Software Architecture for an Exploration Robot based on Urbi

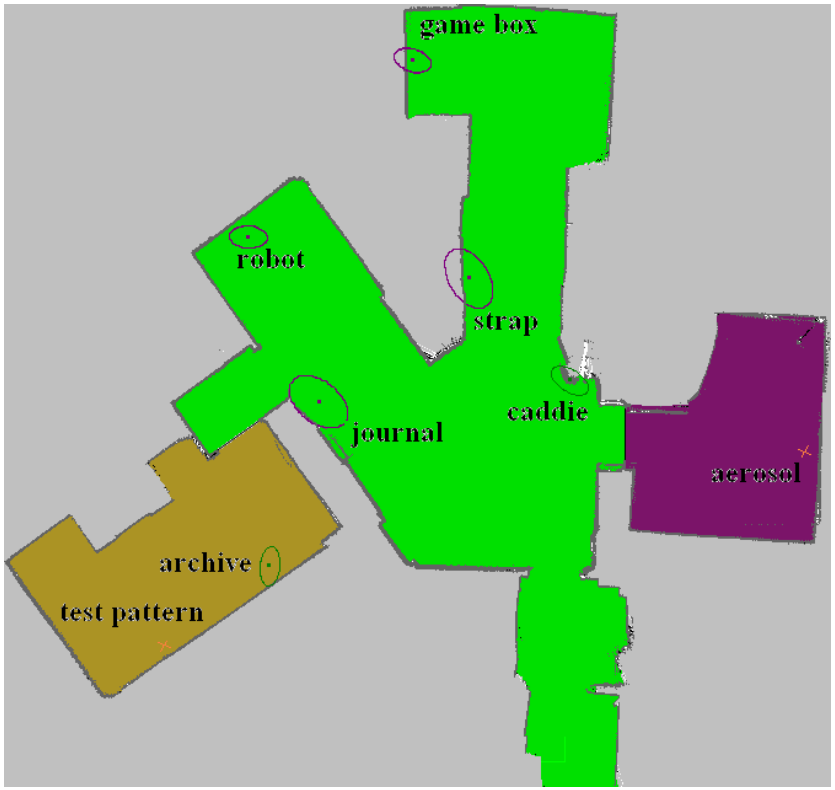
David Filliat Akim Demaille Jean-Christophe Baillie
Guillaume Duceux David Filliat Quentin Hocquet Matthieu Nottale

Ensta-ParisTech, Gostai S.A.S.

Control Architectures of Robots 2011
May 25th, 2011

Motivations: project PACOM

- Participate to the CAROTTE competition organized by the DGA and the ANR



Architecture for a Robot

- Handling a large set of software components
- Dealing with events, data-flow, and failure
- Based on the Urbi framework

