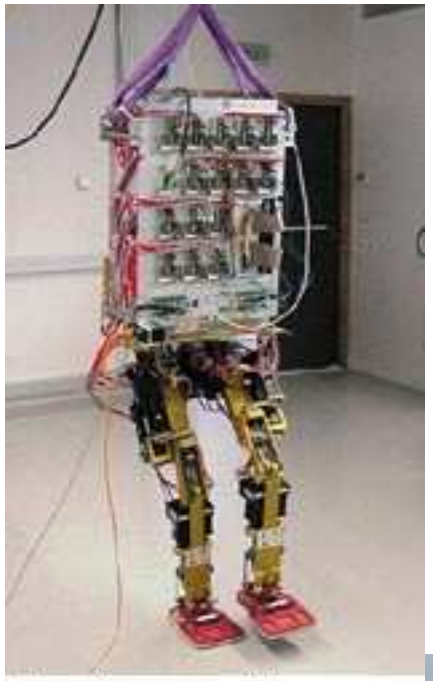
EVOLUTION OF THE ROBOTIC CONTROL FRAMEWORKS AT INRIA RHÔNE-ALPES

May 2011

S. Arias - A. Nègre - N. Turro

Early 2000's



The anthropomorphic robot BIP2000.

Focus on low level :

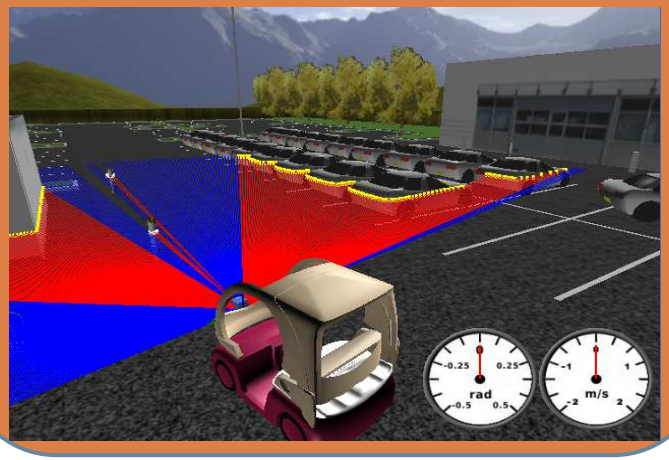
- Realtime OS and architecture
  Research on control loops
- Monolithic applications : Orccad
  Safety, Robust applications

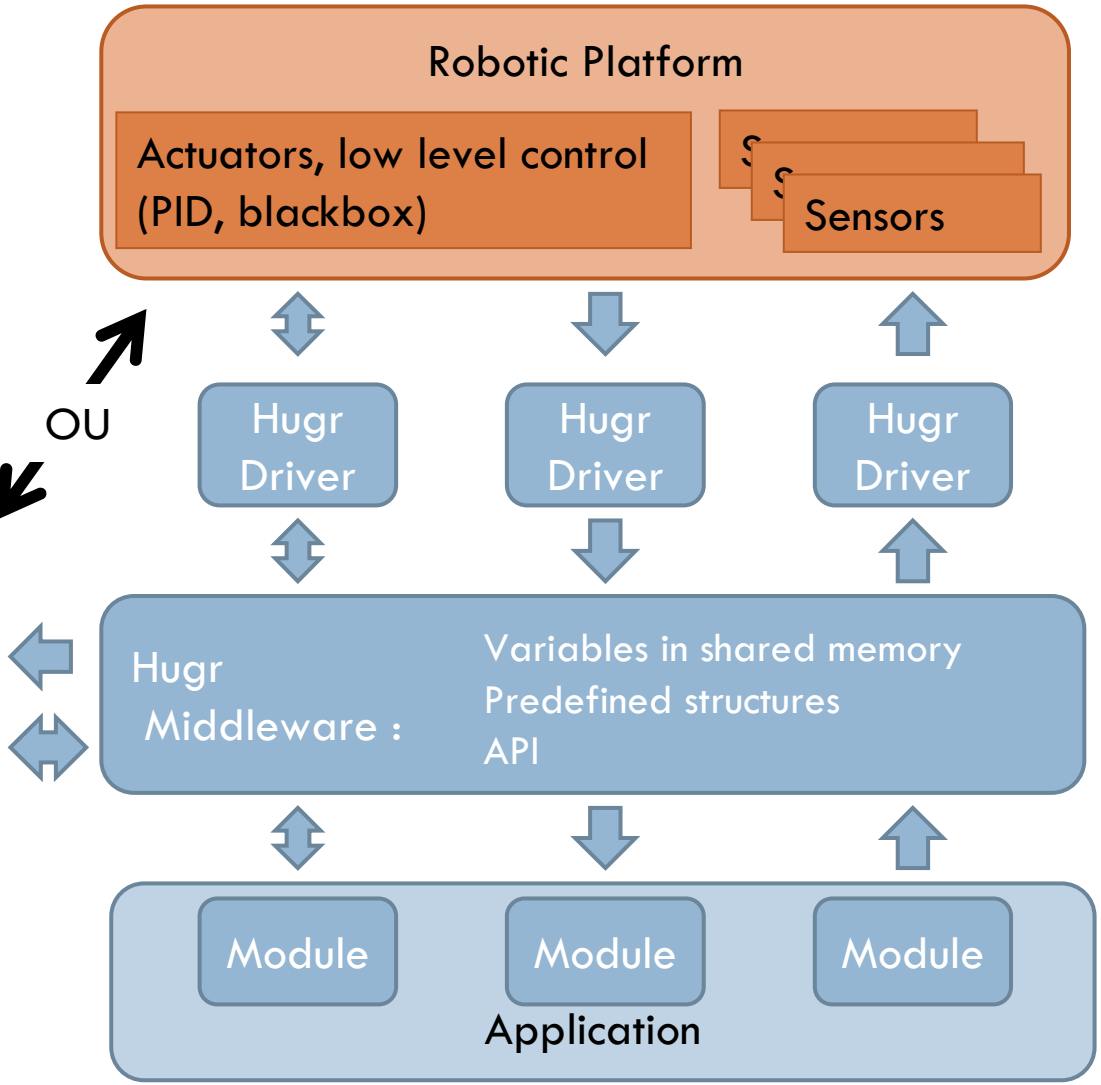
Drawbacks :

