One day with a humanoid robot

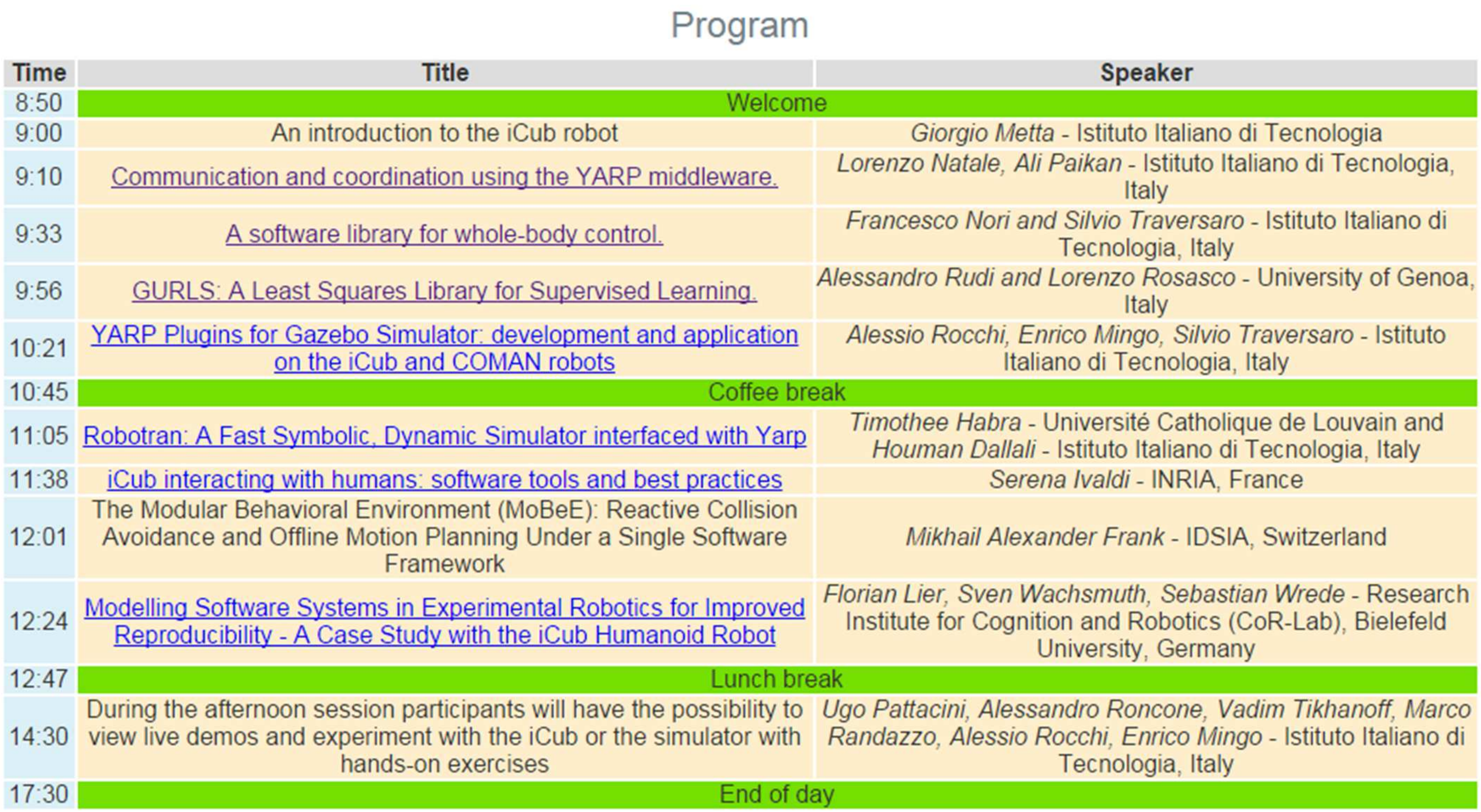
a crash course on the iCub software tools

as part of
2014 IEEE-RAS International Conference
on Humanoid Robots

November, 18th 2014
Madrid, Spain

L. Natale(1), F.Nori(2), U. Pattacini(1), V. Tikhanoff(1), M. Randazzo(1), G. Metta(1)

(1) iCub Facility
(2) Robotics, Brain and Cognitive Sciences
Istituto Italiano di Tecnologia (IIT)
Via Morego, 30 - 16163 Genoa, Italy



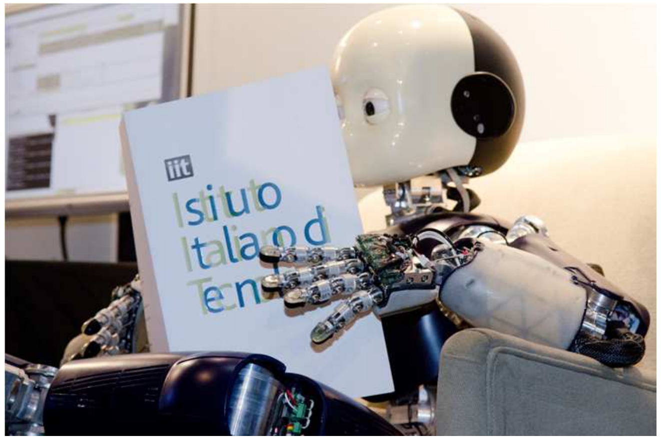
Program

Time	Title	Speaker
8:50	Welcome	
9:00	An introduction to the iCub robot	*Giorgio Metta* - Istituto Italiano di Tecnologia
9:10	Communication and coordination using the YARP middleware.	*Lorenzo Natale, Ali Paikan* - Istituto Italiano di Tecnologia, Italy
9:33	A software library for whole-body control.	*Francesco Nori and Silvio Traversaro* - Istituto Italiano di Tecnologia, Italy
9:56	GURLS: A Least Squares Library for Supervised Learning.	*Alessandro Rudi and Lorenzo Rosasco* - University of Genoa, Italy
10:21	YARP Plugins for Gazebo Simulator: development and application on the iCub and COMAN robots	*Alessio Rocchi, Enrico Mingo, Silvio Traversaro* - Istituto Italiano di Tecnologia, Italy
10:45	Coffee break	
11:05	Robotran: A Fast Symbolic, Dynamic Simulator interfaced with Yarp	*Timothee Habra* - Université Catholique de Louvain and *Houman Dallali* - Istituto Italiano di Tecnologia, Italy
11:38	iCub interacting with humans: software tools and best practices	*Serena Ivaldi* - INRIA, France
12:01	The Modular Behavioral Environment (MoBeE): Reactive Collision Avoidance and Offline Motion Planning Under a Single Software Framework	*Mikhail Alexander Frank* - IDSIA, Switzerland
12:24	Modelling Software Systems in Experimental Robotics for Improved Reproducibility - A Case Study with the iCub Humanoid Robot	*Florian Lier, Sven Wachsmuth, Sebastian Wrede* - Research Institute for Cognition and Robotics (CoR-Lab), Bielefeld University, Germany
12:47	Lunch break	
14:30	During the afternoon session participants will have the possibility to view live demos and experiment with the iCub or the simulator with hands-on exercises	*Ugo Pattacini, Alessandro Roncone, Vadim Tikhanoff, Marco Randazzo, Alessio Rocchi, Enrico Mingo* - Istituto Italiano di Tecnologia, Italy
17:30	End of day	

ISTITUTO ITALIANO
DI TECNOLOGIA

Lorenzo Natale

Coordination and Communication with the YARP middleware



Humanoids 2015 workshop: A day with a humanoid robot
November 18, Madrid Spain

why is the iCub special?

- hands: we started the design from the hands
  - 5 fingers, 9 degrees of freedom, 19 joints

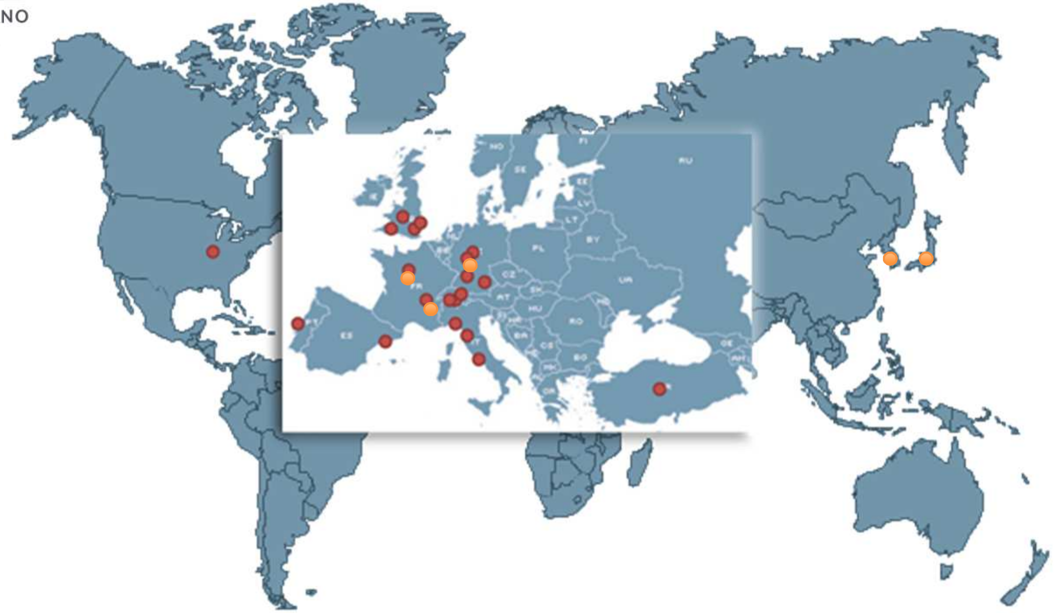
- sensors: human-like, e.g. no lasers
  - cameras, microphones, gyros, encoders, force, tactile…

- electronics: flexibility for research
  - custom electronics, small, programmable (DSPs)

- reproducible platform: community designed
  - reproducible & maintainable yet evolvable platform
  - large software repository (~2M lines of code)

