

# UV LARM

Logiciel et Architecture pour la  
Robotique Mobile

An introduction

Guillaume Lozenguez  
[@imt-nord-europe.fr](mailto:guillaume.lozenguez@imt-nord-europe.fr)



**IMT Nord Europe**  
École Mines-Télécom  
IMT-Université de Lille

1. **What is a Robot ?**
2. **About the UV LARM**
3. **Today: First contact with Linux and ROS**

# 1. **What is a Robot ?**

- Definition

- Modular and Multidisciplinary

# 2. About the UV LARM

# 3. Today: First contact with Linux and ROS

# What is a Robot ?

## On Wikipedia:

**en**

"A robot is a machine—especially one programmable by a computer— capable of carrying out a complex series of actions automatically."

**fr**

"Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) conçue pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines."

# What is a Robot ?

## From my point of view

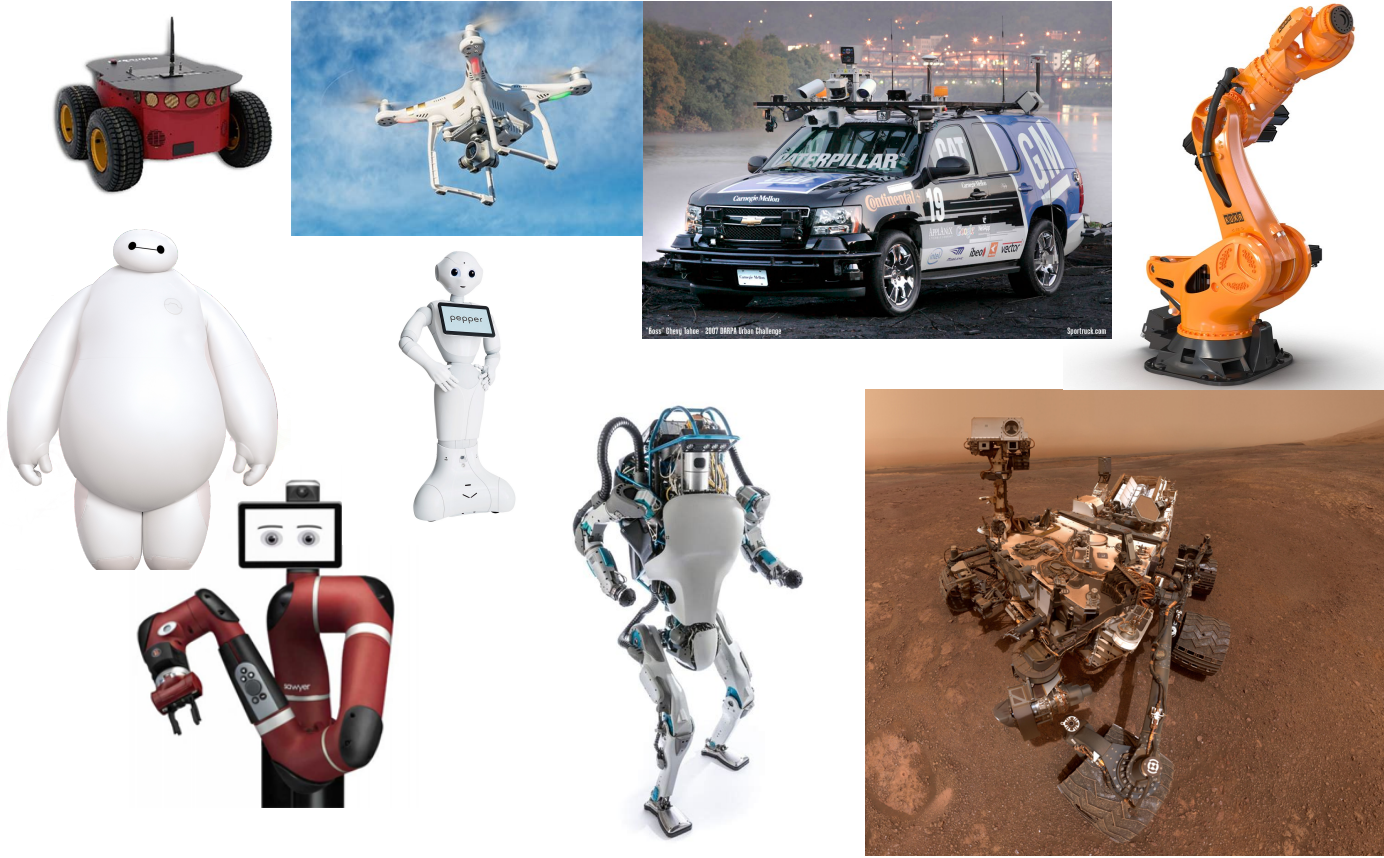
"A **robot** is a **mechatronics** machine capable of autonomously acting in a real environment."

