Advanced Human Machine Interactions

Introduction to multi-agent systems and autonomous agents

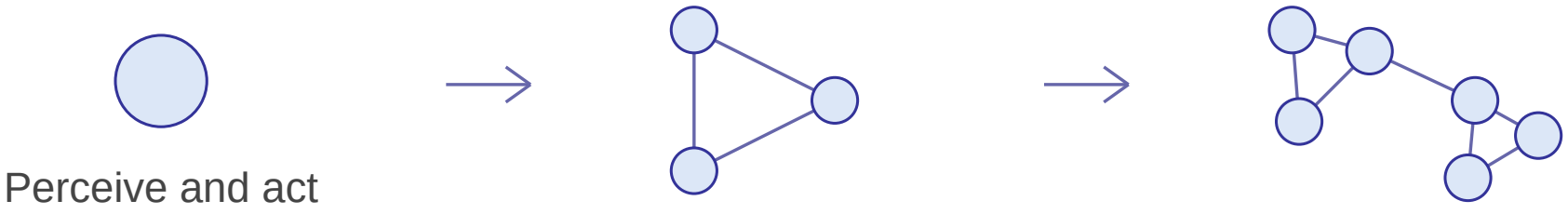
Modelling agents, interactions and collective behaviour

Alexandre Pauchet
INSA Rouen – Département ITI
BO.B.RC.18, alexandre.pauchet@insa-rouen.fr

From interaction data to agent-based models

Previous lectures: analyse traces, sequences and networks of interactions.

Today: explicitly model the entities whose actions produce these interactions.

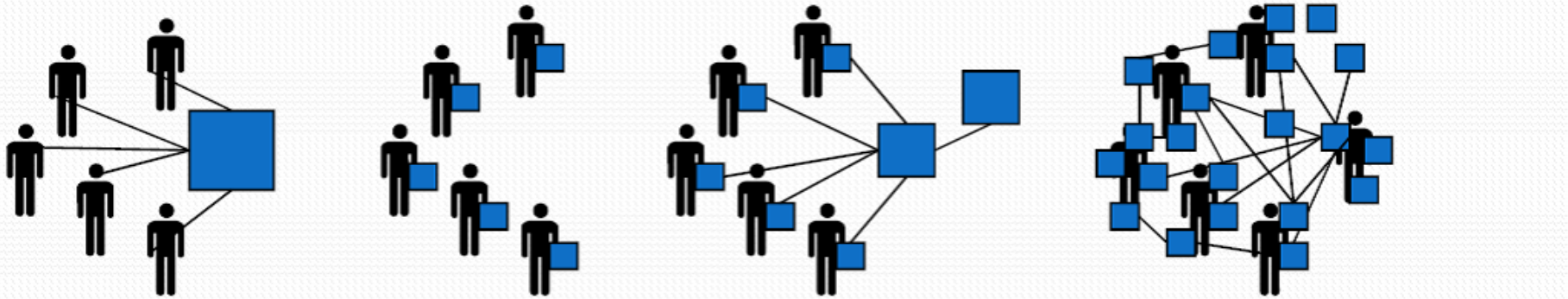


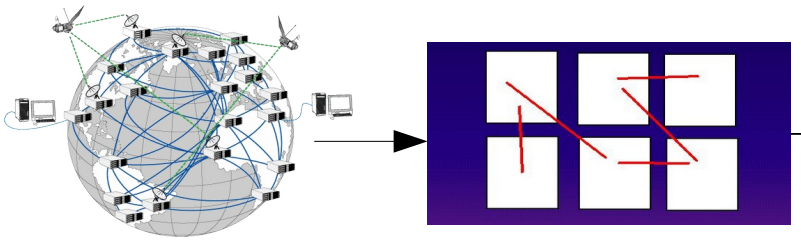
Individual behaviours → **Local interactions** → **Collective patterns**

How can local behaviours produce the patterns we observe?

Systems and communication

- Évolution de l'informatique vers le « Pervasive Computing » (informatique diffuse)





Internet web

- Cloud computing
- Connected objects
- Ubiquitous/pervasive computing
- Internet of things

Interacting entities produce traces at different levels:
individual actions, exchanges between agents, and collective patterns.

Artificial Intelligence

Statistical approach

- Formal neural nets
- Multi-layer perceptron

Symbolic approach

- Turing test, ELIZA
- Lisp, Prolog, …
- Expert systems (Mycin, ...)

