Design Of A Modular Architecture Using ROS : The Pepper Case

RoboBreizh Team

A. Dizet, C. Le Bono, A. Legeleux, M. Neau, C. Buche

Lab-STICC, ENIB

30th September 2020

RoboBreizh team

Teams

- 1 Full professor
- 2 PhD students
- 2 undergraduates students

Competition

- RoboCup@Home Education Online Challenge (@HomeEDU): a good start before the RoboCup@Home competition
- Award: Best Performance Award in Open Platform

RoboBreizh team

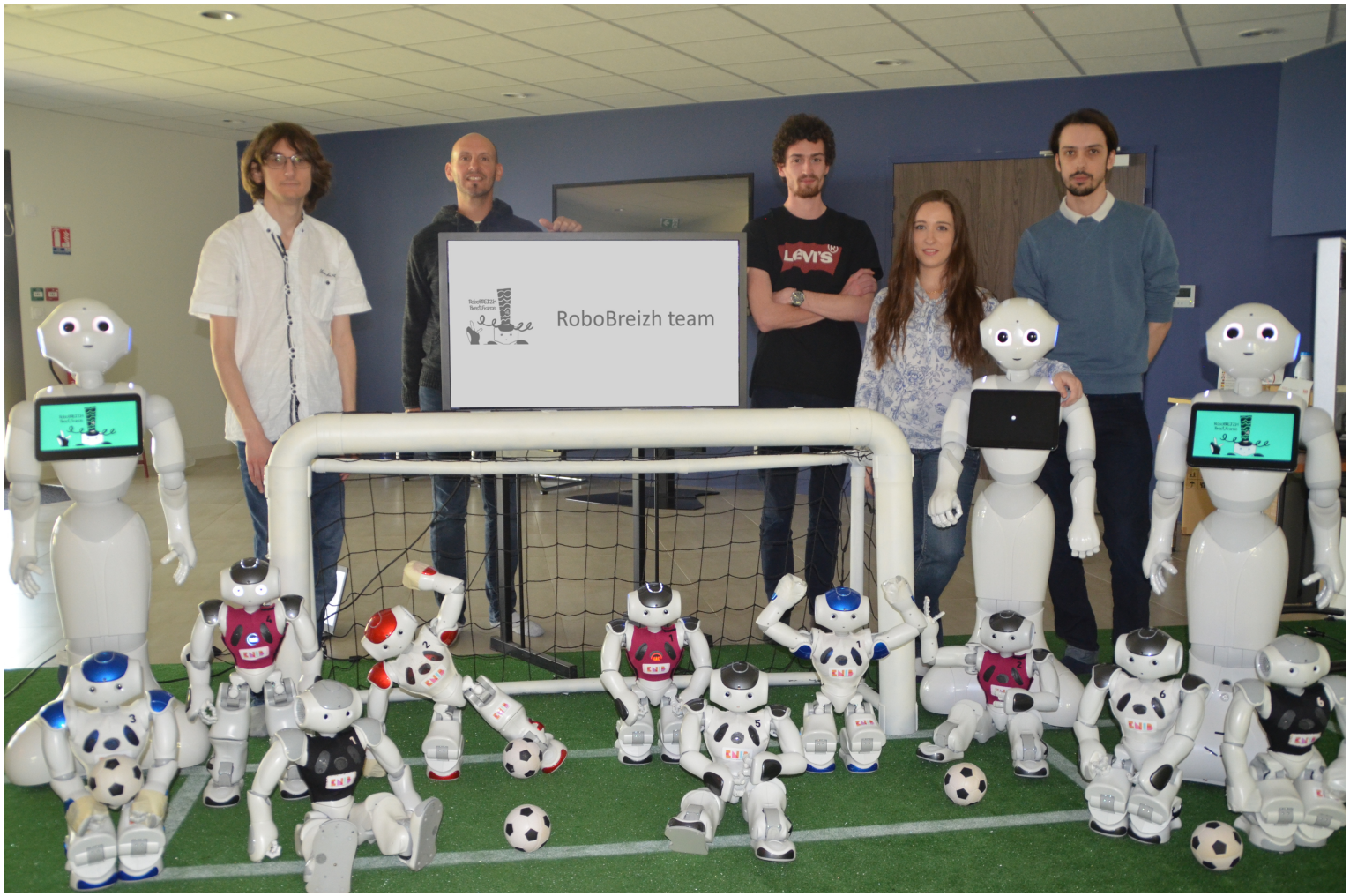


Figure 1: Our team with Pepper and Nao robots

Table of Contents

Architecture design

Examples of required capabilities for @HomeEDU

- Recognize human
- Recognize objects
- Navigate safely
- Understand basics commands
- Take tiny objects
- Bonus : act natural

Architecture design

From needs to functionalities: a Modular Architecture

Modules are good for:

- Division of work
- Behaviour control

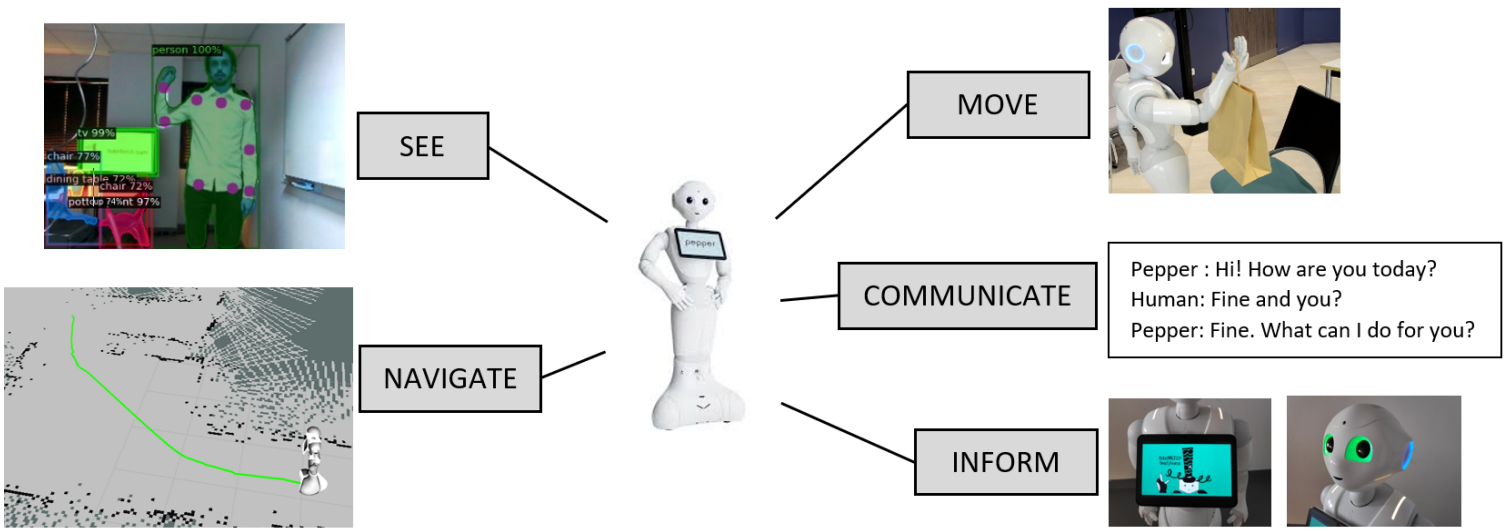


Figure 2: Needed robot's skills

Architecture design

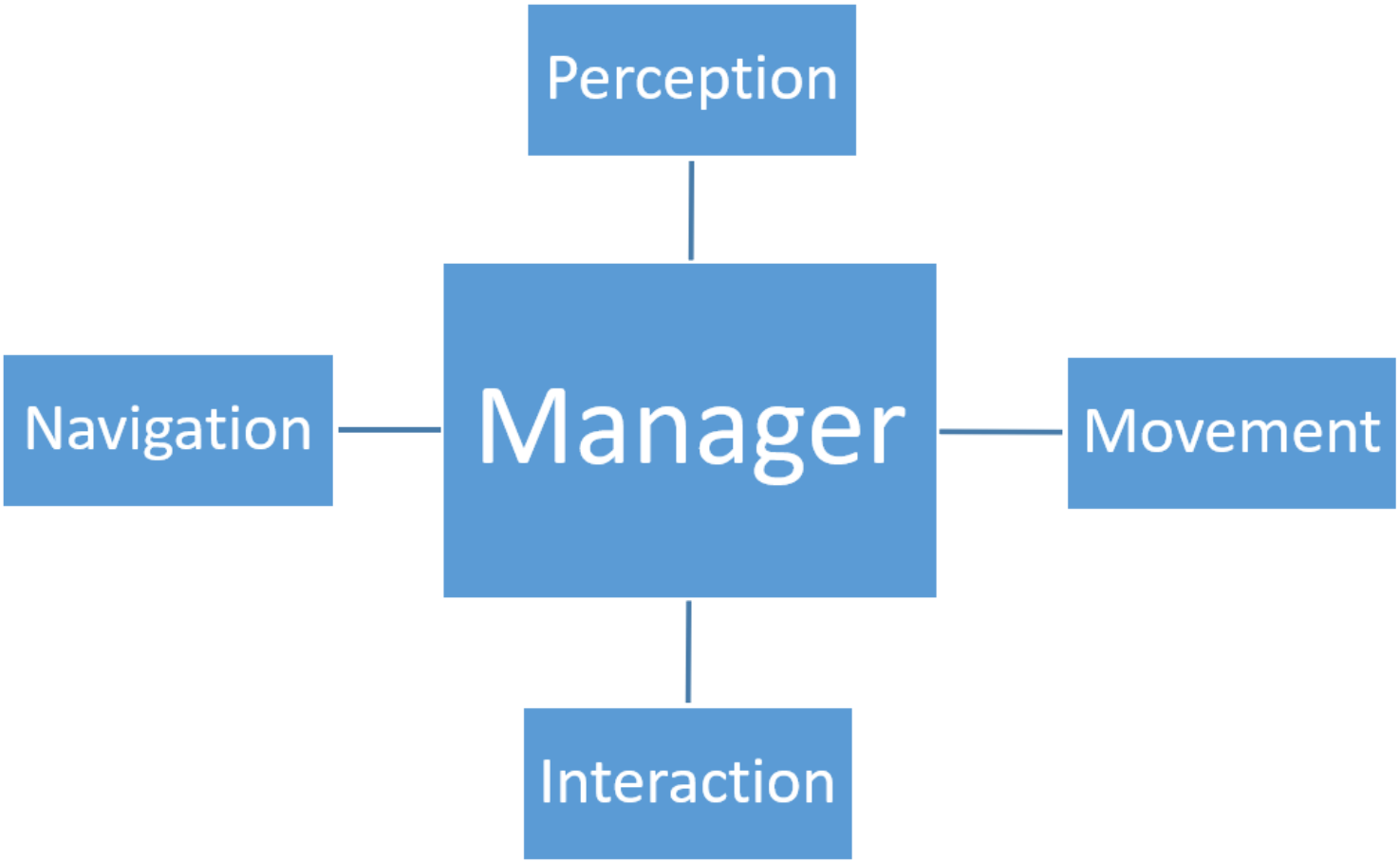


Figure 3: **First step**: split into modules

Architecture design

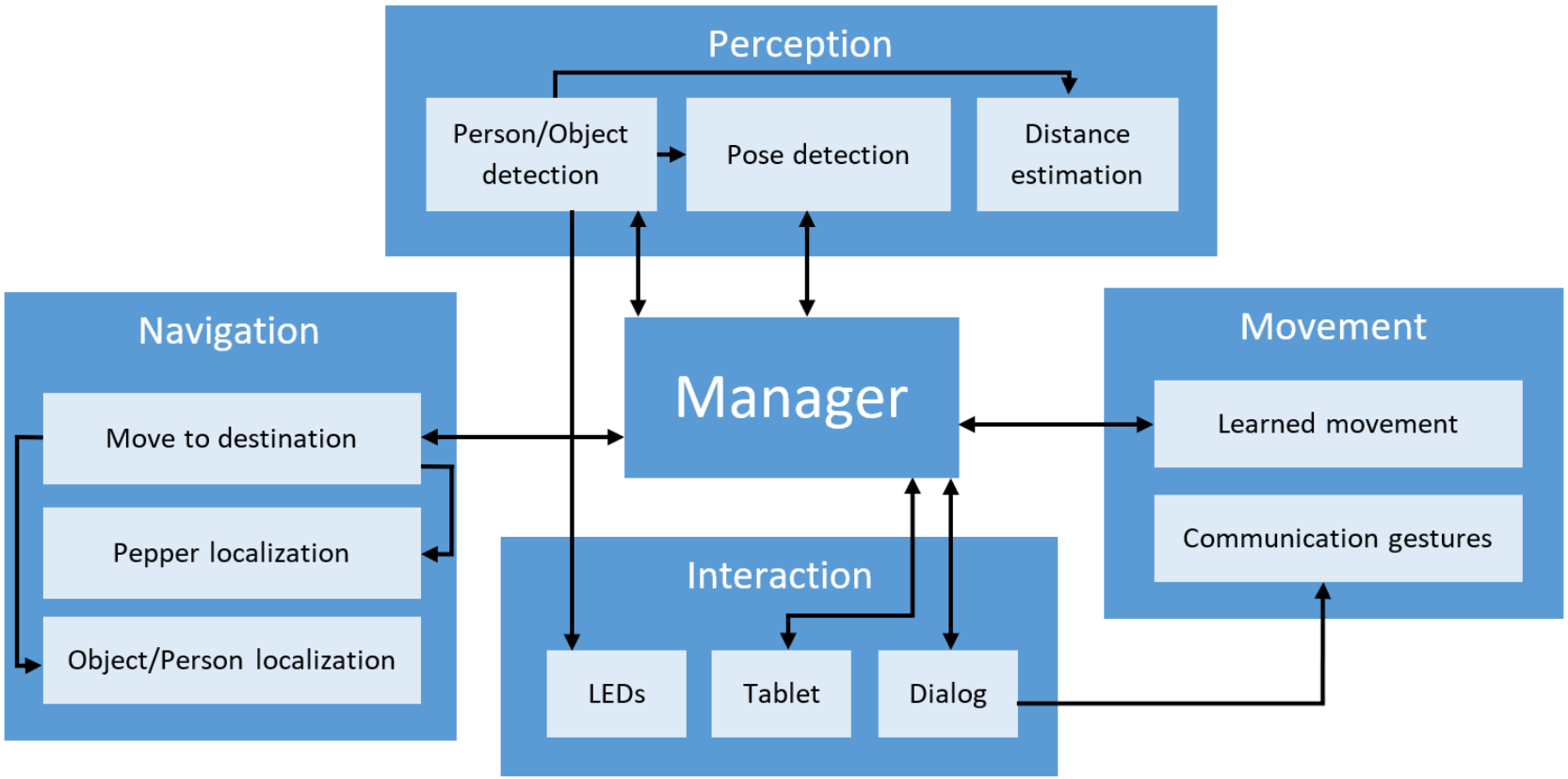


Figure 4: **Second step**: precise interactions between modules

Architecture design

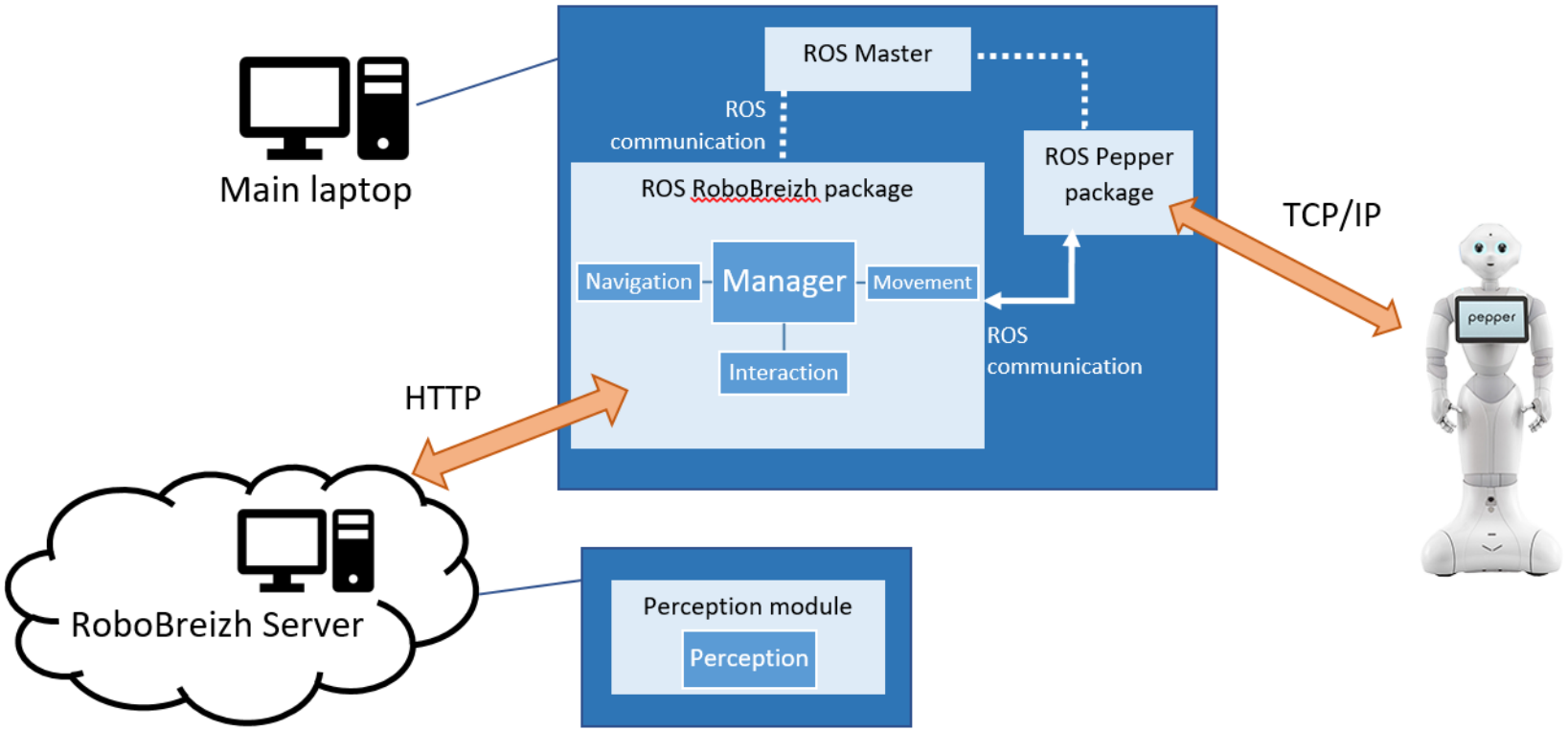


Figure 5: **Third step**: wrap it up

Developments during lockdown period

Use of a simulator

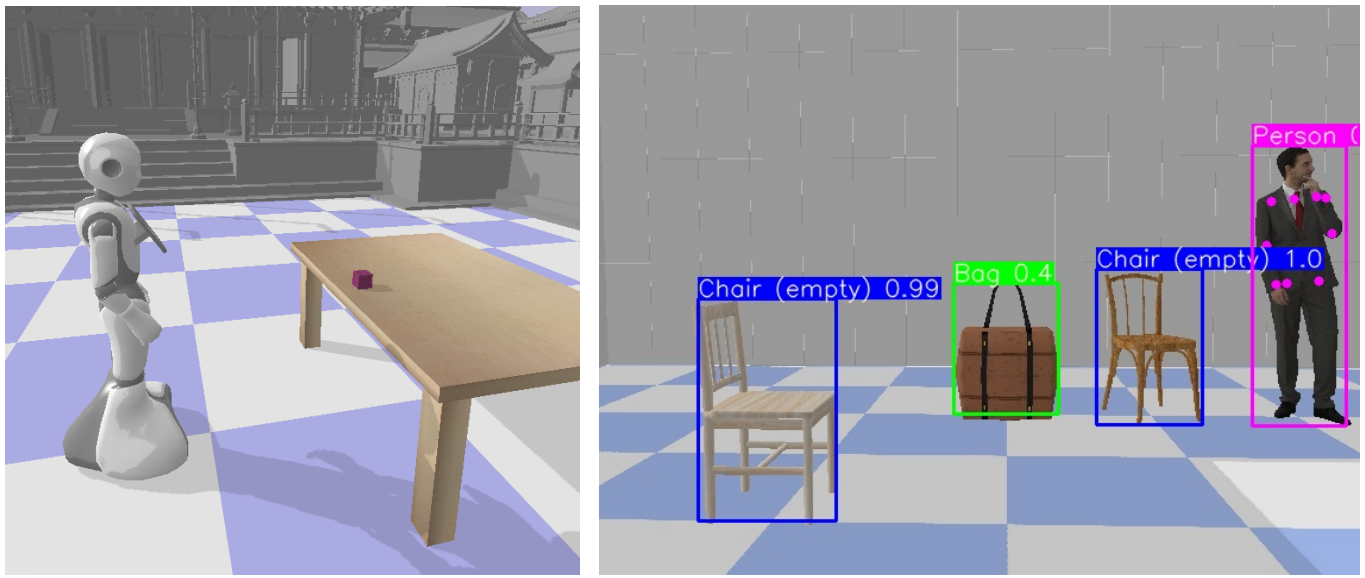


Figure 6: qiBullet simulator [?] used to develop our first algorithms

ROS Basics

How to control every robot the same way?

