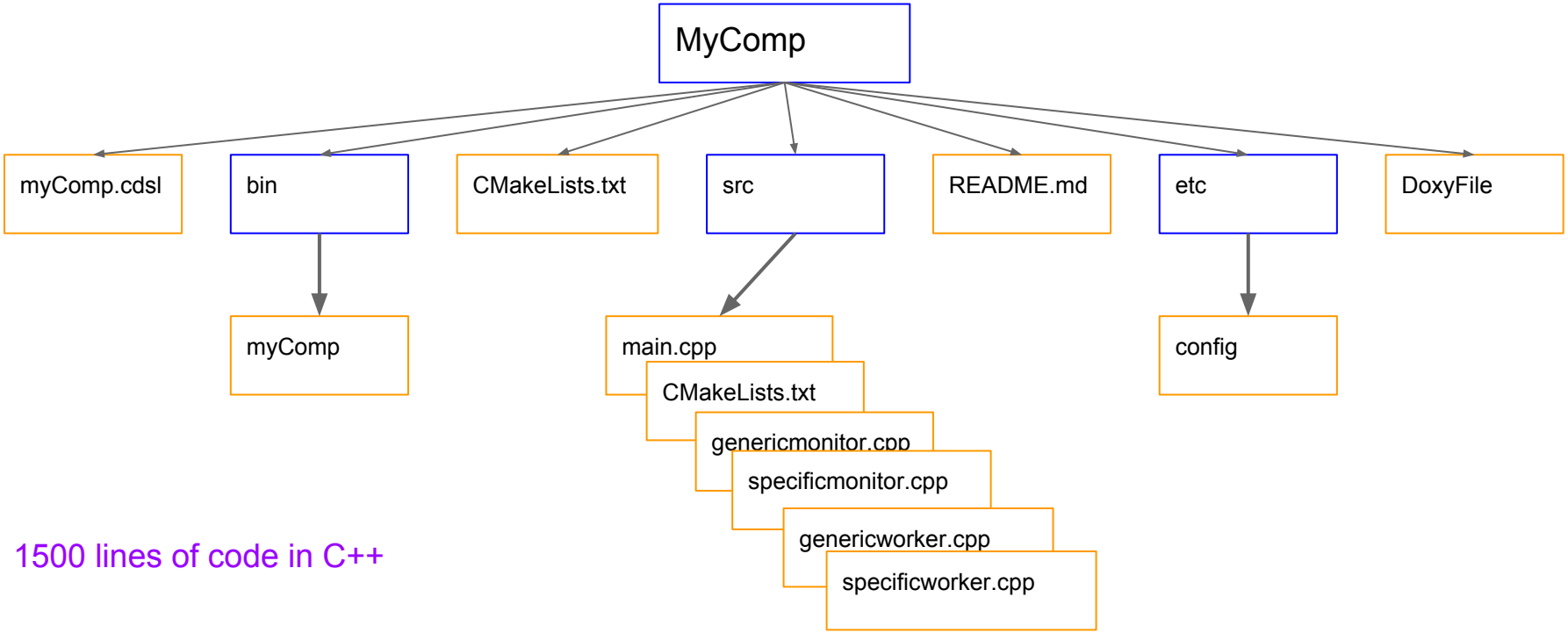
communication primitives

```
import "/robocomp/interfaces/IDSLs/AprilTags.idsl";
import "/robocomp/interfaces/IDSLs/AprilBasedLocalization.idsl";
import "/robocomp/interfaces/IDSLs/Laser.idsl";
import "/robocomp/interfaces/IDSLs/NewAprilPose.idsl";

Component MyComp
{
        Communications
        {
                requires Laser;
                implements NewAprilPose;
                subscribesTo AprilTags;
                publishes AprilBasedLocalization;
        };

        language Cpp;
        statemachine statemachine.smdsl;
};
```