Rise of **distributed** AI and **collective intelligence** to solve more and more **complex** problems, subject to **uncertainties** and **incomplete knowledge**

What is an agent? (according to Sycara)

An **agent** is an entity that perceives its environment and selects actions autonomously.

- **Situated**: receives local information and acts on an environment.
- **Autonomy**: controls its actions according to its behaviour rules or decision mechanisms.
- **Reactive and/or proactive**: responds to changes and/or pursues goals.
- **Sociability**: can affect other agents and be affected by them.

Note: Learning is possible, but not required!

MAS definition (according to Ferber)

A **MAS** is made up of:

- an **environment** *E*
- a **set of situated objects** *O* (*a position into E is associated with them*)
- **a set of agents** *A* (*A included into O ; they are active compared to other passive objects of O*)
- **a set of relations** *R* uniting objects
- **a set of operations** *Op* allowing agents *A* to perceive and manipulate objects in *O*
- **operators** representing the application of those operations and the world reaction to the resulting modification attempts (called the universe laws)

A multi-agent system combines agents, an environment, and mechanisms governing their interactions.

Why multi-agent models in IHME/IDA?

Describe	Identify actors, available information, actions and interactions.
Simulate	Generate interaction traces from explicit behavioural assumptions.
Analyse	Study how local behaviours contribute to collective patterns.
Compare	Compare simulated patterns with observations.

Heterogeneous agents may differ in capabilities, information or behaviour.

Mixed human–agent systems involve both humans and artificial agents.

Multi-agent application examples

Example: MASSIVE
From individual behaviour to crowd simulation

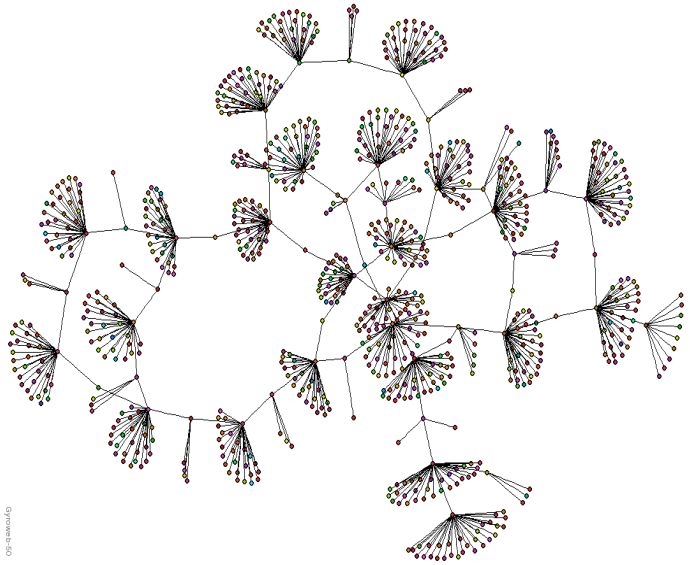
While watching, identify:

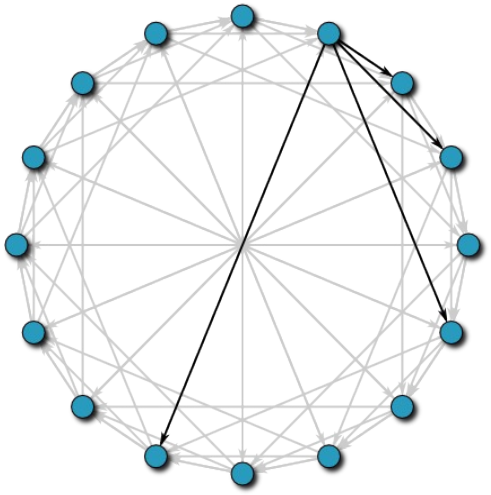
- What can an individual agent perceive?
- What actions can it select?
- Which collective patterns can you observe?

After watching:

What would you measure to compare this simulation with a real crowd?

Peer-to-peer networks





- *Aim*: giving access to resources located on the network nodes
- *Working principle*: distributed indexing or discovery algorithms, and/or repository necessitating a collaborative activity of agents/nodes
- *Multi-agent feature*: protocols implemented by interaction rules, open systems, autonomous agents (sometimes heterogeneous)

Multi-robot systems

Robocup



Robocup rescue



- *Aim*: coordinate a mobile robots fleet
- *Working principle*: local decision-making contributing to an overall goal achievement
- *Multi-agent feature*: locally noised and limited perception, located environment, distributed planning

Modelling a multi-agent system

Four perspectives: Agents, Environment, Interactions, Organisation

A MAS is made up of [Demazeau, 95]

- **A**gents (2)
- **E**nvironements (1)
- **I**nteractions (3)
- **O**rganizations (4)

AEIO 1/4: Environment

What does the environment contain?