Robot Operating System

ROS Basics

ROS: Robot Operating System

- Created by Willow Garage in 2007
- An open-ended collaboration framework
- Over 150 compatible robots
- Over 3,000 available packages
- A large community

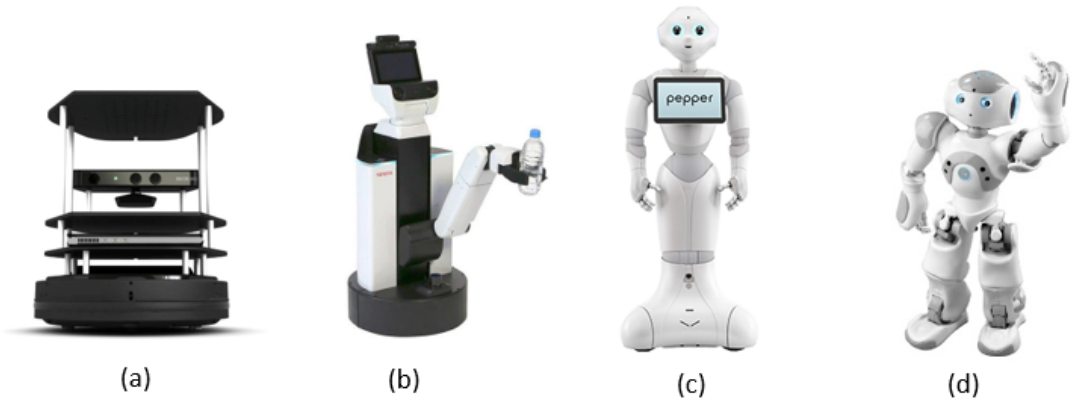


Figure 7: Example of compatible robots: (a) TurtleBot, (b) HSR (c) Pepper, (d) Nao

ROS Basics

Architecture of ROS communication:

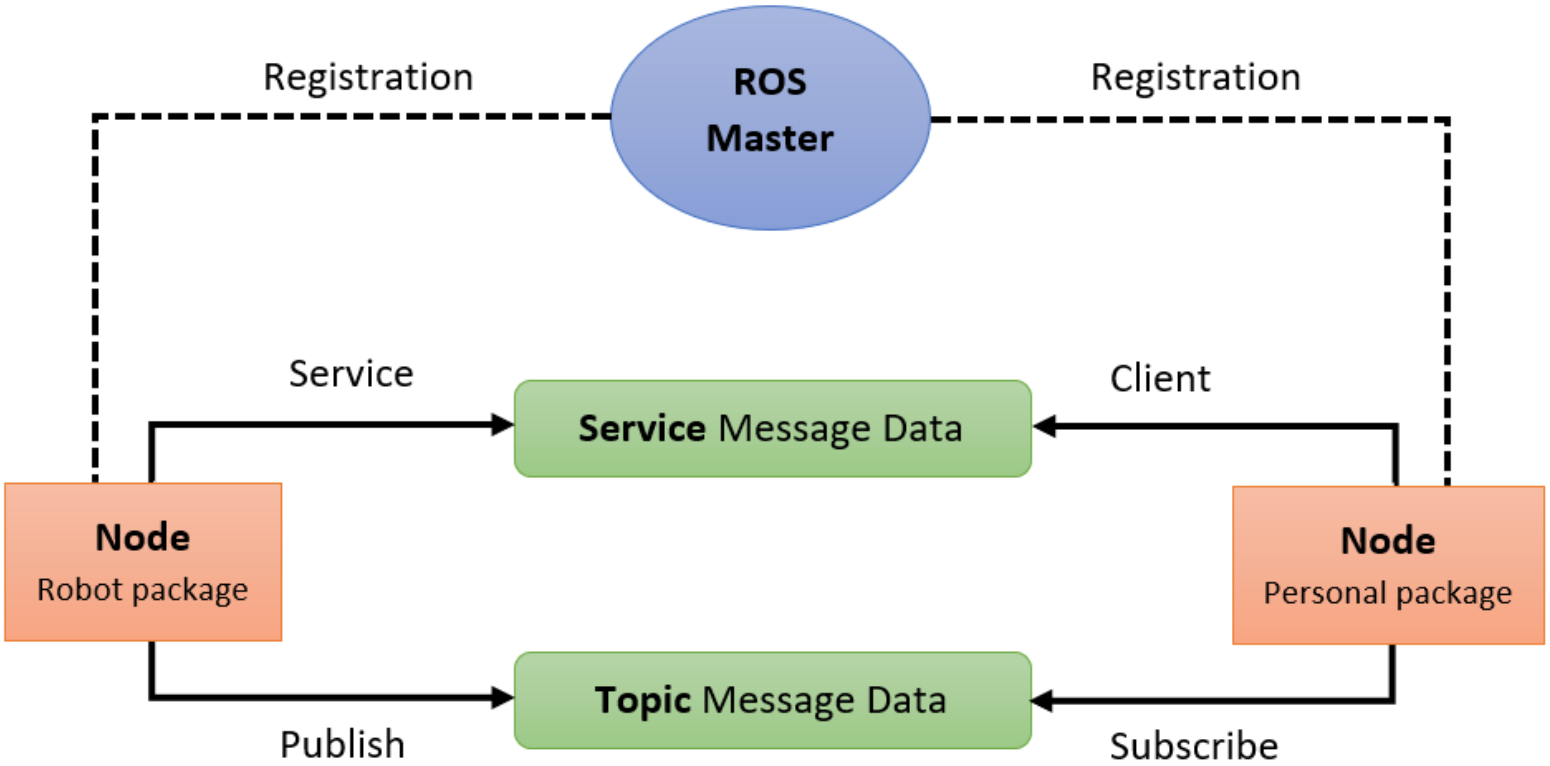


Figure 8: Communication infrastructure

ROS Basics

Lots of tools:

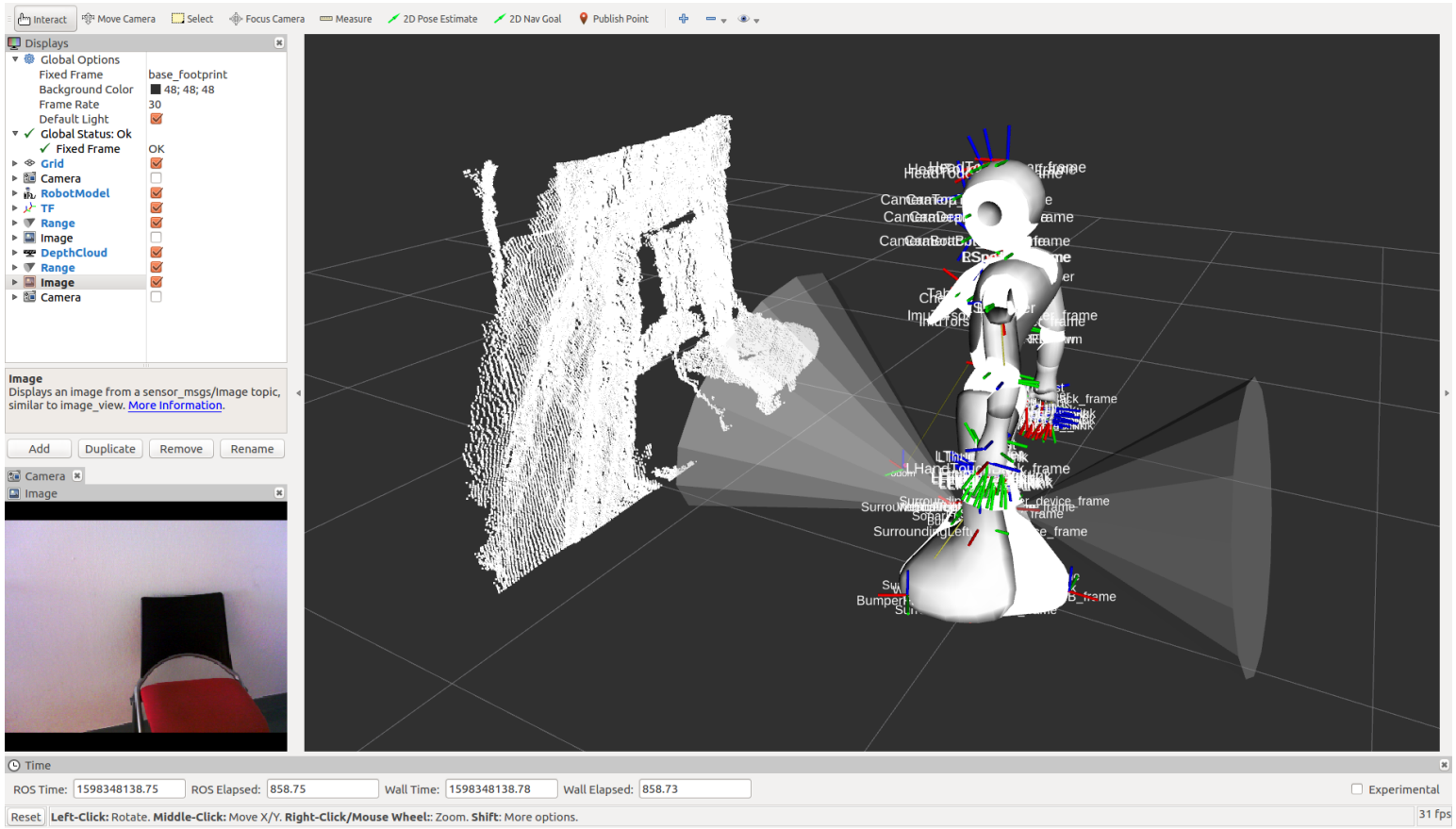
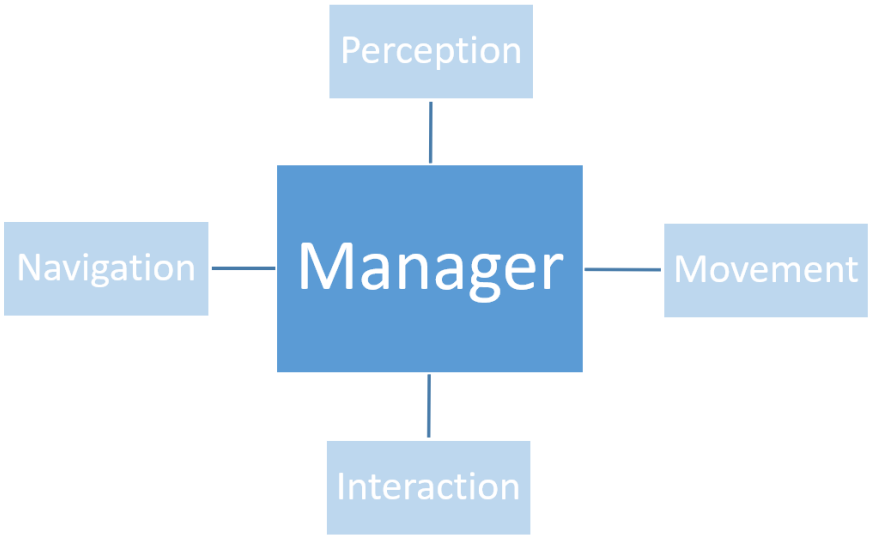