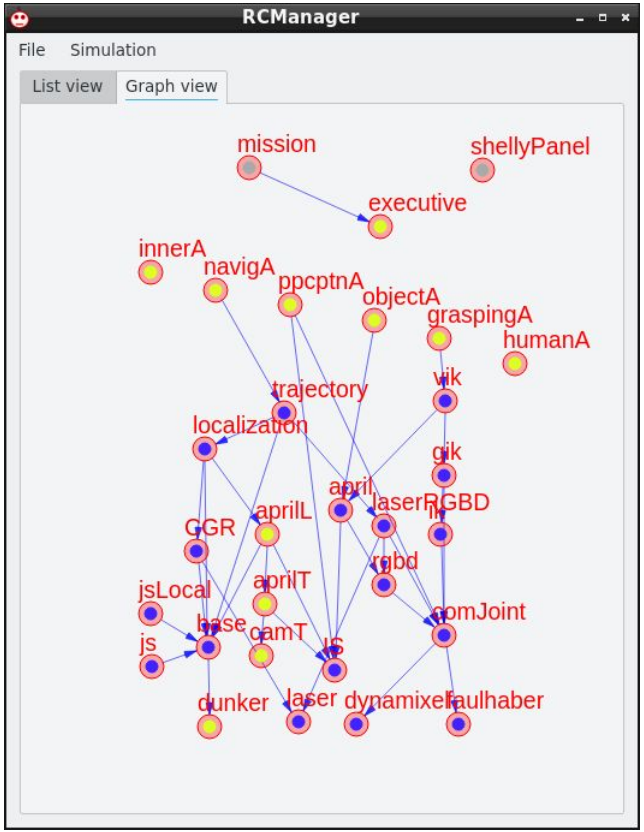
… generates a code directory



1500 lines of code in C++

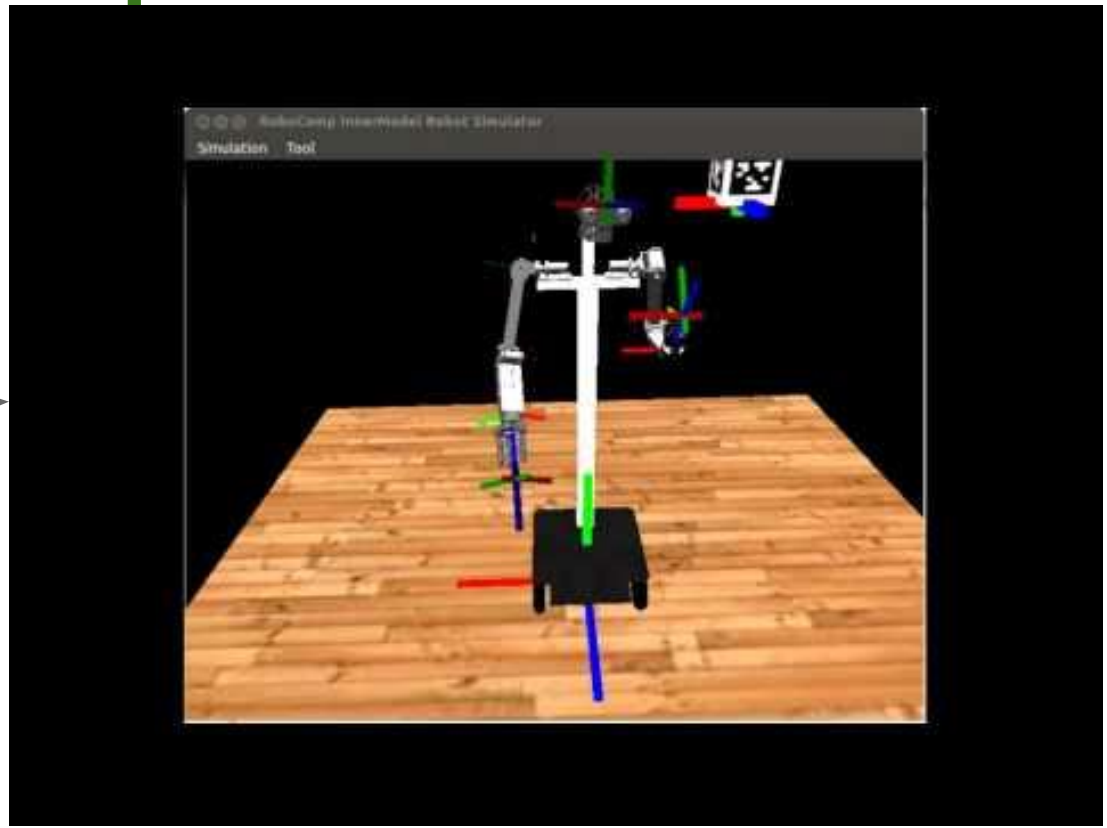
RoboComp Tools

RoboComp's other tools: RCManager



RoboComp other tools: 3D Simulator

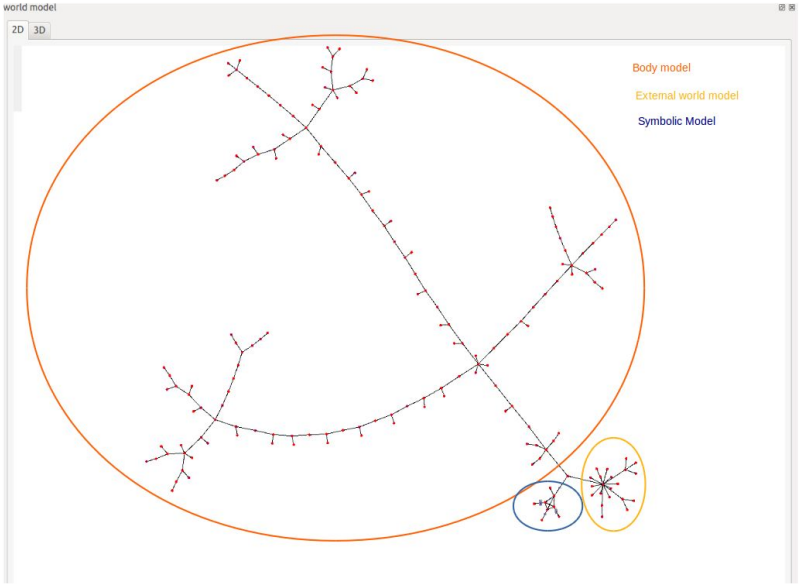
Based on the **Open Scene Graph** engine