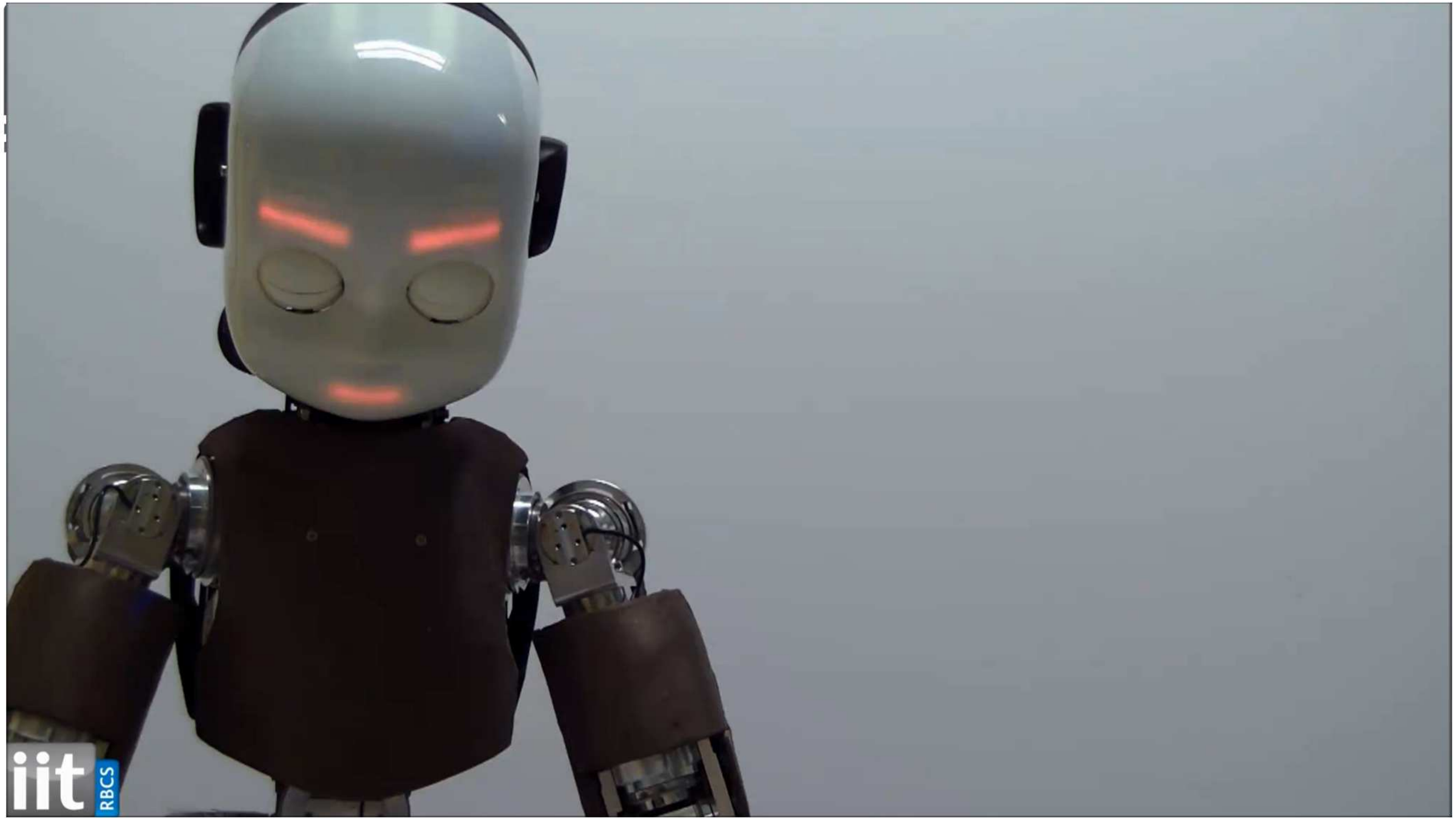
ISTITUTO ITALIANO
DI TECNOLOGIA

the iCub



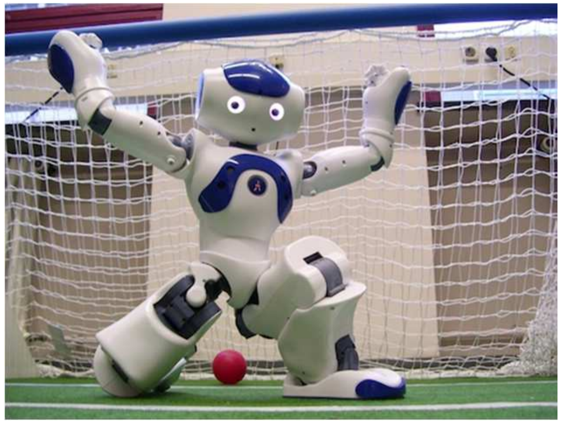
price: 250K€
30 iCub
distributed since 2008
about 3-4 iCub’s/year

open hardware

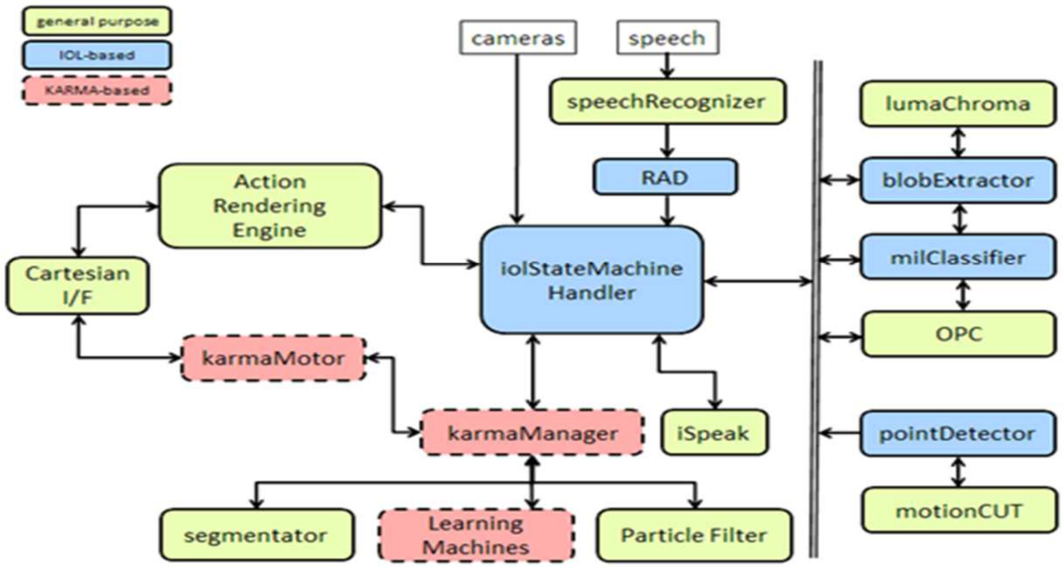


ISTITUTO ITALIANO
DI TECNOLOGIA

Applications for humanoid robotics

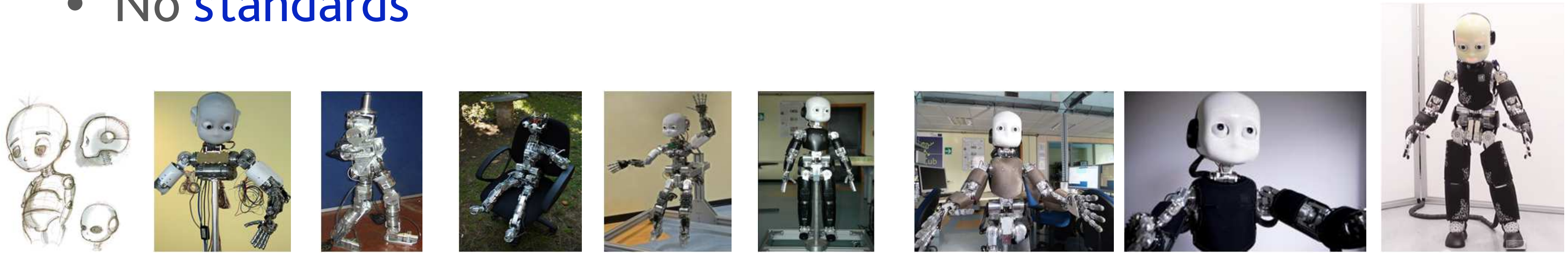


Programming complex behaviors



Key Issues

- Inherent complexity, distributed processing, lots of sensors, real-time
- Asynchronous development
- Various scenarios and platforms
- Fluctuation in hardware and algorithms, lots of open questions
- No standards



Why Yet Another Robot Platform

Why Yet Another Robot Platform

- Started ~2001 as been adopted as the iCub software middleware

CMake
Cross-platform Make
SWIG

Why Yet Another Robot Platform

- Started ~2001 as been adopted as the iCub software middleware
- Peer-to-peer loosely coupled components

CMake
Cross-platform Make
SWIG

Why Yet Another Robot Platform

- Started ~2001 as been adopted as the iCub software middleware
- Peer-to-peer loosely coupled components
- Minimal dependencies/portable

CMake
Cross-platform Make

SWIG

Why Yet Another Robot Platform

- Started ~2001 as been adopted as the iCub software middleware
- Peer-to-peer loosely coupled components
- Minimal dependencies/portable
- Interface for common hardware devices

Why Yet Another Robot Platform

- Started ~2001 as been adopted as the iCub software middleware
- Peer-to-peer loosely coupled components
- Minimal dependencies/portable
- Interface for common hardware devices
- YCM, support for build system based on CMake

Why Yet Another Robot Platform

- Started ~2001 as been adopted as the iCub software middleware
- Peer-to-peer loosely coupled components
- Minimal dependencies/portable
- Interface for common hardware devices
- YCM, support for build system based on CMake
- Facilitate interoperability and coordination

Why Yet Another Robot Platform

- Started ~2001 as been adopted as the iCub software middleware
- Peer-to-peer loosely coupled components
- Minimal dependencies/portable
- Interface for common hardware devices
- YCM, support for build system based on CMake
- Facilitate interoperability and coordination

this talk

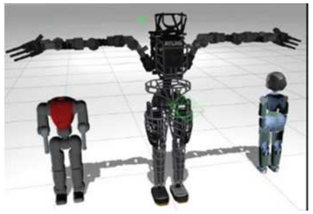
Robot Interface

- Communicating through ports becomes easily complex
- Abstraction layers
  - Separate communication details from interface (streaming, rpc etc)
  - Allows remotization
  - Protect from hardware fluctuations

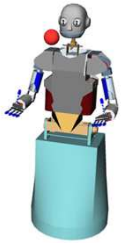
Control loops

```
…
…
read encoders
read IMU
read FT
…
get image
…
set position
```