Figure 9: **RViz**: 3D visualization of many sensors

ROS Basics

Discussion

- ✔ Open-source
- ✔ Standard framework for multiple robots
- ✔ A large community
- ✔ Lots of available packages and tutorials (`http://wiki.ros.org/ROS/Tutorials`)
- ✘ May not have all possibilities given by the native API of the robot
- ✘ Hard to understand for beginner in computer science
- ✘ Asynchronous communication: not adapted to a sequential scenario



The Manager

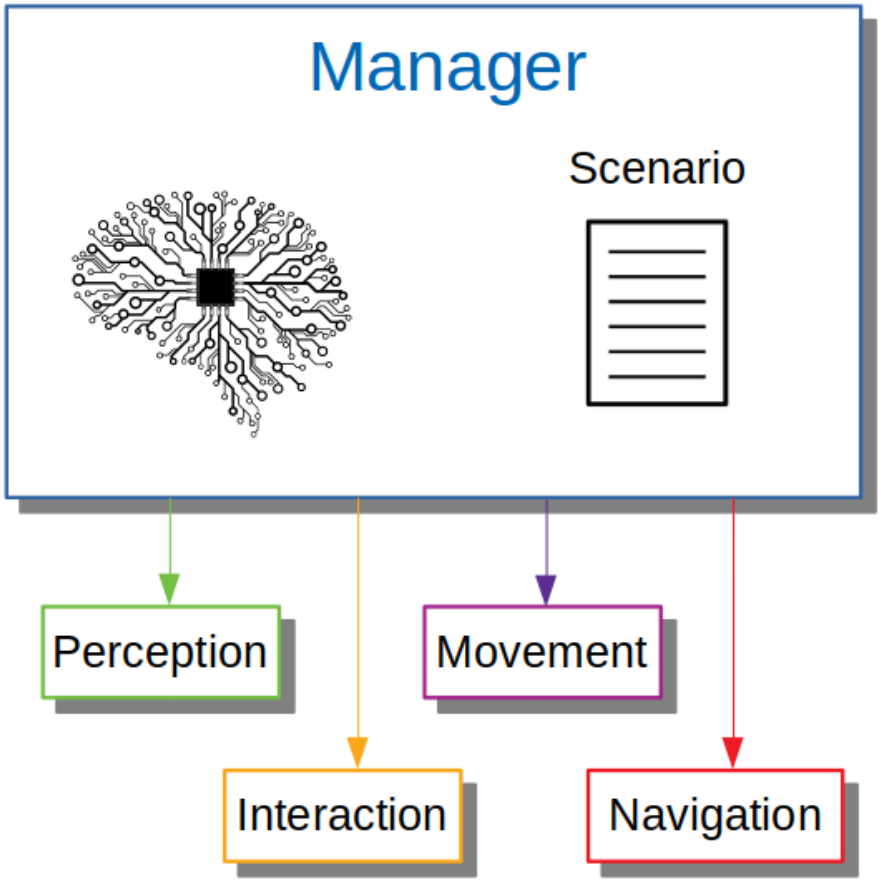


Figure 10: Manager's aim and links

The Manager

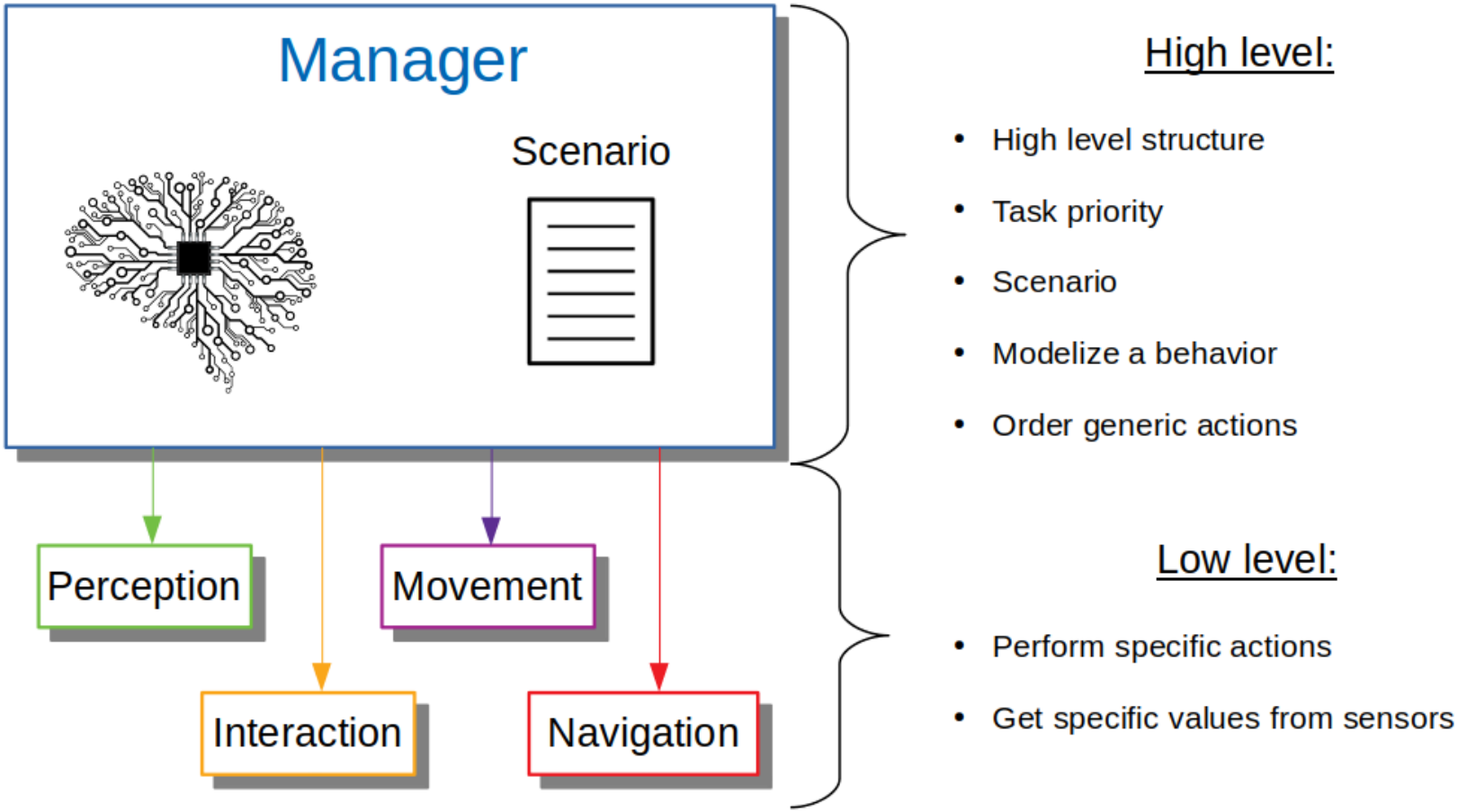


Figure 11: Manager's aim and links

The Manager: Scenario

Our Main Goal

- Create a robot service application for Hotel & Restaurant
- Inspired by RoboCup@Home scenarios Receptionist, Find My Mates and Carry My Luggage

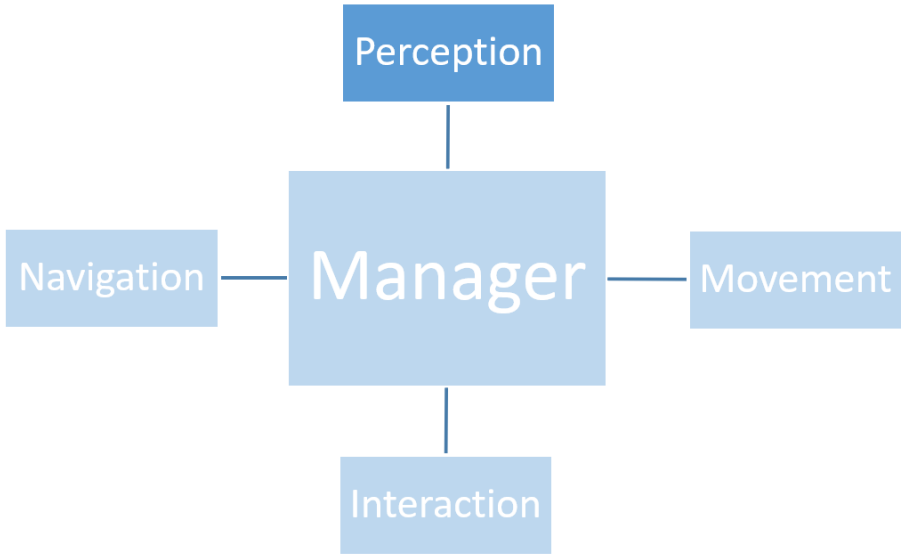
Final behavior (scenario)

1. Pepper is waiting for a new customer
2. Pepper moves to the new customer
3. Pepper ask & detect descriptions about the new customer
4. Pepper offers the client to sit showing him/her an empty chair
5. Pepper move to the receptionist location
6. Pepper repeat those descriptions to the receptionist
7. Pepper return to it's original position

The Manager: Discussion

Discussion

- ✔ Answer to our needs
- ✔ Links modules together
- ✔ High level structure
- ✔ Simple to make and use
- ✔ Time interval limit for order's execution
- ✘ Not a state machine
- ✘ Actions are performed sequentially
- ✘ Scenario paused during order's execution
- ✘ Not fully developed, can be enhance



Perception Module

What is computer vision?

- Aim to reproduce the human's sight
- Deduce whether objects are present in a picture
- Deduce where they are located

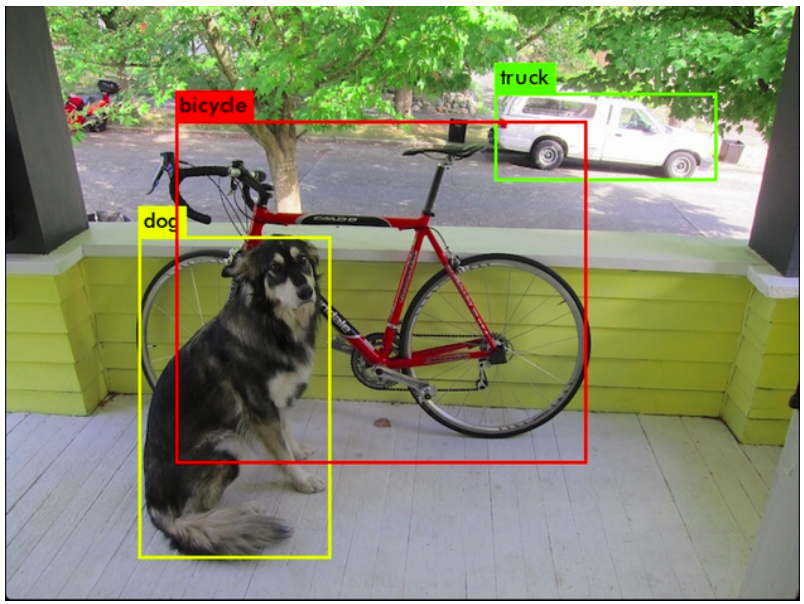


Figure 12: Object detection using YOLOv3 [?]

