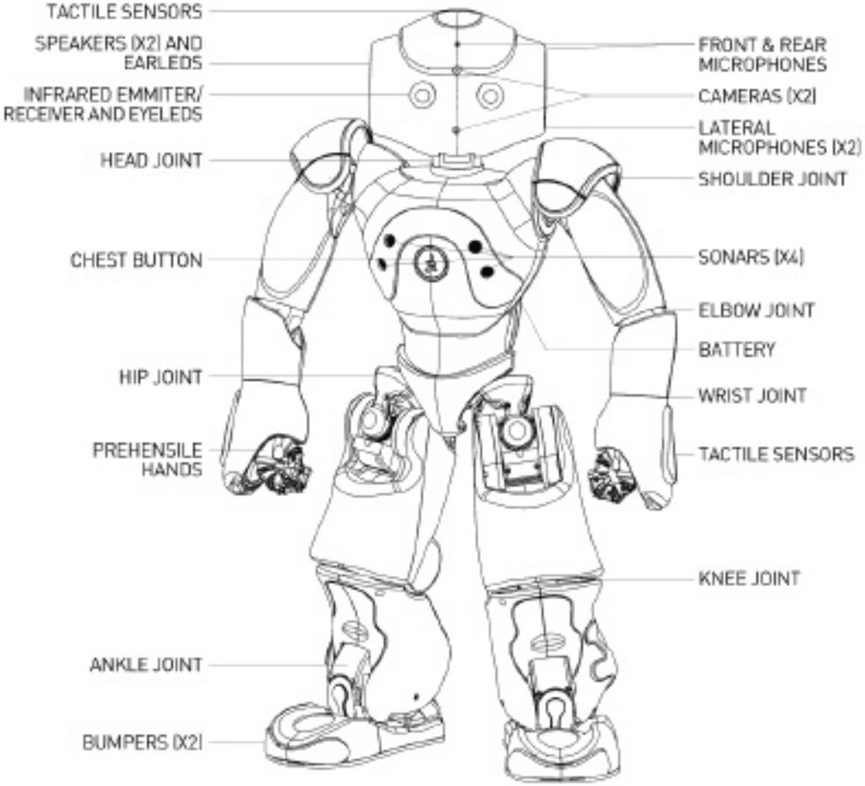
NAO

Programming a humanoid robot

Nao at a glance



ALDEBARAN Robotics

What can NAO do?

MOVE

- 25 degrees of freedom
- Motors controlled by software
- Complex movement capabilities

SENSE

- 2 HD camera
- 4 microphones
- 2 bumpers
- 2 sonars

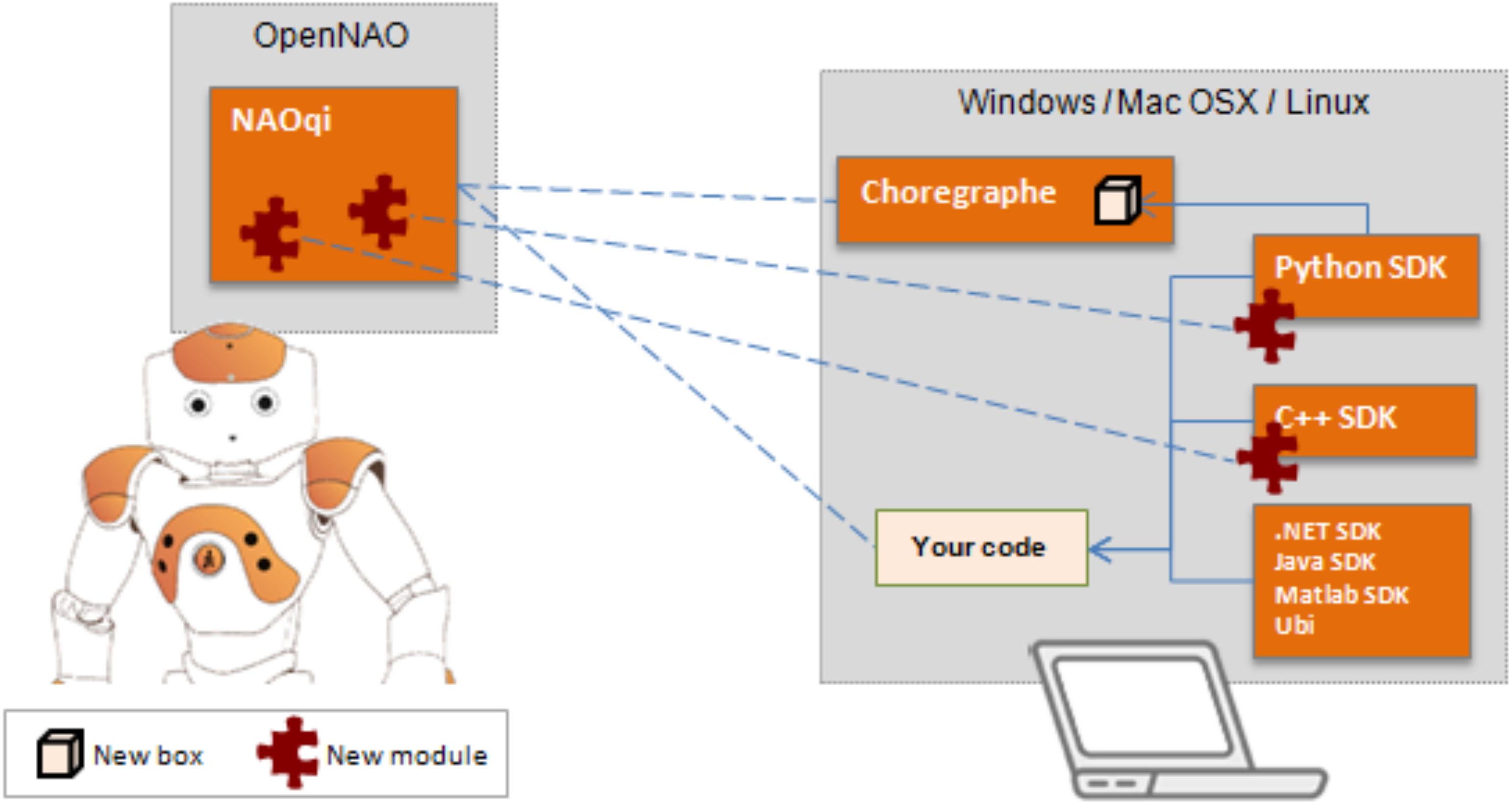
INTERACT

- 2 speakers
- multiple LEDs
- tactile sensors
- prensile hands
- infrared sensors
- WiFi connection