- Physical or virtual entities and resources.
- A state that changes over time.
- Rules governing the effects of actions.

What can each agent perceive and do?

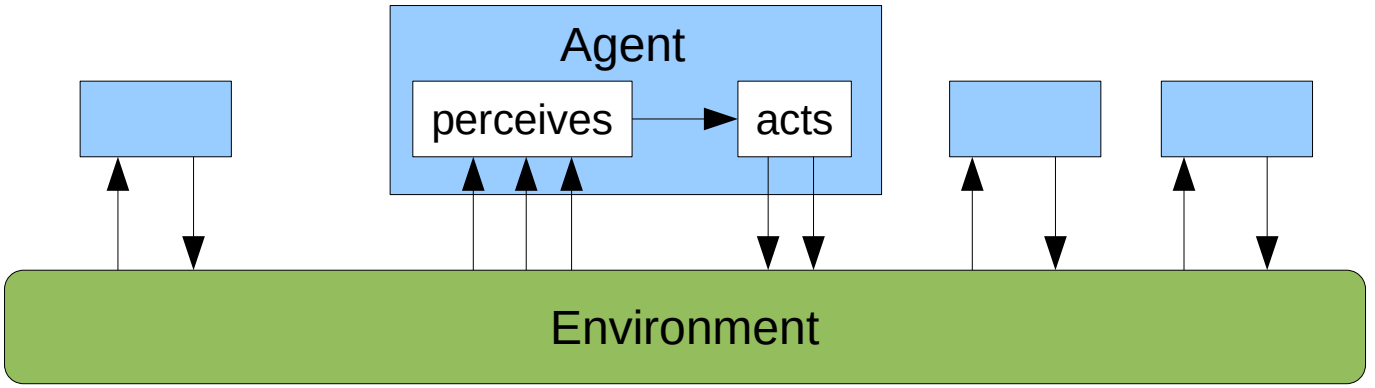
- Available information and perceptual limits.
- Possible actions and their constraints.

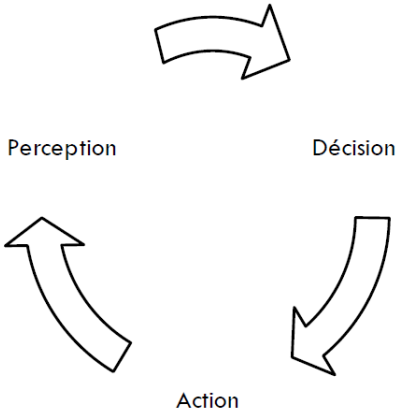
Example: a robot perceives only part of the field and acts through movement or ball handling.

AEIO 2.1/4: Agent architectures
Reactive agents

- Select actions in response to current perceptions.
- Use direct perception-action rules.
- Do not explicitly plan a sequence of future actions.
- May maintain a limited internal state.

Collective behaviour depends on local rules, interactions and the environment.





Difficulties: MAS + settings

Reactive control cycle

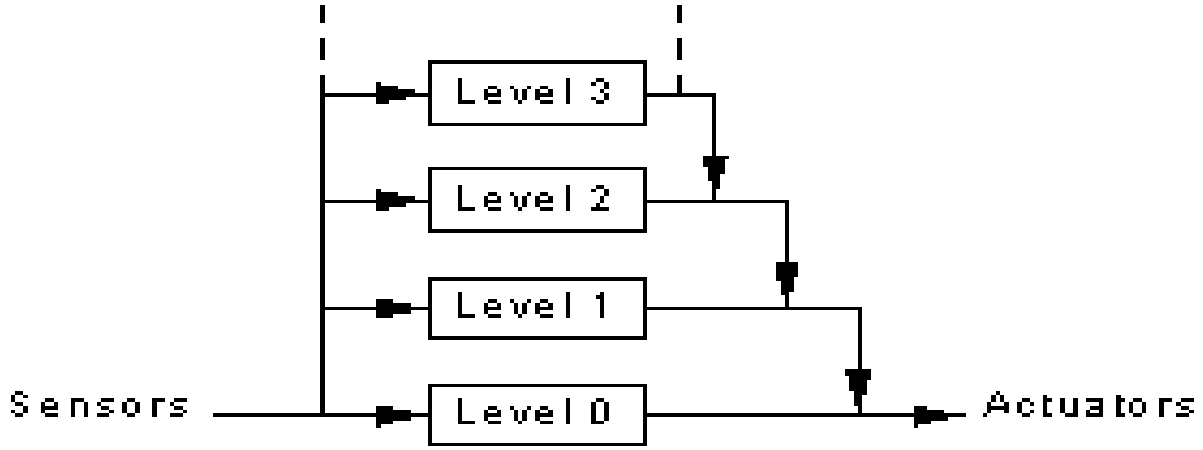
Data:

- Rules “condition => action”
- Set of percepts

```
while (true) {
    percepts := see();
    situation := interpret(percepts);
    rule := match(situation,rules);
    execute(rule[action]);
}
```

The selected action depends on the perceived situation and the available rules.

Example: Subsomption architecture [Brooks, 96]



- Each layer implements a behaviour.
- Higher layers can suppress inputs or inhibit outputs of lower layers.

AEIO 2.2/4: Agent architectures
Cognitive agents

Deliberative agents

- Maintain explicit representations of the environment.
- Represent goals and possible actions.
- Reason about what to do before acting.
- May construct or select plans.

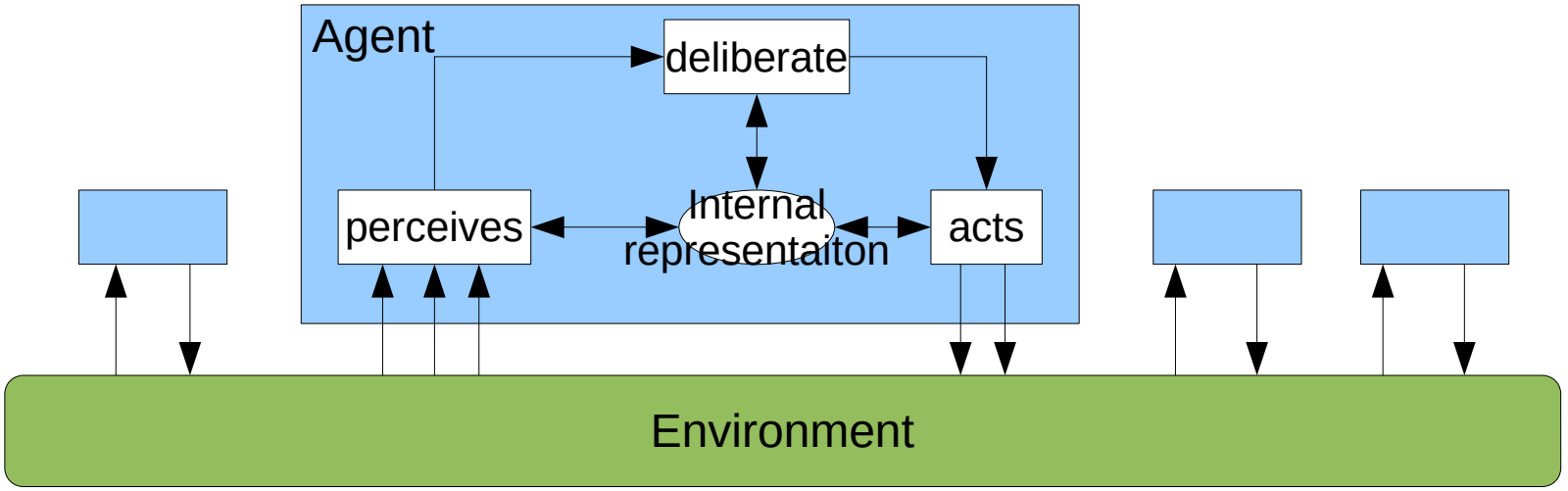
Modelling question: what information and reasoning capabilities do we assign to each agent?

Difficulties: agent architecture + communications

- AHMI: Human - cognitive agent interaction

Cognitive agents

- Perceptions update an internal representation.
- Goals and this representation guide action selection.
- Memory can support reasoning and planning.



Deliberative control system

Data

- Set of states
- Set of percepts
- Set of actions

```
states := initialise_state();
while (true) {
    percepts := see();
    state := update_state(percepts)
    action := deliberate(state, goals);
    execute(action);
}
```

AEIO 3/4: Interactions

Indirect interaction

- Agents affect one another through changes in the environment.
- Example: leaving a trace that another agent can perceive.

Direct interaction

- Agents exchange messages.
- Messages may follow an interaction protocol.
- Interaction is sometimes considered as an action

Both mechanisms can occur in the same system.

Which traces would reveal these interactions?