Perception Module

@HomeEDU requirements

- Detect objects (e.g. bags, chairs...)
- Detect persons
- Bonus: Detect a person's movement

Figure 13: What and how much can be detected?

Perception Module

State-of-the-art

- Object detection: Put objects in boxes, fast
- Instance segmentation: Pixel-perfect localization, slower

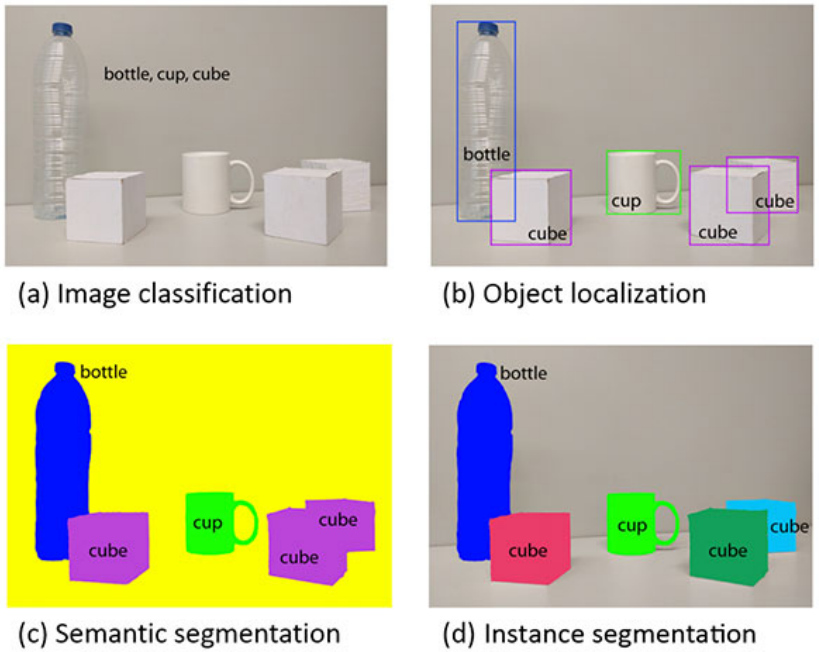


Figure 14: Detecting objects using Mask R-CNN [?]

Perception Module

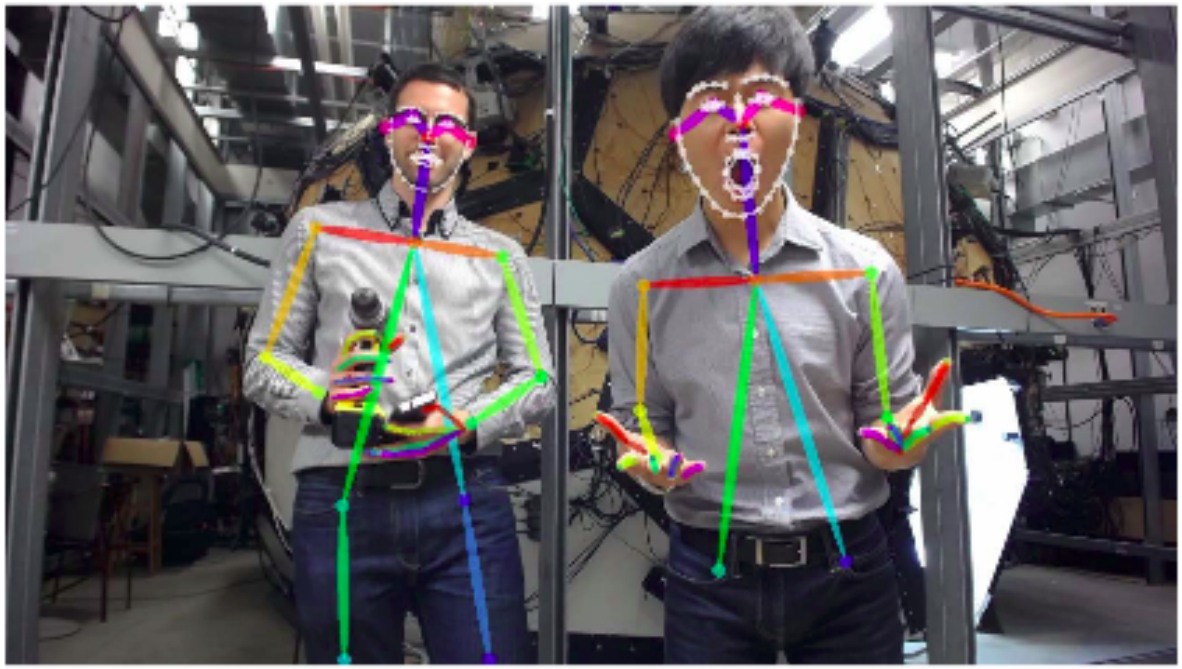


Figure 15: Pose estimation: OpenPose [**?**]

Perception Module

Constraints

- Require computer intensive operations
  What can be performed in real-time?
- Pepper's camera has a poor resolution.
  Is the image quality good enough to extract the information we need?

Perception Module

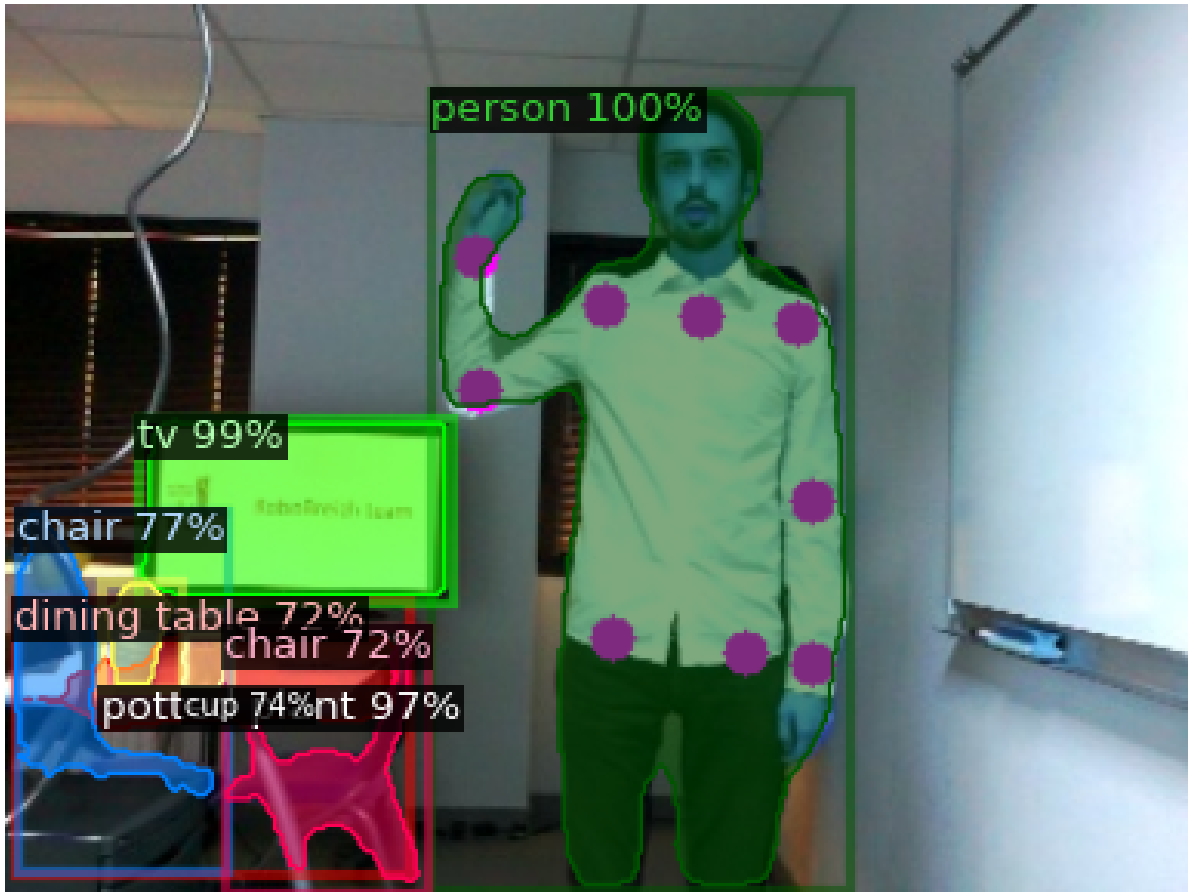
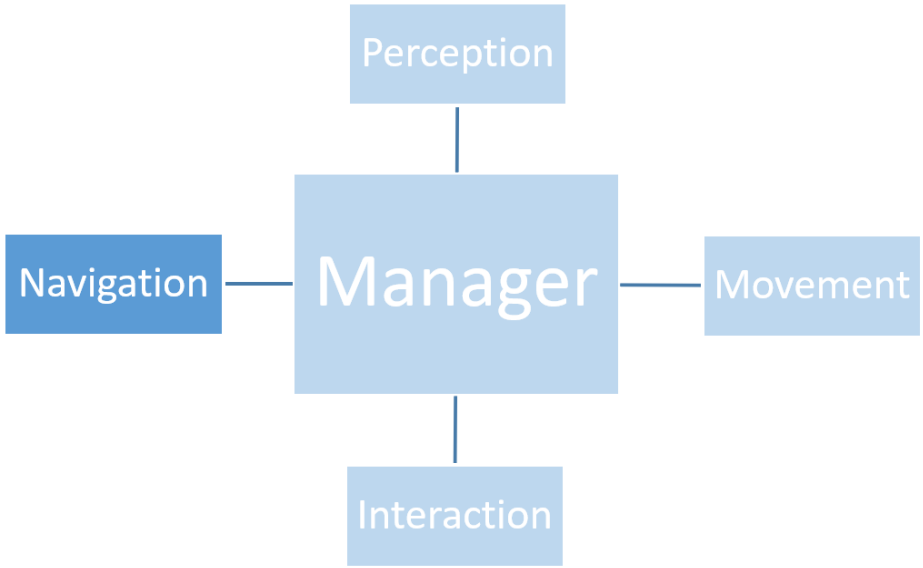


