The 43rd
2012
ISR

# Piaget Environment for the Development and Intelligent Control of Mobile, Cooperative Agents and Industrial Robots

**Jean-Daniel Dessimoz, Pierre-François Gauthey and Hayato Omori**

**http://lara.populus.org/rub/3**

---

Hes·so
Haute Ecole Spécialisée de Suisse occidentale

heig-vd
Haute Ecole d'Ingénieurs et de Gestion du Canton de Vaud

ceTT

Institut d'Automatisation Industrielle
LaRA
Laboratoire de Robotique et Automatisation

# Piaget Environment for the Development and Intelligent Control of Mobile, Cooperative Agents and Industrial Robots

Jean-Daniel.Dessimoz[1], Pierre-François Gauthey[1] and Hayato Omori [2]

{ Jean-Daniel.Dessimoz or Pierre-Francois.Gauthey } @heig-vd.ch

[1]Hesso // Western Switzerland University of Applied Sciences
Heig-vd // School of Business and Engineering, CH-1400 Yverdon-les-Bains, Switzerland
[2]Department of Precision Engineering,
Faculty of Science and Engineering, Nakamura lab, Biomechatronics lab,
Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo,112-8551, Japan
h_omori@bio.mech.chuo-u.ac.jp

43rd International Symposium of Robotics,
Taipei, Taiwan, August 29-31, 2012.

http://www.heig-vd.ch, http://lara.heig-vd.ch

**Keywords**: Cognitics, Cooperative Agent, Piaget, Real-time Intelligent Control, Industrial Robot

# Content

**http://lara.populus.org/rub/3**

---

# 1. Introduction 1 of 3

- **How to make novel smart systems, that change the world and human life?**
- **Much has already been done by mankind, in terms of science and machines; in particular:**
  1. **Robots ;**
  2. **Information processing systems and techniques (computer technologies, microelectronics, communication devices and networks).**

**Current industrial robots can satisfactorily meet most classical requirements in terms of power and accurate motions in space (e.g. current reports in [1]).**

**But robots now face new challenges, in terms of cognitive capabilities.**

# 1. Introduction 2 of 3

- The industrial environment is getting more complex and tends to change more dynamically; real-time cooperation with humans and other resources is envisioned.
- Mankind now needs machines behaving with their own knowledge, artificial cognition, cognitics. cognitive agents acting in the real world (e.g. current theme of [1]).
- After years of activity in industrial robotics (Unimation-Stäubli, ABB, etc.), we had additionally started, back in 1998, to design an environment, denoted Piaget, for the most competitive development and control of autonomous robots, aiming at annually defined, novel tasks [2].

---

# 1. Introduction 3 of 3

- We have then kept developing Piaget achieving the type of smart solutions just mentionned
- For this same purpose, we have joined, for 5 years (the first 5 years of the "@Home league), the Robocup initiative [3]:
  - "integrate robotics and AI"
  - objective of long-term design of smart robots, capable of individual initiative, collective cooperation with other robots and humans, as well as fast locomotion and effective action in the real world.
  - Target performance levels comparable to best human soccer-players. Or best domestic helpers [4].
- Now time has come to extend our successful proposals to new-coming industrial challenges.

# Content

1. Introduction
2. Requirements and theoretical aspects of intelligent control
3. Piaget
4. Conclusion

**http://lara.populus.org/rub/3**

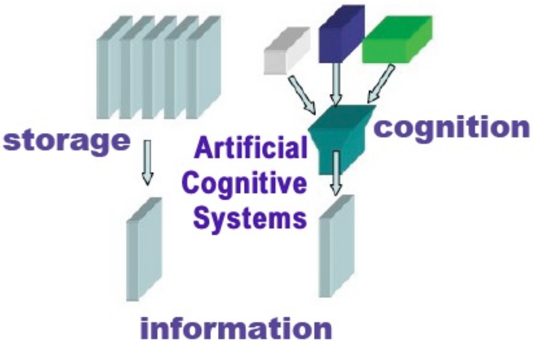
---

# 2. Requirements and theoretical aspects of intelligent control

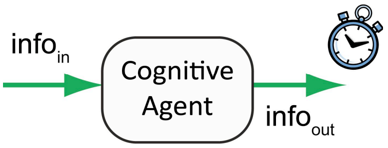
A. First stage in the exploration of cognition
B. Requirements for a new set: architecture and language
C. Metric estimation and benchmarks for best approaches
D. MCS theory for cognition

## 2.A. First stage in the exploration of cognition

### *A.1 What is Cognition? intelligence? and Intelligent Control?*



**Cognition and, effectively, cognitive systems, allow for generating relevant information, exactly similar to pre-stored information - when the latter is available**



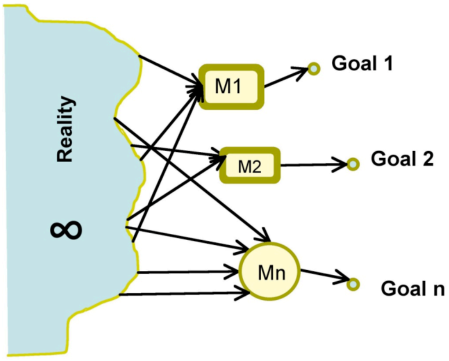
---

## 2.A. First stage in the exploration of cognition

### *A.2 What kind of applications is addressed?*

**Reality is very complex but selecting a goal typically allows for convenient, infinitely simpler models**

**We aim at designing smart cooperating robots**



## 2.A. First stage in the exploration of cognition 1 of 2

***A.3 What strategies are appropriate?*** **The goal just stated in previous paragraph calls for a very complex system, embedded in the real world, and in particular operational in real-time, capable to address the most advanced applications in terms of automation and cognitive, human-related tasks.**

**To be tractable, the proposed system must be organized as a hierarchy of coordinated, specialized resources, contexts, and points of view, each being individually much simpler.**

**Another strategy is, at all levels of the hierarchy, starting from the very top, to rely in as much as possible on existing elementary solutions – subsystems.**

---

## 2.A. First stage in the exploration of cognition 2 of 2

**Here where lots of integration must be done, the first priority in selecting potential components, strangely, is less on the top functional capabilities of these elements than on their safe availability and operational robustness.**

**Possible candidates in terms of possible components may be found, from case to case,**

- **on the market,**
- **in scientific and technological publications,**
- **or other sources yet, including , where necessary,**
    - **new proprietary developments**

# 2. Requirements and theoretical aspects of intelligent control

A. First stage in the exploration of cognition

B. Requirements for a new set: architecture and language

C. Metric estimation and benchmarks for best approaches

D. MCS theory for cognition

---

# 2. B. Requirements for a new set: architecture and language

On day one, back in 1998, like today still, the system we aimed at could not be found, ready made, on the market or in other labs. Nevertheless, more and more powerful components have been developed. At the hinge between these two realities, the first component to appear as necessary for our goal has been the design of a novel set, architecture and language, which we have called “Piaget” in reference to the famous psychologist of same name, recognized scientist of human cognition, who has made major contribution especially in the context of young children development.

The “Piaget” concept for architecture and language has evolved in two or three major stages, as described in detail in Section 3.