Robot Interface



Gazebo



Armar III

iCub

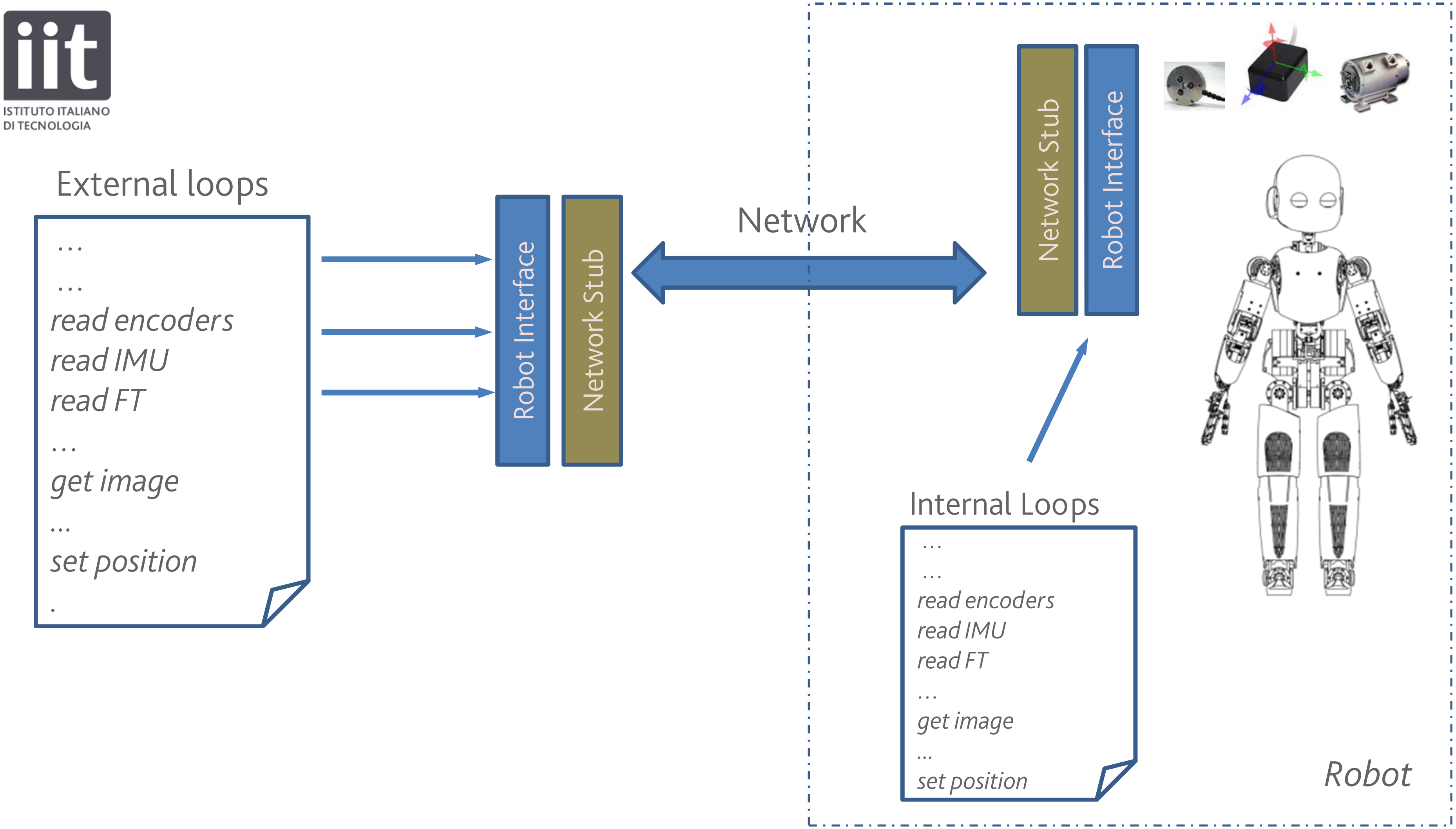
COMAN

Robotran

Multibody Models Online

iit

ISTITUTO ITALIANO
DI TECNOLOGIA



External loops

…
…
read encoders
read IMU
read FT
…
get image
…
set position
.

Robot Interface

Network Stub

Network

Network Stub

Robot Interface

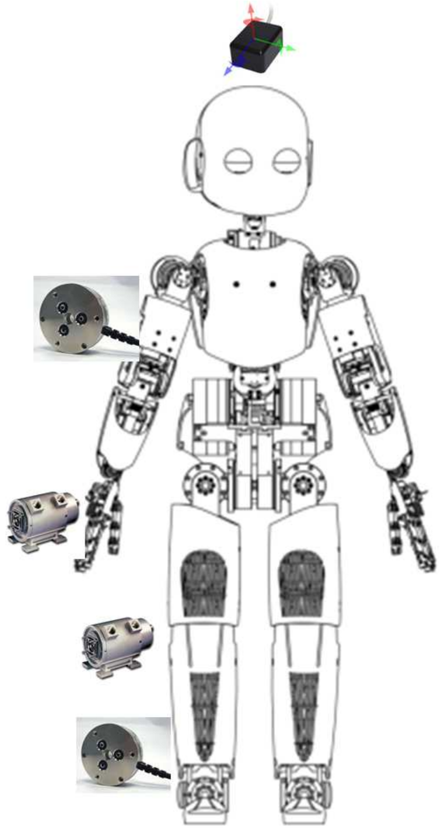
Internal Loops

…
…
read encoders
read IMU
read FT
…
get image
…
set position

Robot

Type of interfaces

- Motor control (position, velocity, open-loop, torque, impedance)
- Sensors: IMU, cameras, torques, F/T, encoders, skin
- Devices can also be virtual



ISTITUTO ITALIANO DI TECNOLOGIA

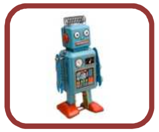
…more on this later this morning

YCM distributed development

- Development is distributed in small repositories
- Libraries and modules are agglomerated in large builds
- Share code not binaries
- Mixed software management tools (GIT, svn, …)

- Built on top of CMake (several patches contributed upstream)

ISTITUTO ITALIANO
DI TECNOLOGIA



FooProject

foo-project.org

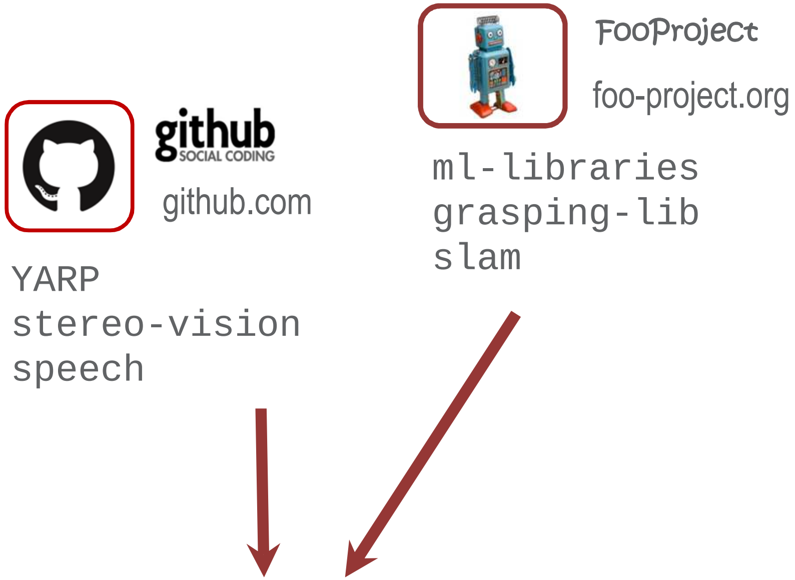
```
ml-libraries
grasping-lib
slam
```

github
SOCIAL CODING

github.com

```
YARP
stereo-vision
speech
```

ISTITUTO ITALIANO
DI TECNOLOGIA



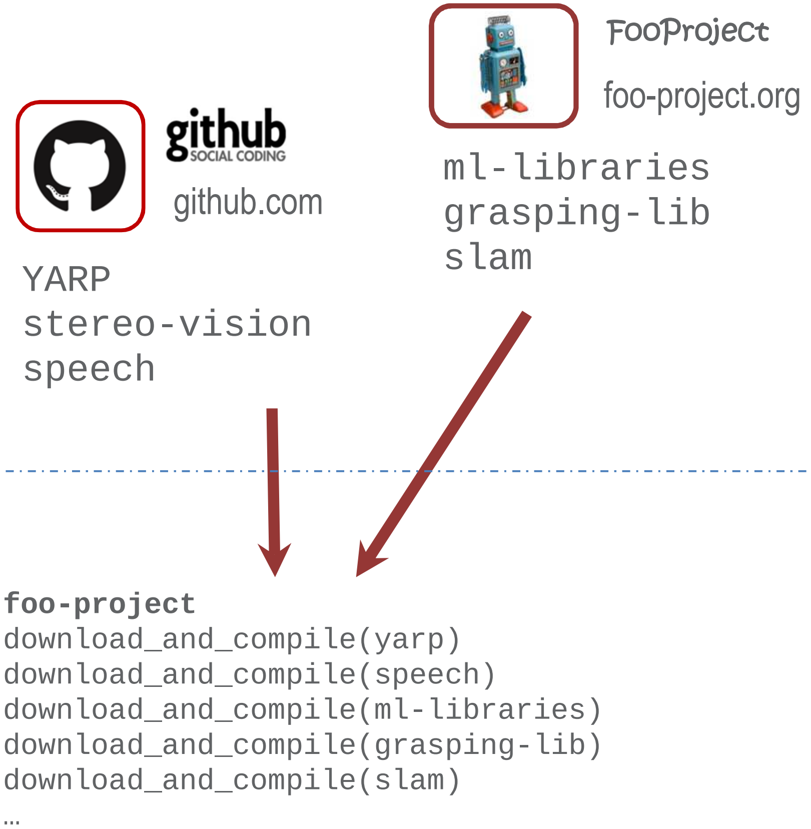
github
SOCIAL CODING
github.com

YARP
stereo-vision
speech

FooProject
foo-project.org

ml-libraries
grasping-lib
slam

```
foo-project
download_and_compile(yarp)
download_and_compile(speech)
download_and_compile(ml-libraries)
download_and_compile(grasping-lib)
download_and_compile(slam)
…
```



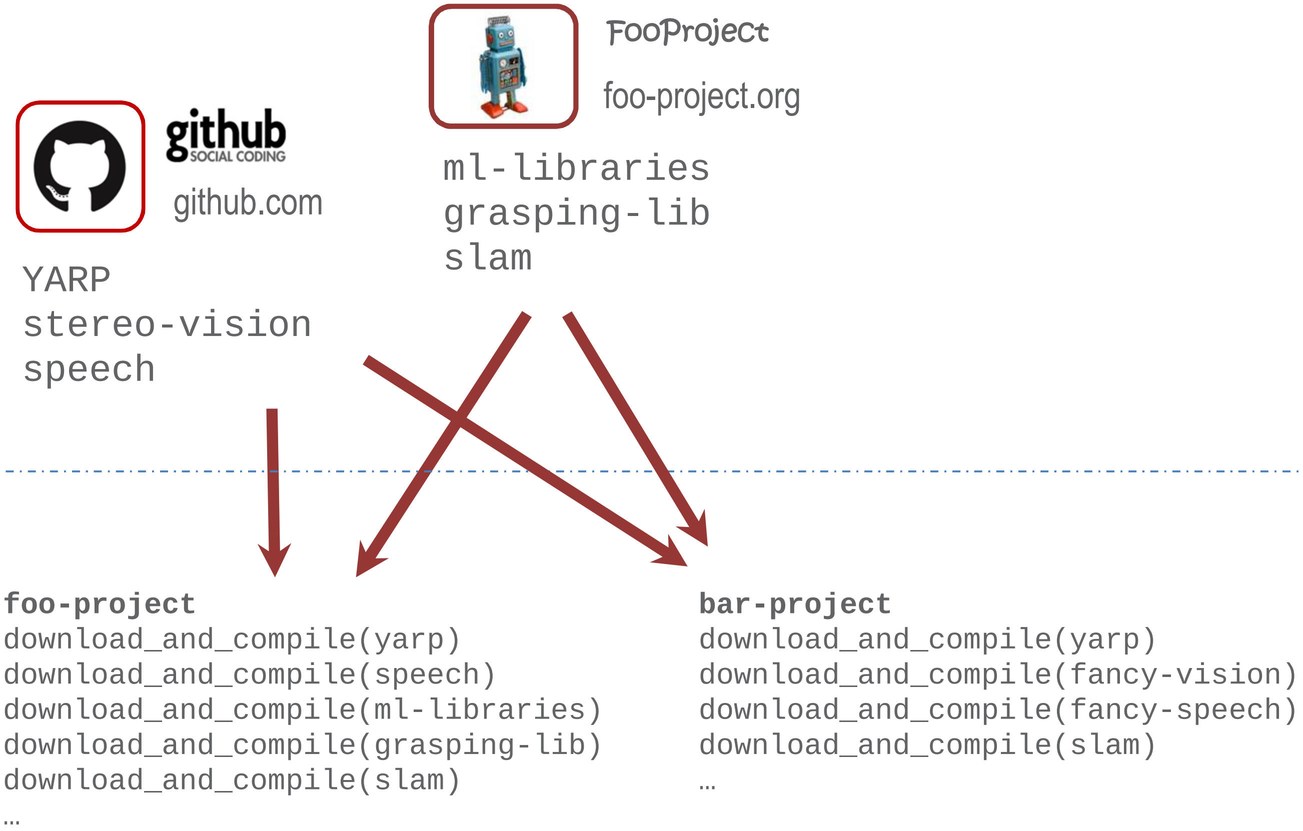
```
YARP
stereo-vision
speech
```

```
ml-libraries
grasping-lib
slam
```

```
foo-project
download_and_compile(yarp)
download_and_compile(speech)
download_and_compile(ml-libraries)
download_and_compile(grasping-lib)
download_and_compile(slam)
…
```