Figure 16: Mask-RCNN and OPenPose video demonstration

Perception Module

Discussion

- ✔ High performance in real-time
- ✔ Can detect on a pixel level
- ✔ Can detect poses
- ✘ Requires a computer with a GPU
- ✘ Communication between the robot and the computer adds latency



Navigation Module

Navigation in Robotics:

- Environment representation (i.e. Mapping)
- Real Time Localization
- Pathfinding

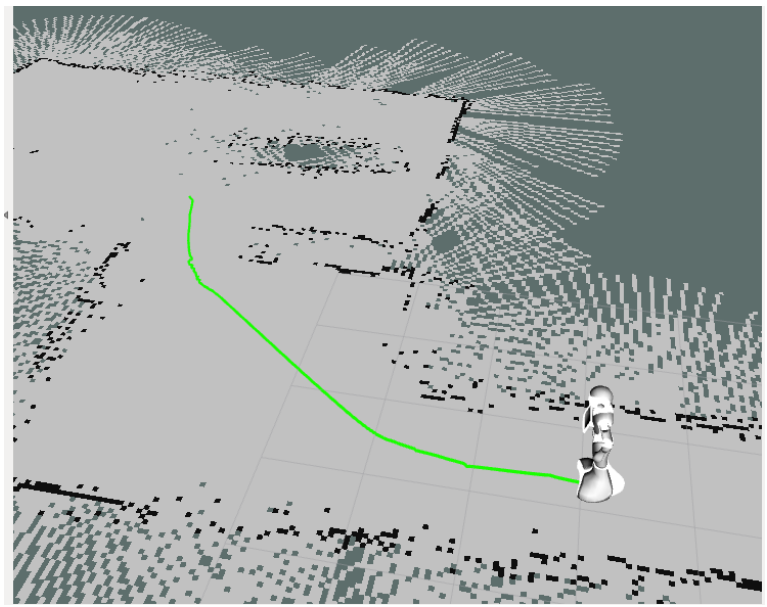


Figure 17: Navigation example

Navigation Module

@HomeEDU Requirements

Safely navigate to a given 2D coordinates

Others

- Navigate in known environment
- Easy-to-use and fast implementation
- Deal with known obstacles

Navigation Module

State-of-the-art (using ROS)

Simultaneous Localization And Mapping (SLAM)

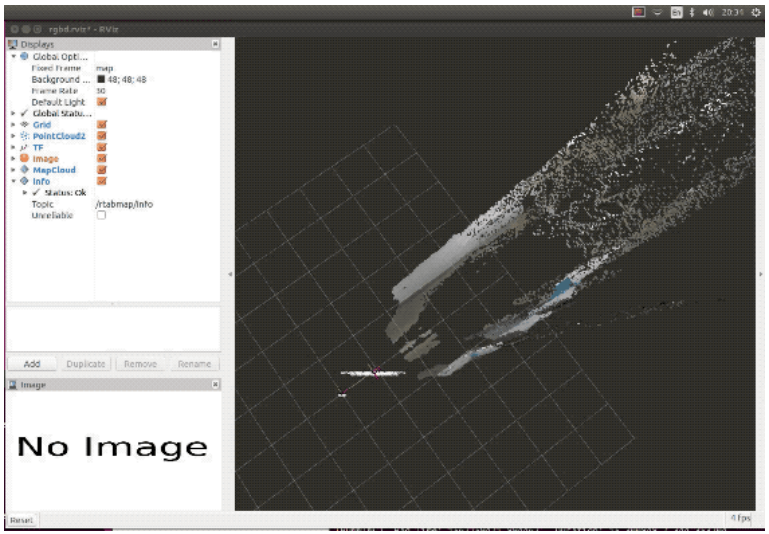


Figure 18: Visual SLAM [?]

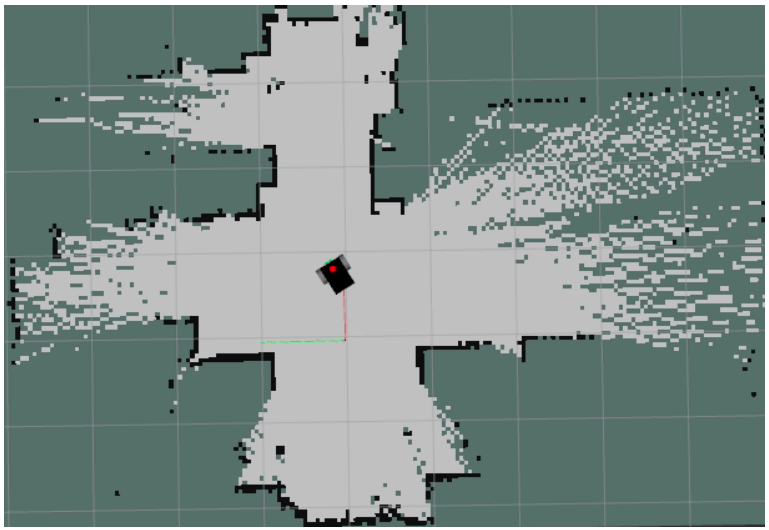


Figure 19: LiDAR-based SLAM [?]

Navigation Module

Constraints

- Accuracy of Pepper depth camera
- Pepper number and range of lasers

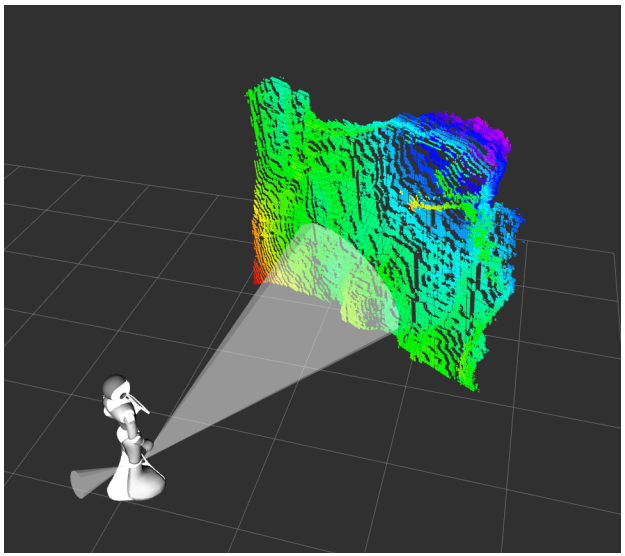


Figure 20: Pepper looking at a flat wall - Rviz Visualization of depth camera

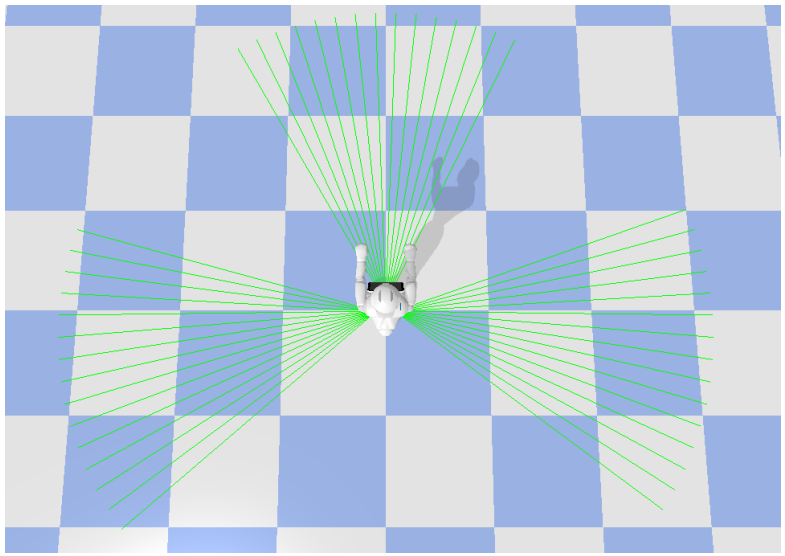


Figure 21: Pepper lasers' range and position

Navigation Module

Our approach with Pepper

- LiDAR based SLAM instead of Visual SLAM
- Mapping: Gmapping ROS node [**?**]
- Navigation: ROS Navigation Stack using AMCL (Adaptive Monte-Carlo Localization) [**?**] & move_base [**?**]