1. System Overview
2. Urbi as a middleware
3. Practical Implementation
4. Conclusion and Future Work

System Overview

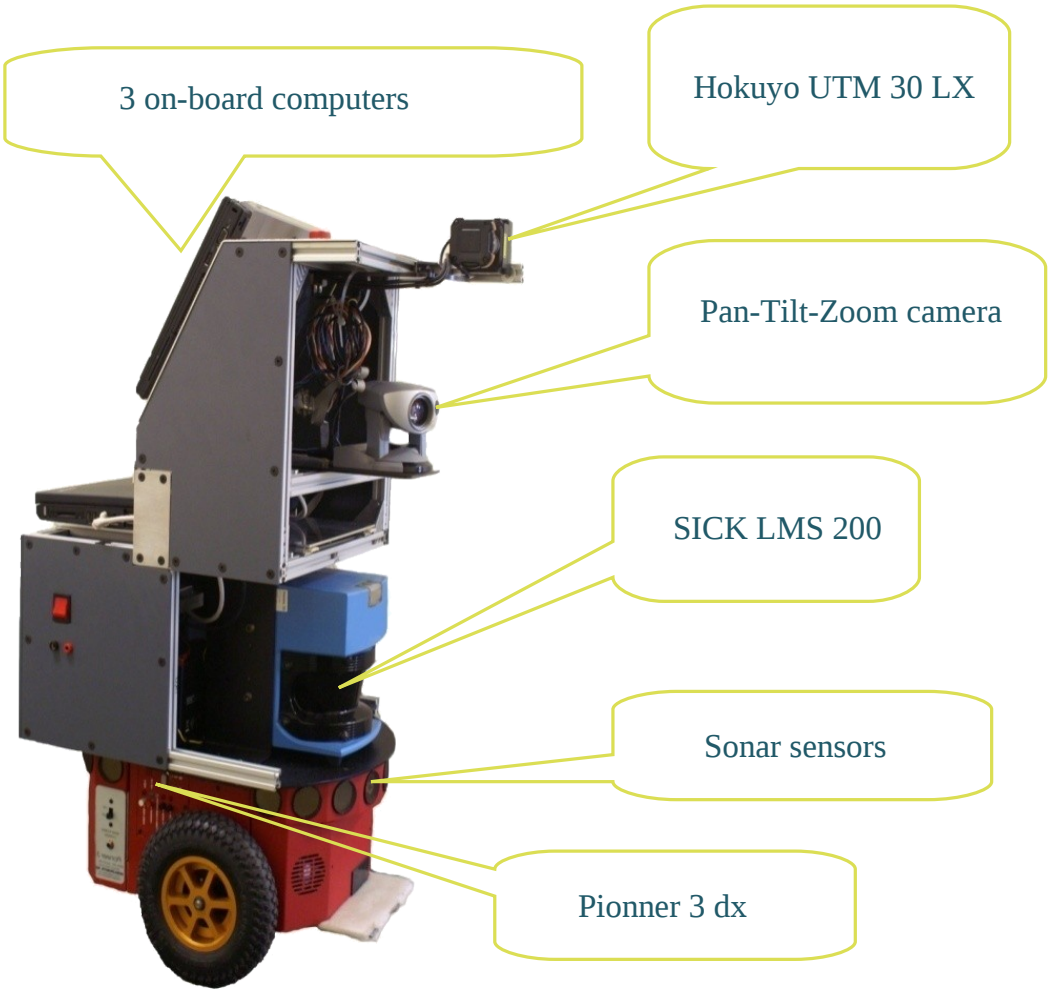
1 System Overview

2 Urbi as a middleware

3 Practical Implementation

4 Conclusion and Future Work

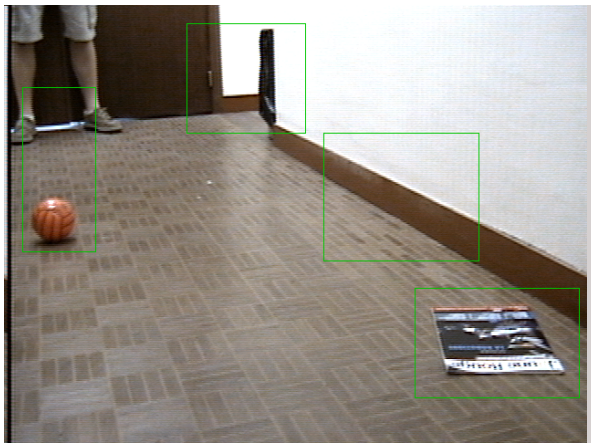
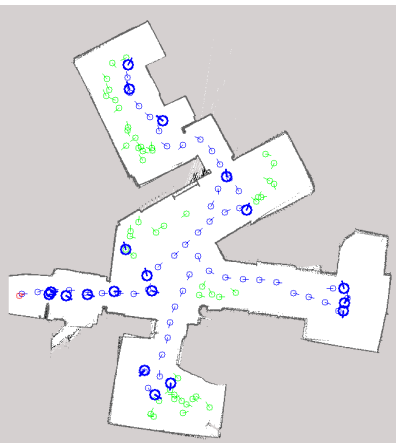
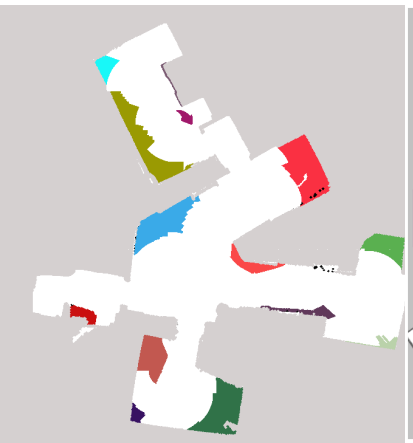
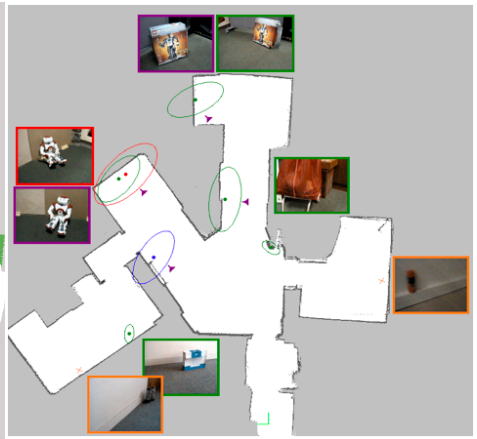
Hardware architecture



