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

1. Active Objects
2. A Generic Architecture of Agent
3. From Actalk to DIMA
4. Experiments
5. Discussion and Future Work

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).

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

~ T. Maruichi
  • a message interpreter
  • the notion of environment

~ T. Bouron (MAGES IV)
  • speech acts

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

~ S. Giroux (ReActalk)
  • reflexive and adaptive active objects

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 decisions, ...).
~ 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

Supervision Module
ATN (states, transitions)

behavior 1
Module 1

behavior 2
Module 2

behavior 3
Module 3

behavior n
Module n

Implementation (Actalk --> DIMA)

An Active Object

body
[true]while True:
  [self acceptNextMessage]

createProcess
^[self body] newProcess

An Agent Structure

body
self atnInterpreter

createProcess
^[self body] newProcess

A Generic Agent Architecture
Self-Control Mechanism

Adaptive Control ==> Autonomy

Intelligent Control

Process Control

Deliberative Module

Behavior Hierarchy

ReactiveBehavior

Perception

SimpleCommunication

DeliberativeBehavior (aKB)

SpeechActsCommunication

Reasoning

RealTimeReasoning

Implementation (Actalk --> DIMA)

An Active Object

body
[true]while True:
  [self acceptNextMessage]

createProcess
^[self body] newProcess

An Agent Structure

body
self atnInterpreter

createProcess
^[self body] newProcess
## Meta-Behavior Hierarchy

- **Reactive Agents**
  - BasicAgent
    - ReasoningAgent
    - CommunicatingReasoningAgent
    - PerceivingReasoningAgent
  - ActiveObject

- **Deliberative Agents**
  - CommunicatingPerceivingReasoningAgent
    - ReasoningModule
    - PerceptionModule

- **Hybrid Agents**
  - Reactive Agents
  - Deliberative Agents

## The Developed Environment

- **Agents**
  - NéOpusActalk
  - RPCTalk
  - Smalltalk
  - Other machines

- **Discrete Event Simulation**
  - DIMA
  - NeOpus
  - Actalk
  - Smalltalk

## 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

<table>
<thead>
<tr>
<th>Multi-Agent Systems</th>
<th>NéoGanDi</th>
<th>Meveco</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Agents</td>
<td>constant</td>
<td>variable</td>
</tr>
<tr>
<td>Nature of Agents</td>
<td>cooperative</td>
<td>competitive</td>
</tr>
<tr>
<td>Context of Agents</td>
<td>agents activity relies on context</td>
<td>context relies on agents activity</td>
</tr>
<tr>
<td>Temporal Constraints</td>
<td>strong</td>
<td>weak</td>
</tr>
<tr>
<td>Social Reasoning</td>
<td>absent</td>
<td>Model-based</td>
</tr>
</tbody>
</table>

### Use of DIMA

- To study multi-agent problems: real-time (anytime reasoning)

  **Schematic structure**

  - Select the fist package
  - Execute the rules of the current package
  - Evaluate the remaining time
  - Add the next package

  **Execution cycle of the real-time metabase**

  - Sufficient quality or insufficient time to add the next rules base
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