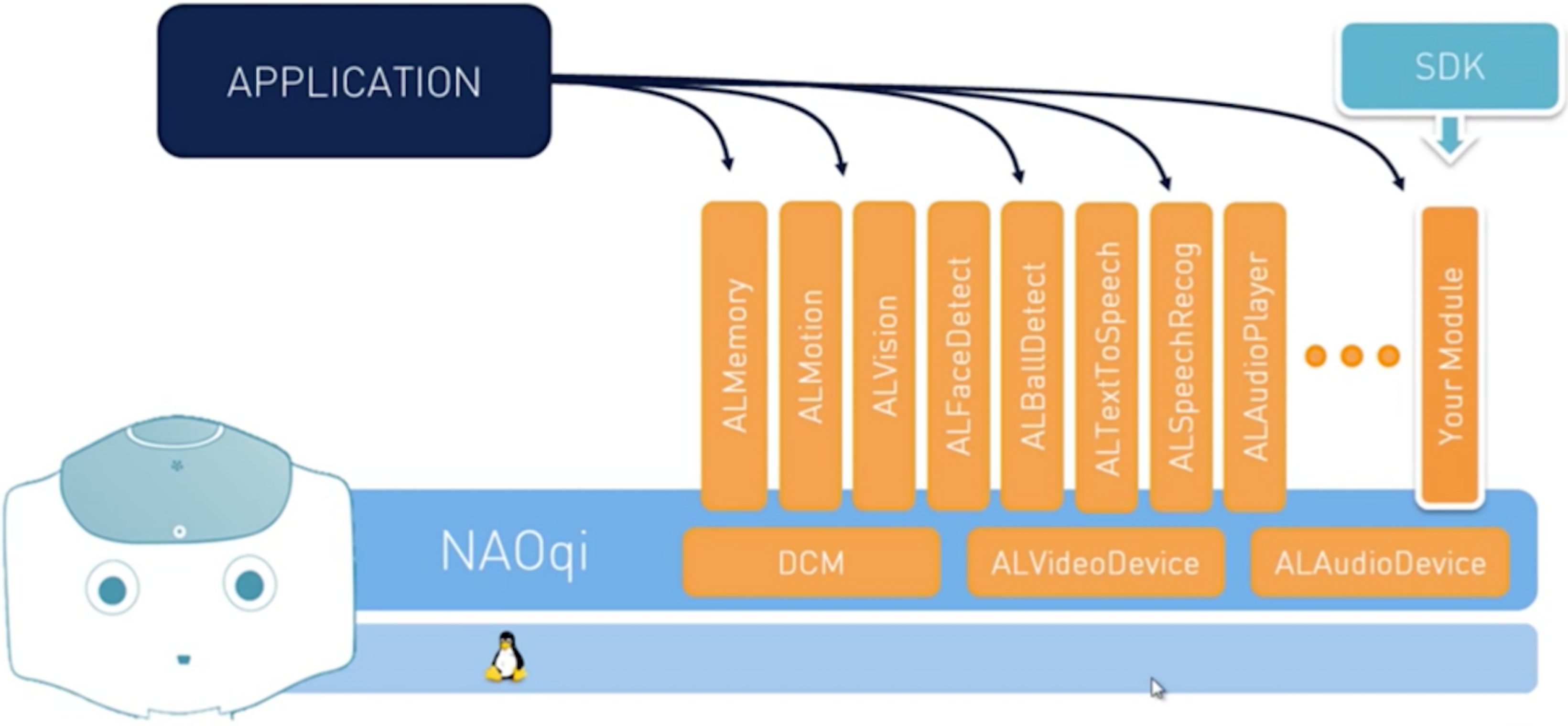
THINK

- Intel Atom 1,6 GHz CPU
- 1 Gb RAM
- 8 Gb Flash Memory
- Software suite

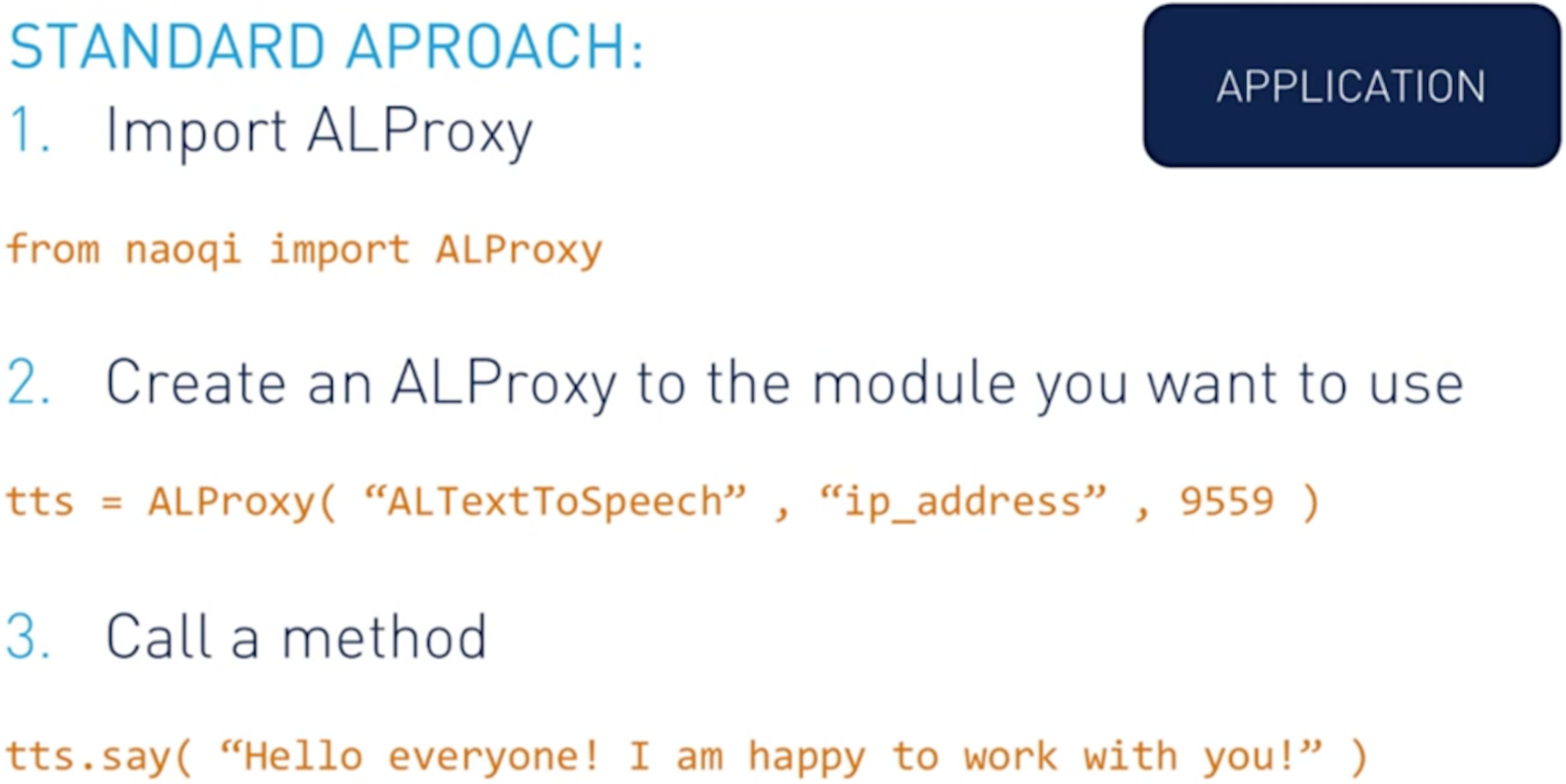
What and where?



