Laboratory Framework for Information Ecosystems

Wamberto Vasconcelos[1] and Jordi Sabater[2]

[1]Division of Informatics,University of Edinburgh, UK
Wamberto.Vasconcelos@ed.ac.uk
[2]Institut d'Investigació en Intel.ligència Artificial, CSIC, Spain
jsabater@iiia.csic.es

20 June 2001

Plan of Presentation

1. Introduction

2. Sequential Simulation

3. Parallel Simulation

   - Multi-Processing
   - Multi-Threading
   - JADE and Multi-Thread

4. Conclusions and Directions of Research

1. Introduction

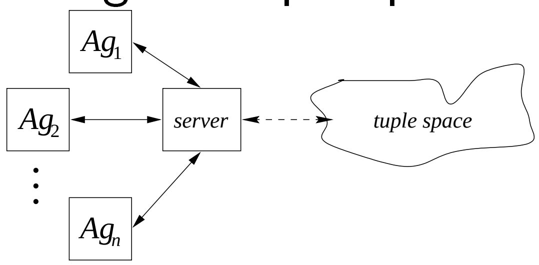
- Purpose: tools to study *populations* of infohabitants
- Basic services
  - run experiments with large number of infohabitants
  - communication among infohabitants
  - means to monitor/record behaviour of infohabitants
- Programming Infohabitants: why Prolog?
  - Programming *intelligent* infohabitants
  - Language design: meta-Programming
  - Experiments: program synthesis and transformation
  - Fast prototyping

2. Sequential Simulation

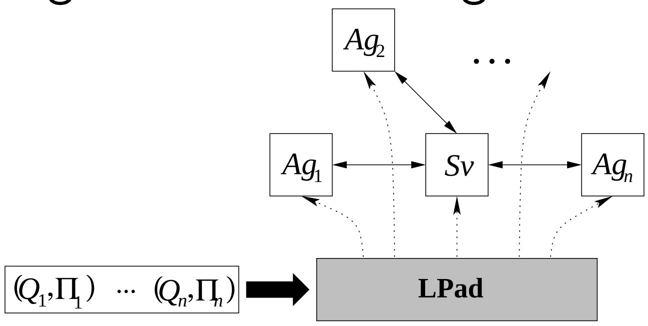
- Possibilities:
  - One complete infoagent at a time (very restricted)
  - One command of an infoagent at a time – interleaving
- Advantages:
  - Runs can be replicated
  - *All* possible runs can be obtained
- Disadvantages:
  - Not realistic
  - Time-sensitive agents not directly simulated
  - Sometimes not feasible: $n \times t$
  - Exponential worst-case

3.1 Parallel Simulation via Multi-Processing

- Infohabitant = independent process
- Linda: message passing via tuple space



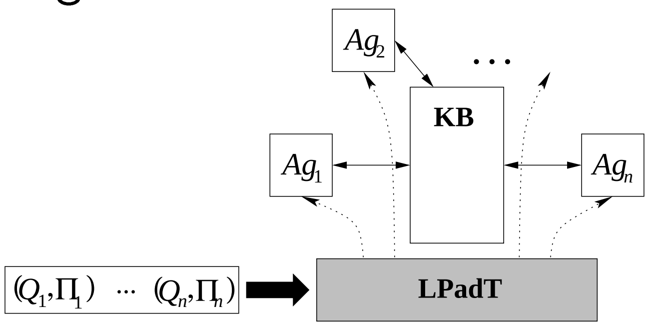
- **LPadL**: Launching Pad for Linda Agents



- Advantages:
  - Time-sensitive agents can be simulated
  - Agents may run in different machines
  - Realistic: message-passing and distributed execution
  - Simple and stable technology
  - Tracing/debugging of individual agents is possible
- Disadvantages:
  - Quick depletion of machine's resources
  - Lack of global account of execution
  - SICStus Prolog is essential
  - Only SICStus Prolog agents

3.2 Parallel Simulation via Multi-Threading

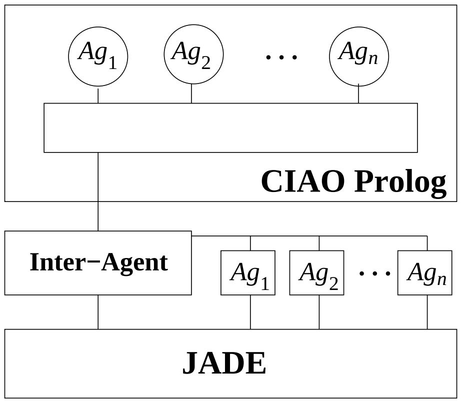
- Thread: independent interpreter
- Infohabitant = execution thread
- n threads sharing single program/knowledge base
- Message passing via shared program/knowledge base
- **LPadT**: Launching Pad for Multi-Threads



- Advantages:
  - More agents per machine (threads are "lightweight")
  - CIAO Prolog essential but free
  - Extension for agents implemented in other languages
  - Extension for agents running in different machines
- Disadvantages:
  - Tracing/debugging not possible
  - Potential "clashes" due to shared KB
  - Unstable technology

3.3 Parallel Simulation: JADE & Multi-Thread

- **JADE**: **J**AVA **A**gent **DE**velopment
- FIPA-compliant agent platform
- JADE + CIAO Prolog:



- Connect different Prolog sessions
  (each possibly running many threads)
- Advantages:
  - Standardisation/compatibility
  - Free
  - Visualisation tools provided
  - Those of multi-threading and additionally:
    * CIAO Prolog sessions in different machines
    * Agents in other languages can join in
- Disadvantages:
  - Those of Multi-threading

4. Conclusions and Directions of Research

- Prolog as a programming language

- 3 approaches for simulating information ecosystems:

  1. Sequential
  2. Multi-Processing
  3. Multi-Threading

  Each with advantages and drawbacks

- JADE + Prolog: the best of both worlds

- Realism $\propto 1/$Control of simulation

- Specification of experiments:
  different platforms for different needs

- Redesign JADE's visualisation tools for our needs

- Integrate laboratory framework with other tools