Issue & Bug Tracking
Documentation
Continuous integration
Better visibility

Easier deployment
Documentation
Continuous Integration



github
SOCIAL CODING
github.com

```
YARP
stereo-vision
speech
```

FooProject
foo-project.org

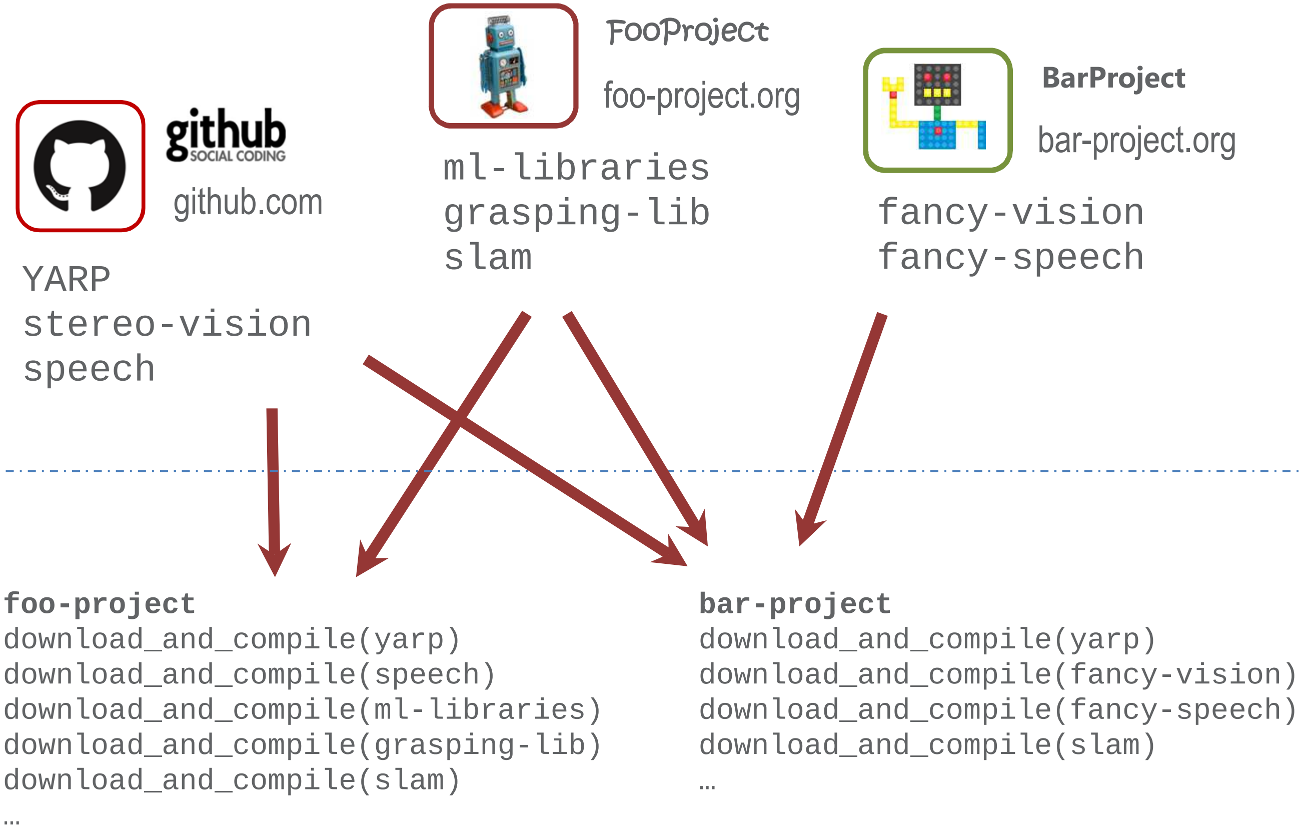
```
ml-libraries
grasping-lib
slam
```

```
foo-project
download_and_compile(yarp)
download_and_compile(speech)
download_and_compile(ml-libraries)
download_and_compile(grasping-lib)
download_and_compile(slam)
…
```

```
bar-project
download_and_compile(yarp)
download_and_compile(fancy-vision)
download_and_compile(fancy-speech)
download_and_compile(slam)
…
```

Issue & Bug Tracking
Documentation
Continuous integration
Better visibility

Easier deployment
Documentation
Continuous Integration



github
SOCIAL CODING
github.com

YARP
stereo-vision
speech

FooProject
foo-project.org

ml-libraries
grasping-lib
slam

BarProject
bar-project.org

fancy-vision
fancy-speech

```
foo-project
download_and_compile(yarp)
download_and_compile(speech)
download_and_compile(ml-libraries)
download_and_compile(grasping-lib)
download_and_compile(slam)
…
```

```
bar-project
download_and_compile(yarp)
download_and_compile(fancy-vision)
download_and_compile(fancy-speech)
download_and_compile(slam)
…
```

Issue & Bug Tracking
Documentation
Continuous integration
Better visibility

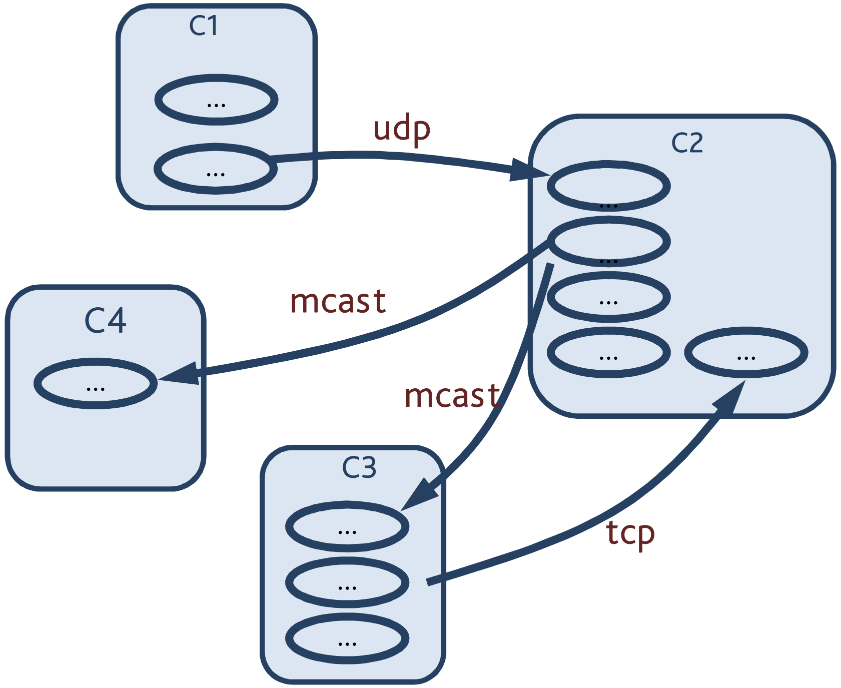
Easier deployment
Documentation
Continuous Integration

ISTITUTO ITALIANO
DI TECNOLOGIA

Communication

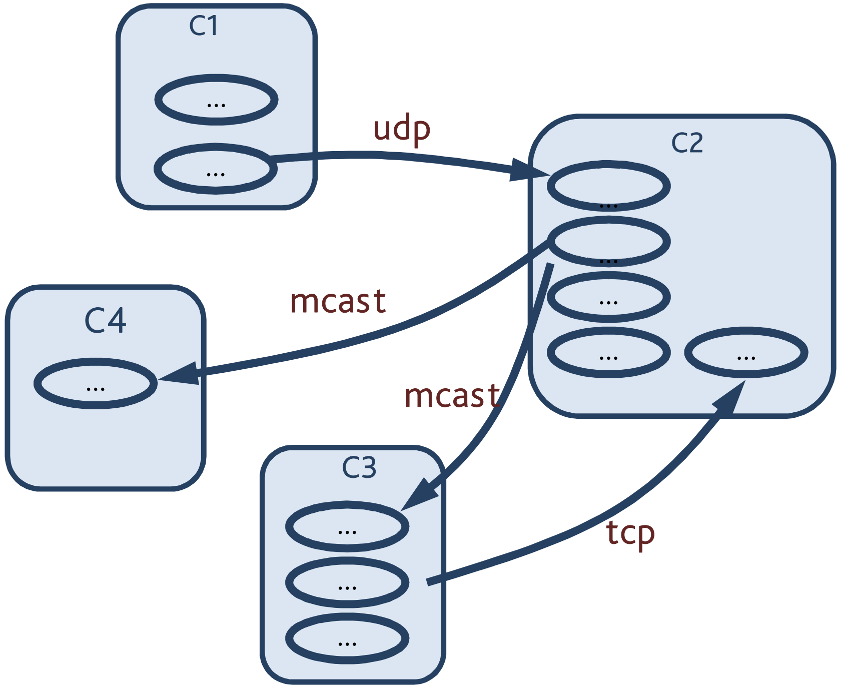
Communication

- Peer-to-peer



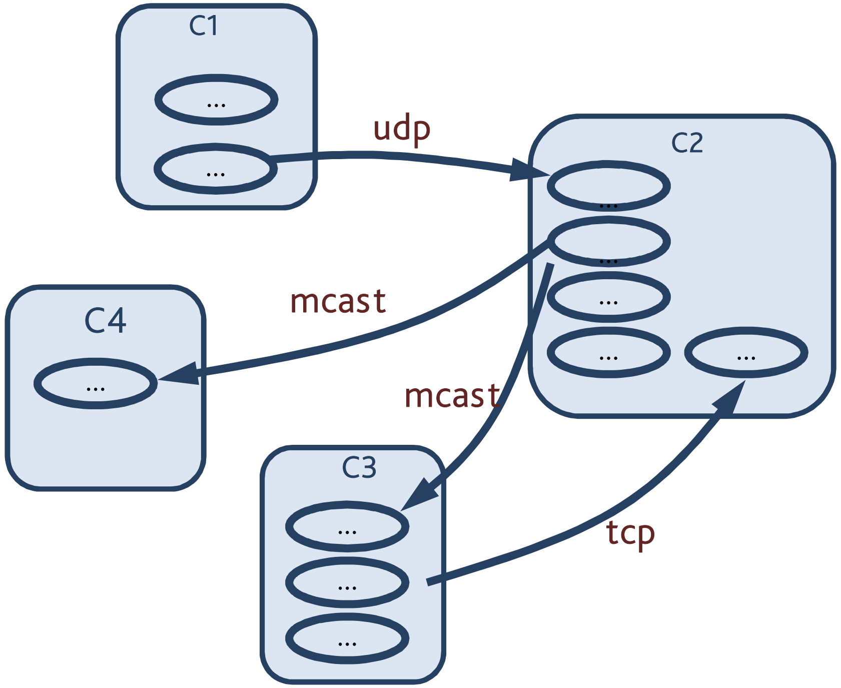
Communication

- Peer-to-peer
- Dynamic topology (can also be statically defined)