# 2. Requirements and theoretical aspects of intelligent control

A. First stage in the exploration of cognition

B. Requirements for a new set: architecture and language

C. Metric estimation and benchmarks for best approaches

D. MCS theory for cognition

---

# 2.C. Metric estimation and benchmarks for best approaches 1 of 3

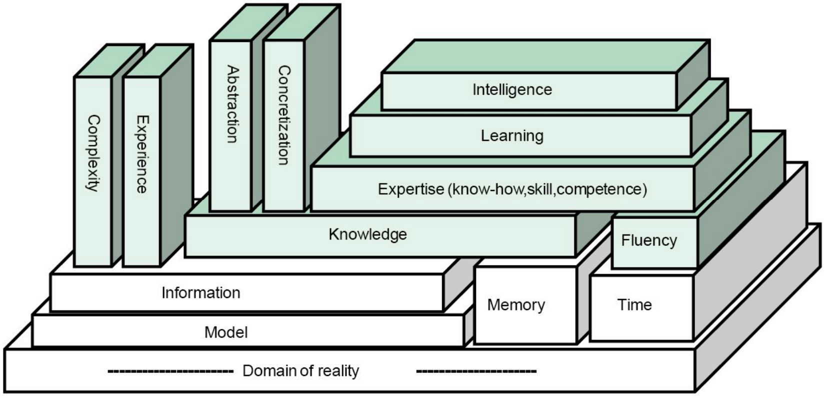
- Consider jumping over a wall: the metric height of the wall is a critical parameter for success. Similarly, the novel possibility of metric assertion of cognitive aspects (complexity, knowledge, expertise, etc.) is very useful. This is a natural merit of the proposed approach in MCS (re. Section II.D).
- Besides, conference attendance and state of the art monitoring bring useful new information. In addition, the methodology of realizing real-world systems allows for concretely implementing proposed theories.
- This can moreover lead to actual competitions on common test applications, thus encouraging active interaction with international experts.
- To address the two latter points, the strategy has notably been for us to join successively two international initiatives- Eurobot and Robocup@Home.

# 2.C. Metric estimation and benchmarks for best approaches 2 of 3

- **The same strategy has brought two apparently opposite benefits: in the early days of our mobile robots, industrial elements were advocated as reliable *components* for our systems: very naturally, industrial components have progressively been integrated in Piaget applications. And in recent time the situation has somehow inversed: now industrial robotic *applications* have become so complex, that they call for solutions of the type of our Piaget development, capable, beyond proprietary robot arm controller level, to effectively and efficiently develop/lead/steer/drive novel, complex applications; thus it now appears that globally Piaget is the "hinge"; both before and after this hinge, standard, industrial and/or commercial components prevail.**
- **In the industrial context, it is the open-market competition, along with the necessary legal framework, that essentially provides the most relevant benchmarks.**

---

# 2.C. Metric estimation and benchmarks for best approaches 3 of 3



Complexity
Experience
Abstraction
Concretization
Intelligence
Learning
Expertise (know-how,skill,competence)
Knowledge
Fluency
Information
Memory
Time
Model
--------------------- Domain of reality ---------------------

**Fig. 3. On the basis of classical concepts (in white;to be revisited though) the green elements are introduced in the formal "MCS" theory for cognition [5]**

**Expertise is a particularly important property, which would deserve a B-Prize.**

# 2. Requirements and theoretical aspects of intelligent control

**A. First stage in the exploration of cognition**

**B. Requirements for a new set: architecture and language**

**C. Metric estimation and benchmarks for best approaches**

**D. MCS theory for cognition**

---

# 2.D. MCS theory for cognition 1 of n

**Our developed, formal framework, “MCS”, allows for the quantitative assessment of cognitive tasks, both as required or as operated by humans and machines.**

Information: $n = \sum p_i \log_2(1/p_i)$ [bit]

Knowledge: $K = \log_2(n_{out} 2^{n_{in}} + 1)$ [lin]

Fluency: $F = 1/\Delta t$ [$s^{-1}$]

Expertise: $E = K \cdot F$ [lin/s]

Learning: $\Delta E = E(t_1) - E(t_0); > 0$ [lin/s]

Experience: $R = r(n_{in} + n_{out})$ [bit]

Intelligence: $I = \Delta E / \Delta R$ [lin/s/bit]

relative Agility: $A_r = \tau / T$

T: $\text{Fluency}^{-1}$ and communication delays

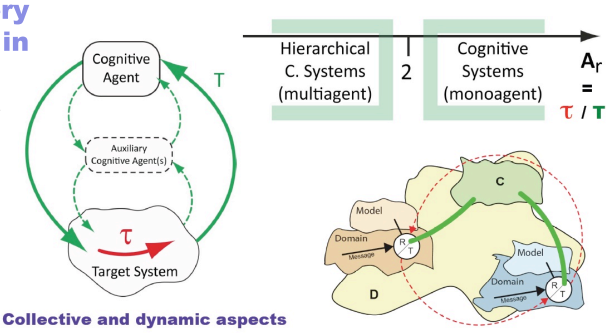
$\tau$: Reaction time of target system, to be controlled

r: number of withnessed experiments

**Fig. 4. Equations for assessing quantitatively the core properties in cognition; re [5]. Information is classical though (re. Shannon).**

# 2.D. MCS theory for cognition 2 of n

Scale and time are very important properties in cognitive systems. In particular. individuals can collectively yield groups; and in all control loops, occurring in single agents or multi-agent systems, strict dynamic constraints allow – or not – for stable results. Partial autonomy may be required.



- Collective and dynamic aspects

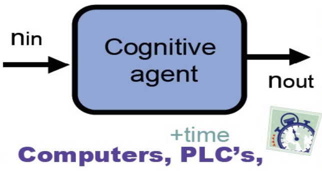
Fig. 5. Collective and dynamic aspects are important, especially when efficience and stability is considered [5].

---

# 2.D. MCS theory for cognition 3 of n

- Sense and perceive!
- Switches
- Microphones and receivers
- Color camera
- « 3D » camera
- Thermal camera
- Planar rangers
- US sensors
- Tactile surfaces
- Force and Torque sensors
- Position encoders
- Etc.

- Think,
  - Understand, decide, plan, design, etc.!



- Computers, PLC's, FPGA's, Specialized devices and circuits, distributed architecture

- Act
  - incl. move, communicate!
- Loudspeakers and transmitters
- Color display
- Stepper and DC motors
- Locomotion devices
- Grasping tools
- Manipulator arms
- Lights and numerous actuators
- Etc.

14

Fig. 6. Smart systems sense, perceive, think and act. The cognitive components of these processes typically relate to large amounts of information (>> 1 Mb), in high speed (up to $10^7$ [1/s] and more).

# Content

1. **Introduction**
2. **Requirements and theoretical aspects of intelligent control**
3. **Piaget**
4. **Conclusion**

**http://lara.populus.org/rub/3**

# 3. Piaget

**Sense and perceive!**

- **Think,**
  - **Understand, decide, model, plan, design, etc.!**
- **Act**
  - **incl. move, grasp, communicate!**

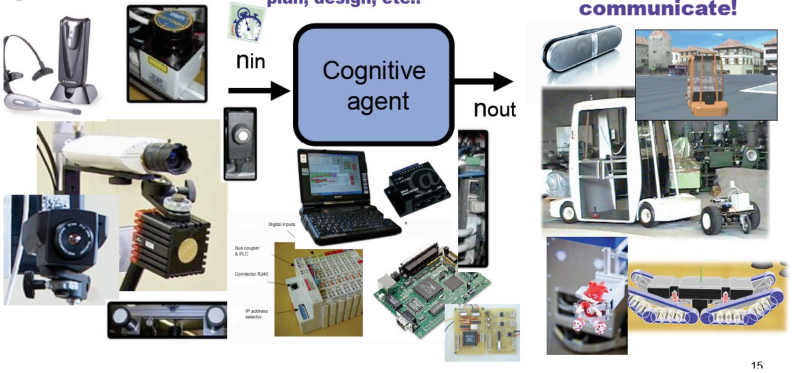


**Fig. 7. Smart cognitive systems sense, perceive, think and act. This relies on a variety of powerful components to flexibly integrate, (re-)configure and operate. Therefore, Piaget!**