Example: testing generalized inverse kinematics using LM non-linear optimization

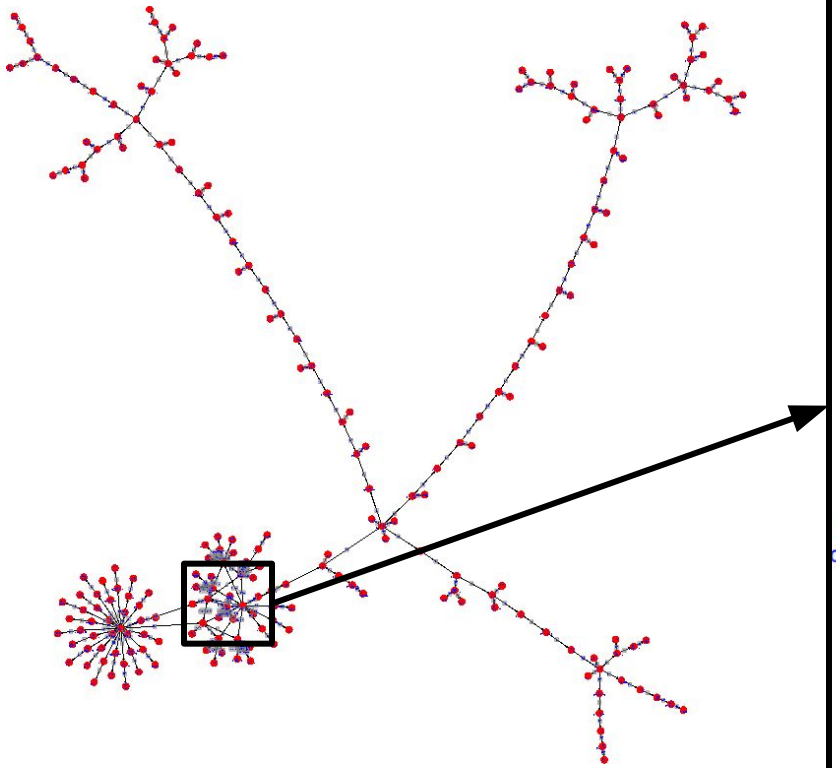
CORTEX

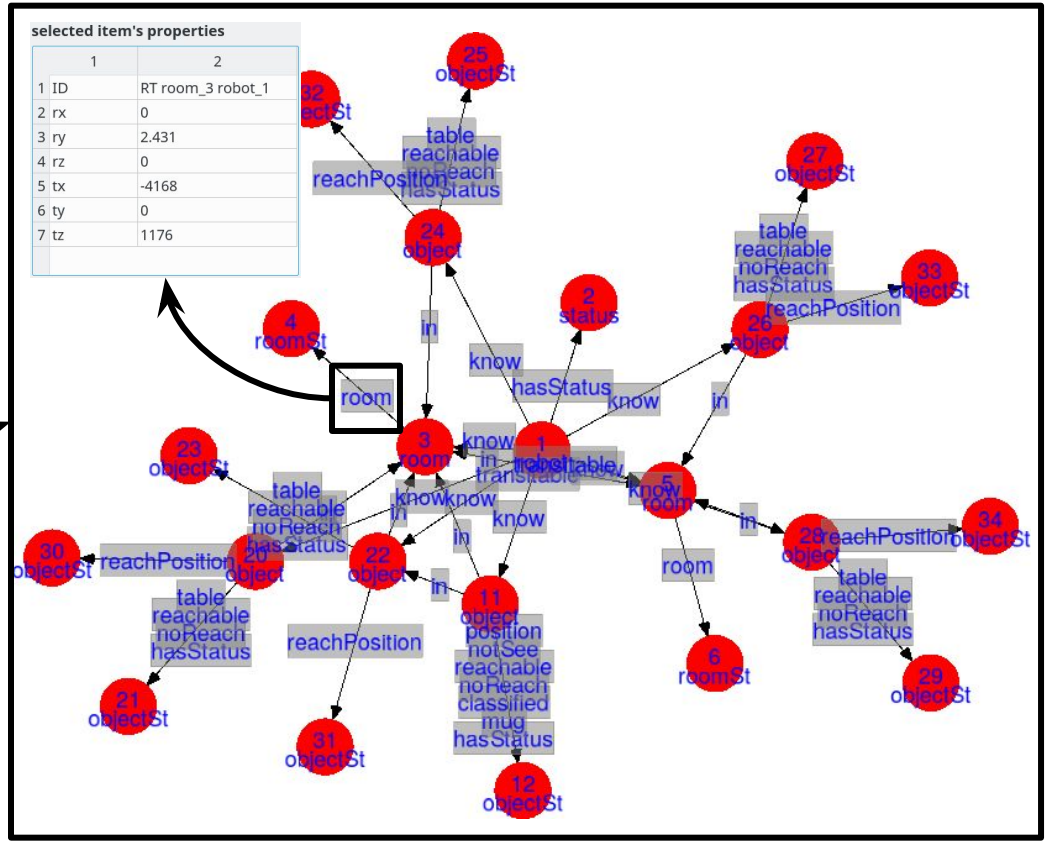