AEIO 4/4: Organisations

Explicit organisation: top-down approach

- Defined roles, rules and responsibilities.
- Constraints on agents' actions and interactions.
- Adaptation through changes in roles or rules.

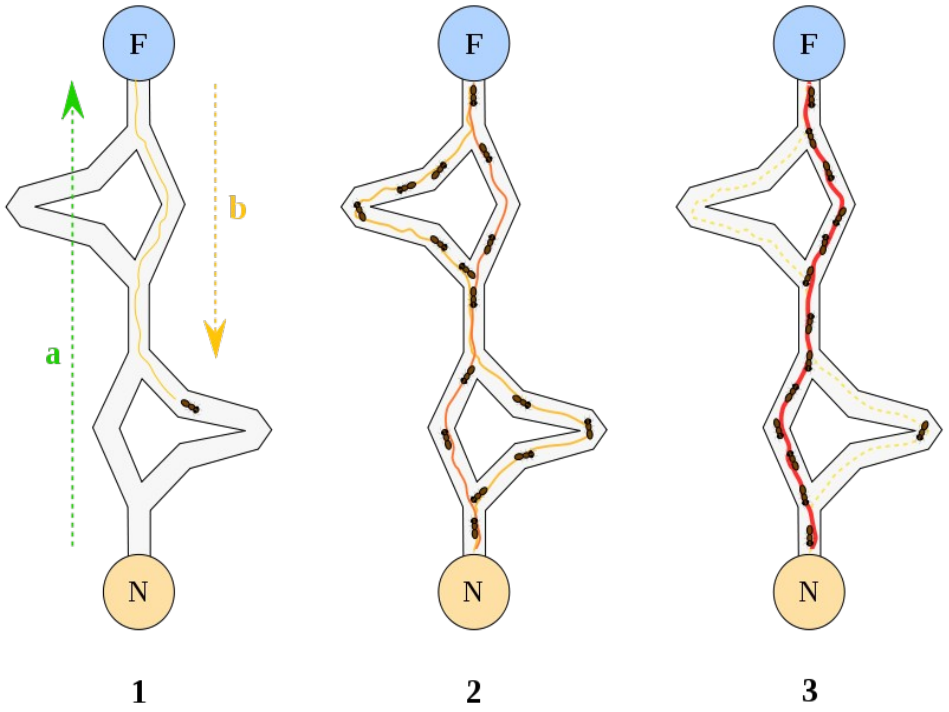
Emergent organisation: bottom-up approach

- Collective structures arise from local interactions.
- No complete organisation is specified in advance.
- Adaptation through changes in local behaviour.

A system may combine both approaches!

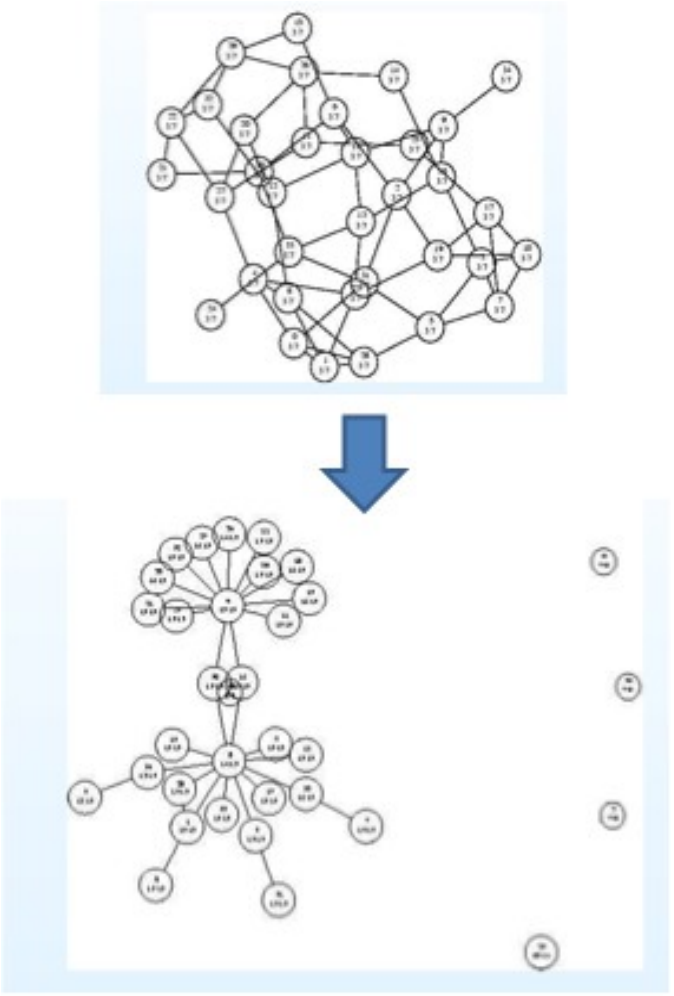
Self-organisation

Ant colony



https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms

Peer-to-peer network regulation [Grizard, 06]



Which local interactions could explain the collective structures?