Navigation Module

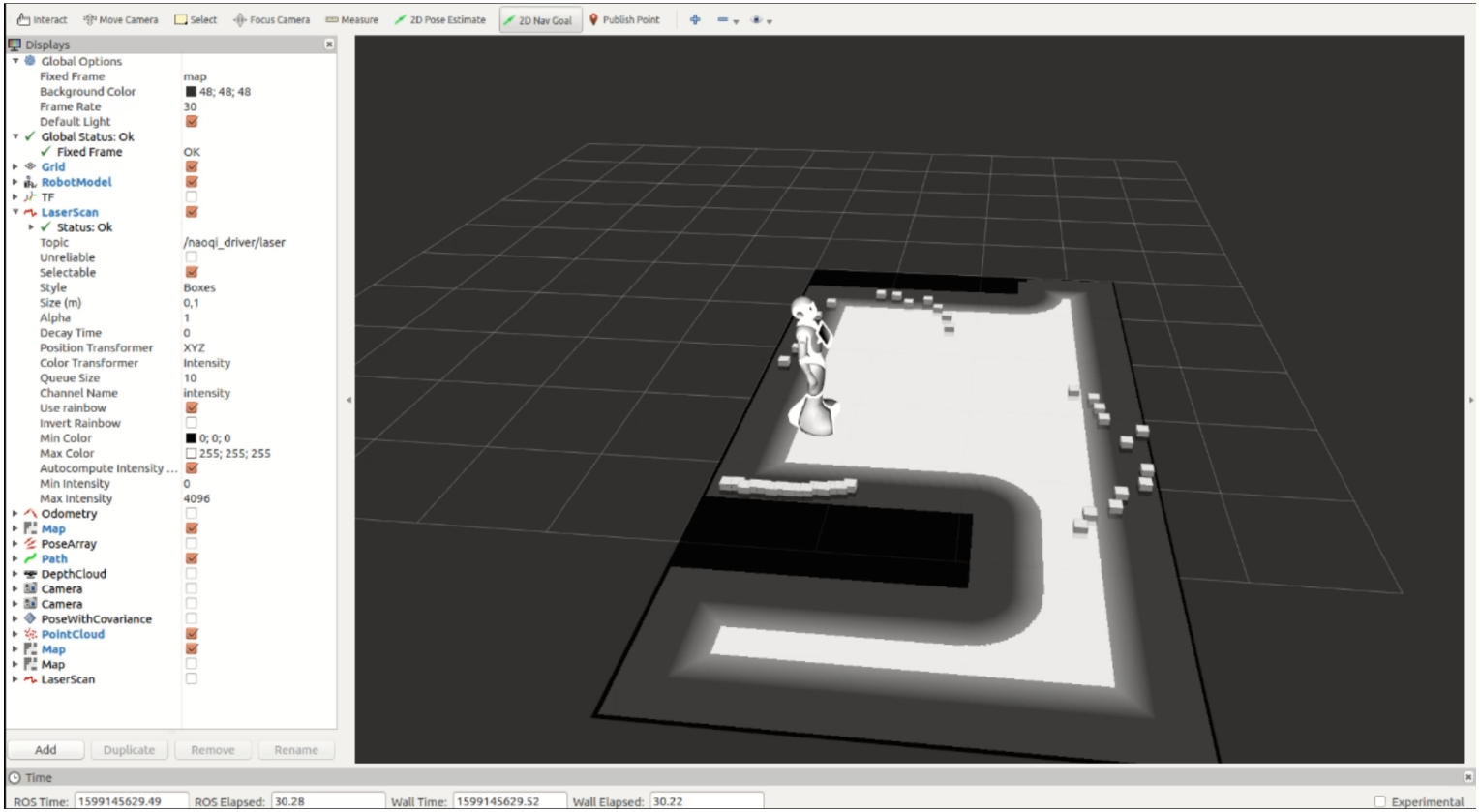


Figure 22: Map and pathfinding visualization through Rviz

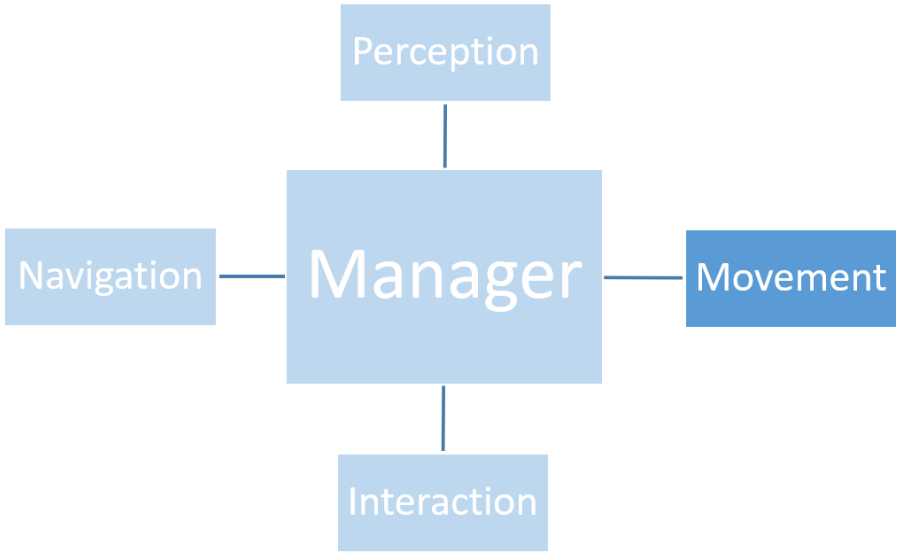
Navigation Module

Figure 23: Autonomous Navigation in known environment

Navigation Module

Discussion

- ✔ Easy-to-use and fast implementation
- ✔ Work well with short path
- ✘ Inaccuracy of depth camera
- ✘ Non-map objects and moving people led sometimes the planner to failure



Movement Module

What is a robotics movement?

- Allow the robot to move its arms and head
- Create a natural interaction with a human
- Interact with the environment

@HomeEDU requirements

- Specific movements (e.g. take objects, point a seat...)

Movement Module

State-of-the-art

- Used tools: naoqi_pose package (ROS)
- Other tools: NAOqi Motion, MoveIt! [?]
- Used learning algorithm: GMM (Gaussian Mixture Model) and GMR (Gaussian Mixture Regression) [?]

Constraints

- Only take light objects with Pepper
- Limited movements due to robot's reachable workspace
- Easy learning a movement

Movement Module

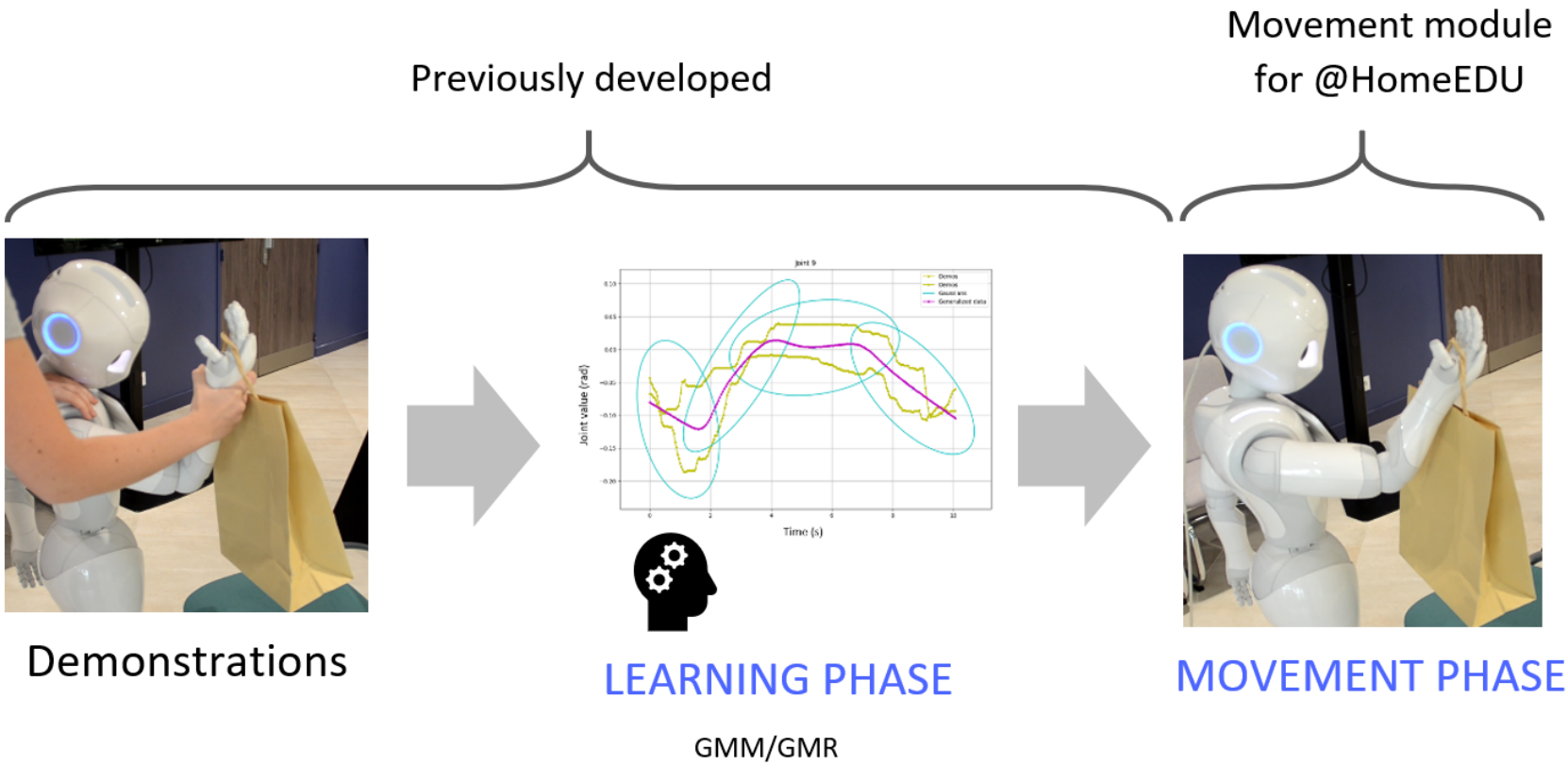


Figure 24: Learning movements process

Movement Module

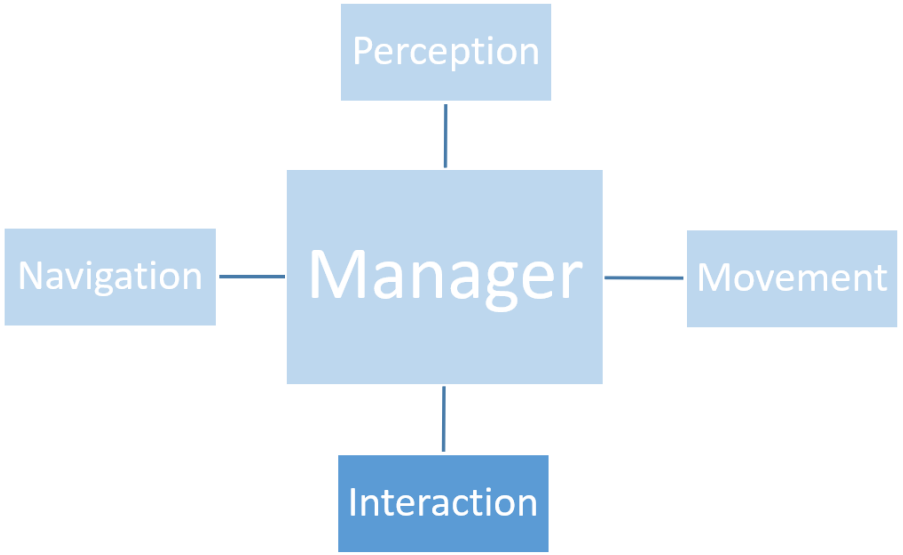
Figure 25: Movement demonstration

Figure 26: Learned movement

Movement Module

Discussion

- ✔ Easy to learn a new movement
- ✔ Interaction with objects
- ✘ Not adaptive to new object position or with obstacles



Interaction Module

What is an interaction?

- A Human-Robot Interaction (HRI): a verbal and non-verbal communication
- **Verbal communication** : dialog
- **Non-verbal communication** : tablet, LEDs, gestures

@HomeEDU requirements

- Understand human speech and his intention
- Answer to a human and memorize information
- Natural interaction with non-verbal communication

Interaction Module

Verbal communication: Dialog

1. Speech-To-Text
2. Natural Language Processing
3. Text-To-Speech

- Used tools: NAOqi
- Other tools: NLTK (Natural Language Toolkit) [?], Mbot [?]

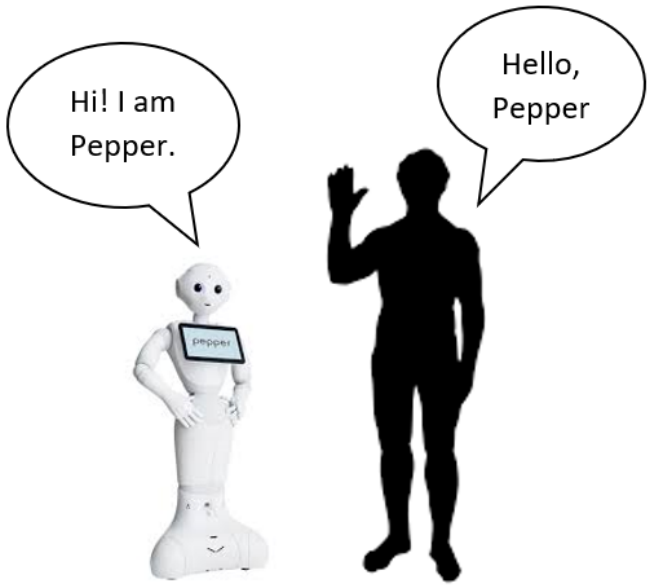


Figure 27: Pepper dialog

Interaction Module

Non-verbal communication

- Display information (tablet)
- Give feedback about its state (tablet, LEDs)
- Show its emotion (tablet, LEDs)
- Communication gestures (arm, head, body)

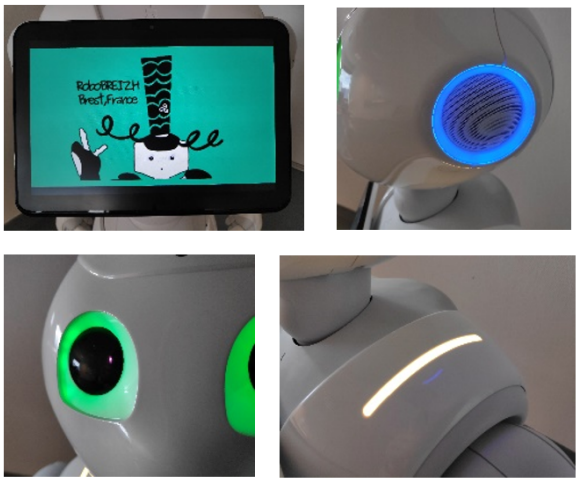


Figure 28: Pepper's tablet and LEDs

Interaction Module

Used communication tools

- Tablet: the hotel welcome
- Eye's LEDs: feedback of the perception module
- Gestures: Autonomous mode from NaoQi

LEDS colour	Type of detection
Red	Detection of a person
Yellow	Detection of a person's physical features
Cyan	Detection of a bag
Green	Detection of an empty chair
Pink	Detection of a taken chair
Purple	Detection of a waving hand

Table 1: Colour code of the Pepper eyes

Interaction Module

Discussion

- ✔ Easy to implement if the interaction is predefined
- ✔ Improve the natural communication with the robot
- ✘ Complex with human natural language (not predefined)
- ✘ Gestures during an interaction are complex to defined

RoboCup@Home Education Online Challenge: Final

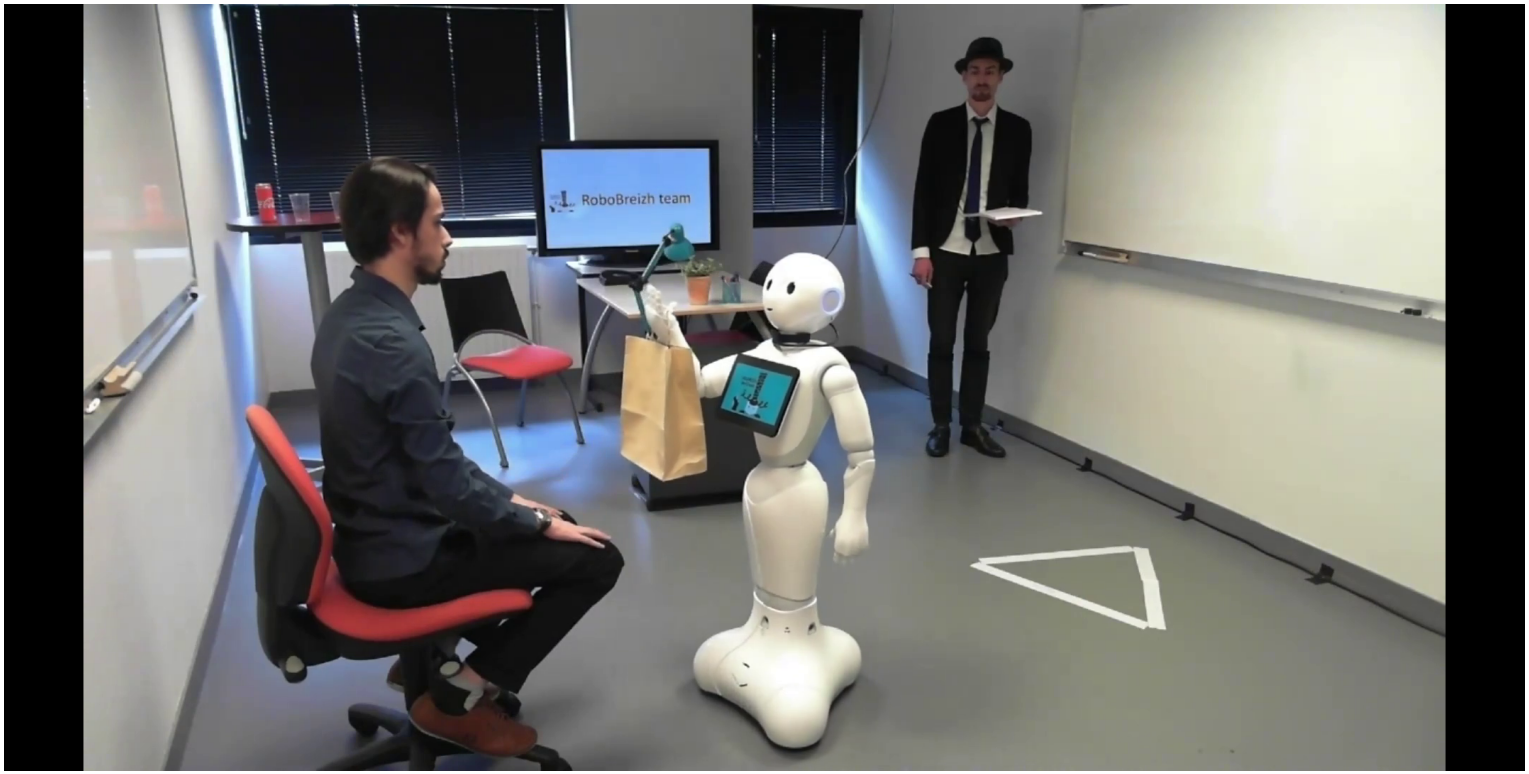


Figure 29: @HomeEDU final of RoboBreizh team

Conclusion

Summing up

- Choose the application and define the needs
- Define the module for the architecture and the used tools to create it (e.g. ROS)
- Be flexible
- RoboBreizh team won the Best Performance Award in Open Platform @HomeEDU

Conclusion

Futur work

- Improve the Dialog part
- Be on-board
- Improve Navigation system
- Participate in the RoboCup@Home competition

Partners

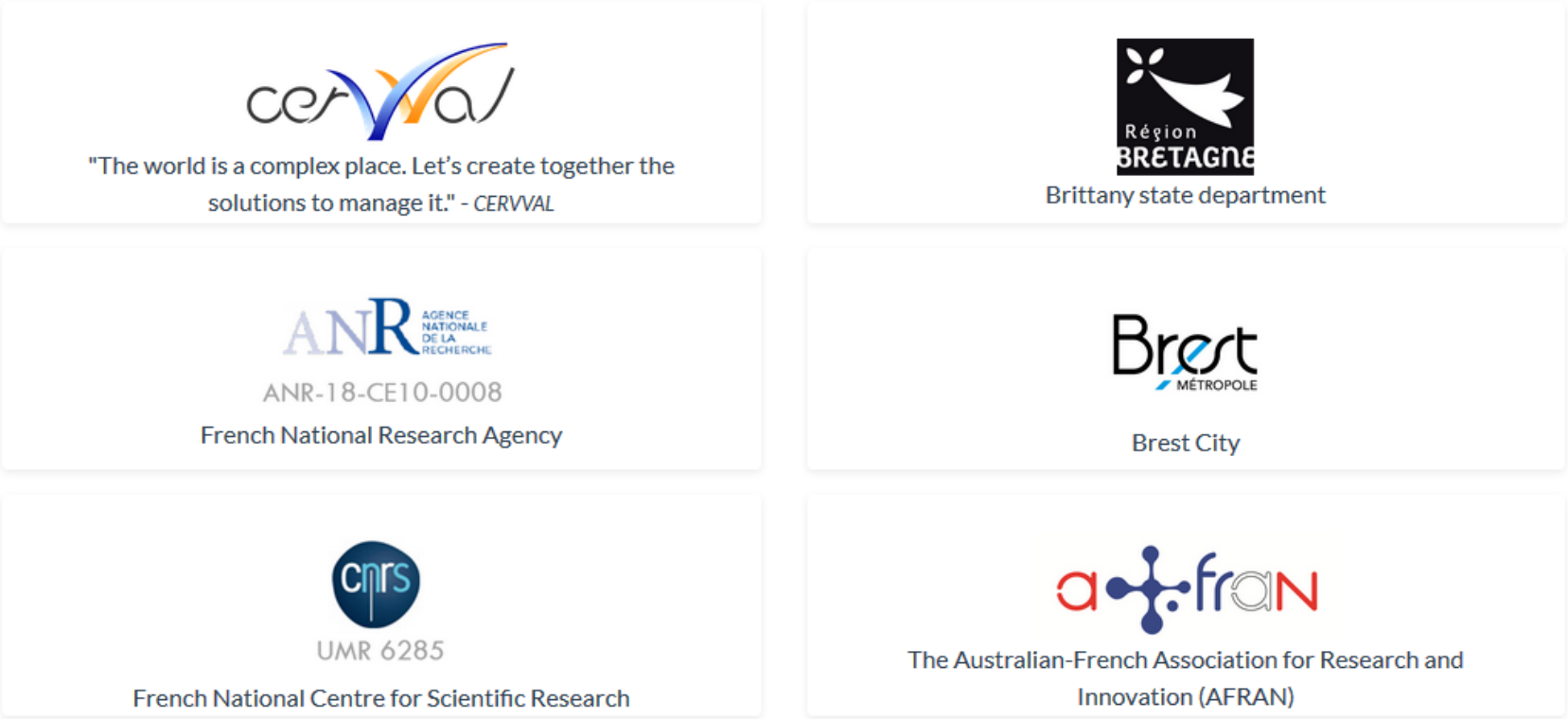


Figure 30: Our partners

Design Of A Modular Architecture Using ROS : The Pepper Case

RoboBreizh Team

A. Dizet, C. Le Bono, A. Legeleux, M. Neau, C. Buche

Lab-STICC, ENIB

30th September 2020

Website:
www.enib.fr/~robobreizh

References I