- Low computing power
- Few sensors
- Cumbersome experiments

2005 – 2010 : Hugr Middleware

OU





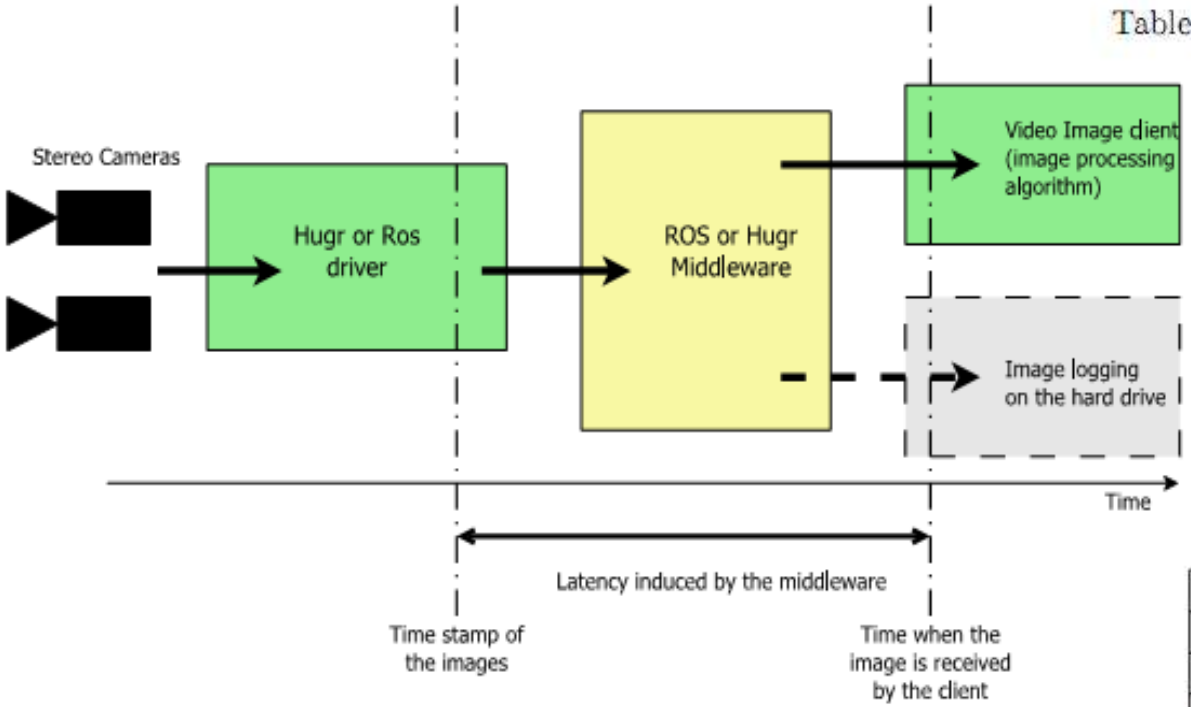
Hugr middleware benefits

- Experiment in your office !
  - Tools to log experimental data and replay them
  - Simulator (dynamic, sensors)
  - Same development environment (Ubuntu Linux)
- No restriction on the computing power
  - Standard PC, GPU/CUDA
  - Hugr can work across the network
- Very light and efficient
- But :
  - No man power for maintenance and further development
  - No visibility outside INRIA Rhône-Alpes, hinders collaboration

Realtime aspects

	Hugr	ROS
Mean	0.35 ms	0.8 ms
Max	2 ms	2 ms
Std Dev	0.03 ms	0.6 ms

Table 1: Latency without any external load



	Hugr	ROS
Mean	0.5 ms	0.8 ms
Max	8 ms	10 ms
Std Dev	0.19 ms	0.9 ms

Table 2: Latency on a stressed environment

2010-… ROS, the good

- Same principles as Hugr (Middleware, predefined data structures, API)
- Free, Open Source, BSD licence
- Widely used :
  - More standard data structures
  - Drivers/algorithms / tools repository
  - Teached in several universities
- Showcase for Inria algorithms, experimental data sharing, performance comparison

2010-… ROS, the not so good

- More complex internals
  - Performance
  - ‘Black box’
  - Custom modules and build system
    - Steeper learning curve
- Simulator ?
  - ROS interface being written