CORTEX: a proposal for a cognitive architecture



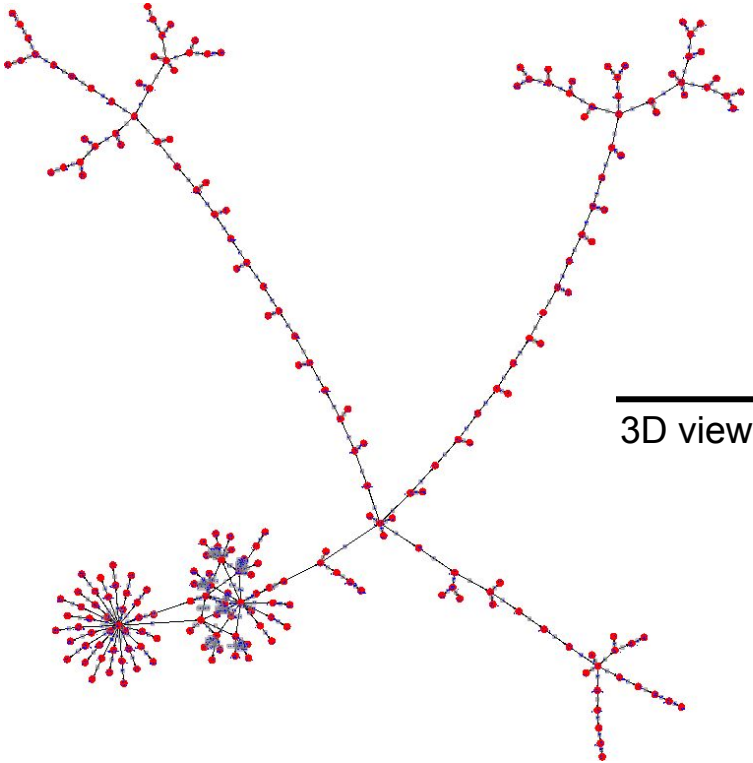
A graph-based representation of the robot and the world, shared by a set of reactive-deliberative agents