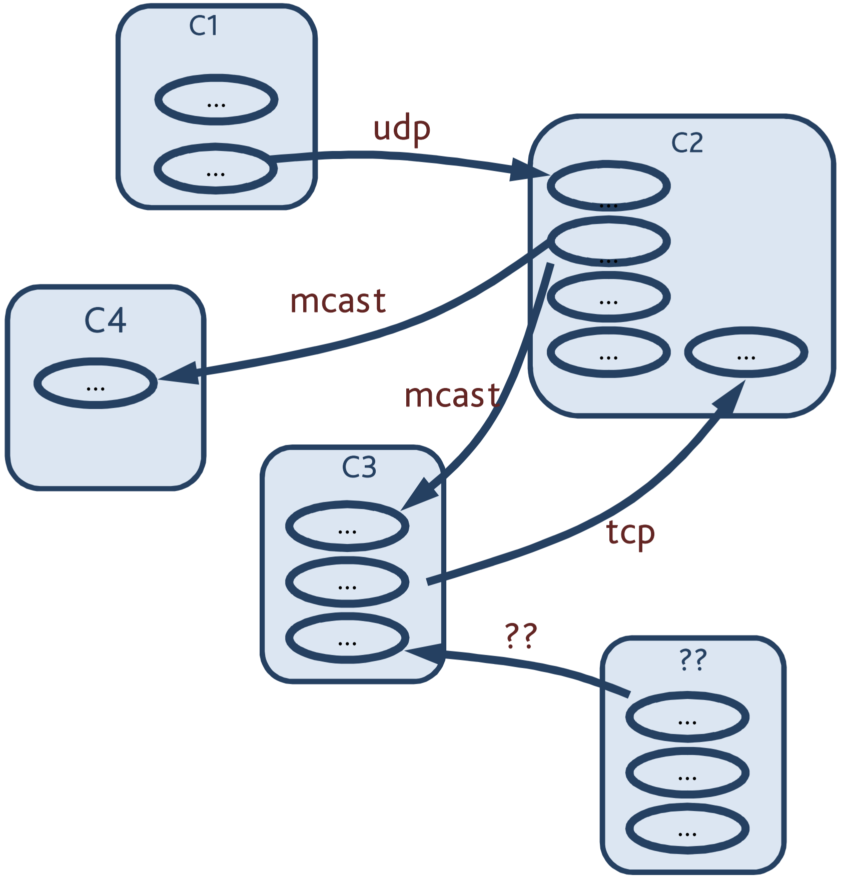
Communication

- Peer-to-peer
- Dynamic topology (can also be statically defined)
- Loosely typed, but IDL language can specify types and interfaces



Communication

- Peer-to-peer
- Dynamic topology (can also be statically defined)
- Loosely typed, but IDL language can specify types and interfaces
- Carriers: protocols can be extended as plugins and configured at runtime



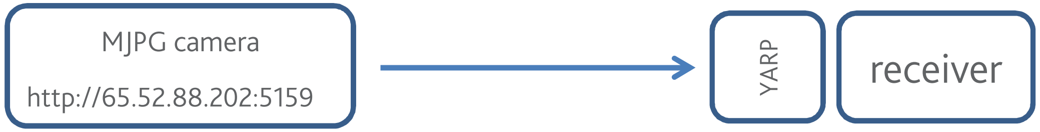
YARP plugins

- YARP includes a plugin system for drivers and protocols (carriers)
- Interchangeable carriers allow:
  - interfacing existing software with ports (without bridges)
  - change significantly port behavior
- Examples:
  - mjpeg, xml rpc, ROS, …
  - Bayer carrier, port-monitor

ISTITUTO ITALIANO
DI TECNOLOGIA

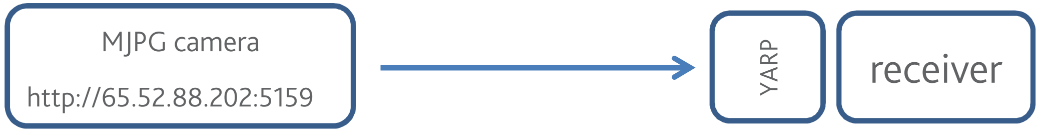
Examples

Examples

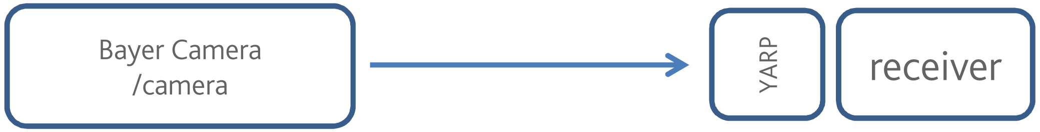


```
yarp  connect  /65.52.88.202:5159  /receiver  mjpeg
```

Examples

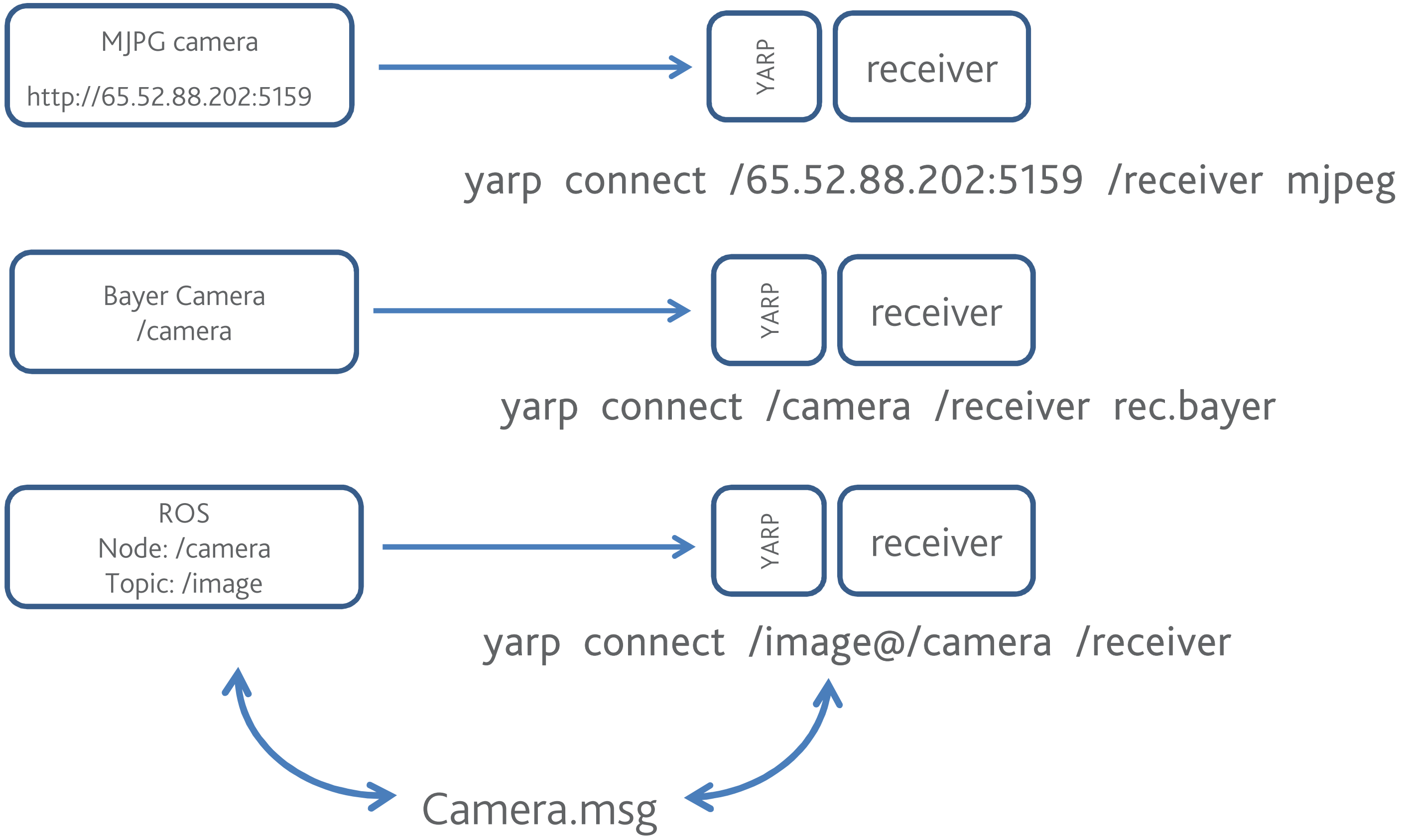


```
yarp  connect  /65.52.88.202:5159  /receiver  mjpeg
```



```
yarp  connect  /camera  /receiver  rec.bayer
```

