NAO
Interfacciamento e programmazione

Simone Costanzi

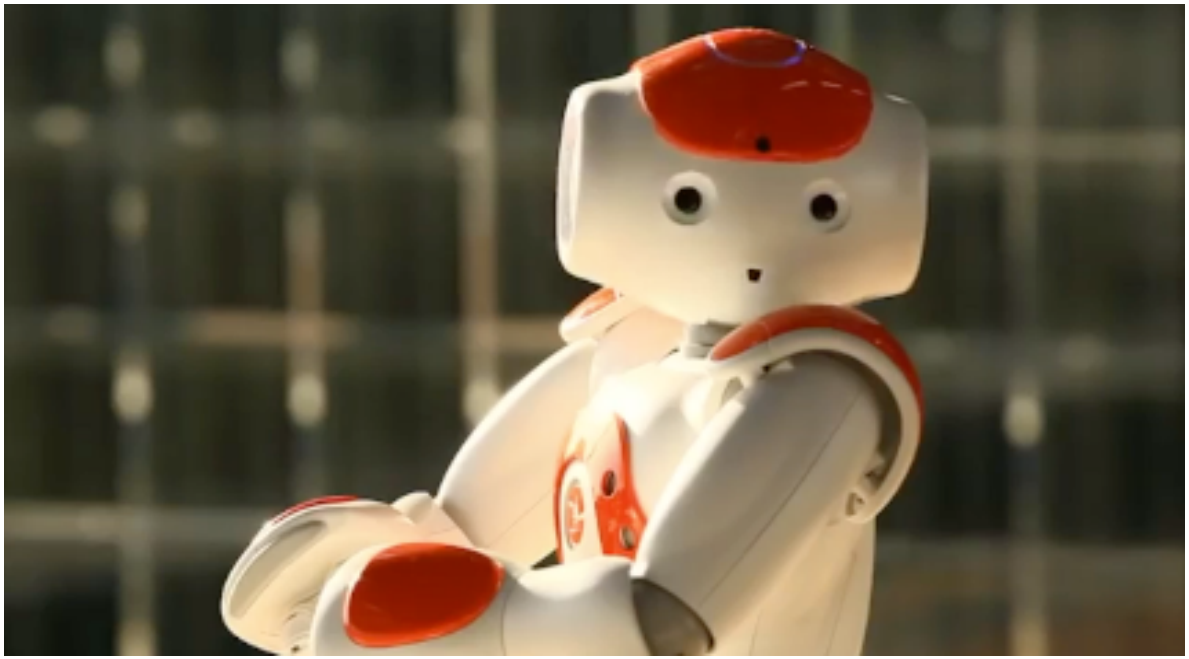
simone.costanzi2@unibo.it

Alma Mater Studiorum—Università di Bologna

March 23, 2021 - Cesena (Italy)

Outline

NAO



Soft Bank Robotics

Outline

Connessione Wi-Fi

Connessione Wi-Fi

- Collegare il robot al PC tramite cavo Ethernet e avviarlo[a]
- Terminata la fase di boot il PC assegnerá un indirizzo IP al robot
- Digitando questo indirizzo nel browser del PC sará possibile configurare la connessione Wi-Fi:
  - Se é il primo avvio verrá visualizzato un apposito wizard
  - altrimenti é possibile impostarla dalla sezione *Connessioni di rete* del sito web
- **Consiglio**: Per qualsiasi problema con la rete Wi-Fi con questi step é sempre possibile riconnettersi al robot e ripristinare la connessione