RoboComp tools: DeepStateRepresentation

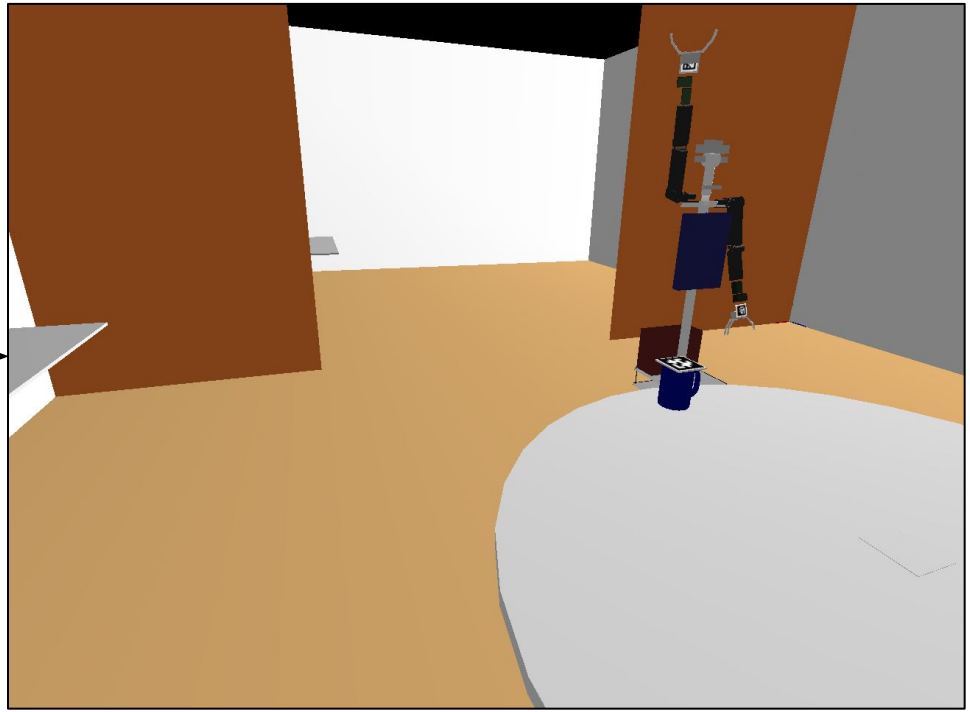




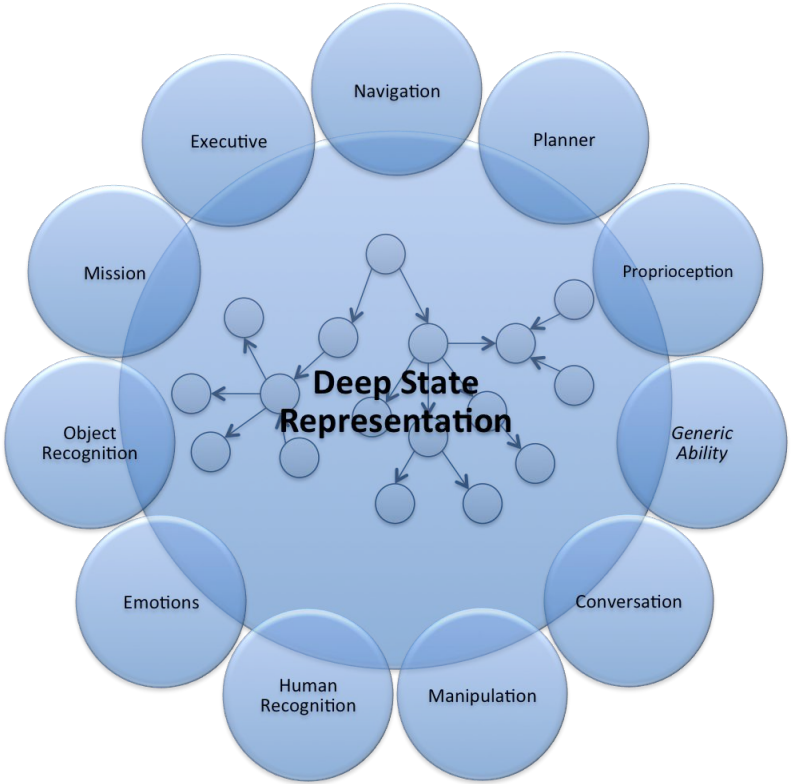
RoboComp tools: DeepStateRepresentation



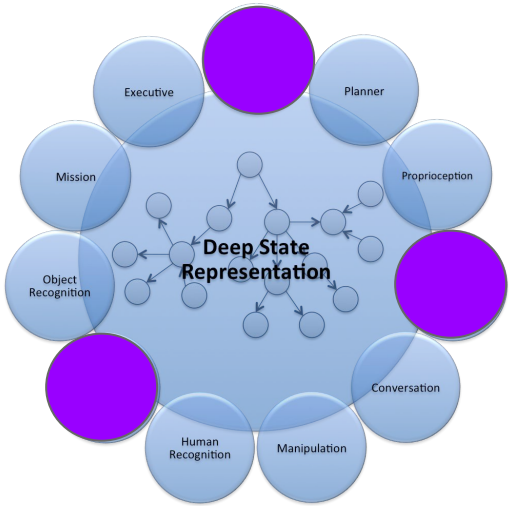
3D view