# 3. Piaget

**A. Core aspects**
- A.1 Parallelism, real-time, and “open” resources.
- A.2 Piaget and VAL
- A.3 Hardware support

**B. Aspects of particular interest**
- B1. Simulation capabilities
- B.2. Interactive actions and language interpreter
- B.3. Four levels of programming techniques “plus”.
- B.4. Multiple degrees of inter-cooperation performance.
- B.5. Test instruction and Test task
- B.6. Examples of application – Piaget for Cognitics

---

# 3. Piaget

**A. Core aspects**
- A.1 Parallelism, real-time, and “open” resources.
- A.2 Piaget and VAL
- A.3 Hardware support

**B. Aspects of particular interest**
- B1. Simulation capabilities
- B.2. Interactive actions and language interpreter
- B.3. Four levels of programming techniques “plus”.
- B.4. Multiple degrees of inter-cooperation performance.
- B.5. Test instruction and Test task
- B.6. Examples of application – Piaget for Cognitics

# 3.A.1 Parallelism, real-time, and "open" resources 1 of 6

**Computers have been around for some time, as well as standard products in electronics, precision engineering and microtechnology.**
***The*** **first crucial component that appeared to be missing though, was an application-oriented environment, with parallelism and real-time capabilities, and very open possibilities for integration with numerous, heterogeneous, products and services.**
**Languages like Pascal, C, C++ or C# are not naturally prone to parallelism. Similarly common operating systems, such as DOS, RTDOS or Windows do not support a parallelism agile enough for our control requirements (e.g. sub millisecond switching time and I/O reactivity for coordination and low-level, micromotor control). Among approaches to attempt solving this problem we may now find Webots [7], ROS, Microsoft Robotics (Developer) Studio[8] (started in 2006), or better in terms of real-world capabilities, many proprietary solutions developed by SME's with ad hoc, application oriented automation constraints.**
**Therefore we created Piaget.**

---

# 3.A.1 Parallelism, real-time, and "open" resources 2 of 6

**Classically, a multitask kernel relies on lots of stack operations for a maximal use of fast registers in each task being restarted. We attempted instead to revisit an old Texas Instrument solution, whereby task switching is efficiently done simply by switching contexts in ordinary memory, betting on progress in cache-memory and improved compilation capabilities.**
**In Piaget instructions are numbered (re. Fig. 8). A metalevel program counter is defined for each task and is typically realized in the implementation, lower level language as a switch paradigm. A possible "AGN" suffix explicitely indicates, when present, that, for the next allocated time slot, the program proceeds at the next numbered Piaget instruction.**

# 3.A.1 Parallelism, real-time, and "open" resources 3 of 6

**A possible "AGN" suffix explicitely indicates, when present, that, for the next allocated time slot, the program proceeds at the next numbered Piaget instruction.**

**Fig. 8. Example of instructions in Piaget language**

---

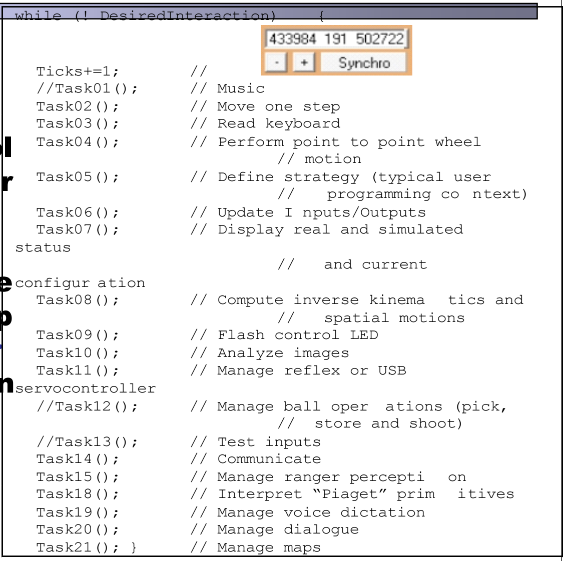
# 3.A.1 Parallelism, real-time, and "open" resources 4 of 6

**Our applications make typically use of about 20 parallel agents, and experience shows that common, current computers can in average visit (enter, do the work, and step out) each task in a single 100 nanosecond long time chunks.**

**Thus in general, internal Piaget task switching is performed many thousand times more often than is possible with the standard threading tools of native operating systems**

# 3.A.1 Parallelism, real-time, and "open" resources 5 of 6

**Fig. 9. For most agile agent switching, control is kept in kernel loop for thousands of iterations before returning to operating system. In the illustrated case the loop was visited 502'272 per second, as estimated on a 191 second basis**



433984 191 502722

```
while (! DesiredInteraction)    {

   Ticks+=1;        //
   //Task01();      // Music
   Task02();        // Move one step
   Task03();        // Read keyboard
   Task04();        // Perform point to point wheel
                           // motion
   Task05();        // Define strategy (typical user
                           //   programming co ntext)
   Task06();        // Update I nputs/Outputs
   Task07();        // Display real and simulated
status
                           //   and current
configur ation
   Task08();        // Compute inverse kinema  tics and
                           //   spatial motions
   Task09();        // Flash control LED
   Task10();        // Analyze images
   Task11();        // Manage reflex or USB
servocontroller
   //Task12();      // Manage ball oper  ations (pick,
                           //  store and shoot)
   //Task13();      // Test inputs
   Task14();        // Communicate
   Task15();        // Manage ranger percepti  on
   Task18();        // Interpret "Piaget" prim  itives
   Task19();        // Manage voice dictation
   Task20();        // Manage dialogue
   Task21(); }      // Manage maps
```

---

# 3.A.1 Parallelism, real-time, and "open" resources 6 of 6

**In our terminology, the task code is knowledge and, when running on the computer, the latter becomes per se the corresponding agent.**
**As shown in Fig. 9, "Ticks" are incremented at each kernel loop. On a regular laptop, such a loop can be visited about 500'000 times per second, including many returns to OS for ordinary system operations (communication , mouse, etc.; re. "DesiredInteraction"). This means that in average one agent exclusively takes only 100 nanosecond per turn, which is excellent for our goals. A "TicksPerSecond" parameter plays a key role for fast event timing in Piaget; it can be adjusted manually or automatically synchronized on the basis of experience.**

## 3.A.2 Piaget and VAL 1 of 3

**The Piaget language includes**

- **in principle** very specific, application-oriented instructions**, for example the "ChooseTheBridgeVisually" instruction.**

**- also useful, a** subset of the excellent VAL language for robotics**.**

**Therefore two main advantages:**

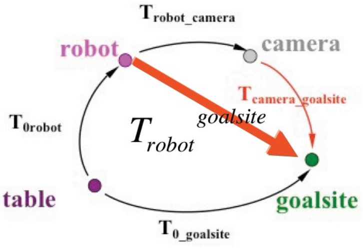
1. **VAL keeps a relatively general view at robotic and automation level (e.g. "Signal i" instruction to turn on the digital output number i) , useful for the early phase of a new application.**
2. **this paves the way to a common standard for novel, mobile agents and classical, industrial robots. Val [9] can be traced back to the beginning of industrial robotics, and keeps evolving [10].**

- **Piaget supports direct and inverse kinematics as well as extensive support for transformation and frame ancillary computations, in matrix form and homogeneous coordinates (re. Fig. 10).**