Software components

- SLAM
- Path planning
- Exploration
- Semantic mapping

- Robot
- Servoing
- Object recognition
- Windows detector

Handling different situations

- Normal ongoing mission
  - Find a target
  - Find a path to the target
  - Avoid Obstacles
  - Reach the target
  - While detecting Objects
  - At the end generate the semantic map
- Situations to handle
  - Target unreachable
  - Emergency stop
  - Robot blocked
  - Component failure

Handling different situations

- Normal ongoing mission
  - Find a target
  - Find a path to the target
  - Avoid Obstacles
  - Reach the target
  - While detecting Objects
  - At the end generate the semantic map
- Situations to handle
  - Target unreachable
  - Emergency stop
  - Robot blocked
  - Component failure

Urbi as a middleware

1 System Overview

2 Urbi as a middleware

3 Practical Implementation

4 Conclusion and Future Work

Software platform for robotic applications

- distributed component architecture
- urbiscript

- Flexible
- Facilitate integration and development

Software platform for robotic applications

- distributed component architecture
- urbiscript

- Flexible
- Facilitate integration and development

UObjects: drivers and software components

- UObjects
  - Written in C++ or Java
  - API
- Plugged UObjects
  - Shared-memory at the expense of responsiveness
  - Possibility to run in separate thread
- Remote UObjects
  - Used seamlessly
  - Parallelism but no shared-memory
  - General failure protection

UObjects: drivers and software components

- UObjects
  - Written in C++ or Java
  - API
- Plugged UObjects
  - Shared-memory at the expense of responsiveness
  - Possibility to run in separate thread
- Remote UObjects
  - Used seamlessly
  - Parallelism but no shared-memory
  - General failure protection

UObjects: drivers and software components

- UObjects
  - Written in C++ or Java
  - API
- Plugged UObjects
  - Shared-memory at the expense of responsiveness
  - Possibility to run in separate thread
- Remote UObjects
  - Used seamlessly
  - Parallelism but no shared-memory
  - General failure protection

UVar: data-flow control features

- Incoming/outgoing interface between components and Urbi
- Transparent serialization of integer, float, list, array, etc...
- Event support: notification of change or access

urbiscript

- Direct interface with the robot
- Scripting the behavior of the robot

Orchestrate and define robot behavior

```
whenever ( ball.visible )
{
    headYaw.val   += camera.xfov * ball.x
  & headPitch.val += camera.yfov * ball.y
},
```