Why programming in Python



Remote control



APPLICATION

STANDARD APROACH:

1. Import ALProxy

```
from naoqi import ALProxy
```

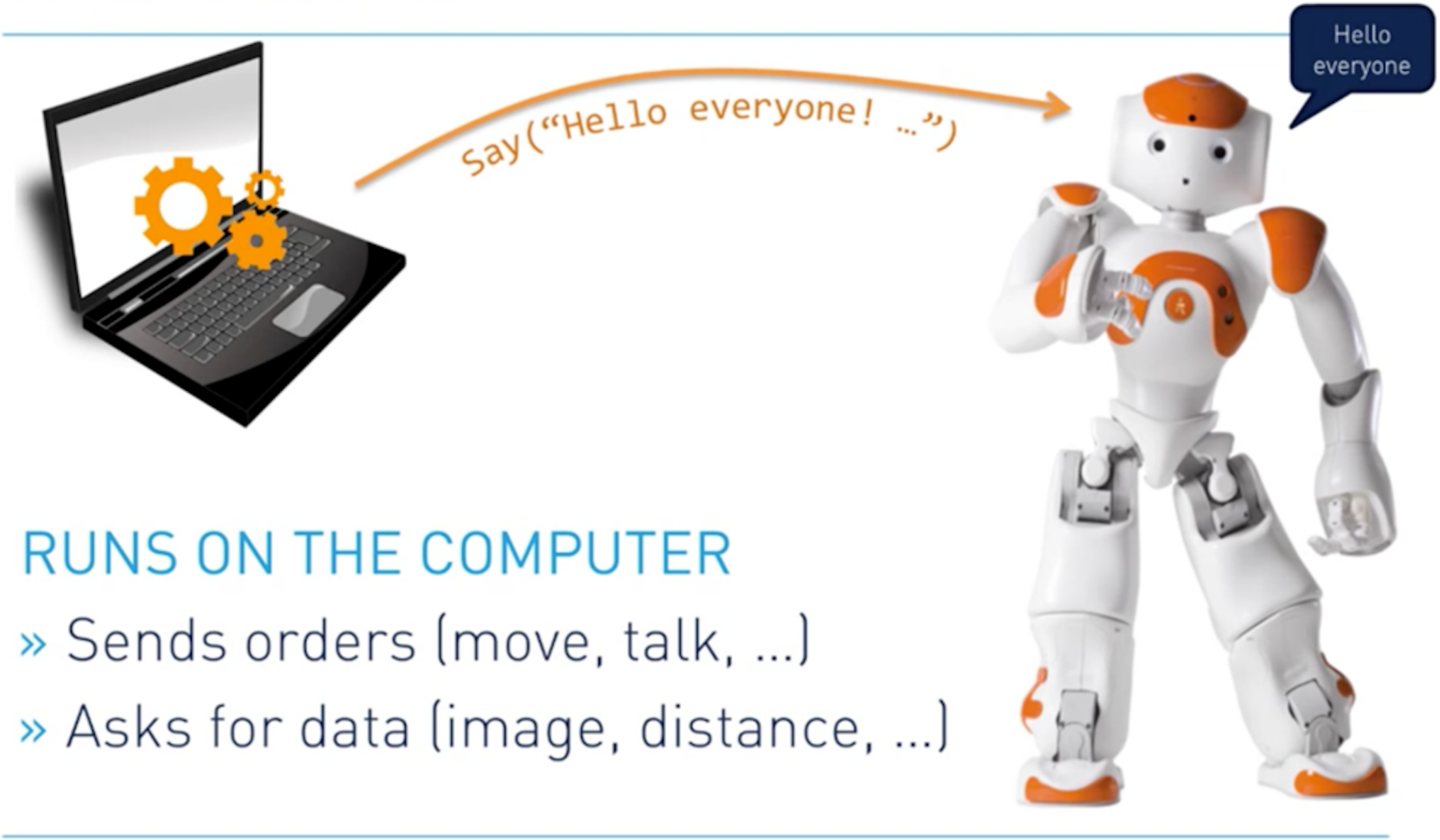
2. Create an ALProxy to the module you want to use

```
tts = ALProxy( “ALTextToSpeech” , “ip_address” , 9559 )
```

3. Call a method

```
tts.say( “Hello everyone! I am happy to work with you!” )
```

Remote control

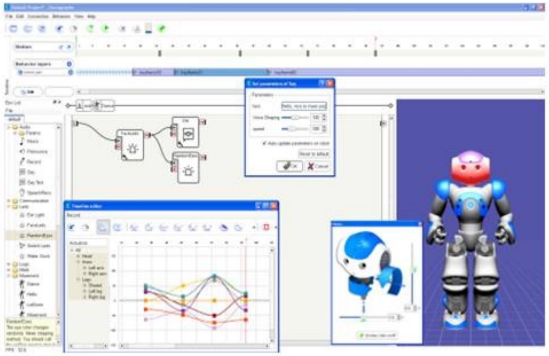


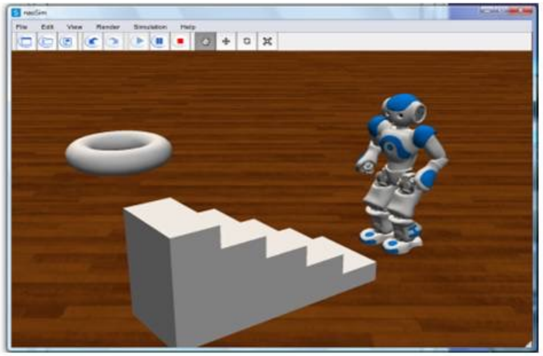
RUNS ON THE COMPUTER

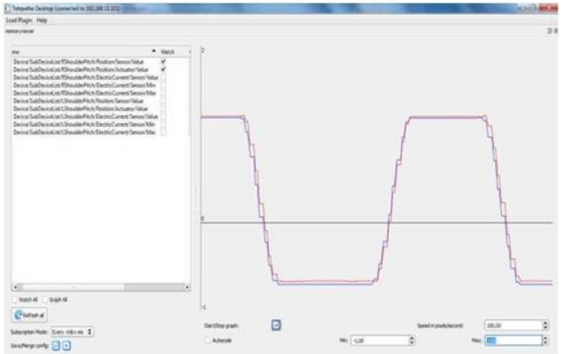
» Sends orders (move, talk, …)

» Asks for data (image, distance, …)

Software Suite







Choregraphe

- ✓ Graphical Development of Behaviors
- ✓ Ergonomic and user-friendly Interface

NAOsim

- ✓ Physical Simulation Engine
- ✓ Behaviors Simulation and validation

Monitor

- ✓ Ergonomic Interface to monitor actuators and sensors data

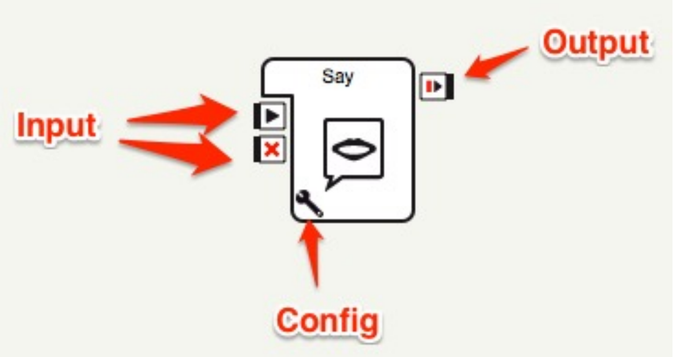
SDK

- ✓ Compilation and debugging tools
- ✓ MatLab, Java, Python, C++, .NET, MS Robotics Studio