Examples



MJPG camera

http://65.52.88.202:5159

YARP

receiver

```
yarp  connect  /65.52.88.202:5159  /receiver  mjpeg
```

Bayer Camera
/camera

YARP

receiver

```
yarp  connect  /camera  /receiver  rec.bayer
```

ROS
Node: /camera
Topic: /image

YARP

receiver

```
yarp  connect  /image@/camera  /receiver
```

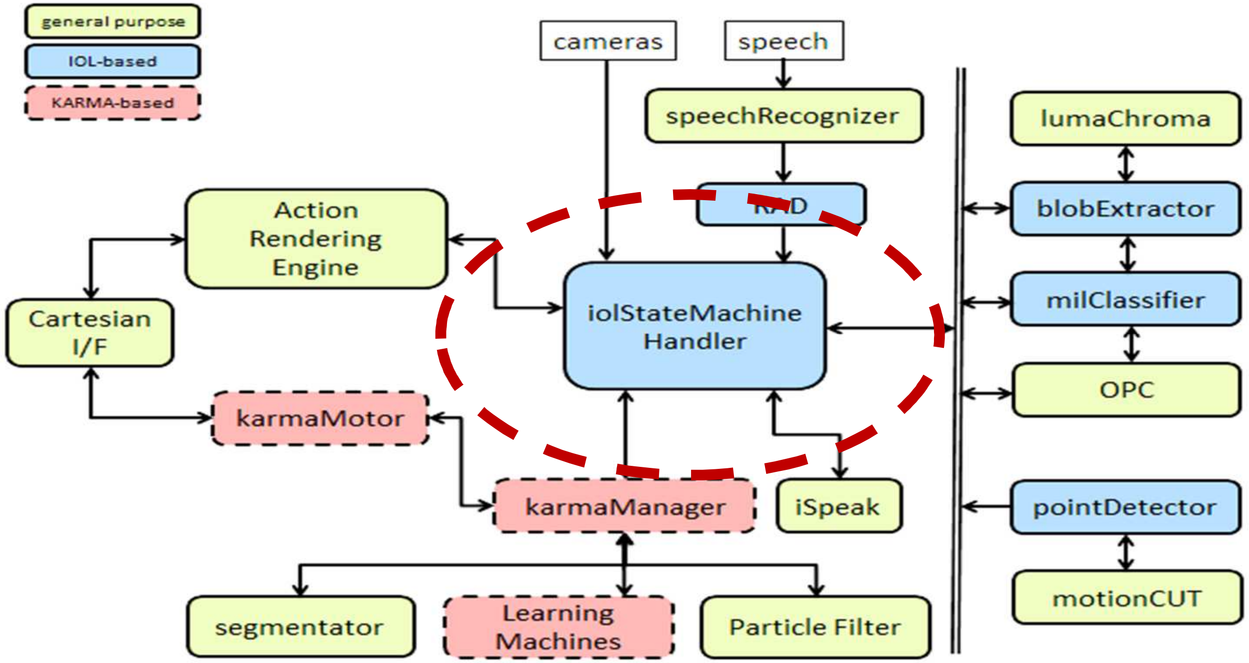
Camera.msg

More on YARP-ROS

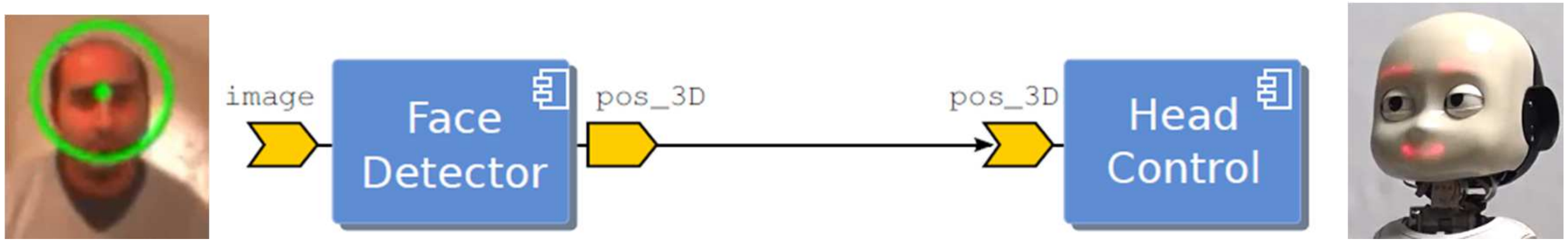
- Type server providing type information at runtime (YARP-ROS without ROS)
- Compatibility with ROS nameserver
- Concept of nodes
- Type and direction information within ports

- Check-out www.yarp.it → YARP with ROS

Coordinating modules



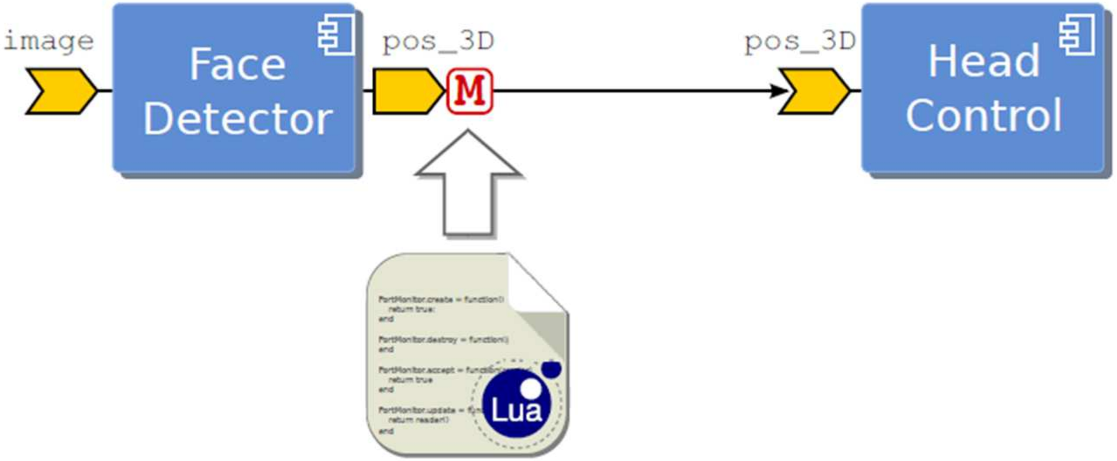
A simple example



Track a face if and only the confidence level (certainty) of the Face Detector is above a desired threshold.

- Simple scenario poses questions on the design of the components
- Some functionalities are application dependent
- Should we:
  - extend Head Control, Face Detector?
  - Add a a separate filtering module?

Port monitor plug-in

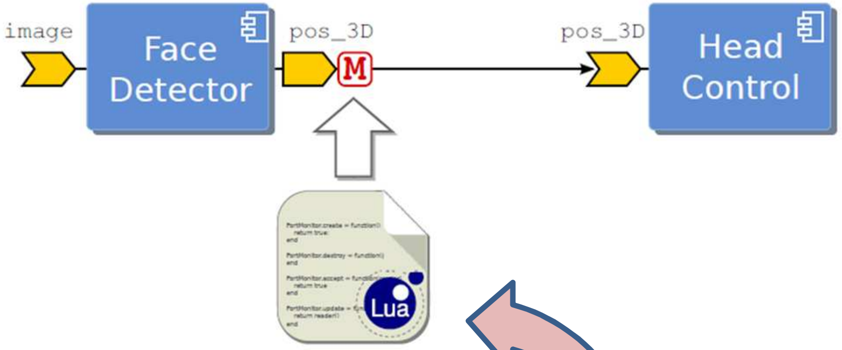


```
PortMonitor.create = function()
    return true;
end

PortMonitor.destroy = function()
end

PortMonitor.accept = functi
    return true
end

PortMonitor.update = f
    return reader()
end
```

The port monitor approach:

- Add code that handles outgoing (or incoming) data
- Dynamically loading or configuring a run-time script (e.g. Lua)
- Monitoring, filtering, and transforming....

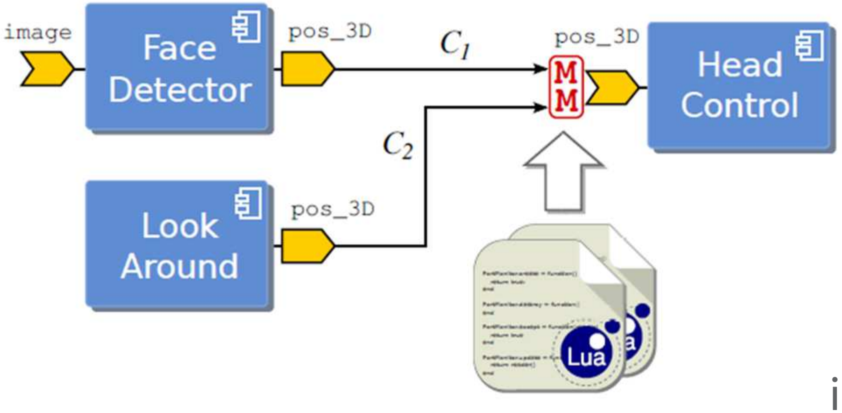
Track a face if and only the confidence level (certainty) of the Face Detector is above a desired threshold (e.g. 80%).



Application configuration (not compile time)

```
if (data.certainty > 0.9)
    accept(data)
else
    drop(data)
```

Port arbitration using plug-ins



```
if (C1.certainty > 0.9)
    accept(C1)
else
    accept(C2)
```

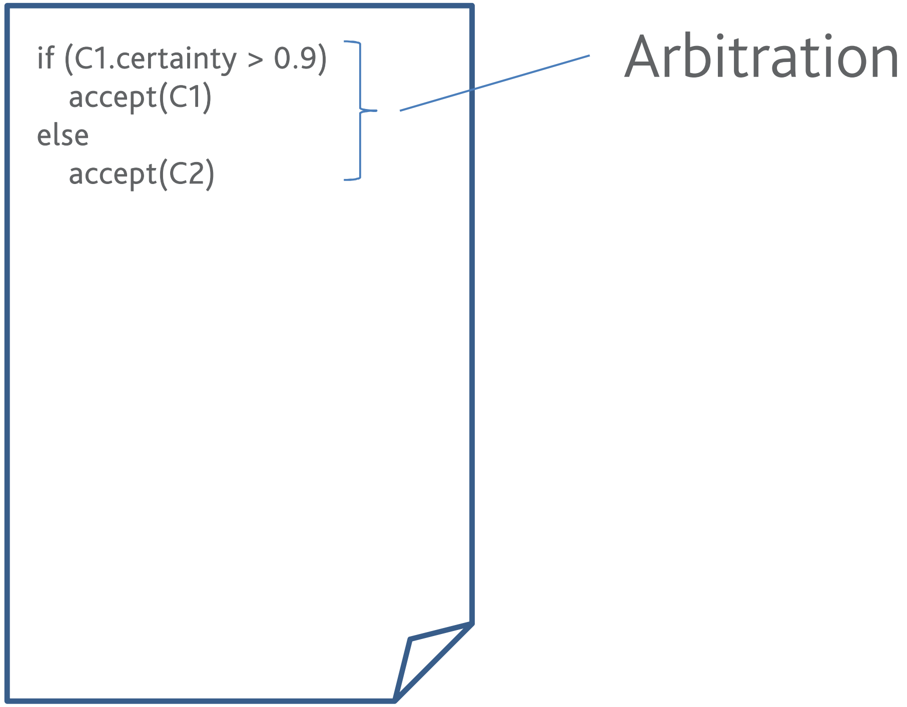
Example:

- Search and Track a face

Requirements:

- Monitoring the confidence level of Face Detector
- Arbitrating the connections

Potential applications

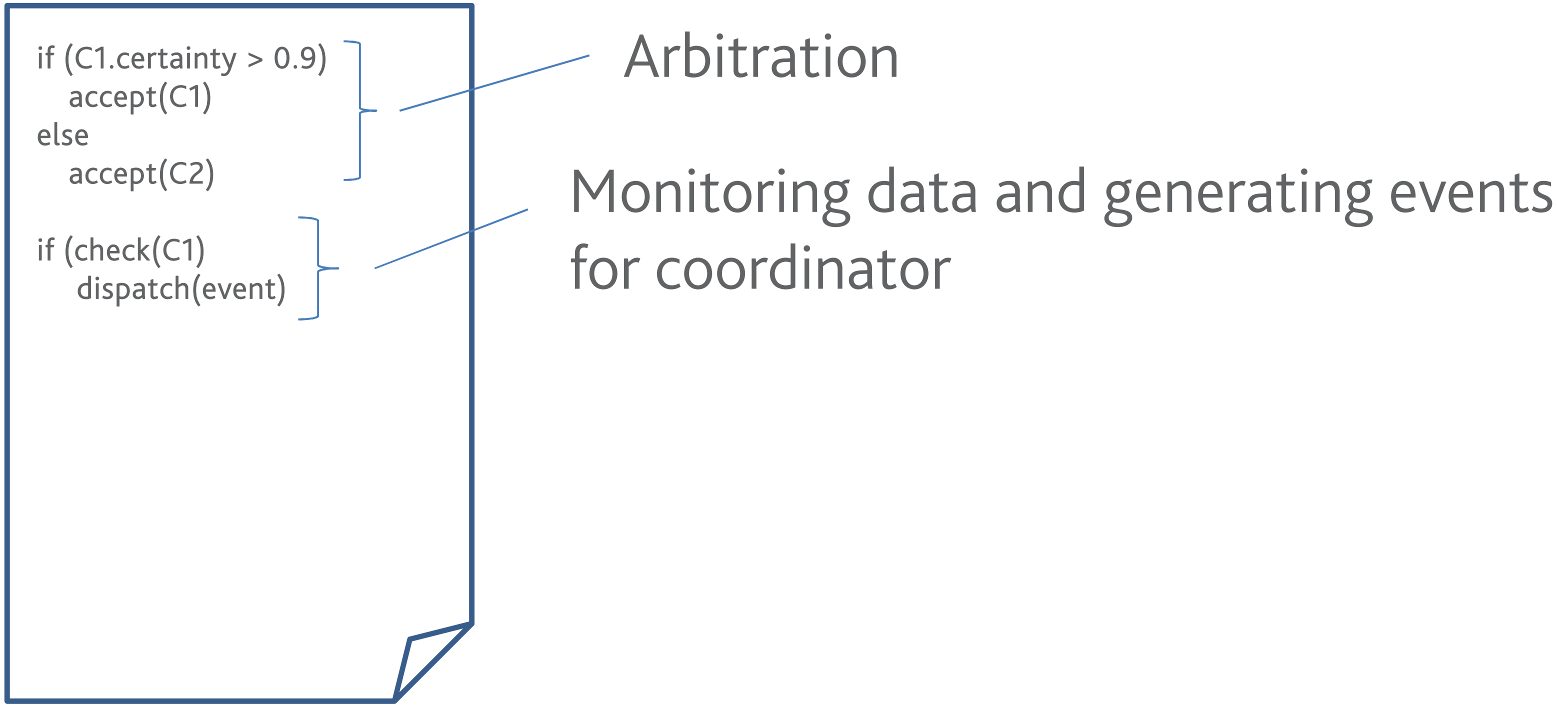


```
if (C1.certainty > 0.9)
    accept(C1)
else
    accept(C2)
```

Arbitration

Online tutorials: www.yarp.it → Port monitoring and arbitration

Potential applications



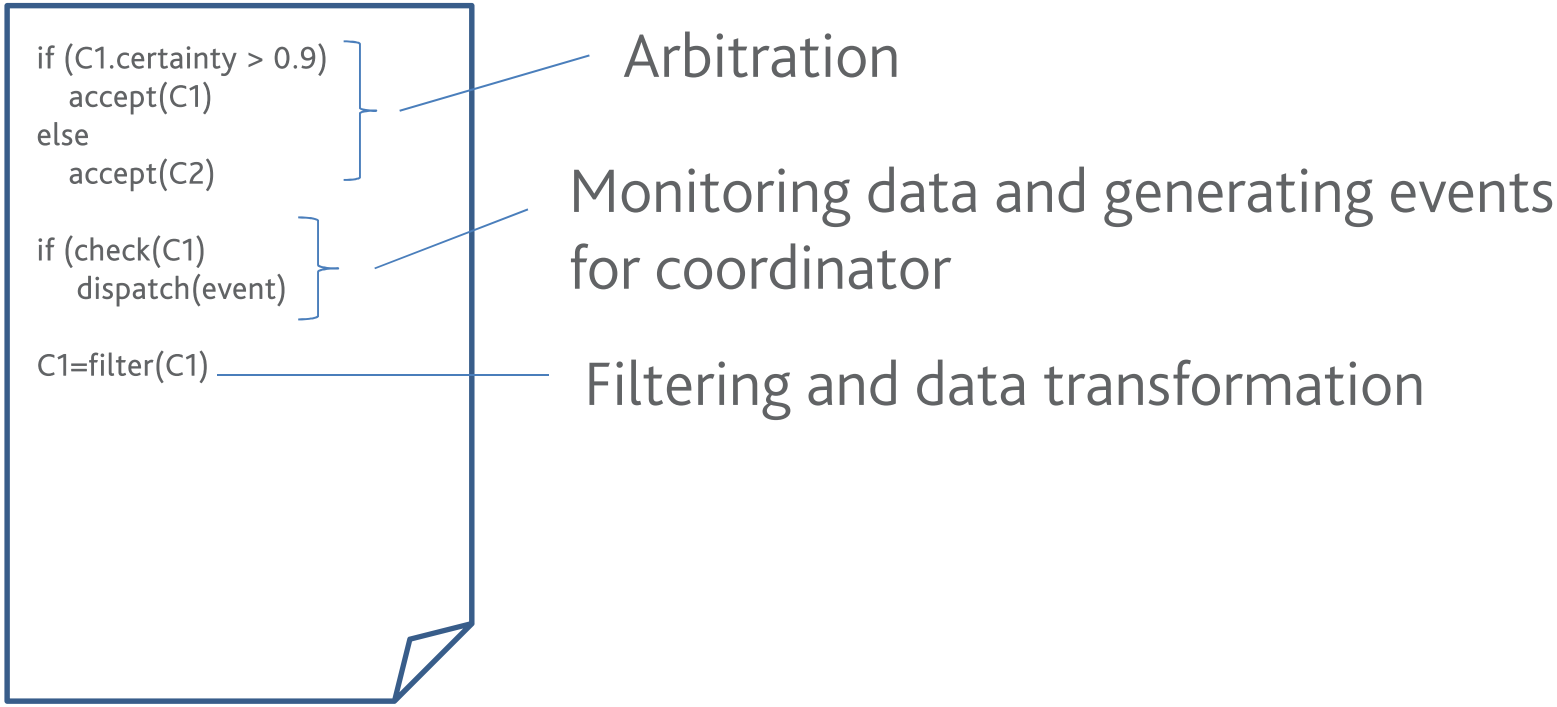
```
if (C1.certainty > 0.9)
    accept(C1)
else
    accept(C2)

if (check(C1)
     dispatch(event)
```

Arbitration

Monitoring data and generating events for coordinator

Online tutorials: www.yarp.it → Port monitoring and arbitration

Potential applications



```
if (C1.certainty > 0.9)
    accept(C1)
else
    accept(C2)

if (check(C1)
     dispatch(event)

C1=filter(C1)
```

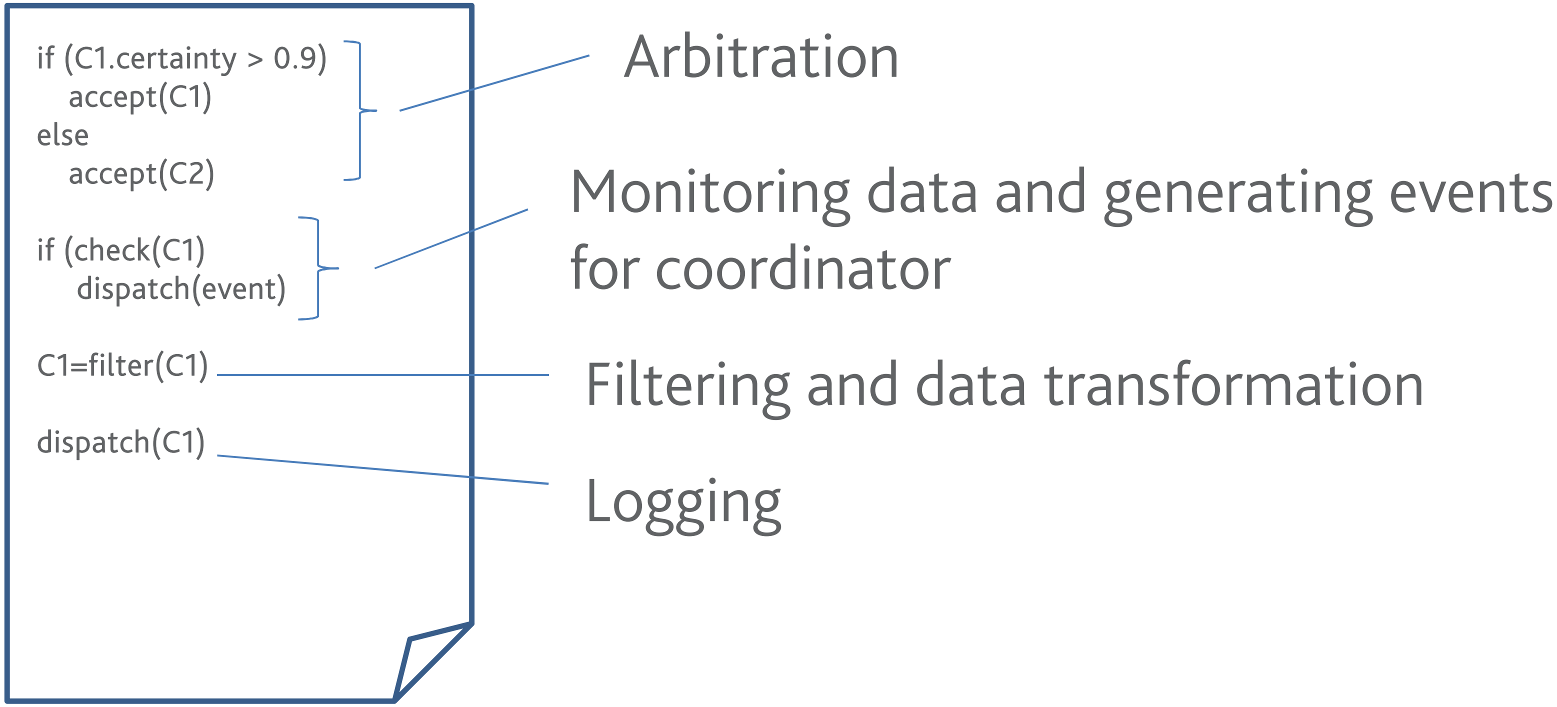
Arbitration

Monitoring data and generating events for coordinator

Filtering and data transformation

Online tutorials: www.yarp.it → Port monitoring and arbitration

Potential applications



```
if (C1.certainty > 0.9)
    accept(C1)
else
    accept(C2)

if (check(C1)
     dispatch(event)

C1=filter(C1)

dispatch(C1)
```

Arbitration

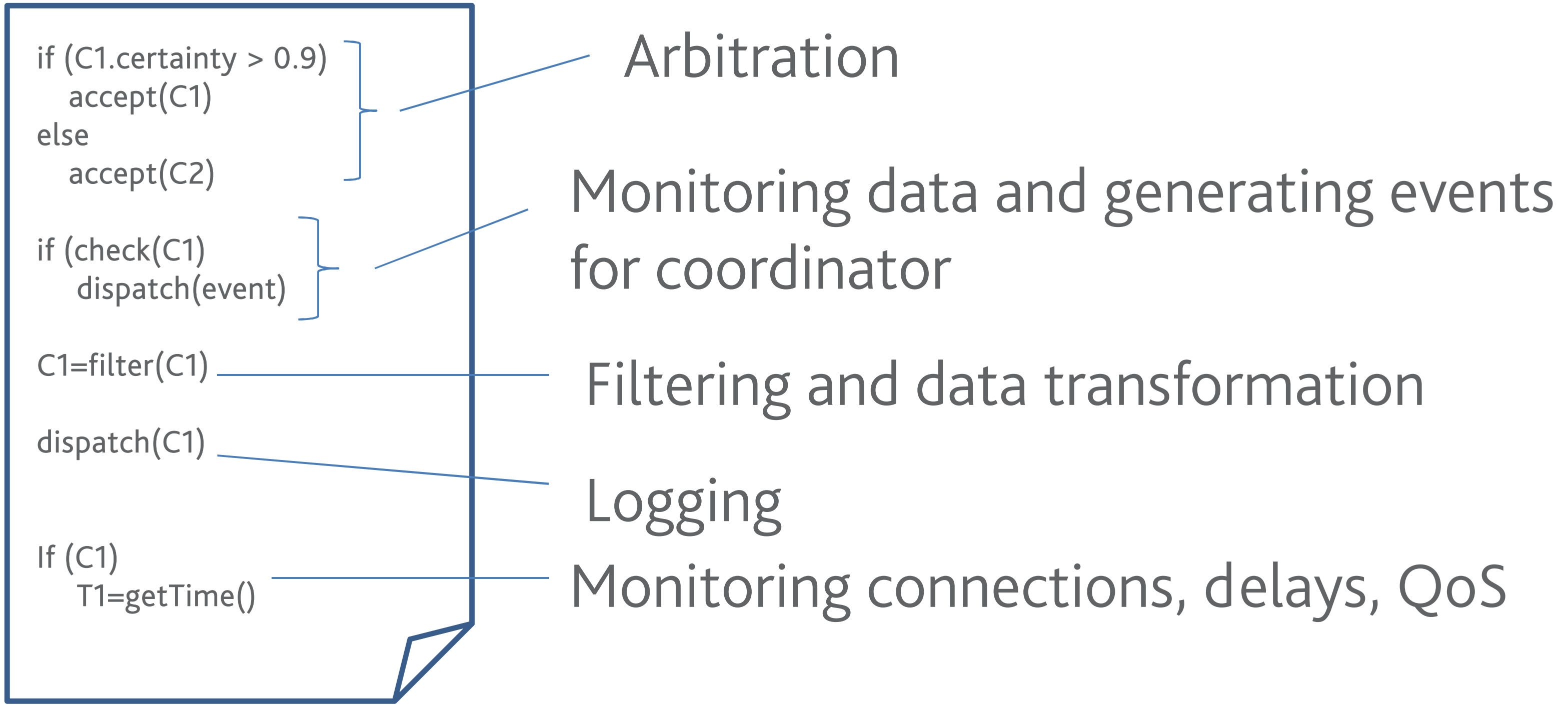
Monitoring data and generating events for coordinator

Filtering and data transformation

Logging

Online tutorials: www.yarp.it → Port monitoring and arbitration

Potential applications



```
if (C1.certainty > 0.9)
   accept(C1)
else
   accept(C2)

if (check(C1)
    dispatch(event)

C1=filter(C1)

dispatch(C1)

If (C1)
    T1=getTime()
```

Arbitration

Monitoring data and generating events for coordinator

Filtering and data transformation

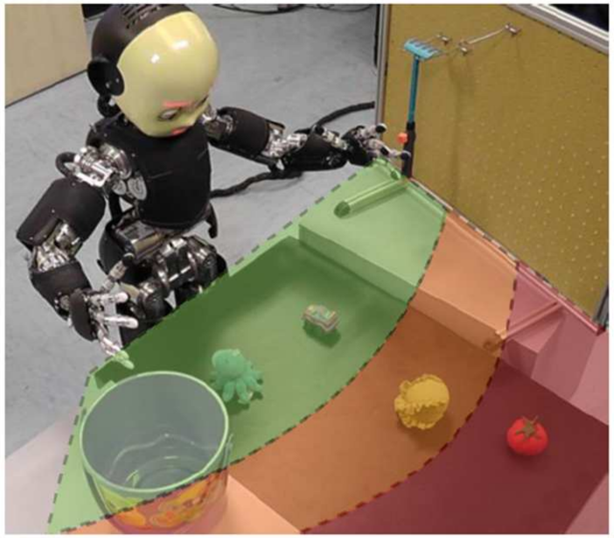
Logging

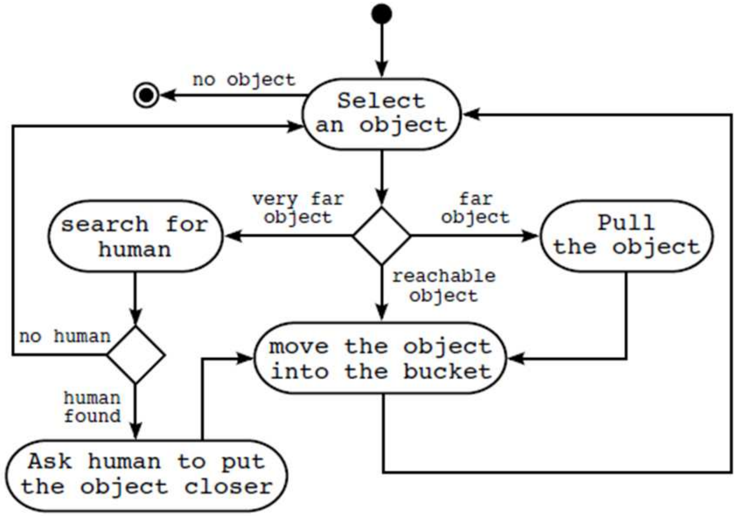
Monitoring connections, delays, QoS

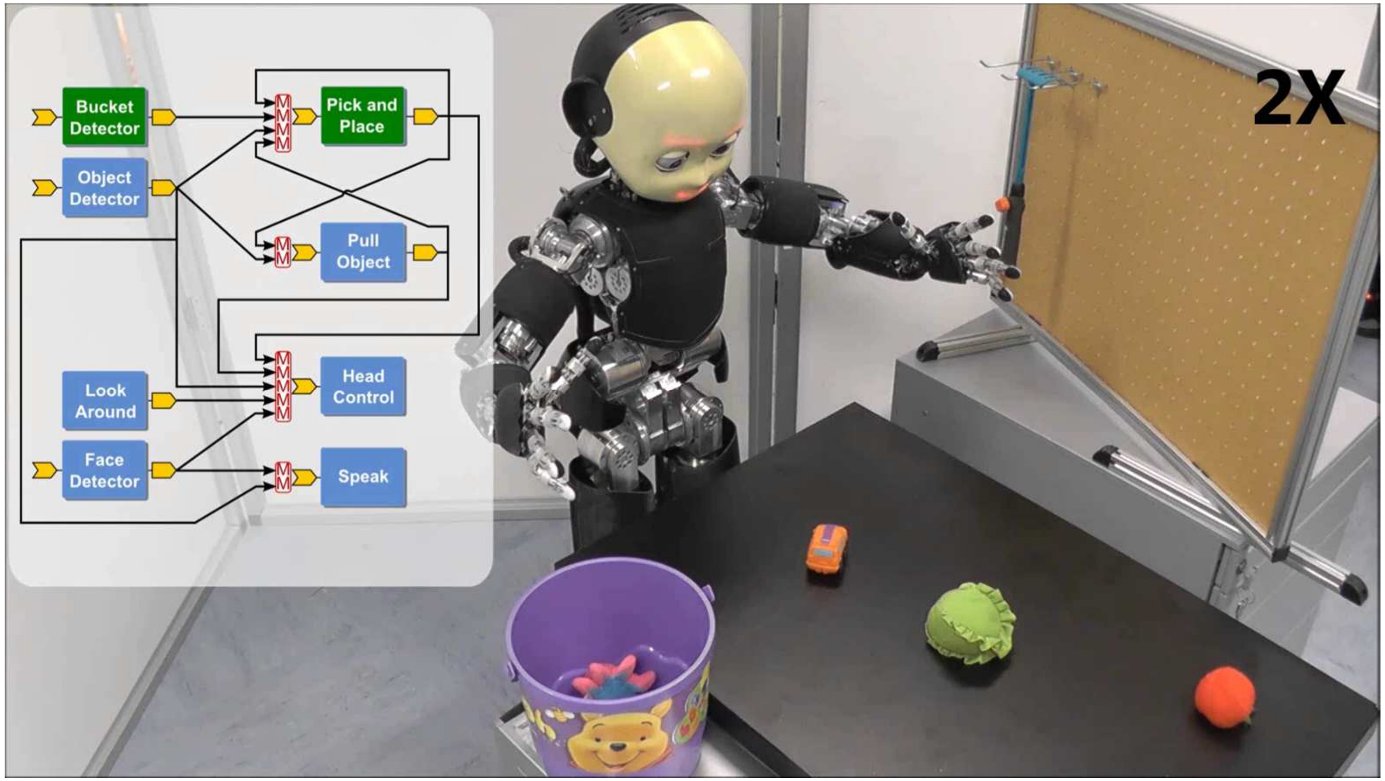
Online tutorials: www.yarp.it → Port monitoring and arbitration

Table-cleaning application

- Completely built using modules from the iCub repository
- No modifications to the existing modules
- Extending the required functionalities (e.g., for coordination) using port plug-ins







A. Paikan, V. Tikhanoff, G. Metta and L. Natale, IROS 2014

iit
ISTITUTO ITALIANO
DI TECNOLOGIA

Acknowledgements

RobotCub

poeticon++

CODYCO

XPERIENCE.ORG

WALK-MAN

KOROIBOT

Giorgio Metta
Ali Paikan
Daniele Domenichelli
Alberto Cardellino
Vadim Tikhanoff
Ugo Pattacini
Marco Randazzo
Paul Fitzpatrick