Practical Implementation

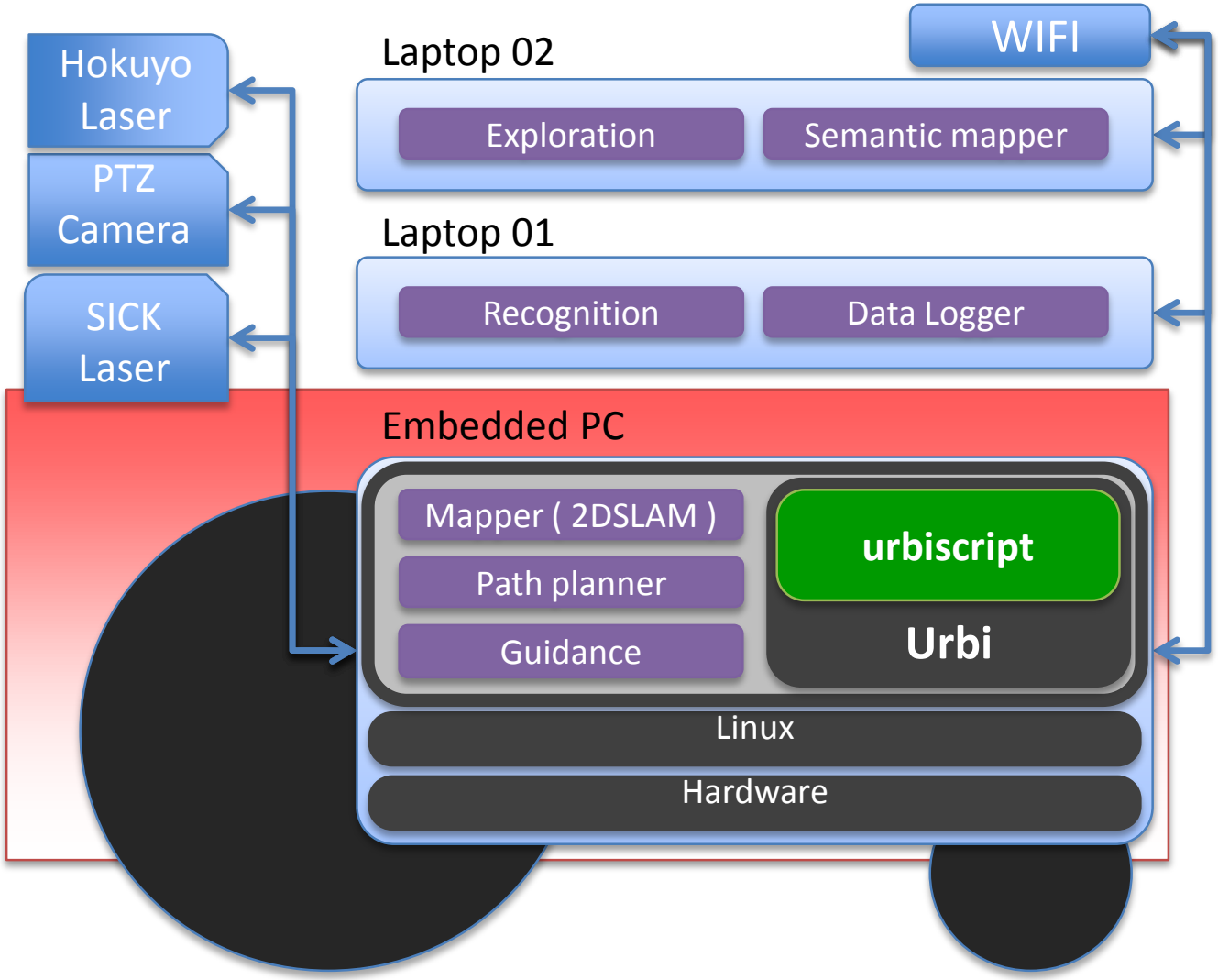
1 System Overview

2 Urbi as a middleware

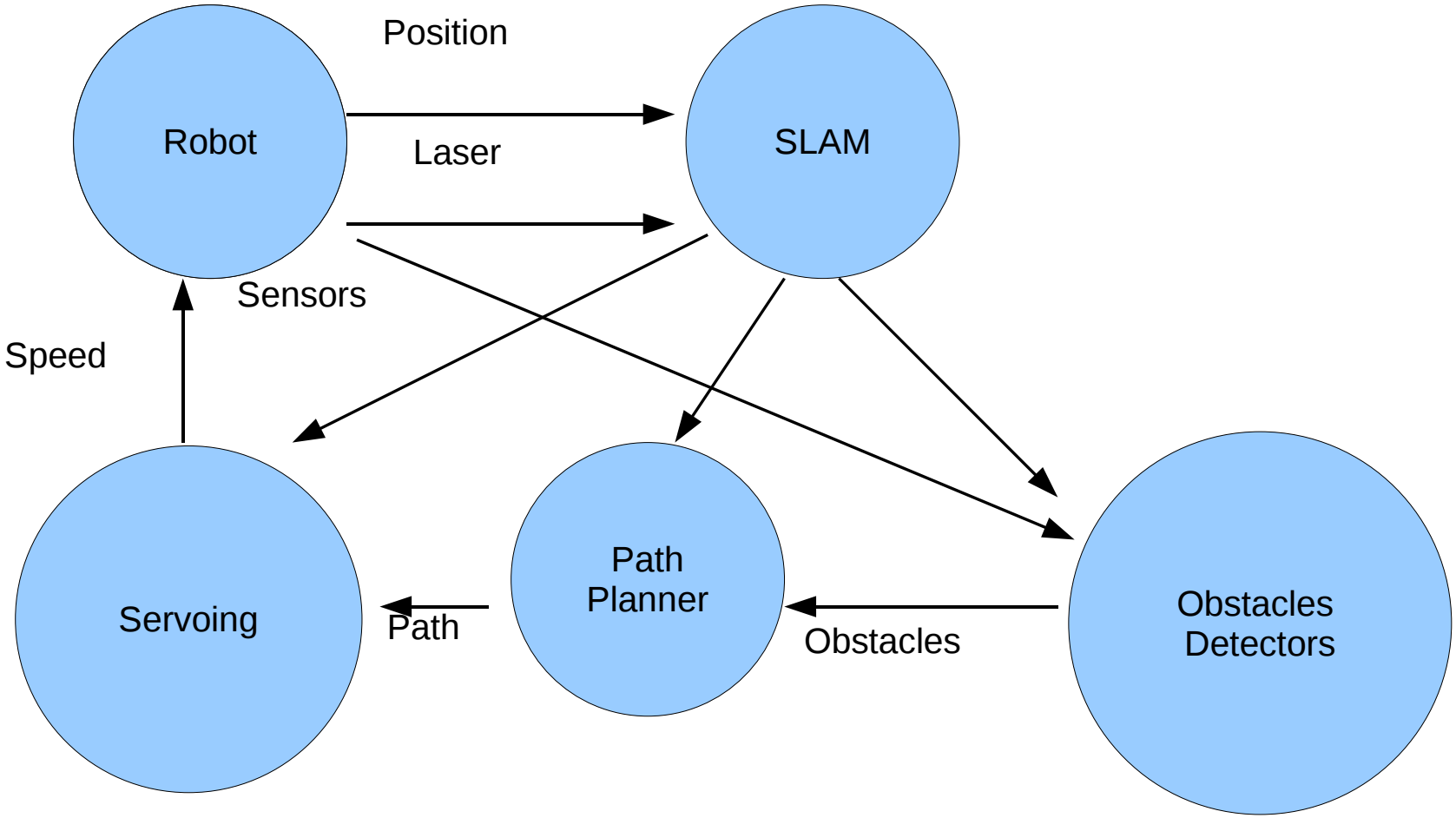
3 Practical Implementation

4 Conclusion and Future Work

Hardware architecture