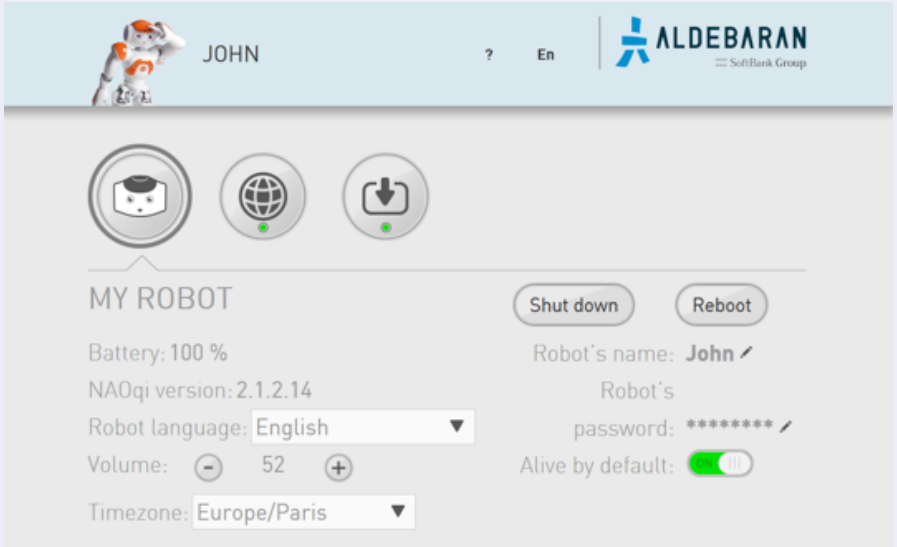
[a] Hello Guide

Outline

Metodi di comunicazione I

Sito Web

Il robot offre un'interfaccia web per alcune impostazioni (lingua, volume, connessione alla rete, ecc.) e per accedere allo store



- Per accedere digitare il suo indirizzo IP nel browser del PC che si trova nella stessa sottorete (User Name: `nao` Password: `nao`)

Metodi di comunicazione II

SSH

E' possibile fare accesso tramite Secure SHell:

- nel PC della stessa sottorete di NAO digitare la linea di comando `ssh nao@169.254.44.123`
- User Name: `nao` Password: `nao`

FTP

Per il trasferimento di file é presente il servizio *File Transfer Protocol*

- per es. FileZilla

Choregraphe

Ambiente di programmazione visuale (slide di sezione 2)

- Monitor

Outline

Programmazione

Riferimenti[1]:

- **NAO V5**
- Sistema operativo **NAOqi 2.1**
  - Distribuzione GNU/Linux basata su **Gentoo**

Linguaggi di programmazione:

Programming Languages	Bindings running on		Choregraphe support	
	Computer	Robot	Build Apps	Edit code
Python	✔	✔	✔	✔
C++	✔	✔	⊘	⊘
Java	✔	⊘	⊘	⊘
JavaScript	✔	✔	✔	⊘

✔	OK
⊘	Not available

[1] Nostro NAO

Outline

Python I

- Python é il principale linguaggio di programmazione del robot[2]
- Scaricare ed installare il framework python per la versione del nostro robot (**NAOqi 2.1**) dalla sezione downlod
- Il framework é in **Python 2.7**

[2] Python SDK

Python II

Suddivisione in moduli

- Il framework NAOqi é strutturato in moduli, ognuno dei quali fa riferimento ad un specifico componente (software, hardware o meccanico) del robot[a]
  - **ALMotion** - movimentazione
  - **ALAudioDevice** - periferica audio
  - **ALVideoDevice** - periferica video
  - **ALVisionRecognition** - training e riconoscimento di oggetti
  - **ALSpeechRecognition** - riconoscimento di parole
  - **...**
- Ognuno di questi offre un specifica API per interagire col relativo componente

[a] NAOqi APIs

Python III

Accesso remoto

Possibilitá di creare un proxy per ogni modulo in modo da potervi accedere anche esternamente al robot (PC nella stessa rete)

- **Esempio**[a]:

```
from naoqi import ALProxy
motion = ALProxy("ALMotion", "169.254.44.123", 9559)
tts    = ALProxy("ALTextToSpeech", "169.254.44.123", 9559)
motion.moveInit()
motion.post.moveTo(0.5, 0, 0)
tts.say("I'm walking")
```