CORTEX: a proposal for a cognitive architecture

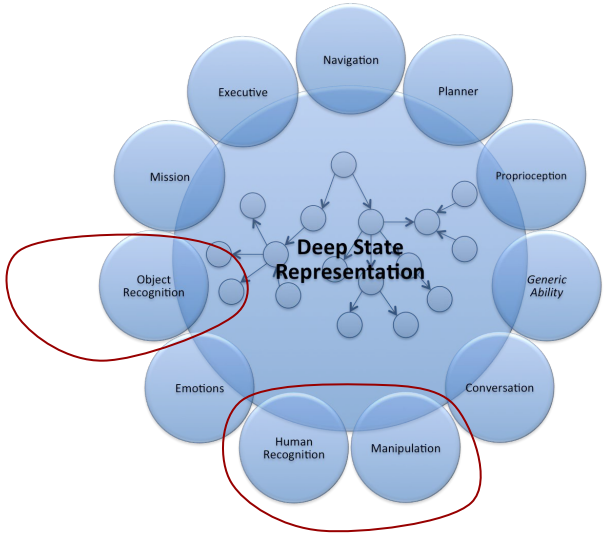


CORTEX: a proposal for a cognitive architecture



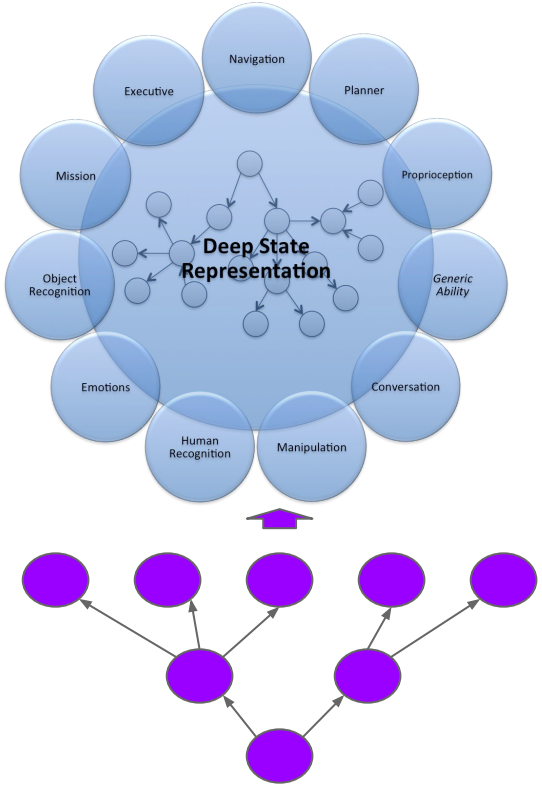
- ¿What is the proper division of functionalities?