Choreographe

- It is a graphical interface to program NAO (for Windows, Mac e Linux)
  (downloadable after creating an account on **Aldebaran community**)

- It is composed by boxes containing some code for specific actions



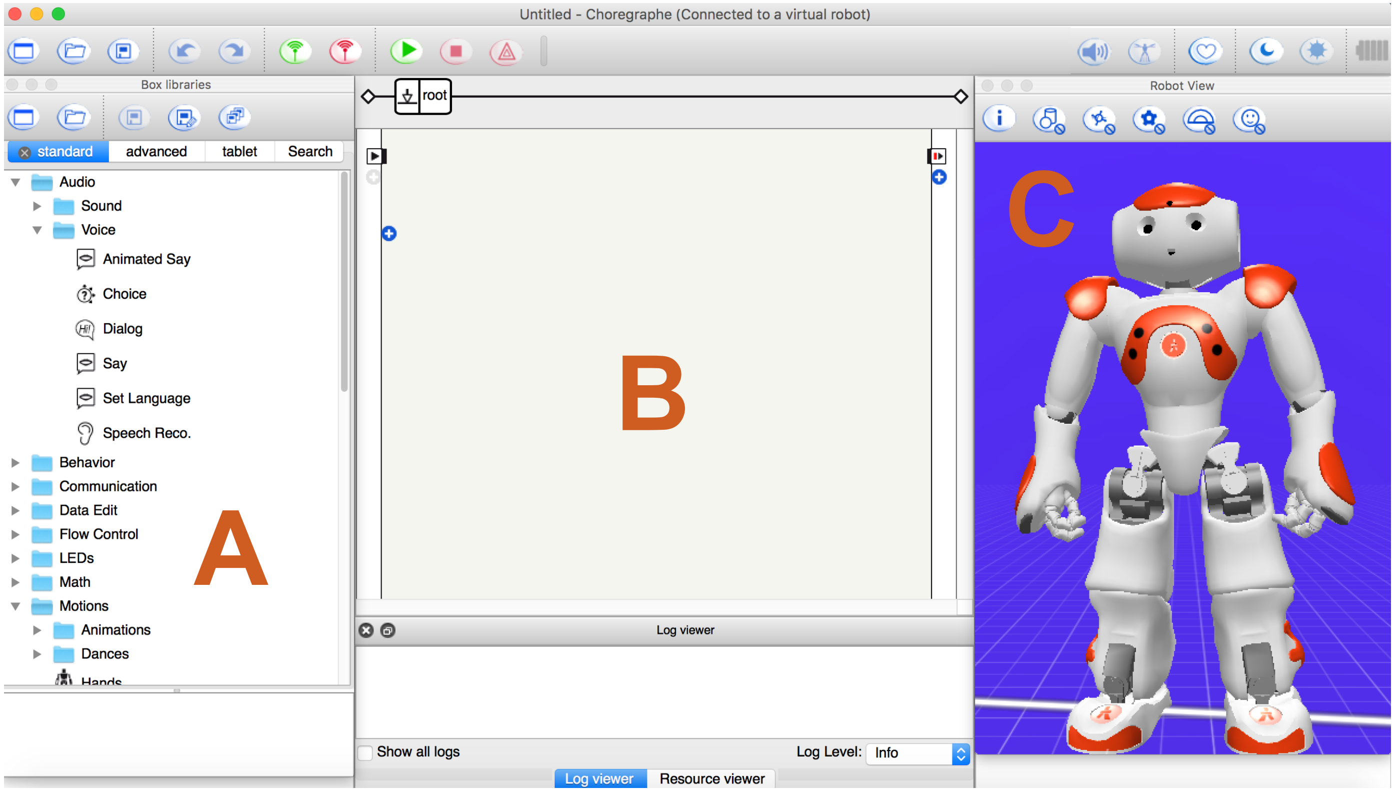
- Python code into boxes
- a box can be made of other boxes
- a box can have different inputs/outputs