[a] Making NAO move and speak

Python IV

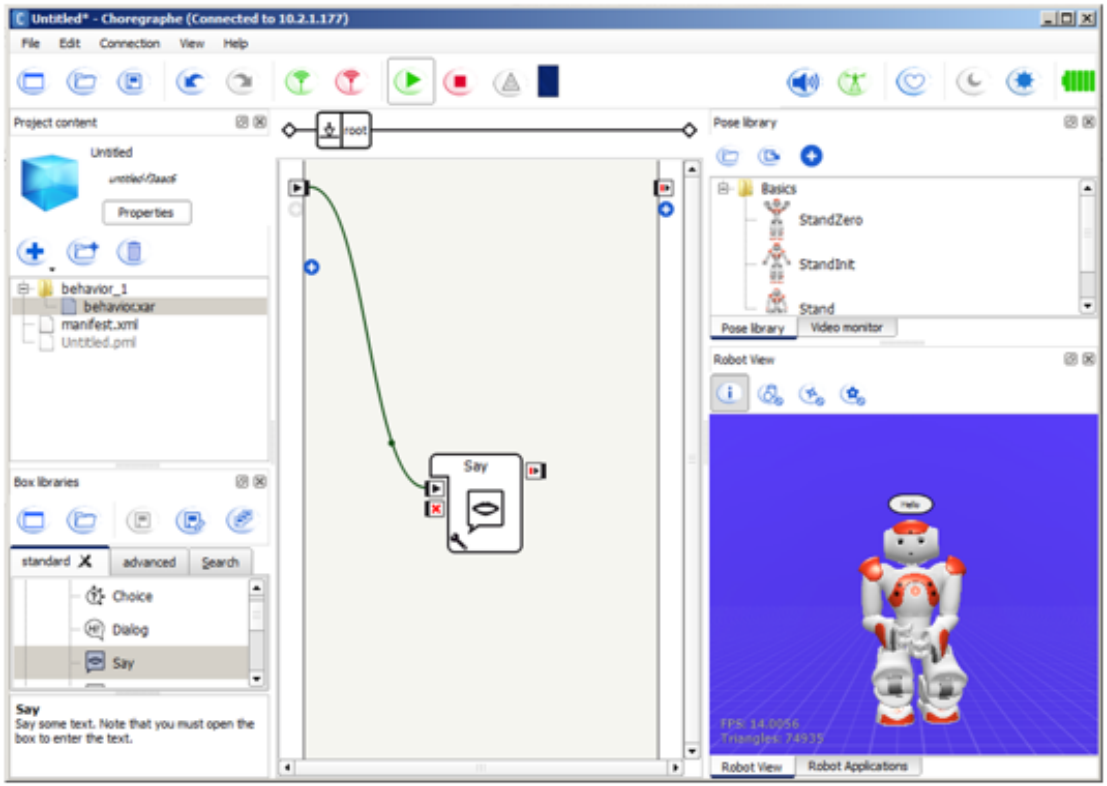
Programmazione ad eventi

- I moduli emettono specifici segnali (**eventi**) ai quali é possibile registrare le proprie **callback**
- **Consiglio**: controllare sempre quali segnali vengono emessi dai moduli e valutare in primis questo approccio
- **Esempio**: Evento FaceDetected del modulo `ALFaceDetection`

Outline

Choregraphe I

Programmazione visuale del robot NAO[3]



[3] Choregraphe User Guide

Choregraphe II

Behavior

- **Creazione** - Trascinare i blocchi dalla sezione *Box Libraries* al centro per creare la propria applicazione
- **Esecuzione** - Il pulsante Play verde nella parte alta esegue l'applicazione creata sul robot (fisico o virtuale)
- **Caricamento** - Da *Robot Application* in basso a sinistra (al posto della visualizzazione 3D del robot) é possibile installare l'applicazione creata sul robot premendo il pulsante *Install Application*

Nota: In *Robot Application* é possibile visualizzare tutte le app installate (comprese quelle scaricate dallo store)