Data-flow control



Data-flow control

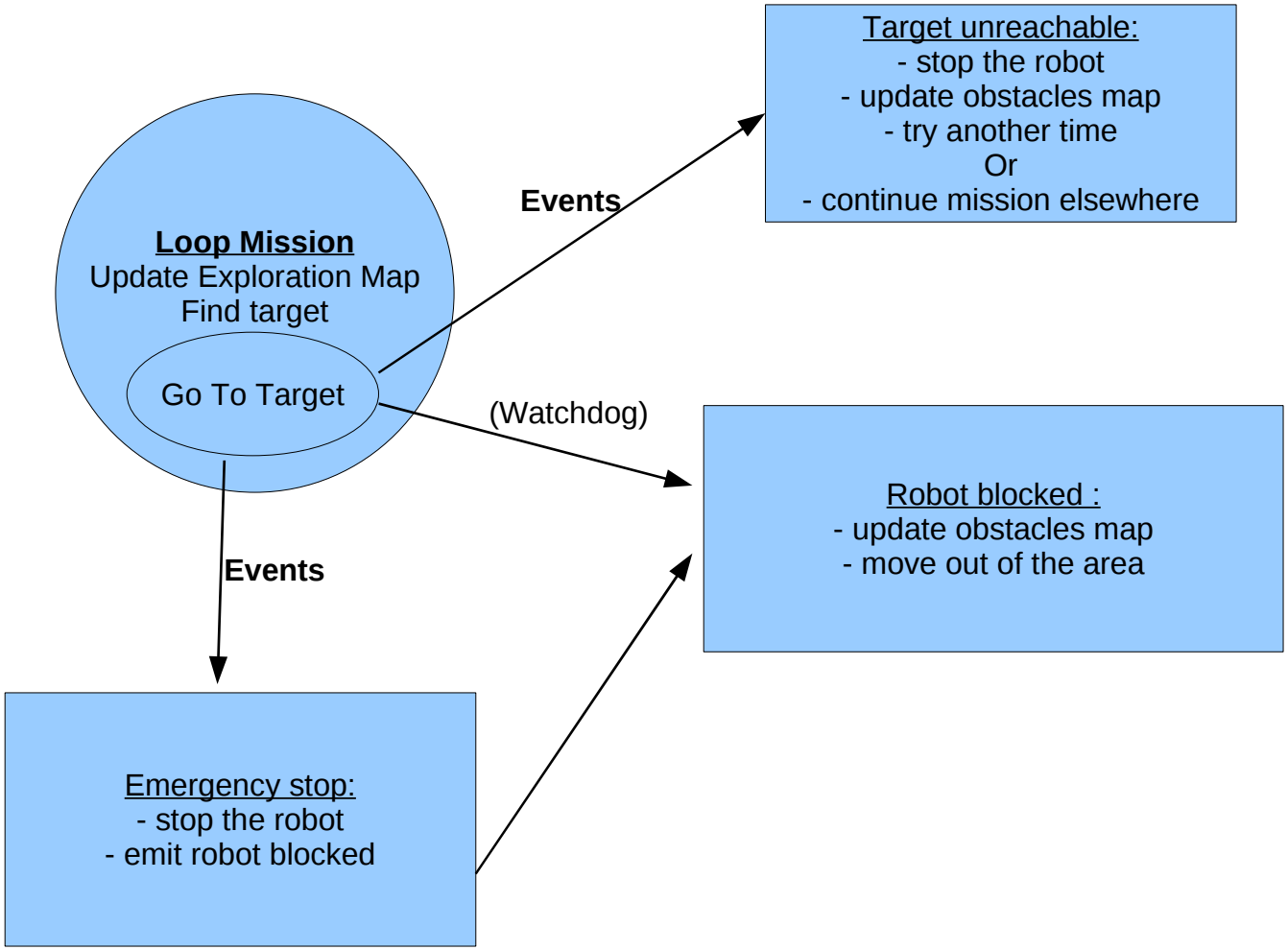
notification of UVar change

```
servoing.&speedDeltaCompute.notifyChange(
  closure()
  {
    robot.speedDelta = servoing.speedDeltaCompute
  }
);
```