CORTEX: a proposal for a cognitive architecture



- ¿What is the proper division of functionalities?
- ¿What is the correct choice of agents?

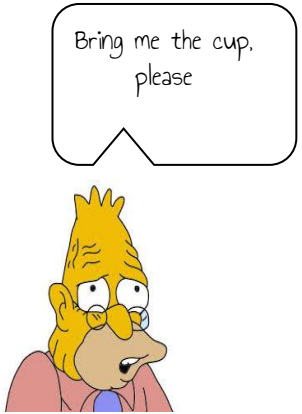
CORTEX: a proposal for a cognitive architecture

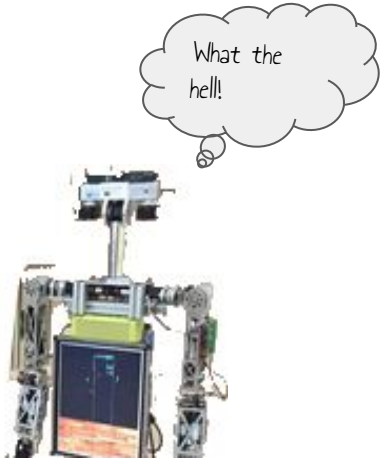


- ¿What is the proper division of functionalities?
- ¿What is the correct choice of agents?
- ¿Is developmental cognition the only way?