Choreographe - Panels

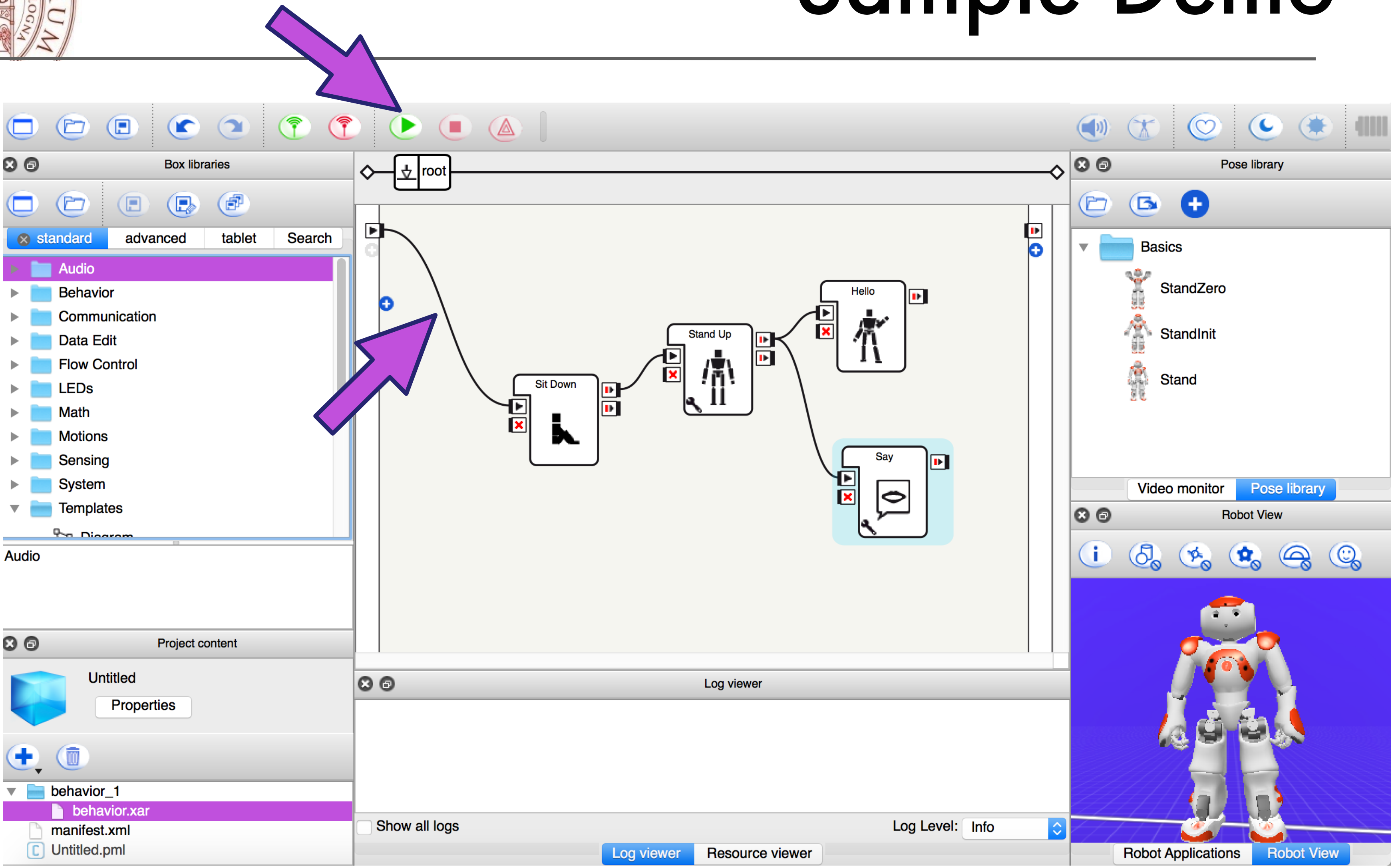
A Box libraries panel

B Flow diagram panel

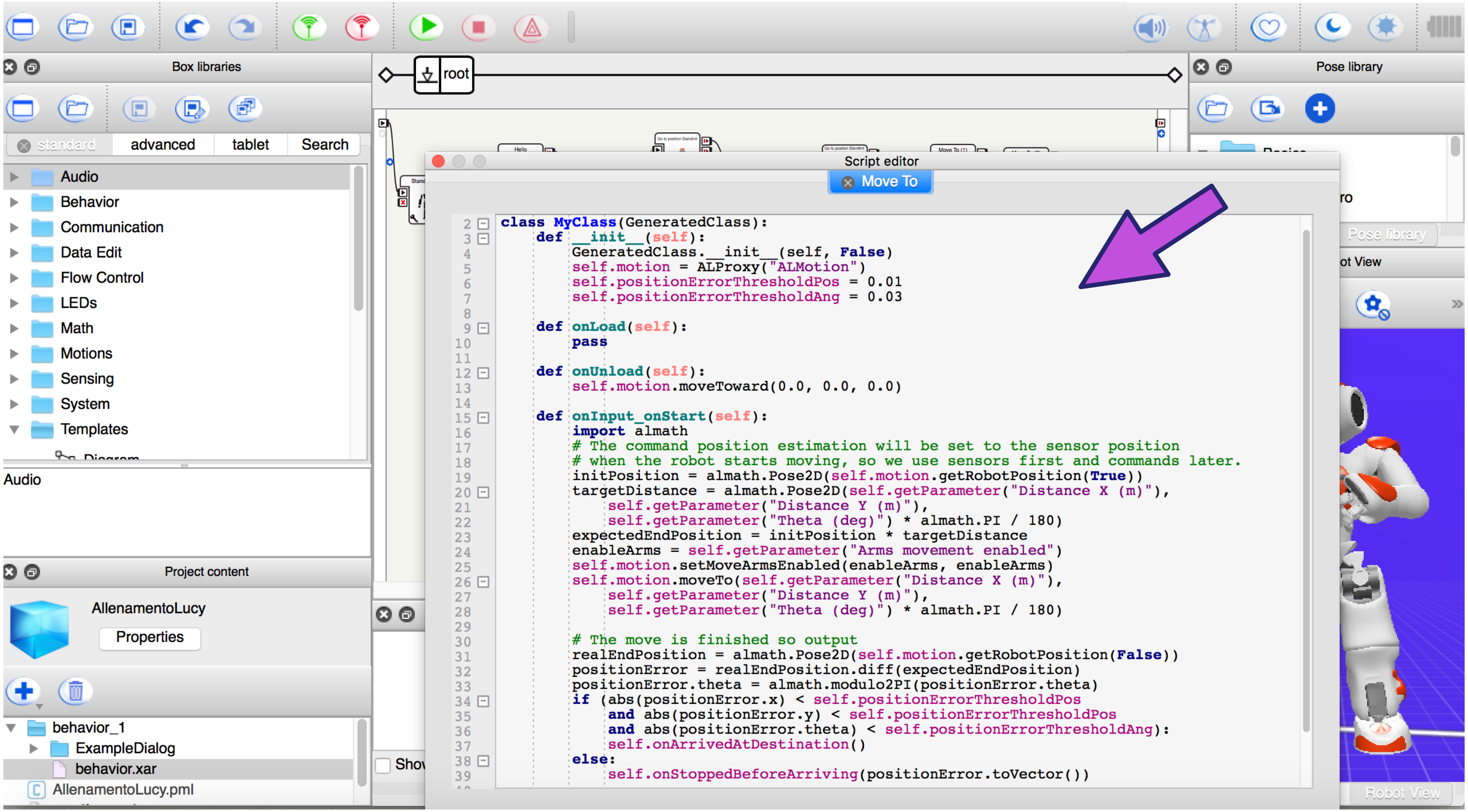
C 3D Robot View



Sample Demo

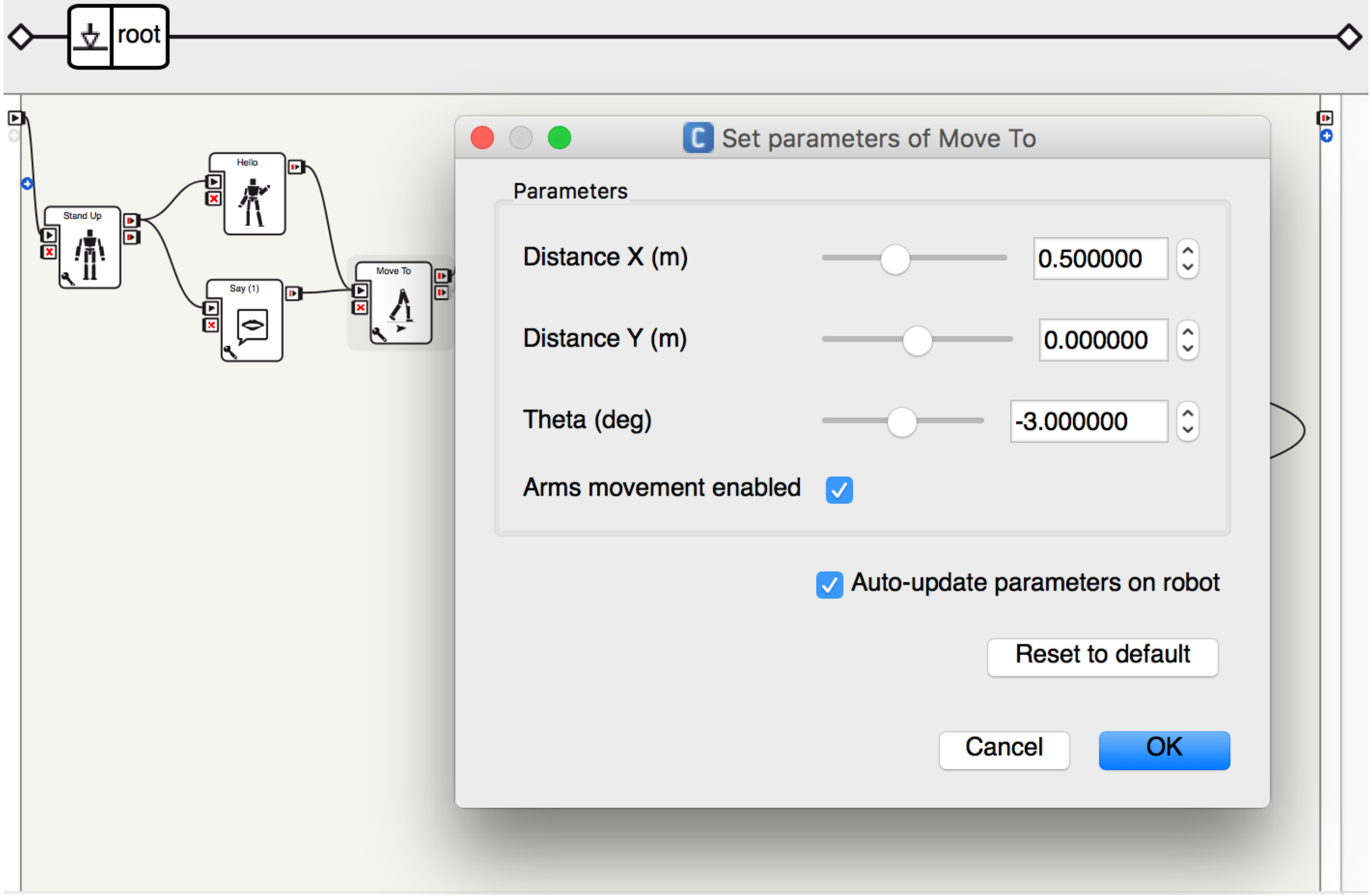

Box libraries
standard
advanced
tablet
Search
Audio
Behavior
Communication
Data Edit
Flow Control
LEDs
Math
Motions
Sensing
System
Templates
Diagram
Audio
Project content
Untitled
Properties
behavior_1
behavior.xar
manifest.xml
Untitled.pml
root
Sit Down
Stand Up
Hello
Say
Log viewer
Show all logs
Log Level:
Info
Log viewer
Resource viewer
Pose library
Basics
StandZero
StandInit
Stand
Video monitor
Pose library
Robot View
Robot Applications
Robot View

Python Boxes



```python
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self, False)
        self.motion = ALProxy("ALMotion")
        self.positionErrorThresholdPos = 0.01
        self.positionErrorThresholdAng = 0.03

    def onLoad(self):
        pass

    def onUnload(self):
        self.motion.moveToward(0.0, 0.0, 0.0)

    def onInput_onStart(self):
        import almath
        # The command position estimation will be set to the sensor position
        # when the robot starts moving, so we use sensors first and commands later.
        initPosition = almath.Pose2D(self.motion.getRobotPosition(True))
        targetDistance = almath.Pose2D(self.getParameter("Distance X (m)"),
