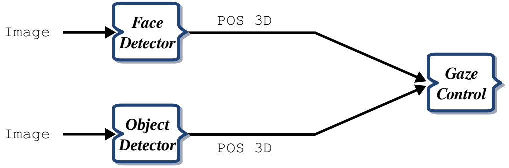
iCub Facility
Italian Institute of Technology

A Representation Of Robotic Behaviors Using Component Port Arbitration

Ali Paikan, Giorgio Metta and Lorenzo Natale

5th International Workshop on Domain-Specific Languages and models for ROBotic systems (DSLRob 2014)
(October 20, Bergamo, Italy)

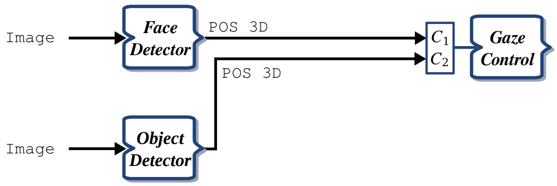
Simple Robot's head control applicaiton



- *'Face Detector'* and *'Object Detector'* can both send 3D position data to *'Gaze Control'* which controls a Robot's head to gaze accordingly.
- Since there is no synchronization among modules, data can be delivered to the input port of *'Gaze Control'* at any time, potentially causing conflicts.

A coordination mechanism should be employed to avoid conflict between these competitive connections!

Using port arbitraion

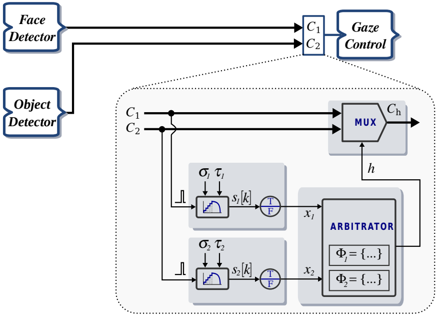


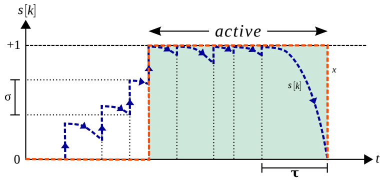
A port arbitrator extends the functionality of an input port to select data from multiple source based on the user–defined constraint.

Imagine we want the robot to track the face if there is no object in the scene:

"SELECT connection C1 IF C2 is not active."

Port Arbitration (inhibition)

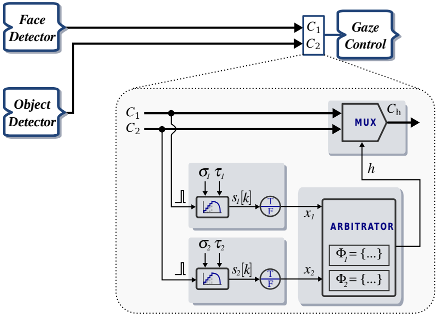


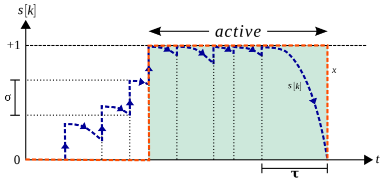


Track the face if there is no object in the scene:

- Φ_1 : C_1 *and not* C_2
- Φ_2 : C_2

Port Arbitration (excitation)





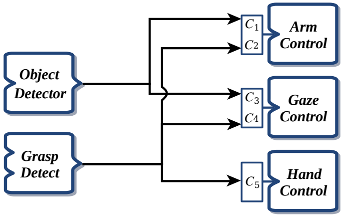
Track the object if there is *also* a person in the scene:

- Φ_1 : *false*
- Φ_2 : C_2 *and* C_1

More complex example (catching an object)

Track an object and grasp it:

- Try to reach for the object by hand and follow it by head
- Continuously check if the object is close enough for grasp
- When grasping the object, inhibit the movement of arm and head of the robot

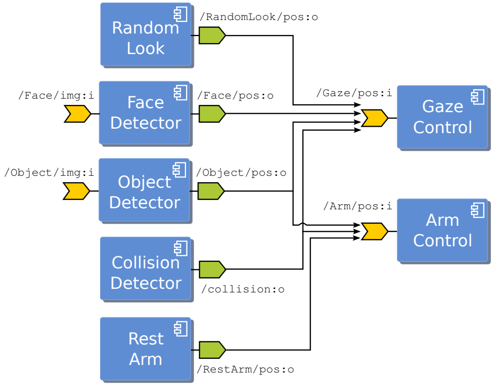


Φ_1 : C_1 *and not* C_2
Φ_2 : *false*

Φ_3 : C_3 *and not* C_4
Φ_4 : *false*

Φ_5 : C_5

Modeling Behaviors using Port Arbitration



- To Implement a behavior called **Follow Face**, the connection from `/Face/pos:o` to `Gaze/pos:i` should be selected by port arbitrator.
- To implement **Track Object** behavior, `/Object/pos:o` to `/Gaze/pos:i` and `/Object/pos:o` to `/Arm/pos:i` should be selected by port arbitrator.

Modeling Behaviors using Port Arbitration

- **Configuration** of a behavior is the list of connections which should be selected by the port arbitrators to implement the behavior.
- **Condition** is an optional property which specifies in, first-order logic, a constraint that should be verified for the behavior to be activated.
- **Inhibition**, specifies inhibitions between behaviors. Specifying inhibitions allows coordinating behaviors that are competing for the same resources.
- **Behaviors** can be grouped to describe a meta behavior.