Outline

Applicazione Dialog

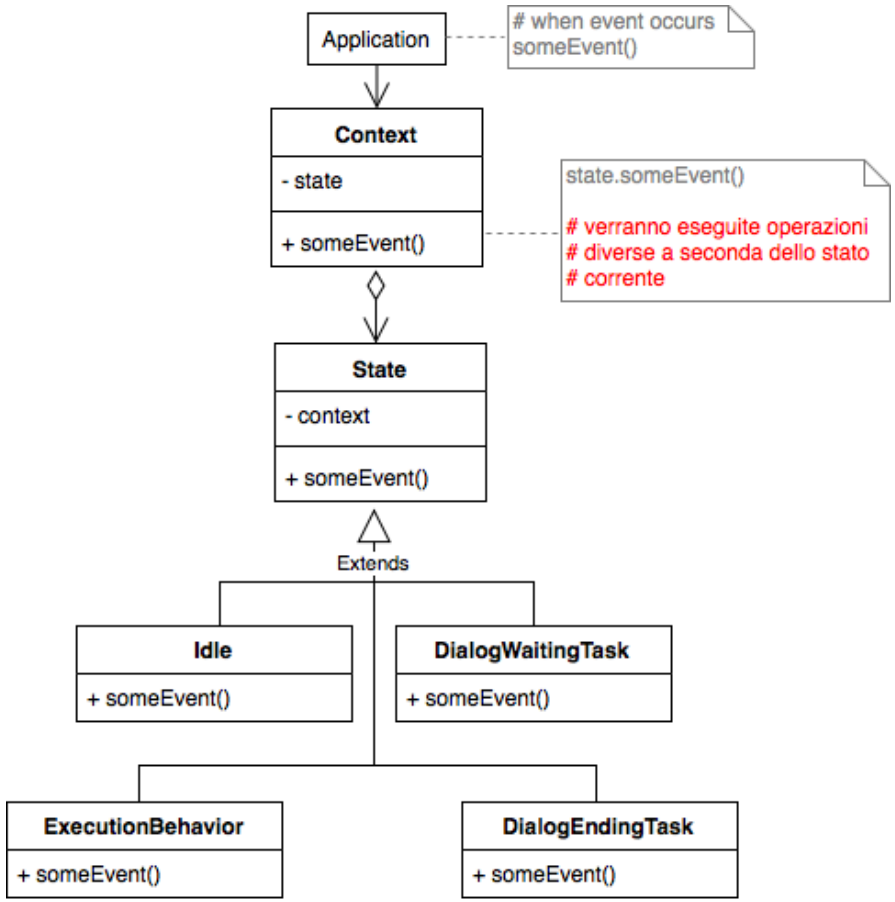
- Esempio concreto di applicazione sviluppata sulla piattaforma NAO[4]
- Si tratta di un vero e proprio dialogo tra NAO e un utente, dove é possibile chiedere al robot di esibirsi in uno dei seguenti task:
  - riconoscere oggetti
  - ballare la Maccarena
  - afferrare oggetti
  - eseguire il saggio di Tai-Chi
  - camminare insieme
- Nota: I behavior di afferrare gli oggetti, camminare con NAO ed il saggio di Tai-Chi sono stati scaricati dallo store ufficiale del robot