To sum up, at agent level

At agent level: what does the model assume?

- What can the agent perceive?
- What information does it retain?
- How does it select an action? What can it do?
- How can it interact with others?

A software **agent** has several features

- It has its own decision ability (**autonomous)**
- **It can take action** on and **perceive** its environment (**pro-activity**)
- It **interacts** with other agents

To sum up, at MAS level

At system level: what do we want to understand?

- Which collective patterns arise from the interactions?
- How do the environment and organisation shape them?
- Which traces and measures describe these patterns?
- How can we compare the model with observations?

A **multi-agent system**

- is an **open** system
- that works in a **decentralised** manner
- is a **compromise** between **control/regulation** and **emergence**

Local rules are modelling assumptions

Collective patterns are outcomes to examine

A reactive approach:
Eco Problem Solving (EPS)

Problem solving (1/5)
Formalisation

Definition

A *problem* is defined by :

- an initial state,
- a goal (final state),
- a set of operators allowing to pass from one state to another.

$\Longrightarrow$ To solve a problem : give a sequence of operators allowing to pass from the initial state to the final one.

Representation

The states of the problem are represented by an oriented graph whose vertices are states.
There is an edge between nodes u and v iff there is an operator that turns u into v.

Problem solving (2/5)
Examples of addressed problems

Exemples

- Proving theorems, solving equations
- Cube world, tower of Hanoi
- Eight queens puzzle
- River crossing puzzle (wolf, goat and cabbage problem, Seven Bridges of Koenigsberg)
- Timetables
- Path search in a maze
- Magic squares, sudoku, crosswords
- Travelling salesman problem

Remarque

Most of "real" problems can be identified as one of the above formalisation.

Problem solving (3/5)
Exact solving

Principle

- Calculation of all possible paths going from the initial state to the final one.
- Exponential computational complexity
  $\Rightarrow$ Combinatorial explosion (memory and/or time).

Example : chess

- State space : all authorized configurations of the board
- Initial state : initial configuration of chess
- Final state : all configurations of the board implying checkmate
- Rules such as : "*if(white pawn is in square(i,2), and square(i,3) is empty, and square(i,4) is empty) then move white pawn from square(i,2) to square(i,4)*"

Problem solving (4/5)
Heuristic methods

Principle

Localized search view and stop after a certain amount of time.
⇒ Approximation of solution
⇒ Admissible quality solution, obtained in minimal time

Exemples

- Simulated annealing,
- Tabu search
- Evolutionary algorithms (or genetic algorithms)
- **Multi-Agent systems (eco-problem solving)**

Problem solving (5/5)

Problem solving and Human-Machine Interactions

Principle

Problem solving

- Dynamic
  - human intervention,
  - open or evolving environment
- with human interaction
  - cooperation,
  - collaboration,
  - competition

Exemple

- Timetable bargaining
- Games

Eco-problem solving (1/5)
Principles

- Used in problem solving
- Based on a set of reactive agents sharing the same behaviour, called *eco-agents*
- As for any MAS, decentralised solving, and emergence of a large-scale behaviour
- Also interesting for dynamic problem solving
- **Goal : reach a stable state (the problem's solution)**

Eco-problem solving (2/5)

Eco-agents

- Their behaviour can be compared to a sequence of perception/action instructions
- Localised environment perception : network of dependencies made up of neighbour agents
- Each agent has a goal : *satisfaction*
- An agent is dissatisfied when *obstructed* by other agents
- If an agent is obstructed, it can *attack* another one
- An attacked agent is dissatisfied and can *flee*