---

## 3.A.2 Piaget and VAL 2 of 3

The



$$T_{robot}^{goalsite} = T_{robot}^{camera} \cdot T_{camera}^{goalsite}$$

**Fig. 10. In internal stages, robot applications require extensive location, frame and trajectory computations, which are practically impossible to solve without state-of-the art knowledge. In particular, Piaget supports transformation graphs reasoning and homogeneous matrices computations**

# 3.A.2 Piaget and VAL 3 of 3

**Motion control is typically hierarchized in three levels:**

**programming,**
**coordination and**
**joint control,**
**with elementary cycle speed respectively situated at about 500, 15, and 0.5 milliseconds.**

**The Piaget "CallAGN(number)" is particularly important. While in Val, the "Call" instruction launches another program , and continues after completion, in Piaget the instruction has additional, crucial properties for parallel systems: it moreover allows for switching through all other agents at each instruction, and stepping for debugging purpose in the single, strategy agent of major interest.**

---

# 3.A.3. Hardware support 1 of 4

**Our Piaget environment has been operational in three main configurations.**

1. **Initially, a PC-base with parallel port capabilities was used, under DOS or Windows OS, for its large basis of compatible products and services, protocols and drivers.**
2. **Then a Piaget-light version has been implemented on a tiny integrated PC (Beck IPC) with an additional, proprietary FPGA, for encoder management and PWM motor control, under constraints of small volume availability [11].**

# 3.A.3. Hardware support 2 of 4

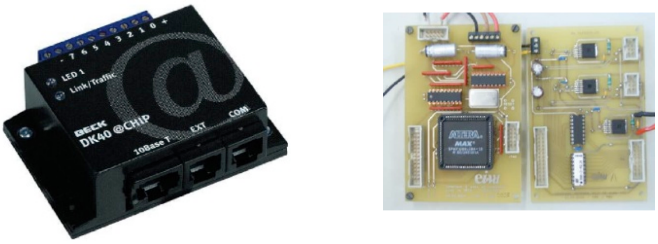


**Fig. 11. Integrated PC and FPGA for Piaget-light implementation in small volume robots [11].**

---

# 3.A.3. Hardware support 3 of 4

3. **Now Piaget is typically running on an heterogeneous system including powerful components in principle interconnected with Ethernet and TCP-IP capabilities; due to lack of availability in this standard, quite a few resources are similarly connected in a complementary, USB mode.**
   - **At supervisory level, a PC in Windows context is the rule, still for reasons of compatibility with complementary existing resources.**
   - **Closer to physical action, specialized components such as**
     - **motioncontrollers, PLC, cameras, rangers provide their own information processing resources, with power and robustness, in their own environment (re. Fig. 12)**

## 3.A.3. Hardware support 4 of 4

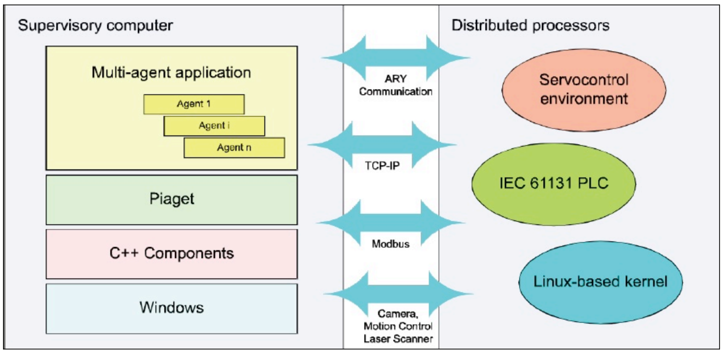


**Fig. 12. The high cognitive and action requirements of our complex applications in the real world call for a great sophistication of structures, and a contingent heterogeneity of resources, communcation channels, and protocols**

---

# 3. Piaget

**A. Core aspects**
- **A.1 Parallelism, real-time, and “open” resources.**
- **A.2 Piaget and VAL**
- **A.3 Hardware support**

**B. Aspects of particular interest**
- **B1. Simulation capabilities**
- **B.2. Interactive actions and language interpreter**
- **B.3. Four levels of programming techniques “plus”.**
- **B.4. Multiple degrees of inter-cooperation performance.**
- **B.5. Test instruction and Test task**
- **B.6. Examples of application – Piaget for Cognitics**

## 3.B.1. Simulation capabilities in Piaget
## 1 of 4

- Extensive simulation capabilities, globally, or by segments (Fig. 13 and 14), things are simpler (e.g. no energy management on our robots; no programmer motions towards physical switches, no robot displacements and orientations in the lab for “resetting” situations), easily replicated, and more robust, which is precious in some development phases.
- Nevertheless, the same environment can, when the corresponding physical resources are available, be turned operational in the real world. This is even the ultimate must.
- Some people advocate in-situ automata, but this cannot be done when past and future are considered, not to talk about if-worlds and ubiquitous presence and accounting of non-physical dimensions.

---

## 3.B.1. Simulation capabilities in Piaget
## 2 of 4

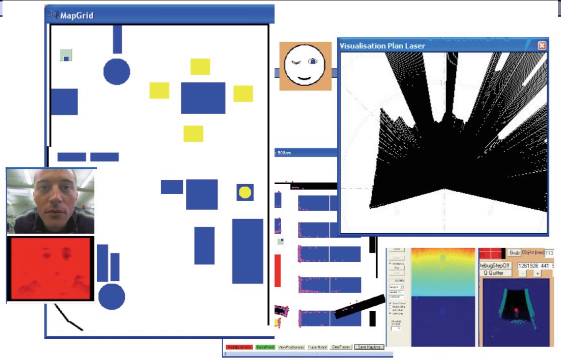


**Fig. 13. Piaget includes numerous possibilities in simulation mode. For example 5 pictures can routinely be selected when the camera is not online, noisy spirals are generated for virtual rangers, maps support motion analysis and perception in virtual world**

# 3.B.1. Simulation capabilities in Piaget
## 3 of 4

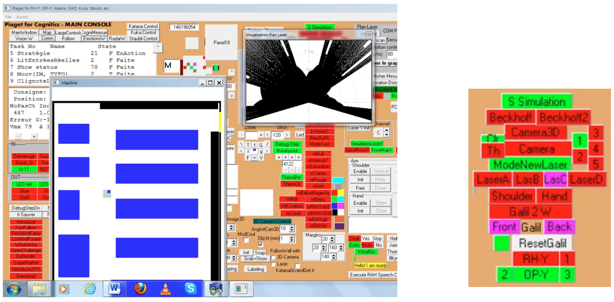


**Fig. 14. In this example, no physical resource is currently connected to the supervisory computer (right window); ranger data (re. upper window in center) are computed from virtual robot motions and objects represented in the map (lower left window).**

# 3.B.1. Simulation capabilities in Piaget
## 4 of 4

**Therefore Piaget environment brings uncomparable advantages, not only in terms of modelling as an *alternative* to direct access to reality but also for complementing techniques, thereby fostering a kind of *augmented reality*.**
**For Piaget, simulation possibilities keep being developed with a focus on results, i.e. when and only when it is expected to bring more effective and efficient results in target application domain.**

# 3.B.2. Interactive actions and language interpreter in Piaget 1 of 3

**Our Piaget environment has extensive interactive control capabilities Nevertheless, the same environment can also, often by hitting a single key or clicking the mouse, be turned operational, autonomous and possibly cooperative in the real world.**