… a real world problem

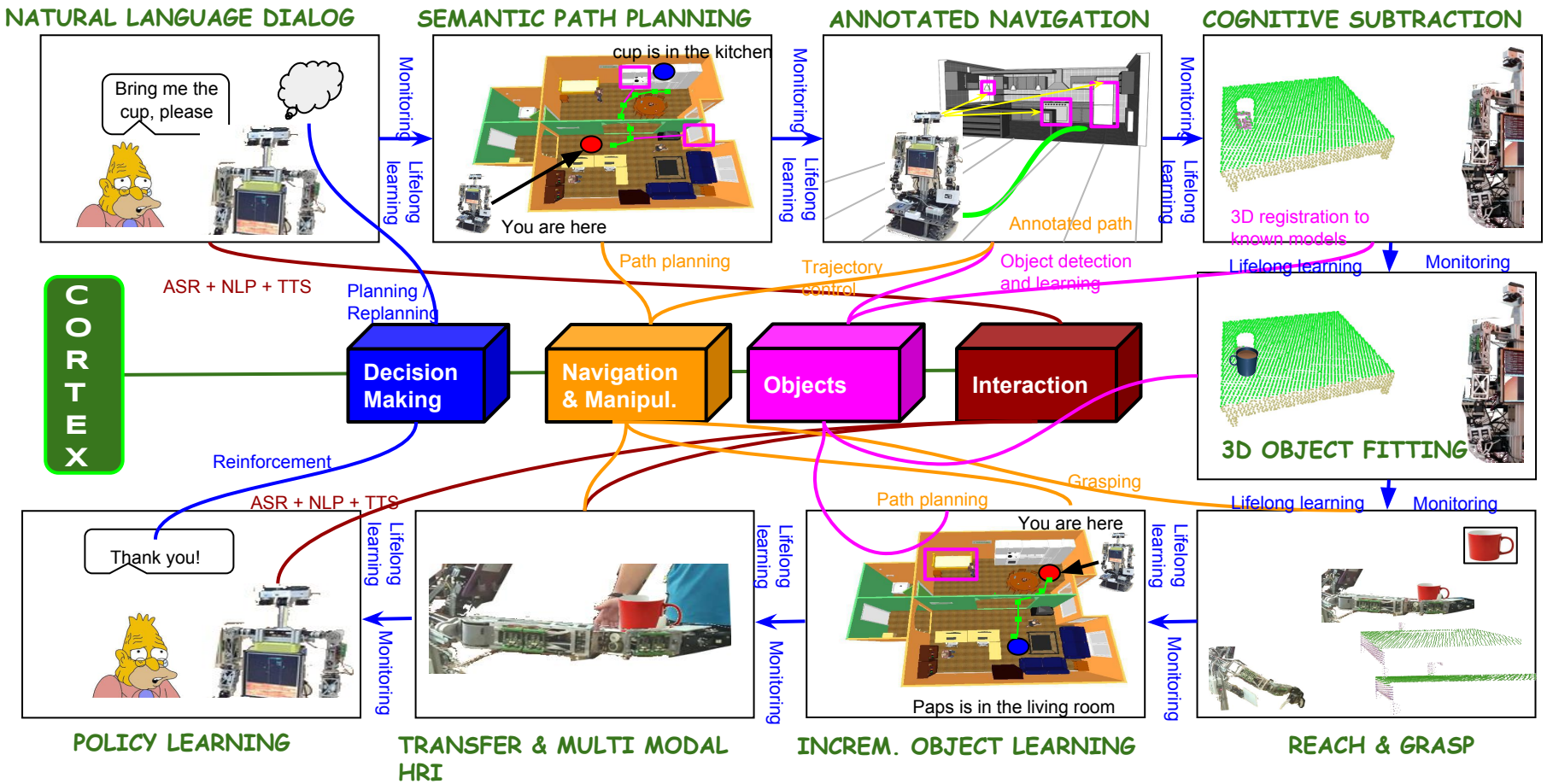
The BringMe(X) function






robolab
x4

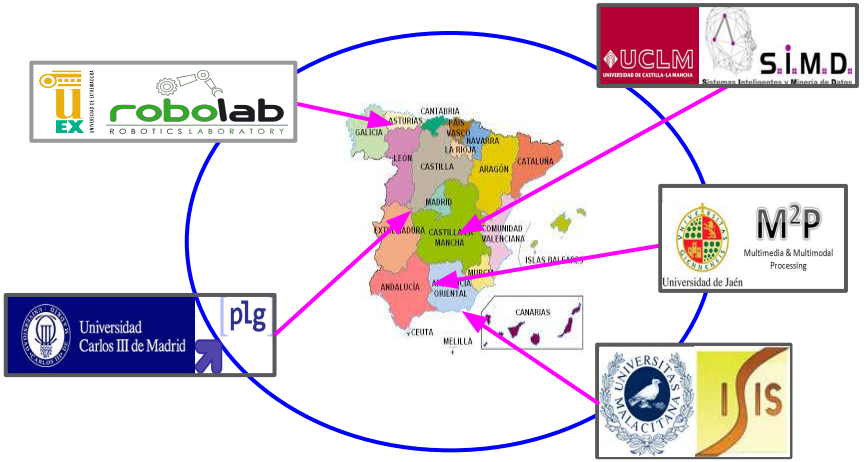
The problem


NATURAL LANGUAGE DIALOG
Bring me the cup, please
Monitoring
Lifelong learning
SEMANTIC PATH PLANNING
cup is in the kitchen
You are here
Monitoring
Lifelong learning
ANNOTATED NAVIGATION
Annotated path
Monitoring
Lifelong learning
COGNITIVE SUBTRACTION
3D registration to known models
Lifelong learning
Monitoring
3D OBJECT FITTING
Lifelong learning
Monitoring
CORTEX
ASR + NLP + TTS
Planning / Replanning
Decision Making
Navigation & Manipul.
Objects
Interaction
Path planning
Trajectory control
Object detection and learning
Grasping
Reinforcement
ASR + NLP + TTS
Thank you!
Lifelong learning
Monitoring
POLICY LEARNING
Lifelong learning
Monitoring
TRANSFER & MULTI MODAL HRI
Path planning
You are here
Paps is in the living room
Lifelong learning
Monitoring
INCREM. OBJECT LEARNING
Lifelong learning
Monitoring
REACH & GRASP

New project

LifeBots: Lifelong Technologies for Social Robots in Smart Homes

MINECO
Retos de la Sociedad
2016-2019



Thanks