Representation with finite-state automata

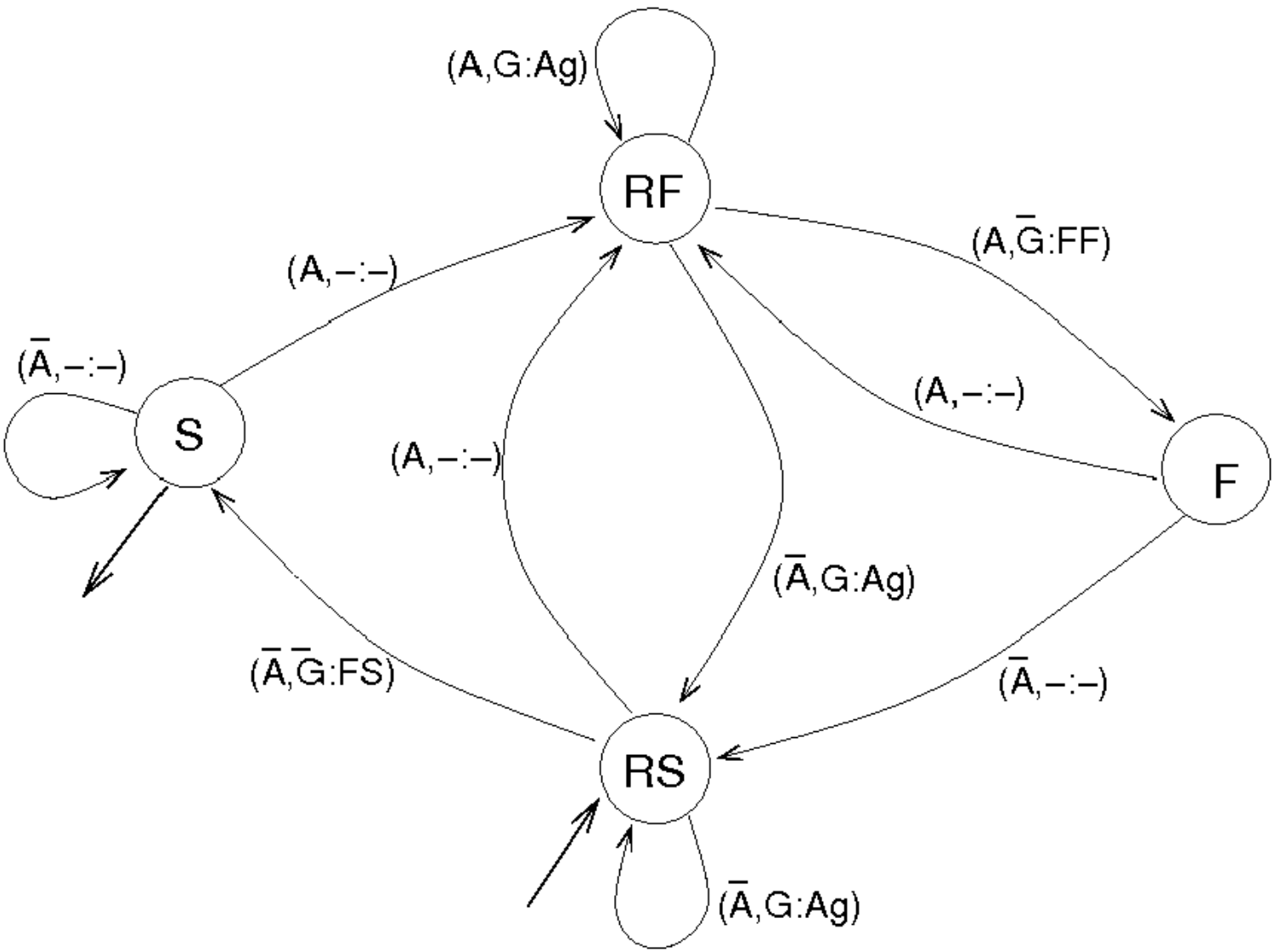
Eco-problem solving (3/5)

Eco-agents architecture

- 4 internal states
  - S Satisfaction (final state)
  - SS Seeking satisfaction (initial state)
  - SE Seeking escape
  - E Escape state
- Perceptions
  - $A/\overline{A}$ attacked/not attacked by an other agent
  - $G/\overline{G}$ obstructed/not obstructed by an other agent
- Actions
  - FS Do satisfaction (action achieving its goal)
  - AT Attack an agent
  - FF Flee

Eco-solving problem (4/5)

Eco-agents behaviour automaton



Eco-problem solving (5/5)