---

# 3.B.2. Interactive actions and language interpreter in Piaget 2 of 3

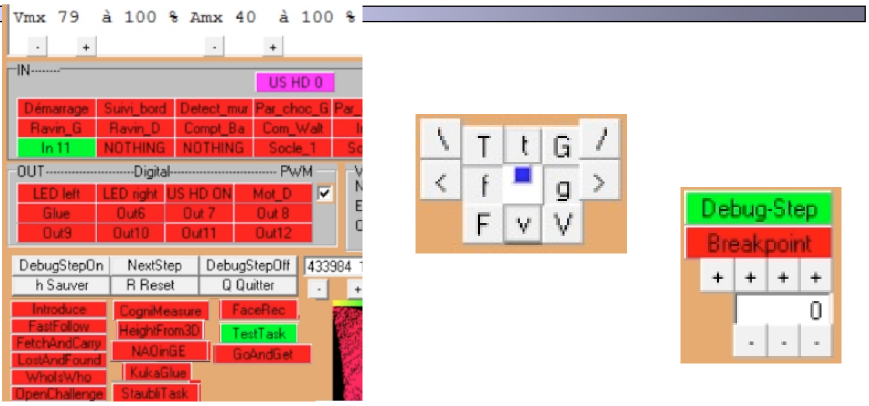


**Fig. 15. Many actions can be ordered by hitting a single key (e.g. “h”) or clicking on a button or panel. The program can be debugged with Piaget steps and breakpoints on Piaget instructions.**

## 3.B.2. Interactive actions and language interpreter in Piaget 3 of 3

In Fig. 15 for example, single letters typed on keyboard (“R”, “h”, “t”, “T”, “+”, “space bar“) allow for tens of immediate actions; similarly, clicking on buttons and panels allow for hundreds of actions. The “h” control has the further advantage of storing the current configuration for later use, in future sessions. These commands are interpreted in real-time, both in the immediate, interactive mode, and also when those controls are referenced in the “execute” phase of pre-compiled programmes

---

## 3.B.3. Four levels of programming techniques in Piaget. And more. 1 of 2

Programming modes in 4 levels of increasing complexities.

1. The first technique (level “0”) is ***interactive*** , as described in previous section, B2, and brings multiple advantages. The first use is immediately operational, the second use is an exploration capability, for training or development purpose, and the third use is programming. Although there is no script mode at this stage, the “h” or “save” command allows to “freeze” the configuration, i.e. to ***store numerous interactively modified parameters*** affecting values and modes for later, possibly autonomous, execution sessions.
2. The second technique (level “1”) allows typical users to express novel strategies, in relatively classical way, namely to program in the very high level, application-oriented Piaget language (re. Fig. 8, classically for us, “Task05”). At this level, it is also optionally possible for users to integrate commands of the implementation language (i.e. from case to case, C#, C++, C or Pascal). In particular, all controls interactively practiced at level 0 can be reused as instructions in level 1.

# 3.B.3. Four levels of programming techniques. And more. 2 of 2

**3.** **The third technique (level “2”) allows users to add or remove their own parallel agents. This requires somewhat more expertise from users, even though each parallel agent is in principle individually written like in level 1.**

**4.** **The 4th programming technique (level “3”) is reserved for experts who develop and implement Piaget language and environment, e.g. adding new instructions, drivers or controls for new resources. In particular, in this level novel contributions made by users in level 1 mode can often be optimized, better integrated in Piaget and thus better made available to all potential users, including at level 0 and 1.**

---

# 3.B.4. Multiple degrees of inter-cooperation performance in Piaget 1 of 3

**Real world systems require in particular real-time coordination of multiple processes and resources. Piaget schematically allows for five degrees of synchronicity, namely in multitask kernel; at system level; via shared local files; via TCP-IP or USB transfers; relying on physical exploration, visual-gestures or vocal dialogues.**

1. **The fastest cycle of coordination lasts in average for about 2 microsecond (μs), with the multitask kernel described above. For example a (global) motion programmed with Cartesian coordinates triggers several levels of numerous parallel (sub)tasks: inverse kinematics, coordinated motion laws, possible joint level servoing at high rate.**
2. **Parallelism at system level typically implies changes with more than 10 ms periods.**
3. **Coordination is often relying on file exchanges, for example between different programs on the supervisory computer, possibly written in different languages and/or involving different compilers; duration of such cycles is probably better, i.e. shorter, than 50 ms.**

# 3.B.4. Multiple degrees of inter-cooperation performance in Piaget 2 of 3

4. **When exchanges involve peripheral resources such as smart sensors, other computers, or robots, TCP-IP or USD techniques are applicable, and the coordination proceeds with delays on the order of 0.1 second.**

5. **For application-level loops, e.g.**
   - **cooperating robots in domestic environment, or**
   - **industrial robots exploring complex situations,**

   **physical exploration, visual-gestures or possibly vocal dialogues may be required, including header invitations and answer validations. Such a type of action may last in total for a time span on the order of 1 to 10 second, or even more.**

---

# 3.B.4. Multiple degrees of inter-cooperation performance in Piaget 3 of 3

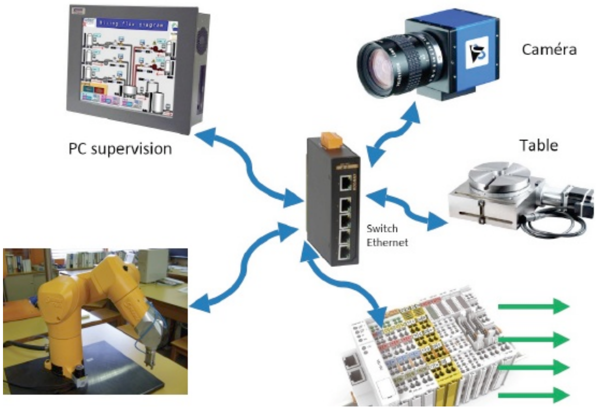


**Fig. 16. Communication between main components in Piaget environment typically relies on TCP/IP and/or USB; here via a hub (or switch).**

# 3.B.5. Test instruction and task in Piaget 1 of 2

**Programming can be performed gradually in terms of complexity.**

**1. For simplicity and a quick start, new users can write their first level-1 program just as a single instruction located in "900". Typically an application starts with a preparation phase in the beginning of the strategy task; then an application-dependant launching phase occurs: continuation may be given in many alternate ways: real boolean input, simulated input ("D" key or similar click), vocal command ("Go", "Yes", etc.; by clicking or speaking in robot's microphone.); or finally, of particular interest here, the "*ModeTest*" control allows to launch the code located in area 900.**

**2. Then this can be similarly expanded as several instructions from that same address.**

---

# 3.B.5. Test instruction and task in Piaget 2 of 2

3. **An application "*TestTask*" is available, as a simple programming example that can be freely modified for new users to acquire experience (re. so-called "sandbox" in other contexts).**

4. **Experience accumulating, expertise also increases and programming becomes more sophisticated, e.g. including the definition of novel elements in Piaget-implementing context (C++, C#, etc.).**

# 3.B.6. Examples of application – Piaget for Cognitics 1 of 13

**This section gives 5 examples of applications developed and driven by Piaget; examples in automated cognition, in cognitics. The first three reflect the main successive application areas of Piaget: Eurobot, Robocup@Home, and industrial robotics; the next two respectively highlight robust vision techniques and estimations in quantitative cognitics [5] as supported in Piaget.**

**1. Piaget was concretely created for Eurobot competitions. As illustrated in Fig. 17, in the "Coconut-rugby", sets of 2 robots had 1.5 minute to defeat an opponent robot set attempting to achieve the similar, "mirror" task: catch coconuts, bring or throw them in opposite net or blue "essai" zone, block own goalsite, and possibly retrieve balls scored by the opponent. One typical skill consists in visually locating randomly located coconut trees and coconuts. Such competences include the fast (0.1 s) perception of colors in 9 robust categories, the recognition of coconuts and trees, as well as the location coordinate transform from picture rows and columns to Cartesian X-Y values on the field (Fig. 17)**

---

# 3.B.6. Examples of application – Piaget for Cognitics 2 of 13

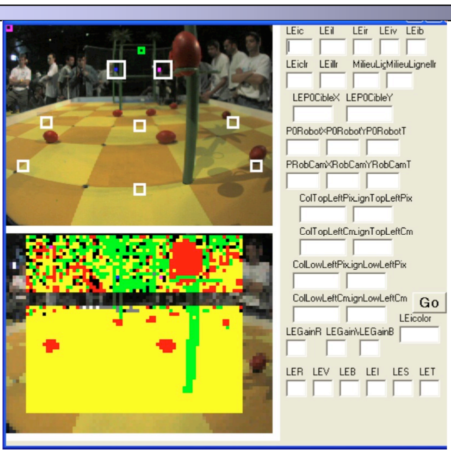


**Fig. 17. Skilled competences include the fast perception of colors and recognition of objects, as well as coordinate transforms from picture onto field.**

# 3.B.6. Examples of application – Piaget for Cognitics 3 of 13

**2. Moving to Robocup@Home called for more complexity. Fig. 18 illustrates vocal and dialogue management as typically supported in Piaget environment and language, as well as a vision-based face recognition**

The path has been completely travelled

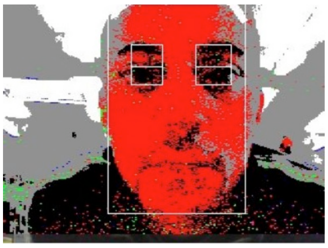


**Fig. 18. On the left, Piaget panels and text-typed fields illustrate typical vocal dialogues : yes/now can simulate microphone inputs; recognized commands are shown in green (here " ") and synthetized text in yellow. On the right a face is recognized for "Who is Who" test**

---

# 3.B.6. Examples of application – Piaget for Cognitics 4 of 13

**Advanced tests in terms of cognitive capabilities and human robot interaction capabilities have been demonstrated in Robocup@Home world competitions, e.g.**

**"CopyCat": programming by showing**

**"FastFollow": leading a robot in new homes just by walking (Fig. 19),**

**"Walk'nTalk": training a robot in new homes just by walking and defining vocally key objects or locations (Fig. 20);**

**"OpenChallenge": e.g. in Singapore our robotic group included three coordinated robots, and in particular a humanoid for the purpose of mediation between human and machines (Fig. 20). Re. http://rahe.populus.ch and YouTube**

# 3.B.6. Examples of application – Piaget for Cognitics 5 of 13

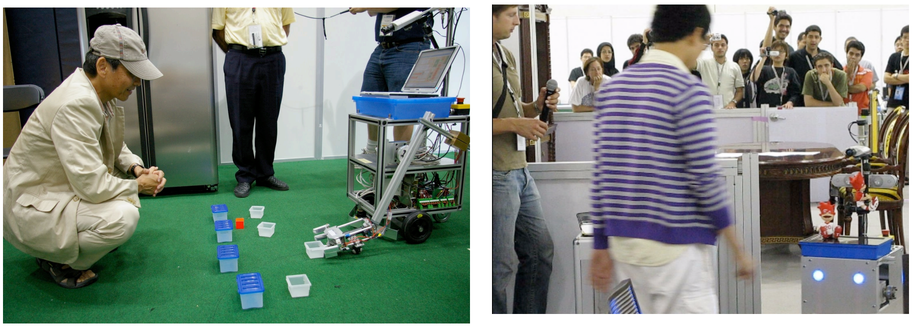
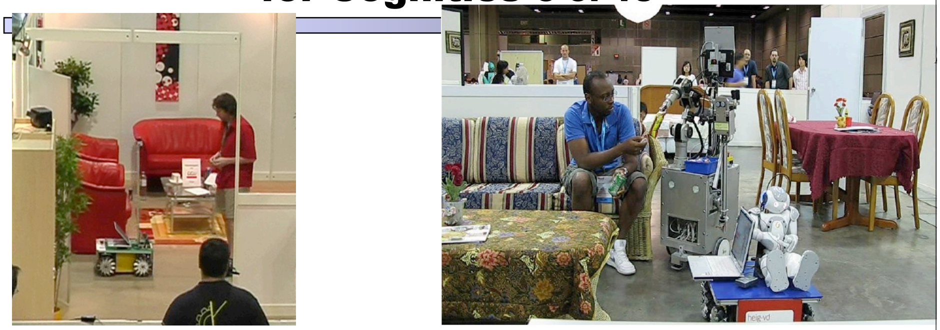


**Fig. 19. On the left, RH-Y robot visually analyzes and immediately replicates each of the object displacements manually performed by President Asada. On the right, RH-Y moves fast, following its guide, crossing another team, and completing first the imposed visit of a home.**

---

# 3.B.6. Examples of application – Piaget for Cognitics 6 of 13



- **Fig. 20 Human and robot share their representations**
  - **On the left, the robot follows first the human, and then they vocally synchronize their respective English names for describing specific locations, such as the plant in the living room (*"Walk'nTalk", Graz, Austria*).**
  - **On the right, Nono-Y, our Nao-typed humanoid mediates humans and other machines (OP-Y platform where Nono-Y sits; and RH-Y robot, which has brought drinks and snacks) (*"Open Cgallenge", Singapore*)**

# 3.B.6. Examples of application – Piaget for Cognitics 7 of 13

**3. Industrial applications can also been driven by Piaget .**

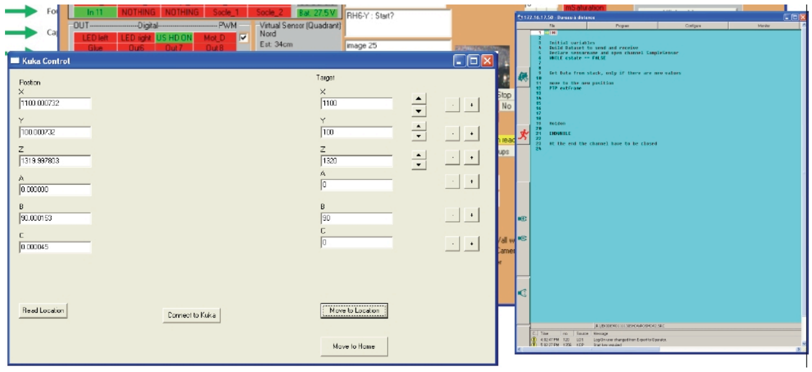
**Fig. 21 and 22 illustrate two cases, the former one involving a Staubli robot and the latter one a Kuka.**

**In both cases, the robot arms are driven, at elementary, lowest level, by manufacturers' controllers (incl. KRL for Kuka; Val3 for Stäubli) and, at higher levels, by a program developed in Piaget environment and expressed in Piaget language**

# 3.B.6. Examples of application – Piaget for Cognitics 8 of 13

**Fig. 21. "Chip count and accuracy test": Application mostly developed and programmed in Piaget, including the industrial robot arm visible on the picture and other resources: optical fiber, PLC, camera, motorized rotating table, servocontroller, Ethernet switch, PC and other components yet.**

# 3.B.6. Examples of application – Piaget for Cognitics 9 of 13



Nord
Est: 34cm

RH6-Y : Start?

image 25

Postion
X 1100.000732
Y 100.000732
Z 1319.997803
A 0.000000
B 90.000153
C 0.000045

Target
X 1100
Y 100
Z 1320
A 0
B 90
C 0

**Fig. 22. Three windows relating to an industrial application involving a Kuka robot (The first two belong to Piaget environment; the third one is a remote desktop linked to Kuka controller)**

---

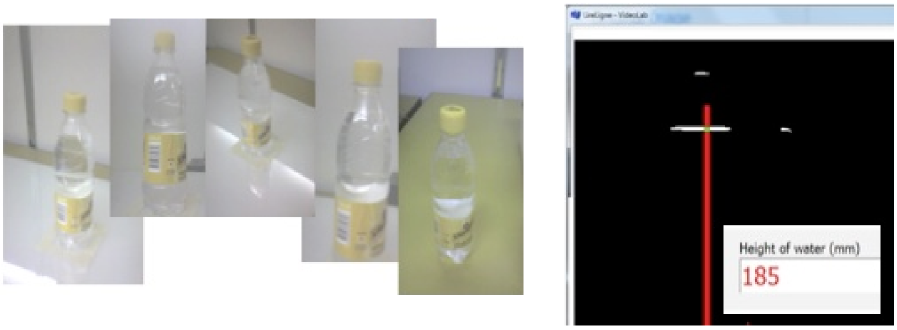
# 3.B.6. Examples of application – Piaget for Cognitics 10 of 13

**4. Piaget supports fast and robust vision, in many modes (infrared/BW, color, thermal, 3D-time of flight, RGB-D sensors; various processes).**

**In the first year edition already (Eurobot context), it could acquire and process 300 pictures per second to locate opponent's robot in real-time.**

**Fig 23 illustrate a key paradigm by which instead of mundane images (left), care is taken to analyze applications in full physical space (here "capillarity" is the most discriminating dimension) before appropriately mapping them into common light domain and processing them specifically.**

# 3.B.6. Examples of application – Piaget for Cognitics 11 of 13



Height of water (mm)
185

**Fig. 23. Left: Piaget discourages naïve vision, supporting goal-oriented image acquisition, processing and analysis. Right: visual, real-time quality control of liquid/water level.**

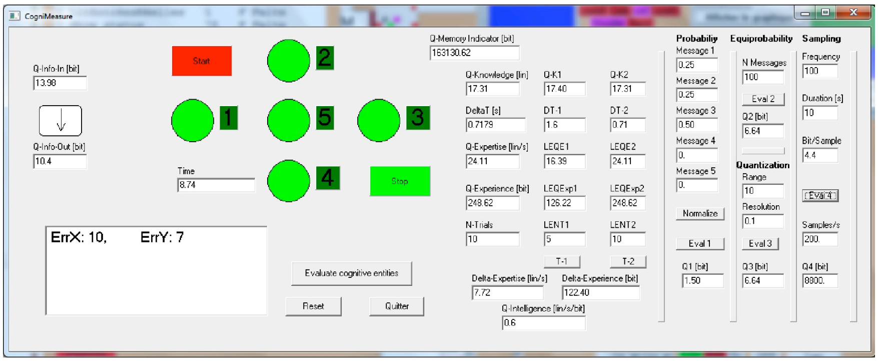
---

# 3.B.6. Examples of application – Piaget for Cognitics 12 of 13

**5. A particular interest of Piaget environment is to provide a tool for convenient, quantitative estimation of core cognitive properties:**

- **knowledge, expertise, experience, speed/fluency, intelligence, as well as low-level ingredients:**
  - **probability calculus, quantization, sampling rate, input and output information signals and quantities,**
- **all this along with an interactive example (Fig. 24)**

# 3.B.6. Examples of application – Piaget for Cognitics 13 of 13



Q-Info-In [bit] 13.98

Q-Info-Out [bit] 10.4

Time 8.74

ErrX: 10, ErrY: 7

Q-Memory Indicator [bit] 163130.62

Q-Knowledge [lin] 17.31
Q-K1 17.40
Q-K2 17.31

DeltaT [s] 0.7179
DT-1 1.6
DT-2 0.71

Q-Expertise [lin/s] 24.11
LEQE1 16.39
LEQE2 24.11

Q-Experience [bit] 248.62
LEQExp1 126.22
LEQExp2 248.62

N-Trials 10
LENT1 5
LENT2 10

Delta-Expertise [lin/s] 7.72
Delta-Experience [bit] 122.40
Q-Intelligence [lin/s/bit] 0.6

**Probabiliy**
Message 1 0.25
Message 2 0.25
Message 3 0.50
Message 4 0.
Message 5 0.
Q1 [bit] 1.50

**Equiprobability**
N Messages 100
Q2 [bit] 6.64

**Quantization**
Range 10
Resolution 0.1
Q3 [bit] 6.64

**Sampling**
Frequency 100
Duration [s] 10
Bit/Sample 4.4
Samples/s 200.
Q4 [bit] 8800.

**Fig. 24. Piaget environment includes a form for the quantitative estimation of cognitive properties in general, along with a specific example: learning how to accurately click in the center of 4 targets**

# Content

1. **Introduction**
2. **Requirements and theoretical aspects of intelligent control**
3. **Piaget**
4. **Conclusion**

**http://lara.populus.org/rub/3**

# 4. Conclusion 1 of 4

**Scientific and technical progress now reaches cognition.**

- **Current industrial robots ok in terms of power and accurate motions in space.**
- **But face new challenges, in terms of cognition:**
  - **industrial environment getting more complex,**
  - **tending to change more dynamically;**
  - **real-time cooperation with humans and other resources is envisioned.**

**Our first stage in the exploration of cognition has been**

- **to define concepts formally and to develop metrics.**

**The second stage has been**

- **to select an architecture and**
- **to develop an environment for the real-time, real-world control of complex systems, capable of addressing the most advanced applications in terms of automation and cognitive, human-related tasks.**

---

# 4. Conclusion 2 of 4

**All this has been done with the following concern**

- **keep connected to world-level expertise and international best practices, for all aspects,**
  - **from the boundaries of scientific and human theories**
  - **to the ones of the market, as involving commercial components and services.**

## 4. Conclusion 3 of 4

- Consider jumping over a wall: the metric height of the wall is a critical parameter for success.
- Similarly, the novel possibility of metric assertion of cognitive aspects (complexity, knowledge, expertise, etc.) is very useful. This is a natural merit of the proposed approach.
- Besides, conference attendance and state of the art monitoring bring useful new information.
- In addition, the methodology of realizing real-world systems allows for concretely implementing proposed theories.
- This can moreover lead to actual competitions on common test applications, thus encouraging active interaction with international experts.

---

## 4. Conclusion 4 of 4

- Our developed, comprehensive framework “MCS” now allows for the quantitative assessment of cognitive tasks, both as required or as operated by humans and machines.
- The proprietary environment “Piaget” has been created, proving to ensure, initially, the convenient control of mobile robots, then “naturally” cooperating with humans. Implemented in different languages (C, C#, C++), with different Operating Systems (incl. RTDOS and Windows) and platforms, “Piaget” has now been successfully added Kuka and Staubli industrial robots to its numerous integrated resources

# Acknowledgements

- **The authors wish to thank numerous engineers and students, as well as members of technical services, at HEIG-VD, who have contributed to the reported projects**

# References 1 of 3

1 **ISR-2012, 43rd International Symposium on Robotics, Taipeh, Taiwan,**

2 **Jean-Daniel Dessimoz, "Ten Years of Experience with Eurobot ; Achievements, Lessons Learned and General Comments", *Proc. Eurobot Conference 2008*, Heidelberg, Germany,22-24 May 2008, ISBN: 978-80-7378-042-5.**

3 **H Kitano, M Asada, Y Kuniyoshi, I Noda, E. Osawa, “Robocup: The robot world cup initiative”, *AGENTS '97 Proceedings of the first international conference on Autonomous agents*, ACM New York, NY, USA ©1997**

4 **Tijn van der Zant and Thomas Wisspeintner, « RoboCup@Home: Creating and Benchmarking », *Tomorrows Service Robot Applications , Robotic Soccer,* Book edited by:Pedro Lima, ISBN978-3-902613-21-9, pp.598, December 2007, ItechEducationandPublishing, Vienna, Austria**

5 **Jean-Daniel Dessimoz, "*Cognitics - Definitions and metrics for cognitive sciences and thinking machines*", Roboptics Editions, Cheseaux-Noréaz, Switzerland, ISBN 978-2-9700629-1-2, pp169, January 2011 http://cognitics.populus.ch**

# References 2 of 3

6 **Nicolas Uebelhart, Florian Glardon and Pierre-François Gauthey, "Lomu, an Autonomous Mobile Robot with Robust Architecture and Components", *« DARH-2005 - 1st International Conference on Dextrous Autonomous Robots and Humanoids», with sponsorship Eurobot, IEEE, CLAWAR, and CTI, HESSO-HEIG* (West Switzerland University of Applied Sciences), Yverdon-les-Bains, Switzerland, May 19-22, 2005. (re: DARH-2005-org )**

7. **O Michel, "WebotsTM: Professional mobile robot simulation", *arxiv.org- Arxiv preprint cs/0412052*, 2004**
8. **J Jackson "Microsoft robotics studio: A technical introduction ", *Robotics & Automation Magazine, IEEE,* 2007 - ieeexplore.ieee.org (started in 2006)**
9. **Shimano, B.. "VAL: a versatile robot programming and control system", Unimation Inc., *Computer Software and Applications Conference, 1979. Proceedings. COMPSAC 79*. The IEEE Computer Society's Third International Conf., 1979, pp. 878 – 883**

---

# References 3 of 3

10. **Stäubli. In *VAL3 robot programming language*. 2012.04.07, from**
11. **Pierre Maurer and Micael Gagnebin, "Advanced control structure for the autonomous mobile robot Lodur", « *DARH-2005 - 1st International Conference on Dextrous Autonomous Robots and Humanoids*», with sponsorship Eurobot, IEEE, CLAWAR, and CTI, HESSO-HEIG (West Switzerland University of Applied Sciences), Yverdon-les-Bains, Switzerland, May 19-22, 2005. (re: DARH-2005.org )**

# Thanks for your attention!

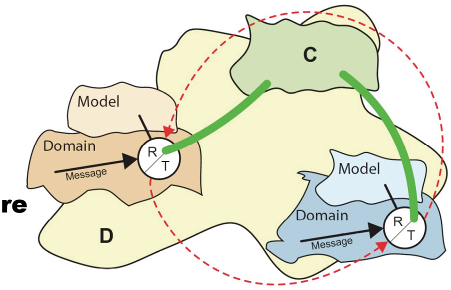
**http://lara.populus.org/rub/3**

---

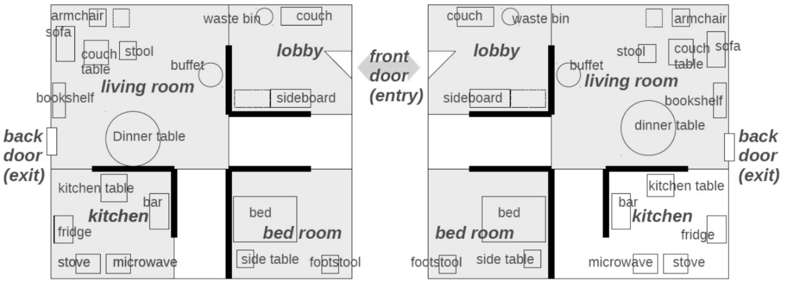
# Appendix 1 – « Robot sociology »

***Group*.**

- **Individual cognitive agents (blue, brown) may coordinate each other, and thus may collectively form a group.**
- **For this purpose,**
  - **a common culture (C, green),**
  - **in reference to some common domain of interest (D, yellow) and**
  - **some communication media are required among agents (R: receive; T: transmit).**
- **At a meta-level, the individual members may be considered as merging, to yield a new individual (the group) with its own collective model (C). (From [6])**



# Appendix 2 « Standard »



| # | location | category | manipulation in GPSR | category placing |
|---|---|---|---|---|
| 1 | sofa | seat | no | |
| 2 | couch table | table | yes | |
| 3 | armchair | seat | no | |
| 4 | stool | seat | no | |
| 5 | dinner table | table | yes | |
| 6 | bookshelf | shelf | yes (2nd height) | |
| 7 | buffet | shelf | yes | |
| 8 | fridge | appliance | yes | |
| 9 | stove | appliance | no | |
| 10 | microwave | appliance | yes (table) | |

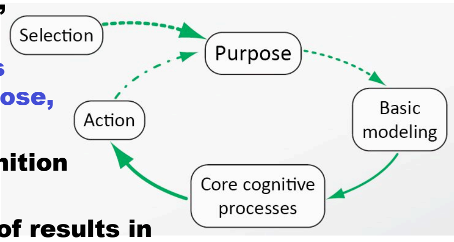
| # | location | category | manipulation in GPSR | category placing |
|---|---|---|---|---|
| 11 | kitchen table | table | yes | food |
| 12 | bar | shelf | yes | drinks |
| 13 | couch | seat | no | |
| 14 | sideboard | shelf | yes | snacks |
| 15 | wastebed | seat | yes | |
| 16 | side table | table | yes | |
| 17 | waste bin | bin | yes (placing) | unkown |
| 18 | bed | seat | yes | bath stuff |
| 19 | side table | table | yes | |
| 20 | footstool | seat | no | |

[12]

---

# Appendix 3 « Purpose-driven modeling »

- **From a cognitic perspective, the strong modeling limit illustrated previously, calls for a pragmatic process**
  - **the complexity of reality requires first the selection of a goal (purpose, ethics)**
  - **only then, can modeling and cognition proceed**
  - **finally, the symmetric necessity of results in the real world requires action (operability, agency), for example by robots**
    - **cognitive results must be put to work, with energy etc.,**
    - **thereby closing of the loop (iteration, and ultimately, in general, survival); from [3].**



## Abstract

Scientific and technical progresses now reach cognitive domains. Current industrial robots face new challenges, in terms of cognitive capabilities. The first stage in the exploration of cognition has been to define concepts clearly and to develop metrics. The second stage has been to select an architecture and to develop an environment for the real-time, real-world control of complex systems, capable of addressing the most advanced applications in terms of automation and cognitive, human-related tasks; with the concern of keeping connected to world-level expertise and international best practices.

Consider jumping over a wall: the metric height of the wall is a critical parameter for success. Similarly, the novel possibility of metric assertion of cognitive aspects (complexity, knowledge, expertise, etc.) is very useful. Our developed, comprehensive framework “MCS” now allows for the quantitative assessment of cognitive tasks, both as required or as operated by humans and machines.

The proprietary environment “Piaget” has been created, proving to ensure, initially, the convenient control of mobile robots, then “naturally” cooperating with humans. Implemented in different languages (C, C#, C++), with different Operating Systems (incl. RTDOS and Windows) and platforms, “Piaget” has now been successfully added Kuka and Stäubli industrial robots to its numerous integrated resources.