InputPort

```
var connection = slam.&intputLaser << robot.&laserReadings
```

Event-driven mission



Urbi Events for handling situation

creation and caption of events

```
var Global.targetUnreachable = Event.new();
at (targetUnreachable?())
{
  echo("target unreachable!");
  driveAuto.freeze;
  explorer.findTargets();
  robot.goTo(explorer.targets.removeFront);
};
```

emitting events

```
targetUnreachable!();
```

Background processing

Handling delay for slow processing

```
var buffer = [];
var imageRecord = Tag.new();

imageRecord : every(100ms)
{
  buffer.insertFront([camera.val,robot.position]);
  // In case of congestion, keep only the most recent images.
  if (buffer.size > 15)
    buffer.removeBack();
},

whenever (!buffer.empty())
{
  objReco.process(buffer.removeBack());
},
```

Failure protection

in urbiscript: checking state of UObject

```
var objReco = nil;

every(1s)
{
  if (uobjects.hasSlot("ObjReco")
      && Global.objReco.protos[0] != uobjects.ObjReco)
    Global.objReco = uobjects.ObjReco.new();
},
```

(re)starting shell

```
#! /bin/sh
while true
do
  urbi-launch --remote ObjReco.so --host robot
done
```

Failure protection

in urbiscript: checking state of UObject

```
var objReco = nil;

every(1s)
{
  if (uobjects.hasSlot("ObjReco")
      && Global.objReco.protos[0] != uobjects.ObjReco)
    Global.objReco = uobjects.ObjReco.new();
},
```

(re)starting shell

```
#! /bin/sh
while true
do
  urbi-launch --remote ObjReco.so --host robot
done
```

Conclusion and Future Work

1 System Overview

2 Urbi as a middleware

3 Practical Implementation

4 Conclusion and Future Work

Conclusion and Future Work

Conclusion

- Control Architectures:
  - Handles data-flow between many modules
  - Deals with different situations
- Urbi
  - Flexible behavior
  - Facilitate integration and development
  - Facilitate cooperative work

Future Work

- Integrate new components
- Change the behavior accordingly
- Take advantage of new Urbi features

Conclusion and Future Work

Conclusion

- Control Architectures:
  - Handles data-flow between many modules
  - Deals with different situations
- Urbi
  - Flexible behavior
  - Facilitate integration and development
  - Facilitate cooperative work

Future Work

- Integrate new components
- Change the behavior accordingly
- Take advantage of new Urbi features

Thank you for your attention!

Questions?