- ▶ perceives with *sensors*
- ▶ models its environment and adapts its behavior with a *computer*
- ▶ acts with *actuators*

Involves generally Artificial-Intelligence:

- ▶ capable to mimic natural (human, animal, insect,...) intelligence

# Some examples



Macro: a large variety of robots

# Some examples



Micro: a large variety of components.

# From a mechanic point of view

## Focus on:

- ▶ Resistance
- ▶ Weight
- ▶ Distortion
- ▶ Vibration absorption
- ▶ Machining, Assembly

## for different robots:

- ▶ Fast
- ▶ Precise
- ▶ Strong
- ▶ resistant (dust, water,...)
- ▶ safe
- ▶ less expensive

## *From an electronic point of view*

Focus on sensors, motor, energy systems and hardware.



# From a automation point of view

## Focus on:

- ▶ Physics science
- ▶ Signal processing
- ▶ Control system

## by manipulating

- ▶ Times series, torques
- ▶ Vector, Matrices

# From a software point of view

## Focus on:

- ▶ Algorithms
- ▶ Knowledge representation
- ▶ Artificial intelligence
- ▶ Software architecture

Robots are complex and singular systems  
which require modular computer programs.

1. What is a Robot ?
2. **About the UV-LARM**
3. Today: First contact with Linux and ROS

## Software and Architecture for Mobile Robots

**Mostly about:** autonomous navigation.

- ▶ Communicate with robot components
- ▶ Control robot movements (nonholonomic robot)
- ▶ Perception of the local environment (laser, vision)
- ▶ SLAM (Simultaneous Localization and Mapping)
- ▶ Path finding and navigation.

## Software and Architecture for Mobile Robots

**With a central need:** modular software

- ▶ An expected complex global behavior
- ▶ Split in piece of programs (modules)
- ▶ Communicating together
- ▶ With dedicated tasks (sensor driver, representation, planning, controlling...)
- ▶ And a bunch of tools making all working together...

## Using ROS API:

**ROS:** The Robot Operating System (ROS) is a set of *software libraries* and *tools* that help you build robot applications.

- ▶ The number one Robotic Middle Ware used in academic
- ▶ Open and oriented toward its *many contributors*
- ▶ Supported by a large number of professional companies

It permits thinking robotic programs in a modular way as independent programs: *nodes* working together by communicating through *topics*.

It comes with useful functionality like *frame* management and *transform*

# ROS API:

## Tools:

To start nodes, to connect them, to visualize the architecture and the data in the pipes (topics).

## An API and Libs:

To develop its own nodes, to use topic-based communication, to help in manipulating spatio-temporal data.

## A community:

To share our contribution and use the one from pairs [wiki.ros.org](http://wiki.ros.org)

# ROS 1 versus ROS 2

## Historical.

- ▶ ROS 1 - 1.0 - 22 janvier 2010
- ▶ ROS 2 - Ardent Apalone - 8 December 2017
- ▶ ROS 1 - Noetic Ninjemys - 23 mai 2020 (dernière version de ROS 1)

## New ROS version.

- ▶ The same philosophy (modular communicative processes - Nodes)
- ▶ ROS 2: decentralized architecture based on DDS (Data Distribution Service norm)
- ▶ No retro-compatibility (as between Python2 and Python3)
  - But a huge community contribution in ROS 1



# Why Ubuntu Linux:

## Because

- ▶ We love *GNU* (and open source in general)
- ▶ ROS supports natively Ubuntu Linux
- ▶ *And mainly:* Linux is efficient, secure and well documented

# The courses:

Mainly based on tutorials on [ceri-num.gitbook.io](https://ceri-num.gitbook.io)

## Notions

- ▶ Modular Software Architecture
- ▶ Perception and Control (obstacle avoidance)
- ▶ Rich perception: Vision
- ▶ Navigation (mapping, localization, planning)

# UV-LARM - Schedule

*1st week:* Introduction, simulation and movement.

*2d week:* Vision and Mapping.

*3d week:* Challenge as your project.

*4th week:* Evaluation through the code you provide.

- ▶ Always from *9:00* to *12:00* and from *14:00* to *17:30*.
- ▶ In *Develter* and *3005*.
- ▶ With or without a teacher.

# Evaluation:

## Challenges

- ▶ *From*: navigate in a cluttered environment.
- ▶ *To*: autonomous explore an unknown area with objects to retrieve.

## 2 Environments

- ▶ Robot: Turtlebot
- ▶ Simulation: Gazebo

1. What is a Robot ?
2. About the UV LARN
3. **Today: First contact with Linux and ROS**
  - Starting with ROS tutorial: <https://ceri-num.gitbook.io/uv-larm/>