[4] Repository dell'applicazione

Outline

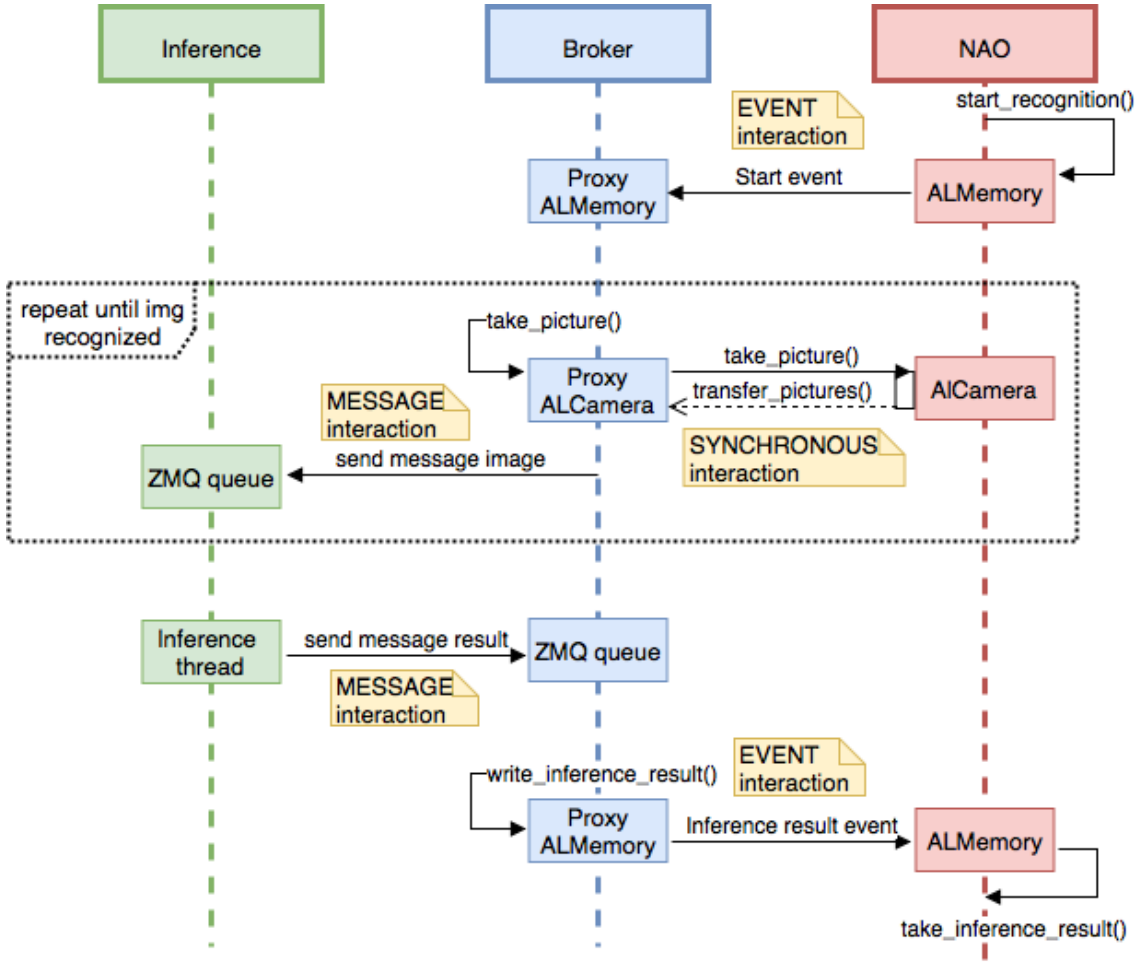
Macchina a stati

- In generale molto utile per definire diverse fasi di un comportamento
- Possibilitá di filtrare gli eventi dei moduli del robot in base allo stato del sistema
- Realizzata seguendo lo stile del **Pattern State**
  - A fianco un diagramma strutturale con i 4 stati usati nell'applicazione



Comunicazione con Python 3

- Interazione del task di riconoscimento
- Tre sottosistemi:
  - NAO (Python 2)
  - Broker (Python 2)
  - Inference (Python 3)
- Modulo Broker (PC) in python 2 che fa da intermediario fra NAO, tramite **naoqi**, e il modulo di Inference (PC) in python 3 tramite ZMQ



Outline

Link ai Repository e alle Applicazioni

- Applicazione principale di dialogo
  - Dialog
- Task di riconoscimento
  - Broker
  - Inference
- Applicazione choregraphe Maccarena e parte choregraphe del task di riconoscimento
  - Applicazioni NAO