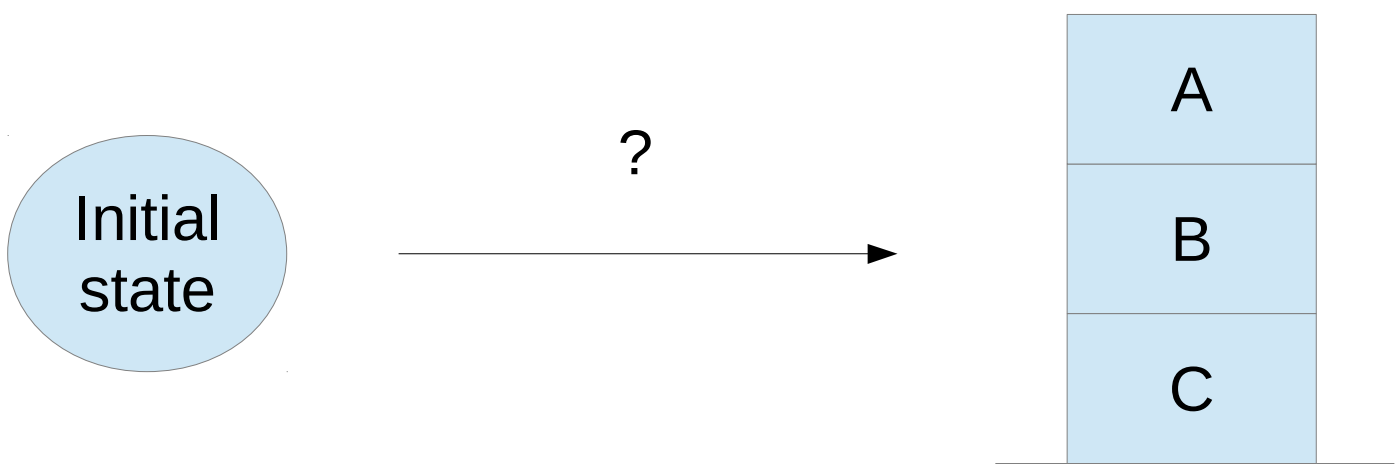
Solving eco-problems

- An eco-problem is defined as
  - A set of agents, featuring a *goal* and a *behaviour* (automaton + basic actions depending upon application)
  - An initial configuration, agents set into their initial state
  - An ending criterion (most of the time : all agents are satisfied)

- Solving a problem through eco-problem solving
  1. Identify the different types of eco-agents
  2. Identify the satisfaction conditions of each eco-agent
  3. Associate a structure describing eco-agents
  4. Specify methods `doEscape`, `doSatisfaction` and `Attack(X)`

Example : cube world

Problem description



From an initial state (a set of cubes on the table), find a sequence of actions to execute in order to reach a final state (a specific configuration of the cubes)

Example : cubes world

Solving through eco-problem solving

- 2 types of eco-agents : the table and the cubes
- Final state : cube A would be satisfied if on the table, cube B if on top of cube A, cube C if on top of cube B
  {A.on(Table), B.on(A), C.on(B)}
- To a cube, fleeing consists in being put on the table

Example : cubes world

Table eco-agent

Attributes

Methods

```
  Method satisfied?():boolean {
    return(True)
  }
```

Example : cubes world

Cube eco-agent

```
Attributes
  over: Cube
  under: EcoAgent (Cube or Table)
  goal: EcoAgent (Cube or Table)

Methods
  Method doSatisfaction() {
    this.goal.over := this
    this.under.over := null
    this.under := goal
  }

  Method doEscape(p: EcoAgent) {
    this.under.over := null
    this.under := p
  }

...
```

Example : cubes world

Cube eco-agent

```
...

  Method satisfied?(): boolean {
    if this.under = this.goal then return(True)
    else return(False)
  }

  Method findPlaceToFlee() {
    return(Table)
  }

  Method obstructorsFlee() {
    if this.over <> null then // return the obstructors list
      return(this.over)
    else
      return(null) // else nothing
  }

...
```

Example : cubes world

Cube eco-agent

```
...

  Method obstructorsSatisfaction() {
    EcoAgent r := null
    if this.over <> null then
      r := this.over
    else if this.goal.over <> null then
      r := this.goal.over
    return(r)
  }
```

Interaction with eco-agents

- Heterogeneous MAS
  - Reactive MAS, easily configurable
  - Mixed community : human / artificial agents
  - Indirect interactions
  - Seeking stable state, human disturbance

- Eco-problem solving through heterogeneous MAS
  - Features identical to a heterogeneous MAS made up of eco-agents and humans
  - Agents and humans cooperate to solve a problem

References

Links

- Link to lesson by C. Bertelle
- Link A. Drogoul

Books and articles

- J. Ferber, "*Les Systèmes Multi-Agents : vers une intelligence collective*", InterEditions, 1995
- K.P. Sycara, "*The many faces of agents*", AI Magazine, 19(2), pp. 11-12, 1998