            self.getParameter("Distance Y (m)"),
            self.getParameter("Theta (deg)") * almath.PI / 180)
        expectedEndPosition = initPosition * targetDistance
        enableArms = self.getParameter("Arms movement enabled")
        self.motion.setMoveArmsEnabled(enableArms, enableArms)
        self.motion.moveTo(self.getParameter("Distance X (m)"),
            self.getParameter("Distance Y (m)"),
            self.getParameter("Theta (deg)") * almath.PI / 180)

        # The move is finished so output
        realEndPosition = almath.Pose2D(self.motion.getRobotPosition(False))
        positionError = realEndPosition.diff(expectedEndPosition)
        positionError.theta = almath.modulo2PI(positionError.theta)
        if (abs(positionError.x) < self.positionErrorThresholdPos
            and abs(positionError.y) < self.positionErrorThresholdPos
            and abs(positionError.theta) < self.positionErrorThresholdAng):
            self.onArrivedAtDestination()
        else:
            self.onStoppedBeforeArriving(positionError.toVector())
```

Configuration Box



Set parameters of Move To

Parameters

Distance X (m) 0.500000

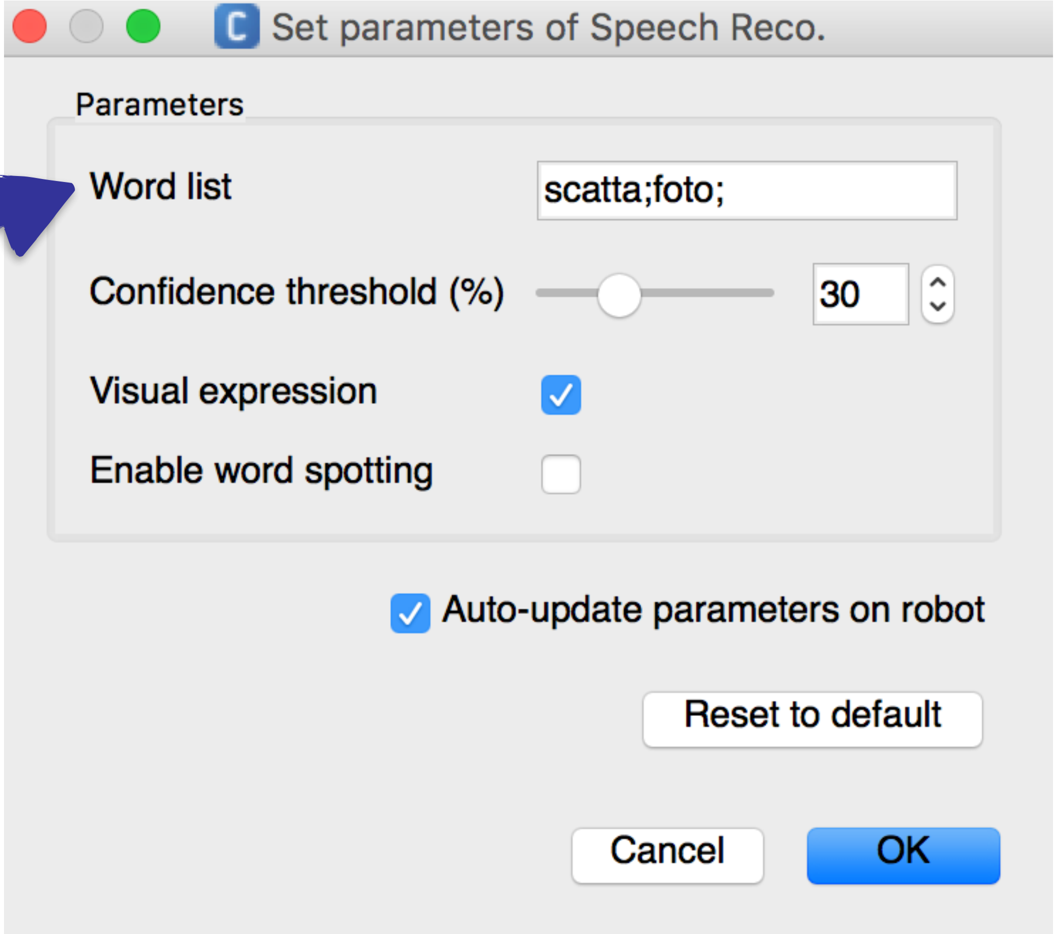
Distance Y (m) 0.000000

Theta (deg) -3.000000

Arms movement enabled ☑

☑ Auto-update parameters on robot

Speech Recognition



Set parameters of Speech Reco.