Track Object

Condition: $\neg$ `/collision:o`

Configuration:

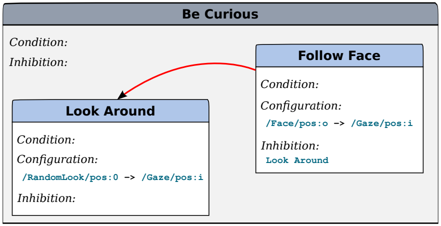
```
/Object/pos:o -> /Gaze/pos:i
/Object/pos:o -> /Arm/pos:i
```

Inhibition:

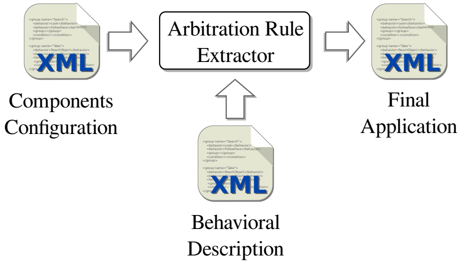
```
Rest Arm
Be Curious
```

Behaviors description in XML

```
<define name="gaze"> /Gaze/pos:i </define>

<meta_behavior name="Be Curious">
    <behavior>Look Around</behavior>
    <behavior>Follow Face</behavior>
    <condition></condition>
    <inhibition></inhibition>
</meta_behavior>

<behavior name="Look Around">
    <config at="$gaze">/RandomLook/pos:o</config>
    <condition></condition>
    <inhibition></inhibition>
</behavior>

<behavior name="Follow Face">
    <config at="$gaze">/Face/pos:o</config>
    <condition></condition>
    <inhibition>Look Around</inhibition>
</behavior>
```

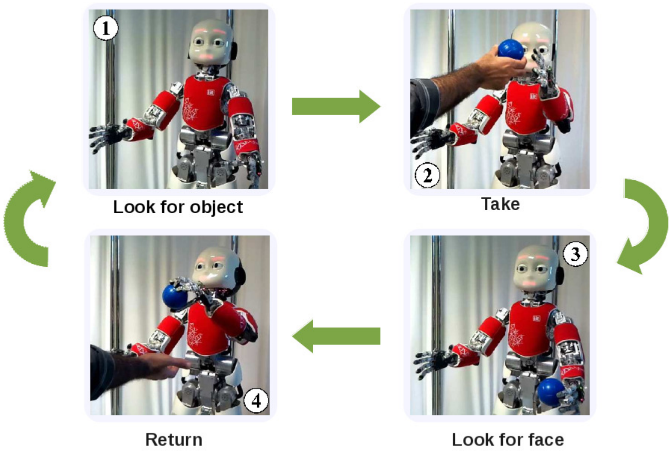


Arbitration rule extraction and application generation

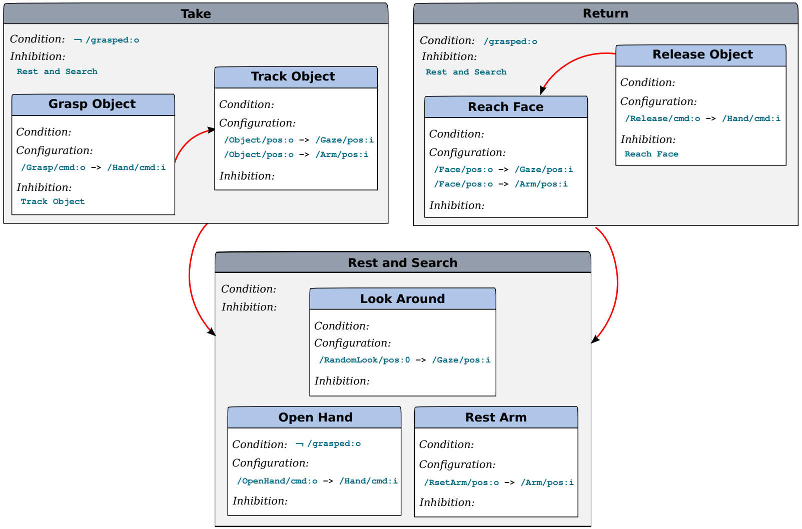


- Separating representation of the behaviors from the composition of the software components.
- Based on different behavioral descriptions, the same software components can be reused to implement different applications.

Catch and Return scenario



Modeling Catch and Return scenario


Take
Condition: ¬ /grasped:o
Inhibition:
Rest and Search
Grasp Object
Condition:
Configuration:
/Grasp/cmd:o -> /Hand/cmd:i
Inhibition:
Track Object
Track Object
Condition:
Configuration:
/Object/pos:o -> /Gaze/pos:i
/Object/pos:o -> /Arm/pos:i
Inhibition:
Return
Condition: /grasped:o
Inhibition:
Rest and Search
Release Object
Condition:
Configuration:
/Release/cmd:o -> /Hand/cmd:i
Inhibition:
Reach Face
Reach Face
Condition:
Configuration:
/Face/pos:o -> /Gaze/pos:i
/Face/pos:o -> /Arm/pos:i
Inhibition:
Rest and Search
Condition:
Inhibition:
Look Around
Condition:
Configuration:
/RandomLook/pos:0 -> /Gaze/pos:i
Inhibition:
Open Hand
Condition: ¬ /grasped:o
Configuration:
/OpenHand/cmd:o -> /Hand/cmd:i
Inhibition:
Rest Arm
Condition:
Configuration:
/RsetArm/pos:o -> /Arm/pos:i
Inhibition: