From Active Objects to Autonomous Agents

Zahia Guessoum & Jean-Pierre Briot

OASIS
(Objets et Agents pour Systèmes d'Information et de Simulation)
LIP6
(Laboratoire d'Informatique de Paris 6)

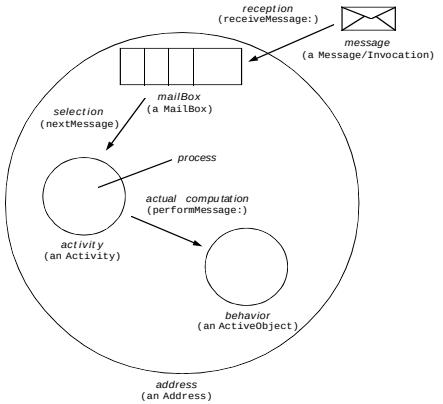
e-mail: {guessoum, briot}@ poleia.lip6.fr
http:\\www-poleia.lip6.fr\{~guessoum, ~briot}

In quest of the missing link between active objects and autonomous agents ...

Plan

Active-Object Models

Example of Active-Objects Model : *Actalk* (Briot 94)



In spite of their communicating subjects appearance, active objects do not reason about their behavior, on their relations and their interactions with other objects... (Ferber 89).

Some Extensions of Active Objects

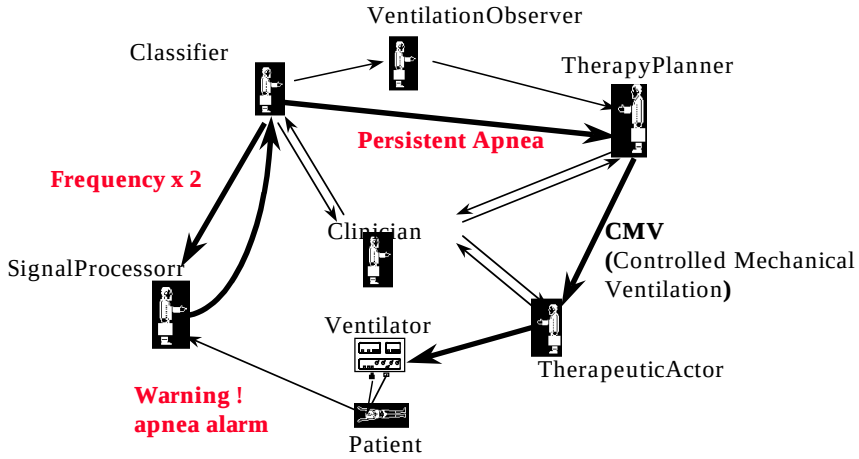
⇨ **T. Maruichi**
- a message interpreter
- the notion of environment

⇨**T. Bouron** (MAGES IV)
- speech actes

⇨**Y. Shoham** (Agent-Oriented Programming)
- mental states to have an interaction-based behavior

⇨**S. Giroux** (ReActalk)
- reflexive and adpative active objects

Example : Intensive Care Patient Monitoring

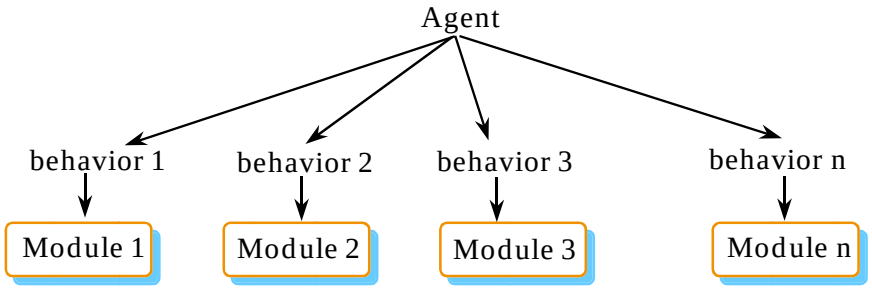


Example : Intensive Care Patient Monitoring

- ⇨ Each agent is an autonomous and active entity
- ⇨ Each agent may have various kinds of behaviors (communicates with other agents, scans its environment, makes decision, ...).
- ⇨ Each agent may have simple or complex reasoning capacities
- ⇨ Each agent owns its control knowledge, the control is not achieved through separated architecture.

A Generic Agent Architecture
Different behaviors

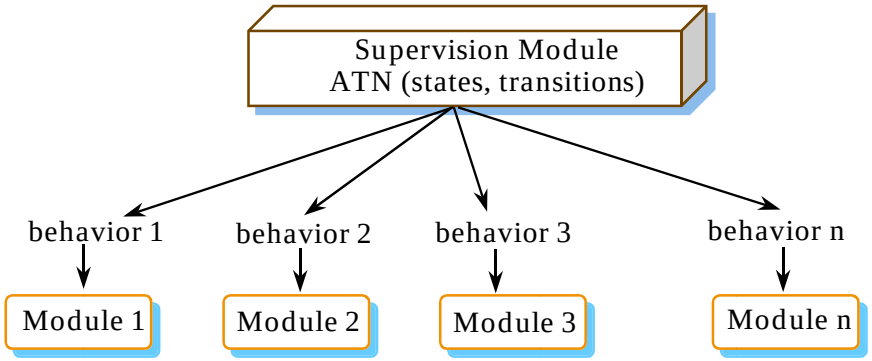
Various kinds of behaviors (reactive or deliberative)
⇨ Different asynchronous and concurrent modules



Examples :
Perception, communication, reasoning, action, learning...

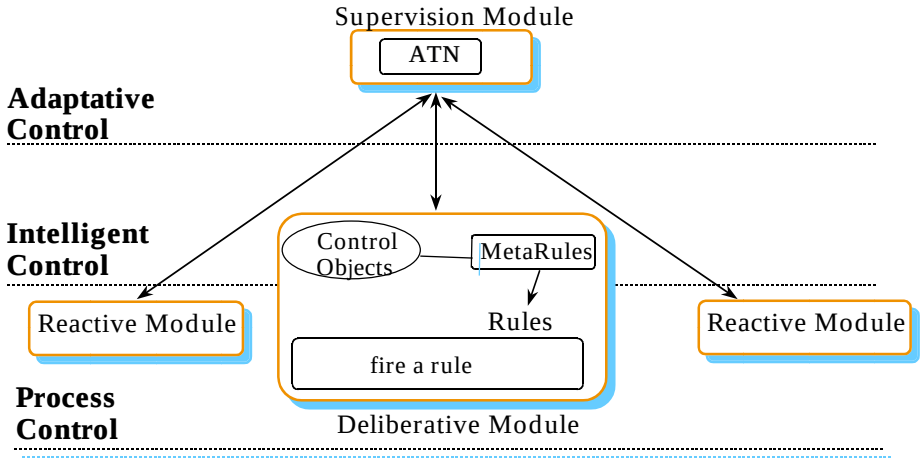
A Generic Agent Architecture
A Meta-Behavior

A Meta-Behavior
⇨ Supervision Module



A Generic Agent Architecture
Self-Control Mechanism

Adaptive Control ==> Autonomy



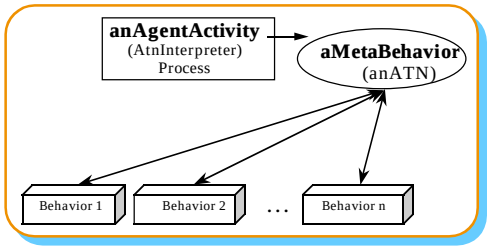
Implementation (*Actalk --> DIMA*)

An Active Object

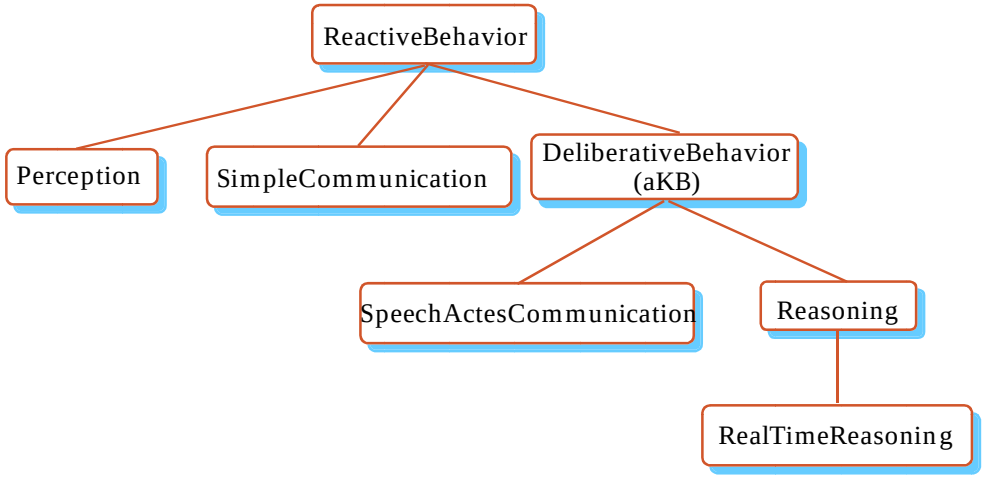
```
body
[true]whileTrue:
    [self acceptNextMessage]

createProcess
^[self body] newProcess
```

An Agent Structure

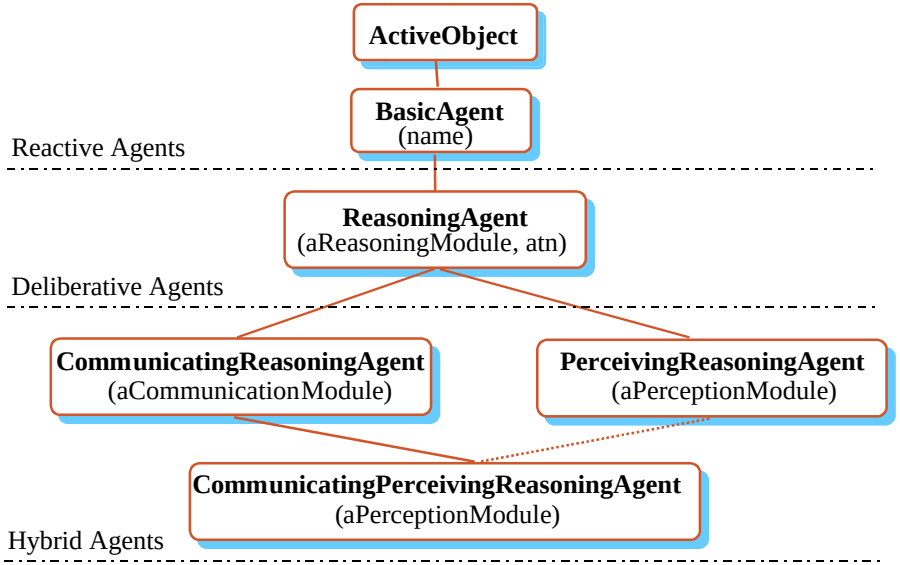
```
body
 self atnInterpreter

createProcess
^[self body] newProcess
```



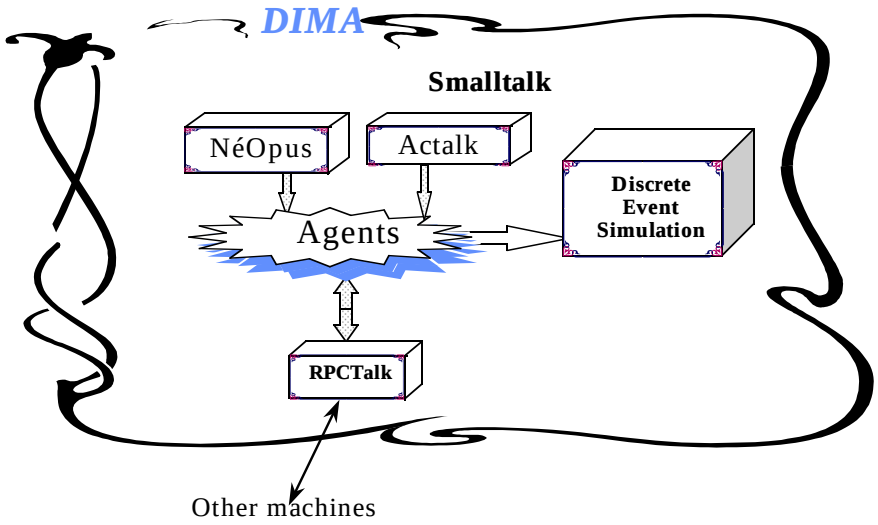
Behavior Hierarchy



Meta-Behavior Hierarchy



The Developed Environment



Use of *DIMA*

➪ **Several real-life applications**

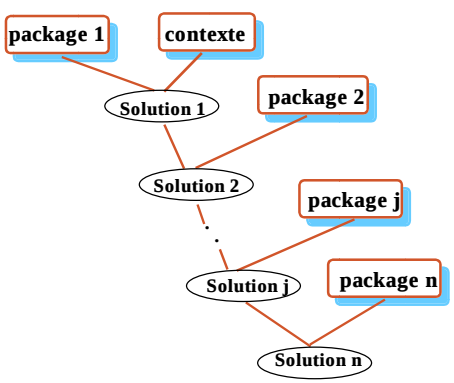
- NéoGanDi : Intensive Care Patient Monitoring (INSERM, Paris)
- Meveco : Economic Agents Evolution (HEC, Banque de France)
- ATM Congestion Simulation
- ...

Multi-Agent Systems	NéoGanDi	Meveco
Number of Agents	constant	variable
Nature of Agents	cooperative	competitive
Context of Agents	agents activity relies on context	context relies on agents activity
Temporal Constraints	strong	weak
Social Reasoning	absent	Model-based

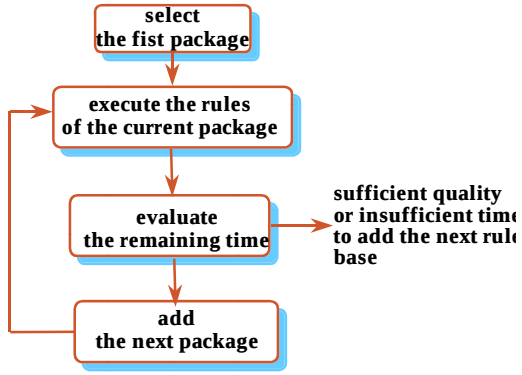
Use of *DIMA*

➪ **to study multi-agent problems :** real-time (anytime reasoning)

Schematic structure



Execution cycle of the real-time metabase



Conclusion & Future Work

Characteristics of the proposed multi-agent platform :

- A generic and modular agent architecture
  - Several paradigms (object, production rules, ATN, ...)
  - Various kinds of agents (reactive, deliberative, hybrid)
- Control mechanisms at two levels
  - At the agent level : a self-control mechanism
  - At the multi-agent level : a coordination mechanism
- Several real-life applications

Future work :

- Real-time multi-agent systems
- Learning behavior