Parameters

Word list: scatta;foto;

Confidence threshold (%): 30

Visual expression ☑

Enable word spotting □

☑ Auto-update parameters on robot

You can enter a list of words to be recognized by voice command to the robot

Projects and Thesis

1. NAO plays Soccer (**RoboCup**)
2. **Thesis and projects**
   - ☑ NAO plays “Guess who”
   - ☑ Planning for NAO actions and learning new movements in Timeline
   - ☑ NAO navigates in different rooms
   - ☑ NAO mathematician
3. **Master Thesis**
   - ☑ Neural networks for
     - ▷ Face recognition
     - ▷ OCR (Optical Character Recognition)
   - ☑ User movements imitation using Kinect

Nao plays Soccer

- RoboCup aims to create, by 2050, **a team of humanoid robots** that can take on and beat the best human players.

- When playing together, the **robots must act autonomously** and are unable to get help from their handlers.

- They also communicate via wi-fi to **co-ordinate teamwork**.

- **Various technologies** have to be developed **in AI**: the robots know who to pass to and how best to defeat an opponent.

Lucy plays “Guess who”

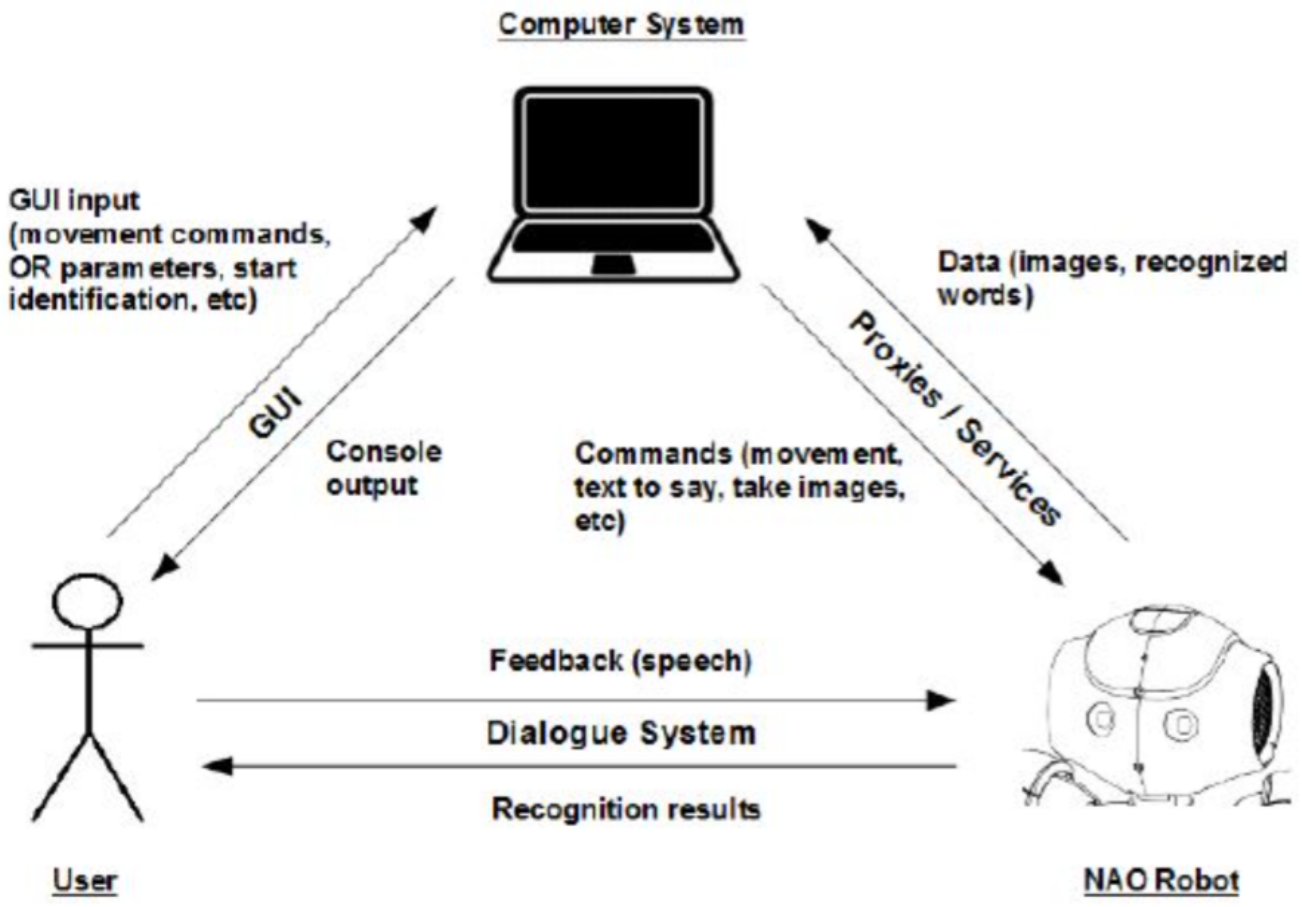
- Lucy is **able to play guess who with a human** using voice recognition.

- Once the sentence pronounced is translated into textual form, **Lucy is able to understand** what was communicated

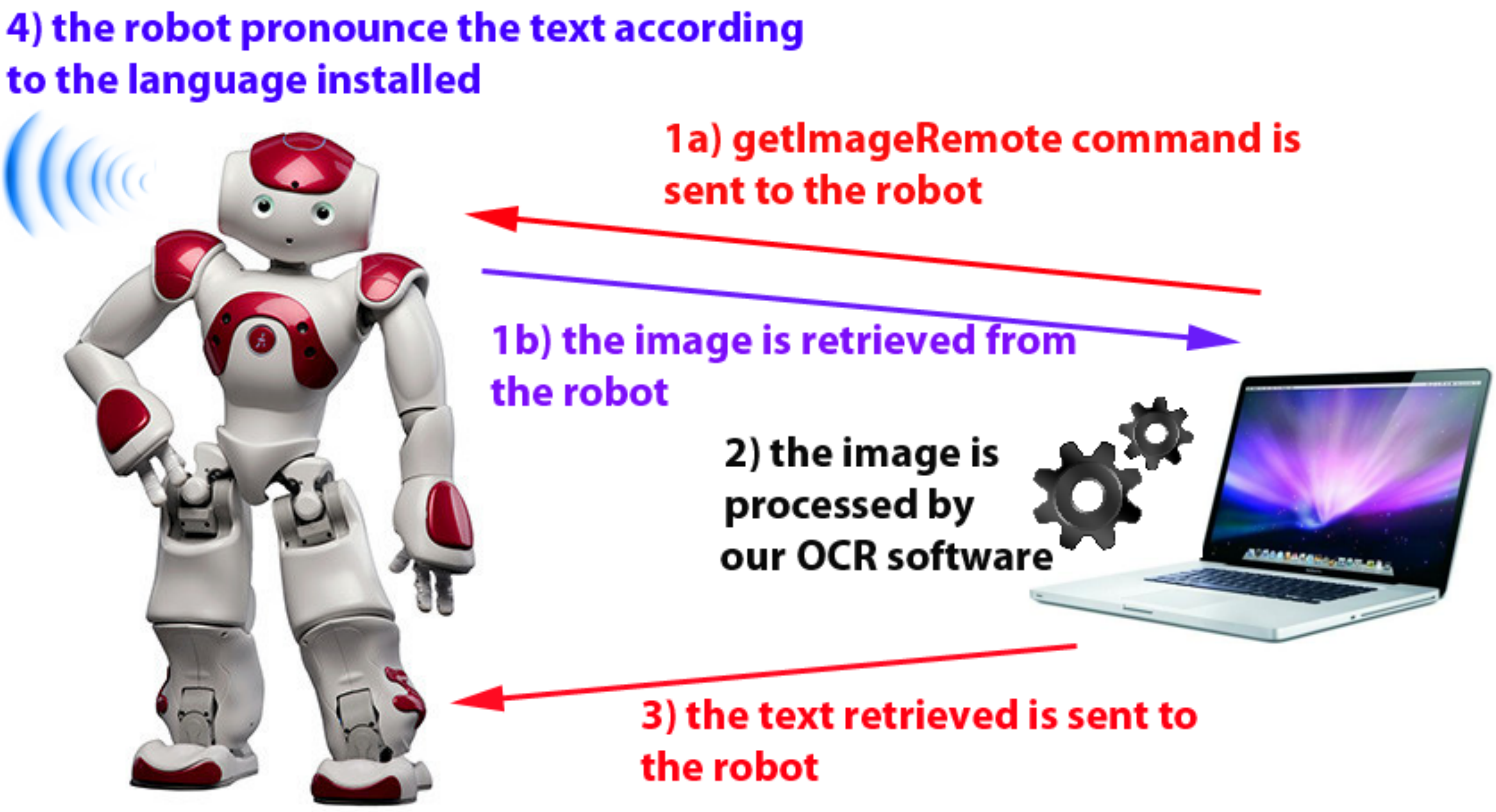
↓

Natural Language Processing

Face Recognition


Computer System
GUI input (movement commands, OR parameters, start identification, etc)
GUI
Console output
Data (images, recognized words)
Proxies / Services
Commands (movement, text to say, take images, etc)
Feedback (speech)
Dialogue System
Recognition results
User
NAO Robot

Lucy reads with OCR



4) the robot pronounce the text according to the language installed

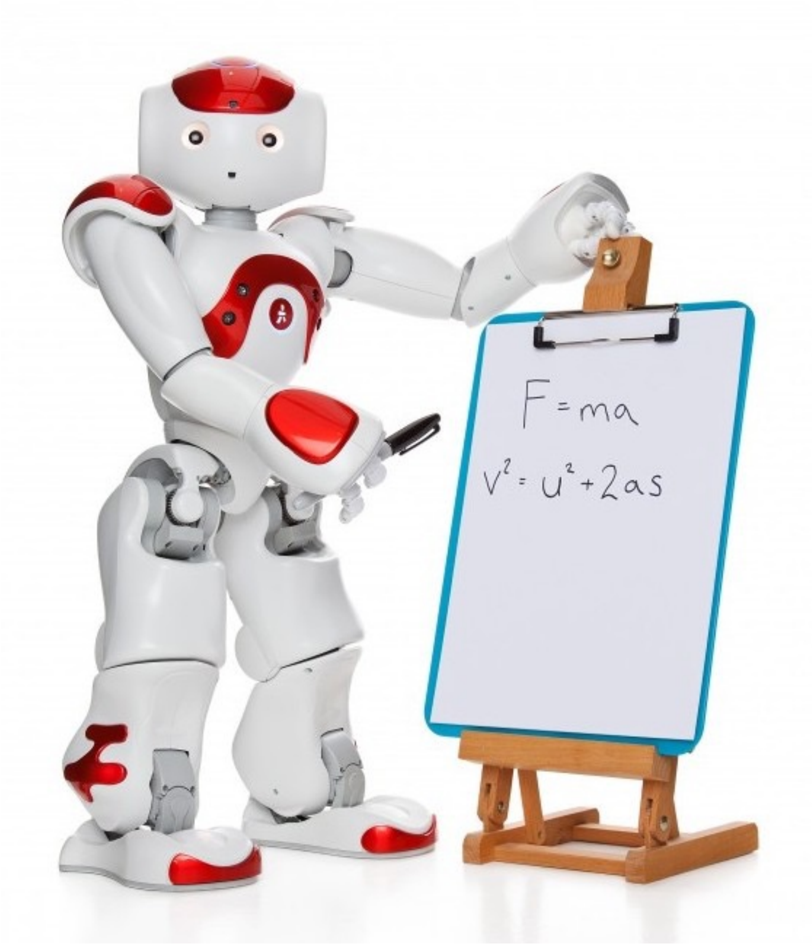
1a) getImageRemote command is sent to the robot

1b) the image is retrieved from the robot

2) the image is processed by our OCR software

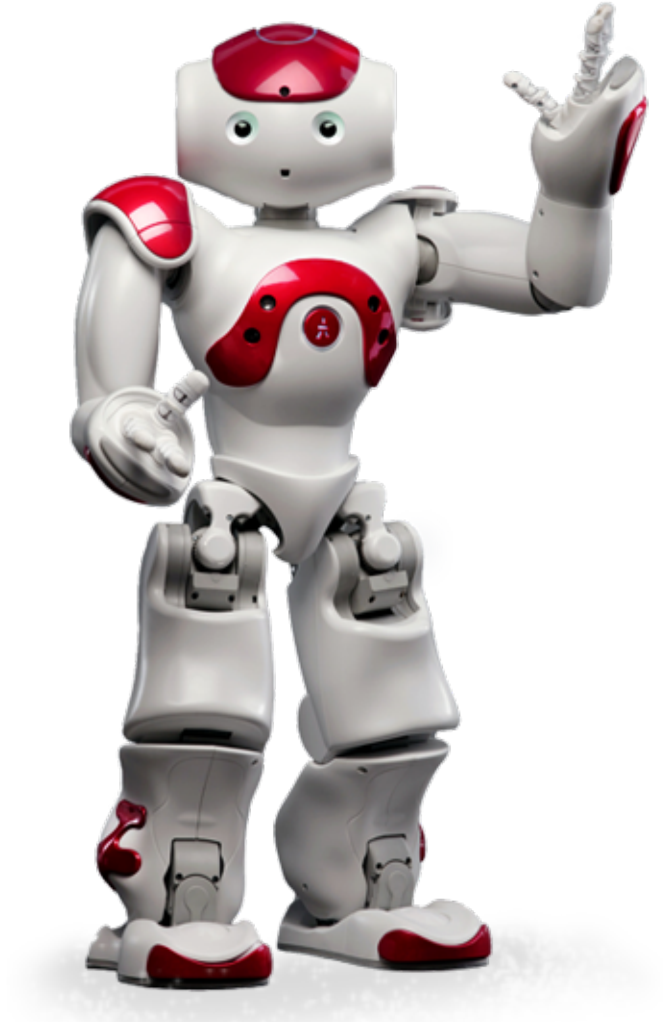
3) the text retrieved is sent to the robot

Presenting our NAO: Lucy



Let Lucy

introduce herself!



Thank you!