MULTI-AGENTS SYSTEMS AND INTERACTIVE SIMULATION

Complex systems
& Agent-based modelling and simulation

Jérémy Rivière
jeremy.riviere@univ-brest.fr

Département d'Informatique
IHSEV Team, Lab-STICC

Plan

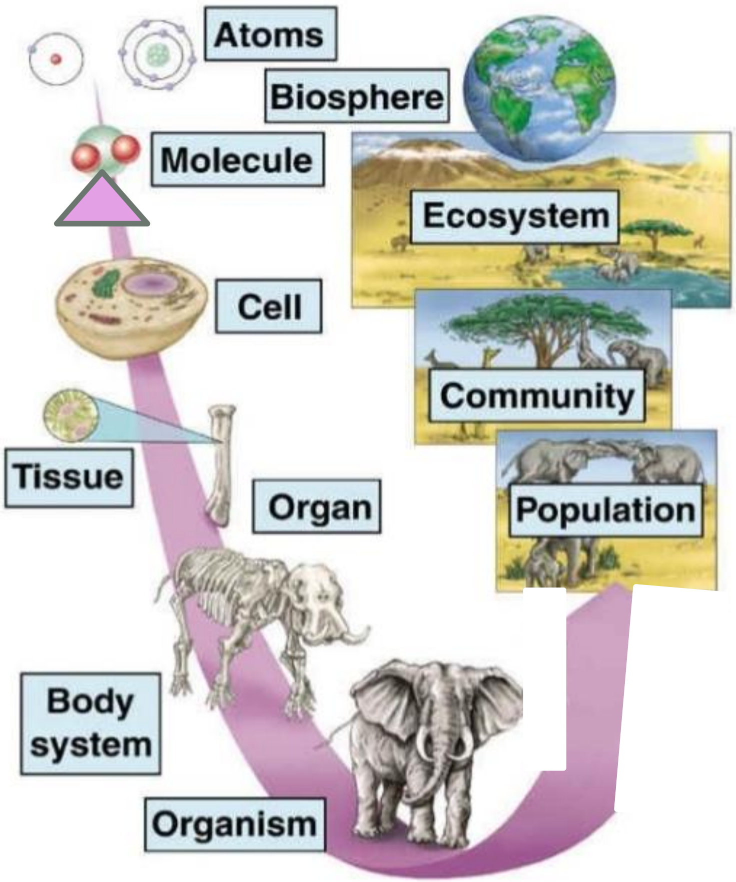
What's complexity?

Example:
the biologic complexity

- How to comprehend it?

⇓

Reductionism since 300 years



What's complexity?

Example:
the biologic complexity

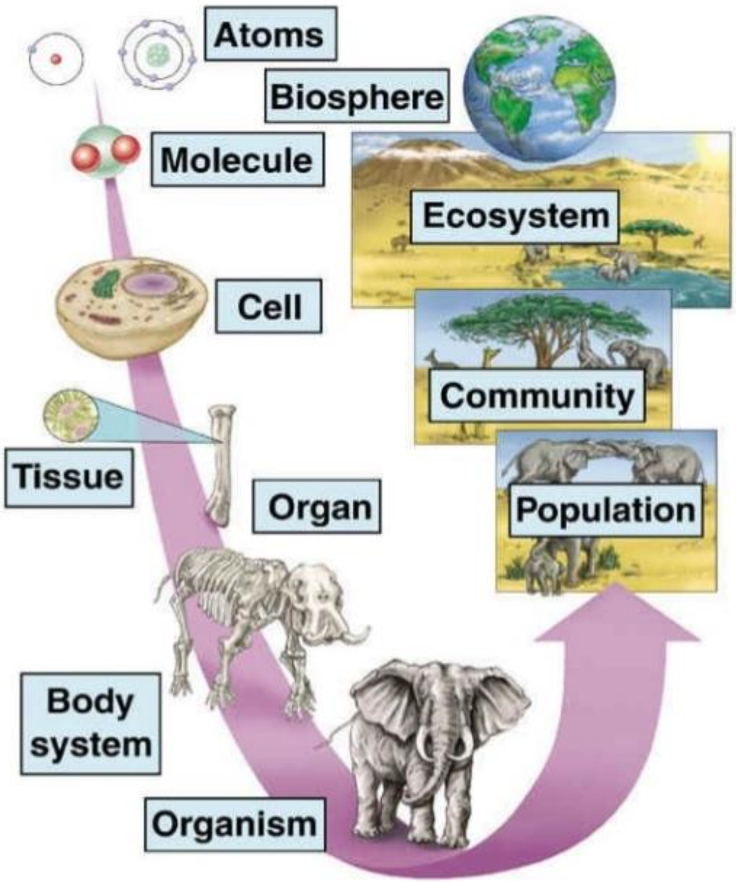
- How to comprehend it?

Reductionism since 300 years

- OK, but how do we go back up?

- It's **complex**:

self-organisation, interdependence, positive/negative feedbacks, *emergent* properties



What's complexity?

Another example: the **brain**

- about 10^{11} neurons
- interconnected by synapses, communicating by electrical and chemical impulses

Complexity: a scientific challenge

THE question that tries to answer the science of complex systems is:

How large systems

- made of (many) "simples" parts,
- with a limited communication between these parts,
- and without any leader or centralized control,

can exhibit **adaptive**, **organised**, **emergent** - well, complex - behaviours?

« *The whole is greater than the sum of its parts* » - Aristote

A concrete question

Why?

A few hypothesis:

- Predators
  - only one (large and threatening) organism
  - harder to target and hunt one individual among thousand
- More effective as predators (cooperative hunt)
- Improve the aerodynamics! (Tour de France!)

And more, non-exclusives, hypothesis

How? (and how to verifiy it?)

We know that 1) there is no leader and 2) no global knowledge

"Boids" **Model** *[Craig Reynolds, 1987]* :

Inspired by **particle system**, used in animation of complex phenomena (e.g. clouds, fire):

- collections of individual particles
- each one has its own state (position, velocity, lifetime …)
- particles' behaviour is defined w/r/t its state

Reynolds' idea is that boids can be seen as particles, that **have to be influenced by the others** to flock in a coherent manner

The boids' behaviour is defined w/r/t its state <u>and</u> its interactions

How? (and how to verifiy it?)

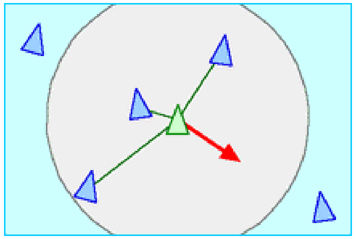
We know that 1) there is no leader and 2) no global knowledge

"Boids" **Model** *[Craig Reynolds, 1987]* :

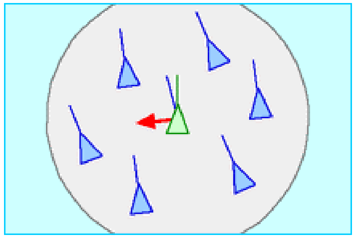
Only 3 rules to define the **behaviour** of each bird

(in order of importance)

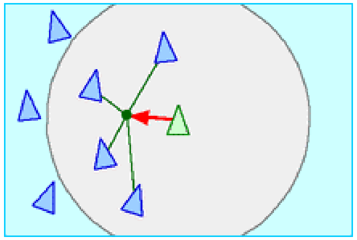
1. Separation: steer to avoid crowding local flockmates.



2. Alignment: steer towards the average heading of local flockmates.



3. Cohesion: steer to move towards the average position of local flockmates.



How? (and how to verifiy it?)

We know that 1) there is no leader and 2) no global knowledge

"Boids" **Model** *[Craig Reynolds, 1987]* :

Only 3 rules to define the **behaviour** of each bird

(in order of importance)

1. Separation: steer to avoid crowding local flockmates.
2. Alignment: steer towards the average heading of local flockmates.
3. Cohesion: steer to move towards the average position of local flockmates.

Flocking.nlogo

And then, what? **Computer simulation**, with for ex. **NetLogo** *(http://ccl.northwestern.edu/netlogo/)*

Experiment the model, test hypothesis, investigate parameters

To resume

Complex systems are composed of **large numbers (and diversity) of individuals/entities**, situated in a shared environment, and that can be seen as **autonomous**, **interconnected** and **interacting** between them and the environment **without any centralized control**.

Difficult (if not impossible) **to explain** their essential functional global properties by simply describing a limited number of parameters or characterizing variables

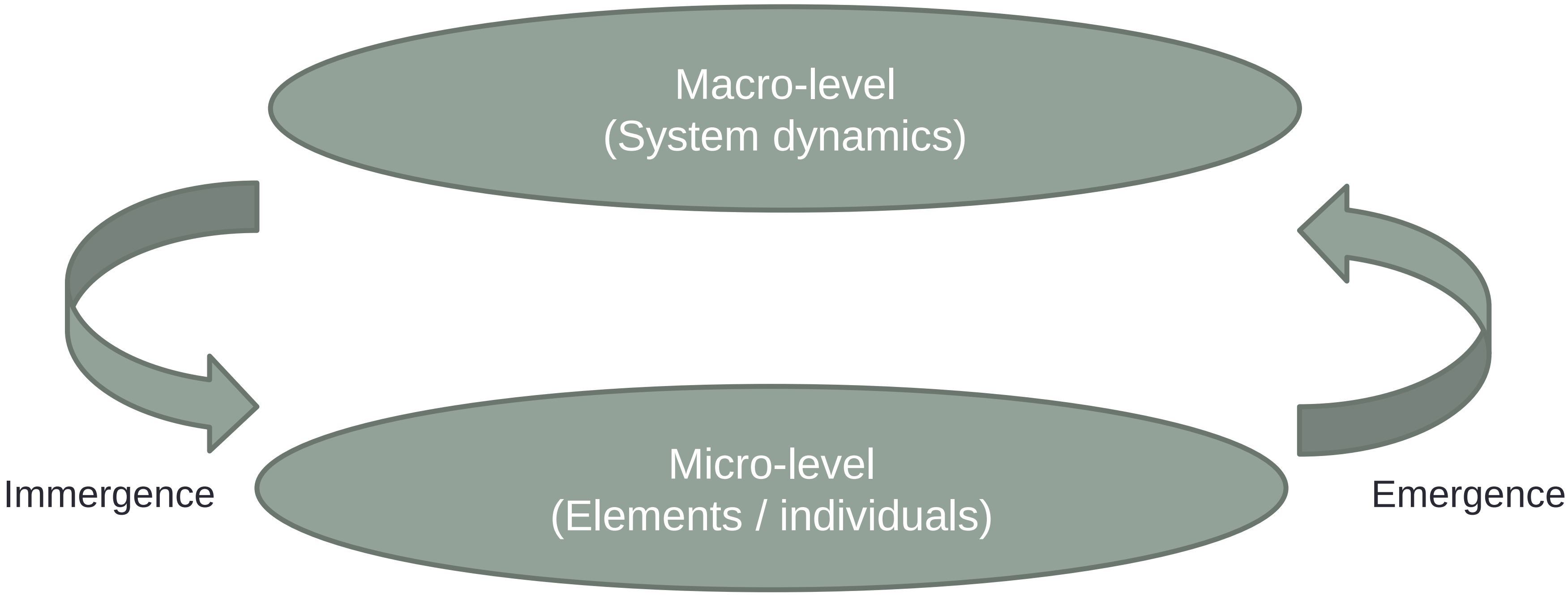
They exhibits *emergent* properties, i.e. properties that result from the interaction between the components
(not the properties of the components themselves).

Fundamental characteristics

- **Emergence**
  - Emergence is the process of deriving some new and coherent structures, patterns and properties in a complex system
  - Emergent phenomena occur due to the non-linear and distributed interactions between the elements of the system
  - Emergent phenomena are observable at a **macro-level**, even though they are generated by **micro-level** elements

Fundamental characteristics

- **Emergence**
  - The system (macro) has properties that the elements (micro) do not have
  - The emergent phenomena can also feedback/influence the elements
    e.g. traffic jam influences drivers, stock market influences buyers…



Fundamental characteristics

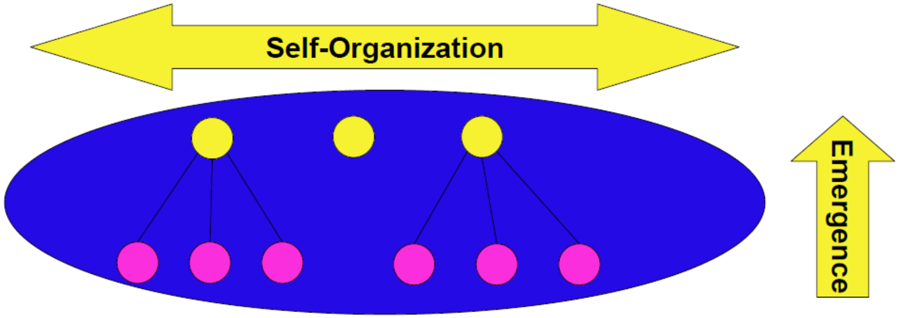
- **Emergence**
- **Self-organisation**
  - Changes to the internal order or organisation of a system without guidance or management from an outside source
  - Neural networks, flocking behaviour, natural section (characteristics that support survival become more common in the species) …

  - Self-organisation by **local interaction** and **feedback loops**
  - direct: entity <-> entity
  - indirect: entity <-> environment <-> entity (stigmergy in social insects)

  - Often confused with emergence!

  e.g. division of labor, flock

AgressorDefender.jar



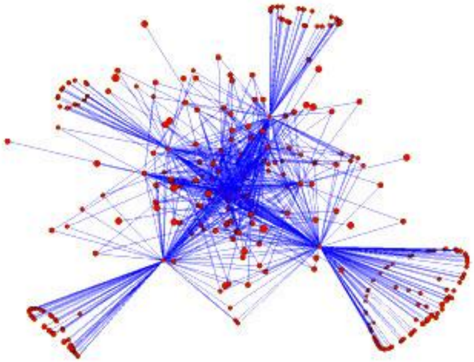
Fundamental characteristics

- **Emergence**
- **Self-organisation**
- **Self-adaptation**
- **Decentralisation**
- **Non-linearity** (random, path-dependance)


Fire.nlogo

- « **Butterfly effect** » : small actions, big consequences


Sea otter and kelp forests



https://www.youtube.com/watch?v=ysa5OBhXz-Q

Fundamental characteristics

Example: Ant nests and "highways"

- Emergence of structures and global behaviour: shortest path in foraging
- Decentralization
- Ant self-organization: roles in the nest, foraging, nest building...
- Self-adaptation of the system in case of disruption (attack, famine...)
- Non-linearity (e.g. in random food search)

Complex systems and other sciences

Complex systems are found in all sciences

- Introduce **universal principles** that can be used to **describe and solve problems** from particle physics to economics, biology, chemistry, social sciences…
- Examples?
- They therefore make the **link** between chemistry, physics, biology, anthropology, sociology, computer science...

→ Transfer of ideas and results between these different research fields

cnrs

Institut des Systèmes Complexes Paris Île de France

COMPLEX SYSTEMS SOCIETY

Understanding complexity
The models' role

Jacques Tisseau

« Describing, explaining, predicting and simulating the behavior of natural or artificial complex systems is, for scientists, one of the major challenges of the 21st century »

- The understanding and/or implementation of a complex system depends on **modeling**
- Study of these models through **simulation**
- ***in virtuo* simulation**: **visualise** the model's behaviour and **interact** whit the simulation

Plan

What's a model?

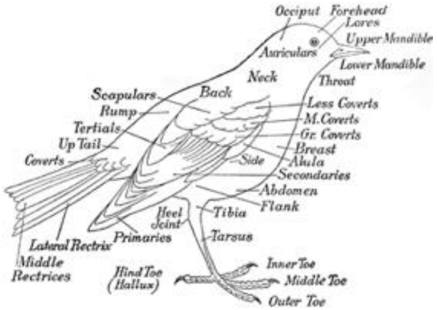
- A model is a simplified representation of reality, a tool for reflection
- One (or several) aspect(s) of the real system
- Exemples : maquette, drawing, robot, equation …

Real bird

MODELLING

Flying maquette

Digesting robot



Schematic description

" Modelling" and "simulating"

- Modelling is

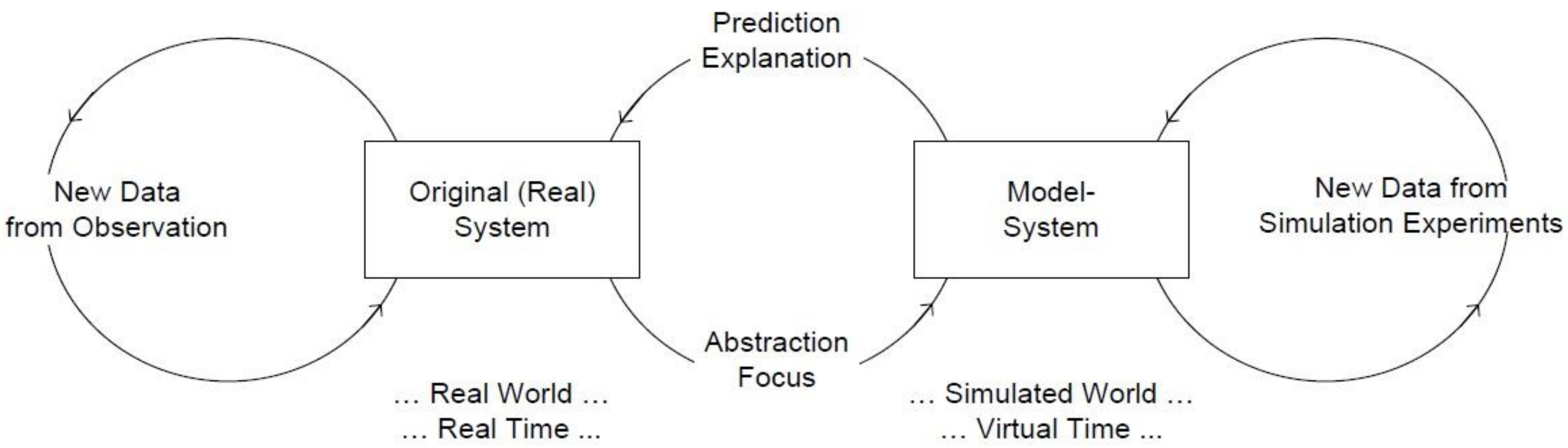
- taking a biased view of <u>one</u> representation of reality
- an abstract process
  - without computational aspects (*e.g.* time management)
  - without technological aspects (*e.g.* platform, language)

- Simulating is to experiment with the model by "immersing" it in time, with the aim of

- understanding the real system
- testing new hypothesis
- predicting the evolution of the real system
- calibrating and validating the model in an experimental approach

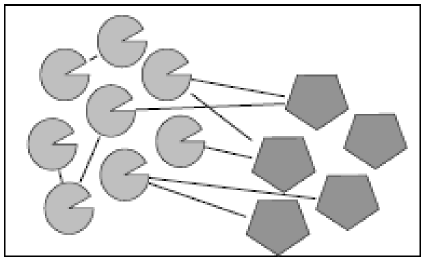
time

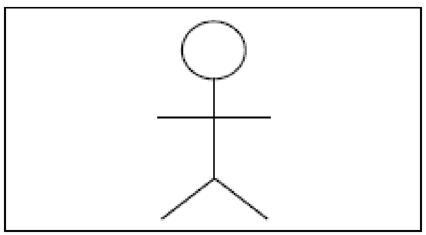
" Modelling" and "simulating"

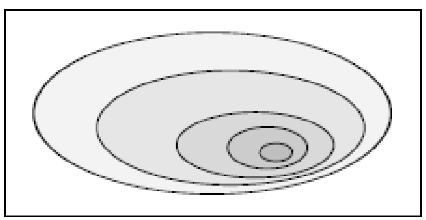


From [F. Klügl 2016]

Concretely
Modelling paradigms







$$y = f(x)$$

$$dz/dt = F(x,y)$$

Models	Global behavior of the system
Agent-Based Models (ABM, IBM) *"bottom-up"*	Results of micro level dynamics (interactions and agents' behavior)
Participatory models played by several experts	Given by the experts
Expanded models Known configurations based on dataset	Given by the data
Population models Or Equation-Based Models (EBM) *"top-down"*	Given by the equations

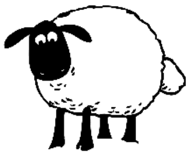
+ hybrids!

Concretely
top-down vs. bottom-up

Example: predator-prey dynamics

- Predators:
  - feed on prey
  - reproduce
  - starve to death

- Prey:
  - feed on grass
  - reproduce
  - die of old age



How evolve the two populations?

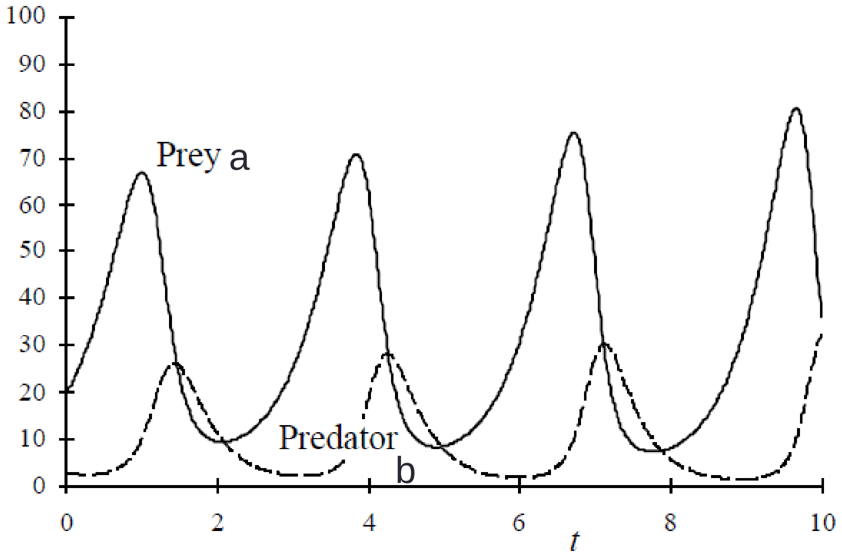
Concretely
top-down vs. bottom-up

Example: predator-prey dynamics

- « ***Top-down*** » approach: differential equation systems

$$\frac{da}{dt} = \lambda_a a - \gamma ab$$

$$\frac{db}{dt} = -\mu b + \lambda_b \gamma ab$$



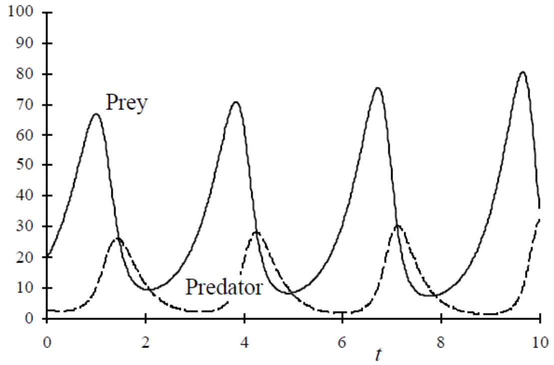
Concretely
top-down vs. bottom-up

Example: predator-prey dynamics

- « ***Top-down*** » approach: differential equation systems

$$\frac{da}{dt} = \lambda_a a - \gamma ab$$

$$\frac{db}{dt} = -\mu b + \lambda_b \gamma ab$$



Does not take into account:

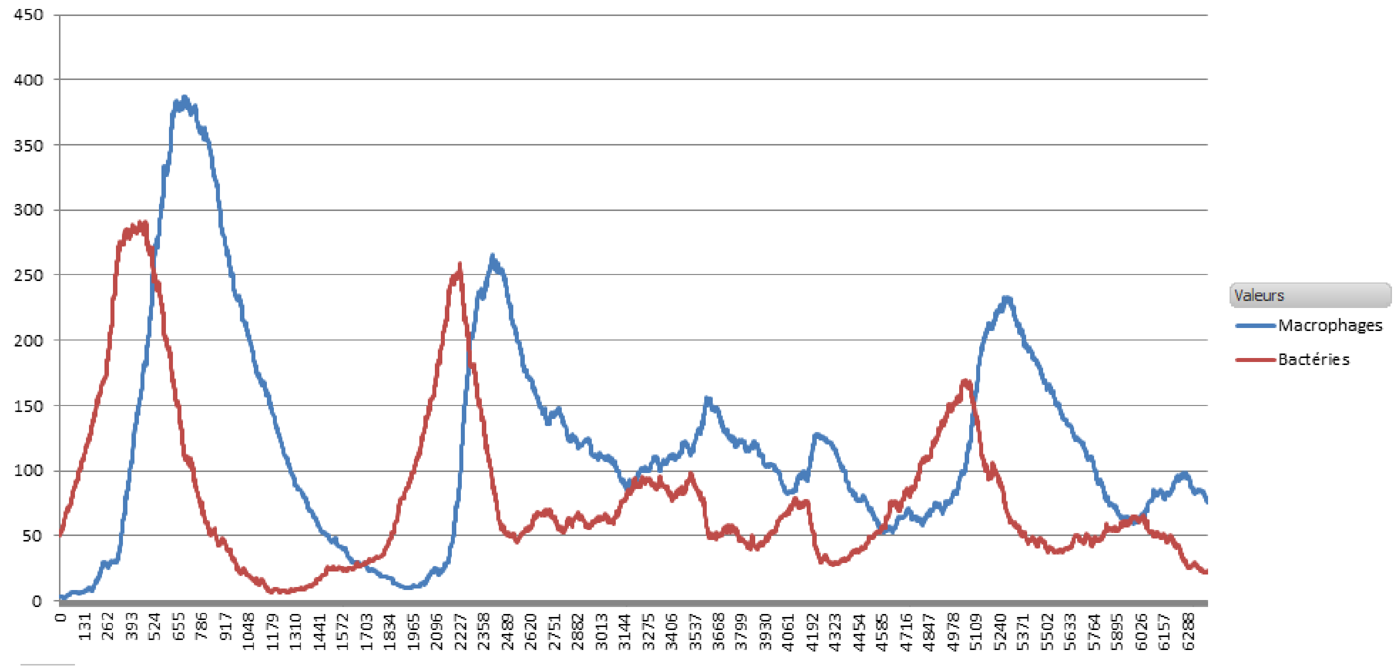
- The space/environment (location of prey, grass, fences, dry periods...)
- The diversity of individuals
- The non-deterministic nature of real life

Does not allow to explain what are the different mechanisms that make up emergent phenomena

Concretely
top-down vs. bottom-up

Example: predator-prey dynamics

- « ***Bottom-up*** » approach: **ABM**
  - Prey and predator as agents
  - Interactions
  - Environment
  - Randomness!



Why model and simulate…

… <u>in general?</u>

- The part of the real world that should be studied is not accessible
- Experimenting with the real world is prohibited
- The time scale of behavior or system size may prohibit observation either because its dynamics advances too fast or system size is too small
- The simulated model and its environment are perfectly controllable (parameters, variables …)
- Putting a scientific theory into practice through modelling then through virtual experimentation
- Predict, approximate by simulating the model

Weather forecasting -> climate model

Why model and simulate…

… <u>complex systems</u>?

- The only way to understand, descibe, explain how emerging, self-* properties arise

« *I can't understand what I can't create* » Richard Feynman

- Create artificial systems that include swarm algorithms, self-organising properties (e.g. task allocation) …
- … and **the human in interaction** with the system

Modelling: 5 General Principles

[Pidd, 1996]

1. **Model Simple, Think Complicated**
   A model does not have to resemble all the complexity of the original system.
   Models -> tools for thinking.

2. **Be Parsimonious : Start Small and Add**
   To be applied when it is not clear how simple a model of a system can be. KISS

3. **Divide and Conquer : Avoid Mega-Models**
   Large models are hard to validate, interpret, calibrate and explain.
   Use a component-based design that allows to tackle parts of the model separately as modules.

4. **Do not Fall in Love with Data**
   The model should drive the data collection and not vice versa.
   Start with a simple model and calibrate/validate it with available data; if the model needs to be refined, then additional data collection may be necessary.

5. **Model Building May Feel Like Modeling Through**
   Modeling is not a linear step-by-step, nor a smooth process.
   90% model assessment, problem definition… 10% model implementation

Strategies for Iterative Model Development KISS

"Keep It Simple, Stupid"

- Keep the model as simple as possible for generating the appropriate behavior
- Simplicity also refers to the modeling and simulation paradigm used
- Not trivial to see how to modify a model for producing any additional phenomena

1. Identify and describe the set of observable properties (statements) about the real system S.
2. Define a model M_0 that is apparently too simple for reproducing the system with all its properties
3. By calibration, determine the set M_S of properties, that are reproduced by M_0.
4. M <- M_0
5. Repeat Until M_S = S
   a) M <- modify model M for producing more elements in S than in the last iteration.
   b) Calibrate M and determine M_S as the set of properties reproduced by M.

Strategies for Iterative Model Development KIDS

"Keep It Descriptive, Stupid"

- « The point is that it is simply not appropriate to make simplifications **before** one knows what is relevant and what not ».
- Models of complex systems must sometimes contain a certain level of detail to be creditable.
- What to do if the model is not sufficiently valid in terms of aspects that are not reproduced ?

1. Repeat until a valid model Ms is constructed
   a) Define a model M that contains all apparently relevant aspects of agent behavior
   b) Identify all assumptions and make explicit all parameter in M
   c) Execute a sensitivity analysis for all parameter of M and eliminate all blocks of behavior that are controlled by a parameter without effect on the overall outcome.
   d) M_s <- M
   e) Test M_s for credibility and validity

Strategies for Iterative Model Development TAPAS

"Take A Previous model, Add Something"

- Pragmatic, focusses on existing **reusable** models
- Related to the KIDS-based strategy but takes an existing model as a starting point
- Selection of the model: reusability, documentation, model description, understandable implementation, data availability …

1. Select an appropriate existing model M
2. if M is not implemented, implement it and validate it using model alignment with respect to published data about M.
3. Add new, additional aspects to produce M_{add}
4. Test and Validate M_{add}

if sufficient, ready, else go back to 3 or - if necessary to 1.

Modelling and Simulating
Global process

Step 1: Modelling

1. Starting point: **a scientific question**
   - problem and positioning
   - a targeted system
   - data (for model **calibration** and **validation**)
   - a scientific hypothesis

2. A theoretical framework

3. Observed phenomena, patterns, properties that the model will have to reproduce

4. Model **verification** (experts)

-> A conceptual model, according to a paradigm, of the targeted system

Modelling and Simulating
Global process

Step 2: Simulating

1. Simulation questions
   - What are the expected answers?
   - Which data to use to compare to the simulation output?
2. An implementation of the model
3. An experimental protocol
4. Analysis of parameter bias and sensitivity
5. Calibration

Wolf Sheep Predation.nlogo

Modelling and Simulating
Global process

Step 3: Results analysis

1. **Which representation** of the results?
   - Evidence of the production of phenomena and expected simulated patterns

2. **Model validation**

3. **Exploitation** of the results**: refutation or validation of the hypothesis**

4. **Reformulation** of the initial scientific question

5. **Enrich the theoretical framework** with new hypotheses

6. **Publish** a scientific article!

Description du modèle ??

Agent-Based Models
When to use them?

1. Agents are a natural metaphor
   - Easy to identify agents as individual entities of a system

2. Individual (micro) behaviors are simpler than the behavior of the overall system (macro)

3. **Numerous** and **strong interactions** between entities

4. The entities' behaviour depends on the **context**
   - Local properties of the environment, neighborhood, internal state...

5. Focus on the **complex** properties of the system
   - Links between processes at different scales (micro-macro)
   - **The aim is to explain** the mechanisms behind these properties

Agent-Based Models
When to use them?

Example: segregation

- In Los Angeles, despite a weak trend towards segregation among residents of different neighborhoods, well-defined ethnic neighborhoods are emerging. How to **explain** it?

1. Residents -> agents
2. Segregation -> interactions between neighbors
3. Hard to explain the macro behavior -> micro is simpler
4. Context -> neighborhood
5. The aim is to explain

-> Agent-Based approach

Agent-Based Models
When to use them?

Example: segregation

- In Los Angeles, despite a weak trend towards segregation among residents of different neighborhoods, well-defined ethnic neighborhoods are emerging. How to **explain** it?

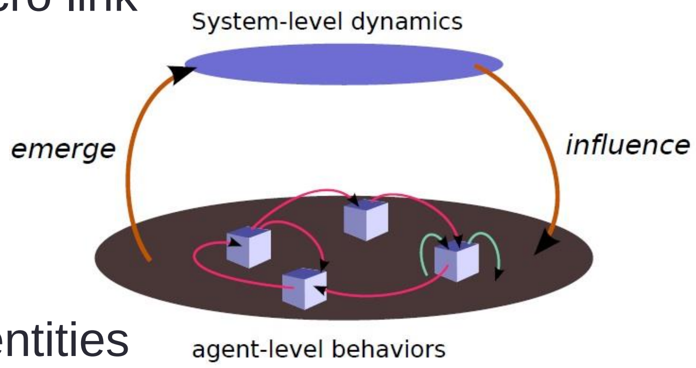
Thomas Schelling's model, 1971 :

- A preference of 30% for its neighborhood is enough to reveal distinct neighborhoods.

Segregation.nlogo

What is the focus of the ABM approach?

- Influence of individual behaviour on the system (emergence) and of the system on individuals (immersion) - macro-micro link



- Inter-individual **variability**, **heterogeneity** of entities
- **Local** interactions
- **Life cycle** of individuals
- **Plasticity** of behaviour, the evolution of entities

Plan

1. System complex: an Introduction

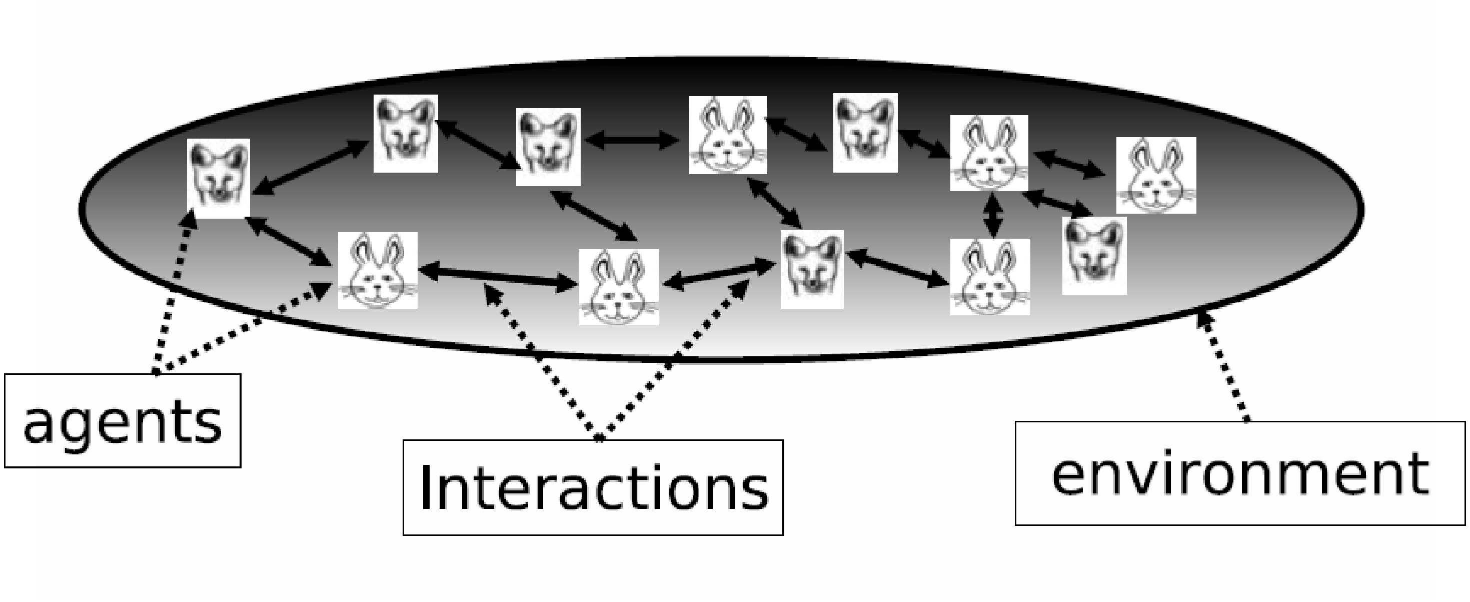
2. Modelling and Simulation

3. Multi-Agent Systems for complex systems modelling and simulation
    1. MAS: reminder and definition
    2. The MAS approach for modelling: principles and case study
    3. Modelling time in simulations

4. Implementing ABM
5. Visualisation and Interactions with an Agent-Based simulation
6. Issues and challenges in Agent-Based Models

Agent definition in MAS

An **agent** is seen as an **autonomous** entity, which can **perceive** and **act** in its **environment**, and which is able to **interact** with other agents



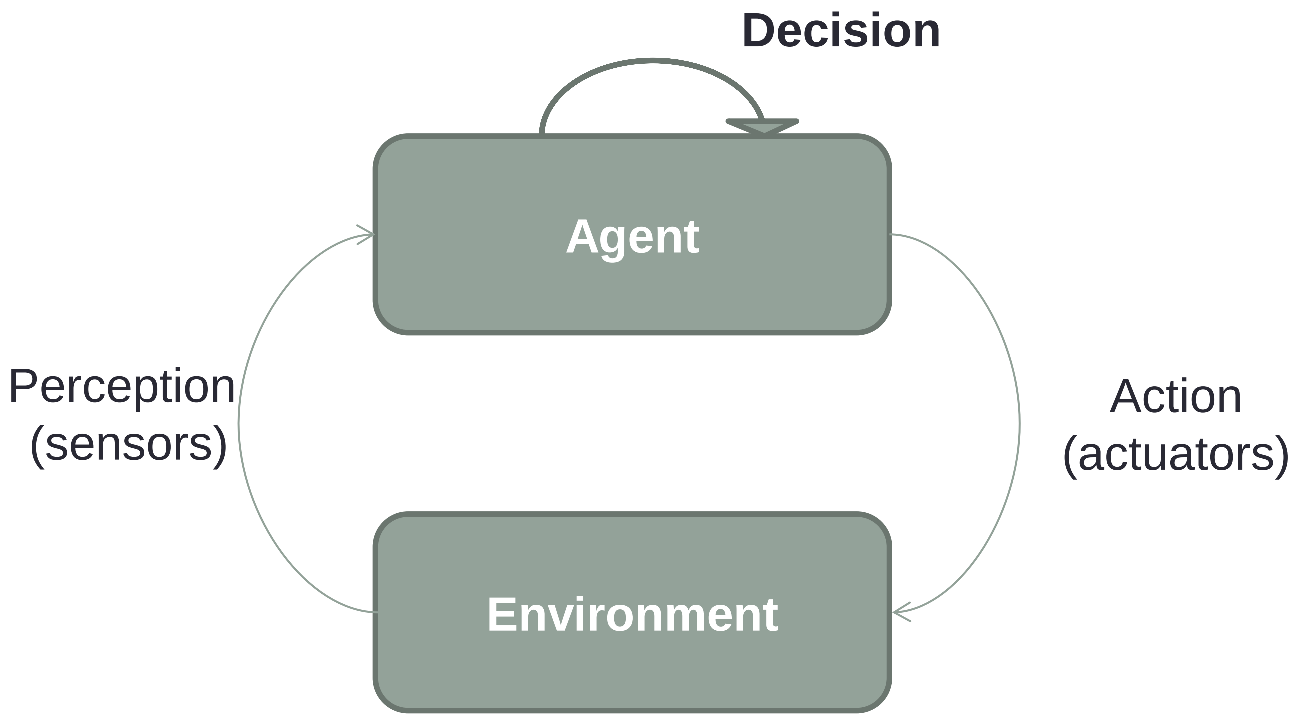
[Michel et al., 2009]

Agent definition in MAS

4 main concepts

1. **Autonomous activity** of the agent

- Able to carry out an action on its own initiative (proactivity)
- Able to decide



Agent definition in MAS

4 main concepts

1. **Autonomous activity** of the agent
2. **Sociability** of the agent

   An agent in a MAS is not an isolated entity, but an element of a society

3. **Interaction**

   Connects the concepts 1 and 2:

   Each intention to interact is decided inside an agent's mind, and creates social forms (cooperation, competition, conflict …)

4. **Situatedness** of the agent

   The fact that agents are placed into an environment

   Defines the conditions in which the agents exist, act and interact

   The glue that connects the agents together

Perception and action capacities
Complex environment

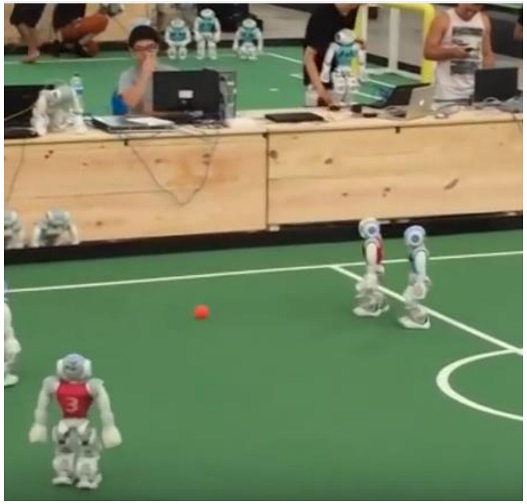
A (physical / virtual / social) **complex** environment is:

- partially known to the agent (*vs.* fully accessible)
- stochastic (*vs.* deterministic)
- dynamic (*vs.* static)
- continuous (*vs.* discrete)
- open (*vs.* closed)

[Russel & Norvig, 2003]

-> An agent has limited perception and action capacities

- No omniscient entities
- A limited repertoire of actions
- Agents with different abilities
- A limited amount of time and resources to make a decision (e.g. within a physical environment)



Interaction capacities

Interaction as actions

- between agents (including humans)
- between the agent and the environment (influence and reactions)
- from and into the environment

Different types of interactions

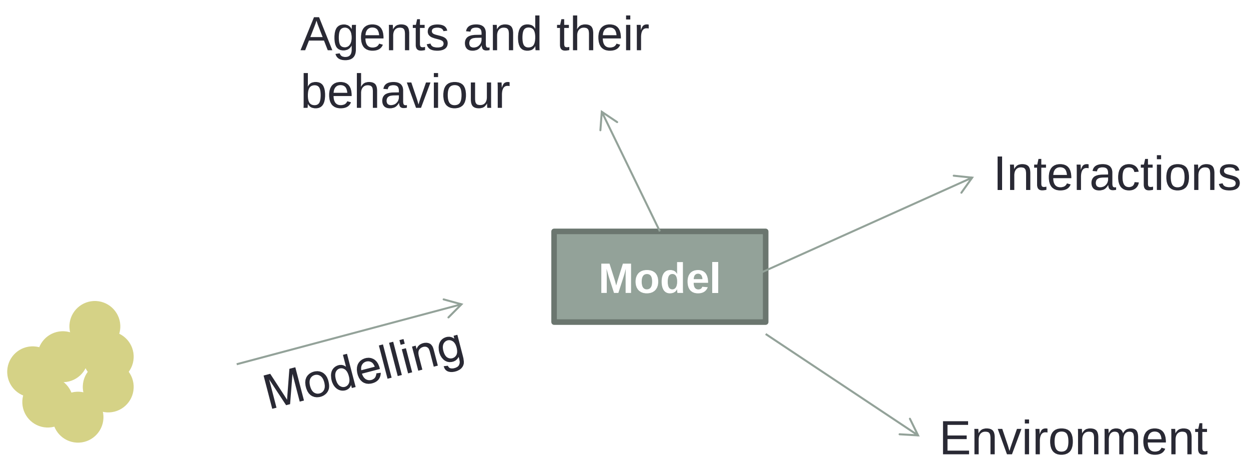
- Direct or indirect communication
- Situations of collaboration, cooperation, competition

Organization of agents

- Induces constraints, rules on interactions
- Example: Simulation of a bee colony

MAS Modelling
Principles and case study

Modelling a complex phenomenon using a multi-agent approach means identifying:

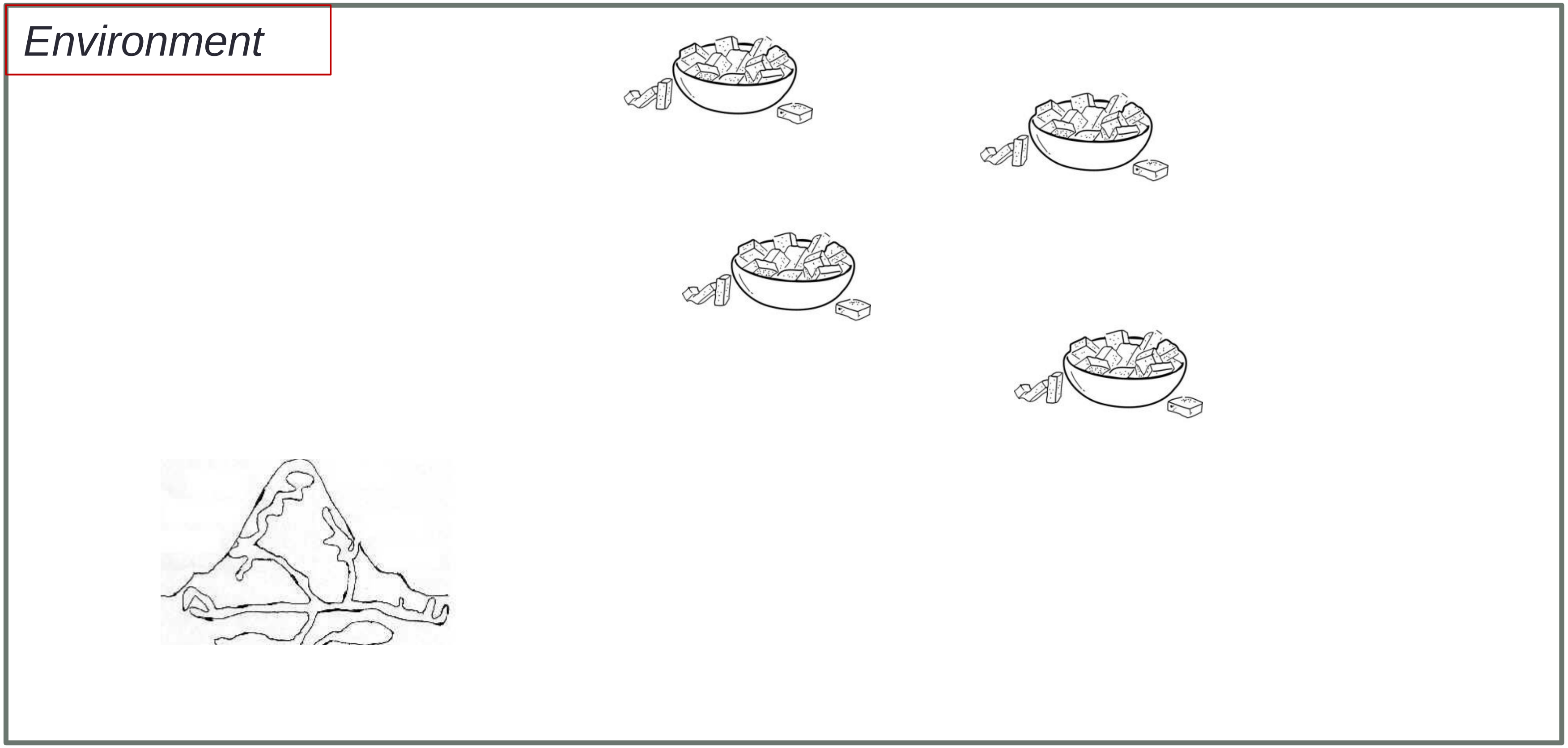


MAS Modelling
The example of ants « highway »

- *How can we **explain** the formation of "highways" of ants from the nest to a food source?*

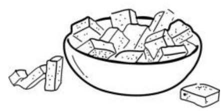
- Biology data and observations
  - Observed behaviour of ants
  - Perception and action skills
  - Indirect and simple communication by pheromones
  - Evaporation of pheromones over time

MAS Modelling
The example of ants « highway »



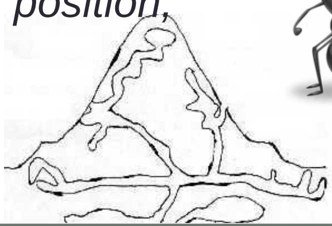
MAS Modelling
The example of ants « highway »

Environment

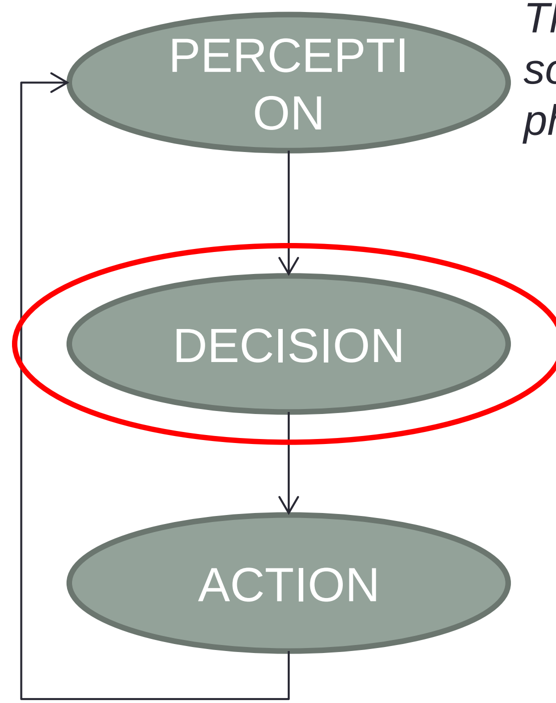


One agent <-> One ant

properties:
Carrying?,
Nest position,
…



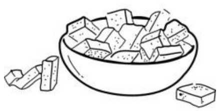
The nest, food sources, pheromone



Moving, Collecting / Depositing food, Depositing pheromone

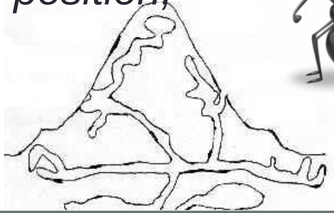
MAS Modelling
The example of ants « highway »

Environment



One agent <-> One ant

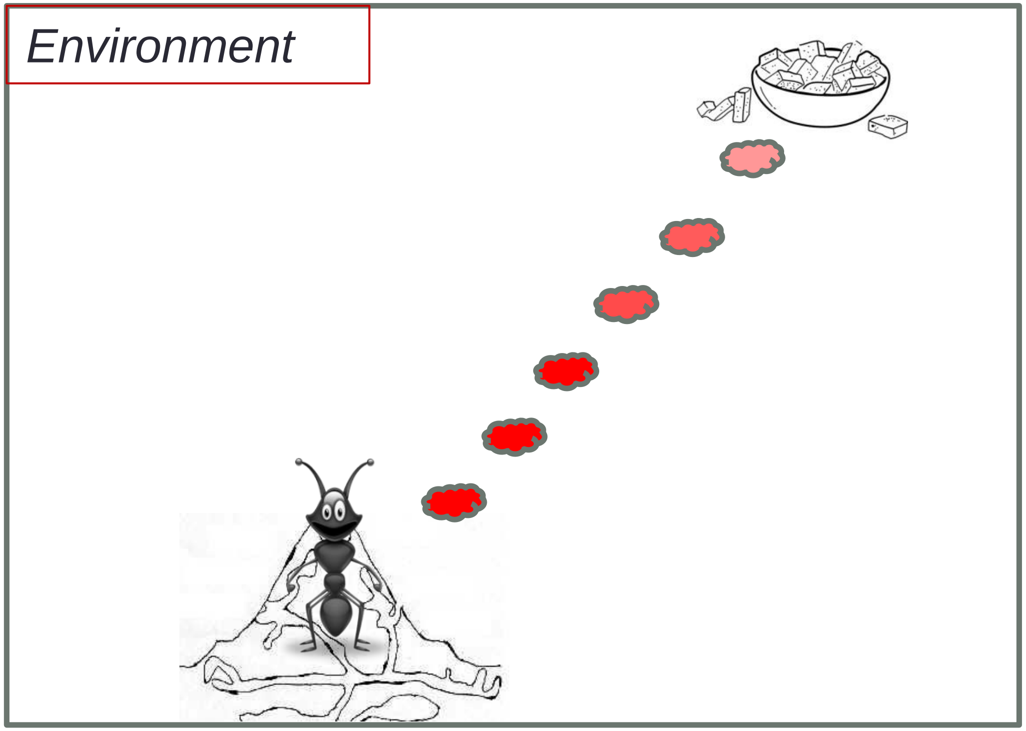
properties:
Carrying?,
Nest position,
…



Objectives
Relate the agent's perceptions (and states) to the actions he can take

-> Decision rules

MAS Modelling
The example of ants « highway »



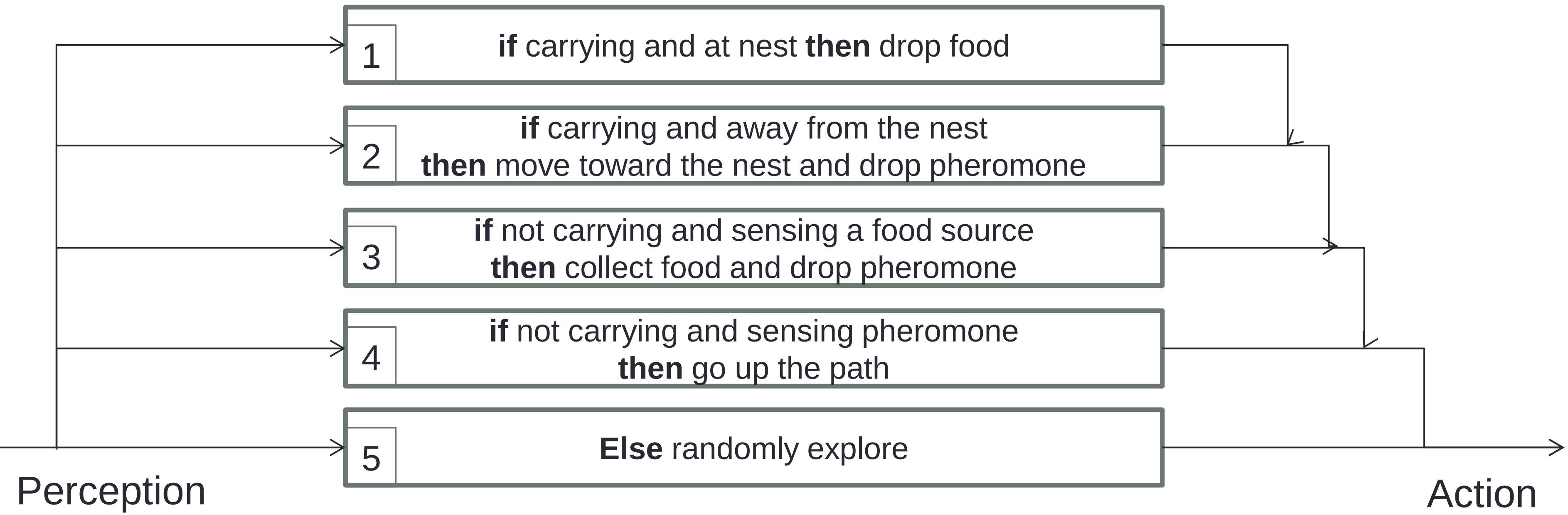
Rules for the agent's decision:

- Randomly explores
- **if** not carrying **and** sensing a food source **then** collect food **and** drop pheromone
- **if** carrying **and** away from the nest **then** move toward the nest **and** drop pheromone
- **if** carrying **and** at nest **then** drop food
- **if** not carrying **and** sensing pheromone **then** go up the path

MAS Modelling
The example of ants « highway »

Organising the decision rules

-> **Subsumption** architecture (the higher the rule, the higher the priority)

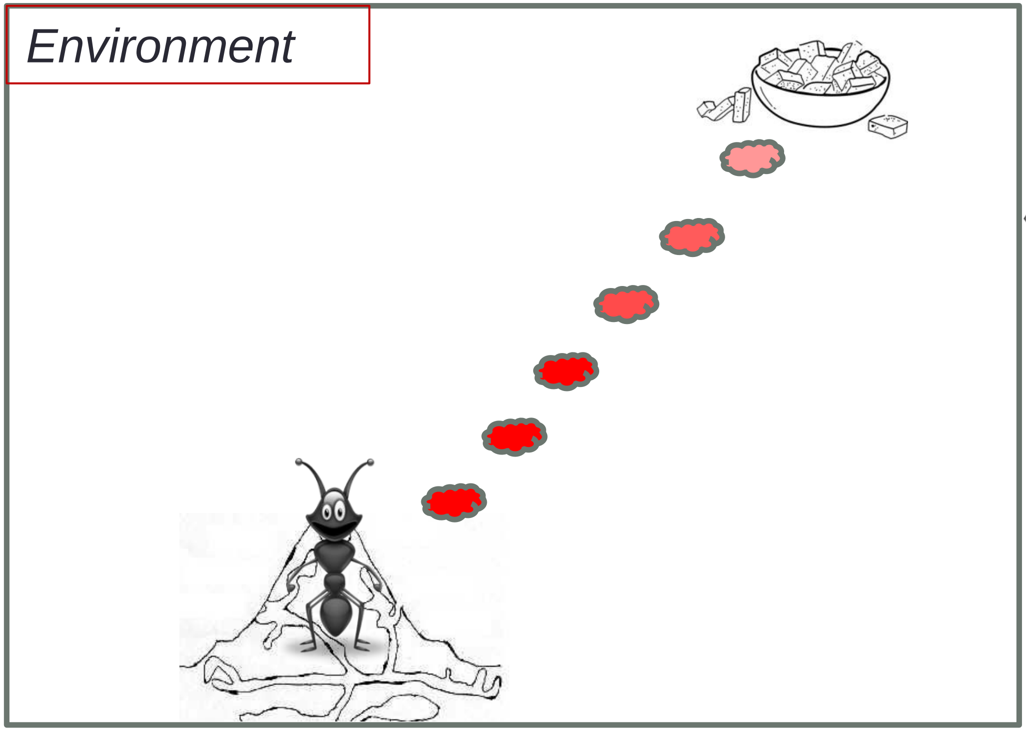


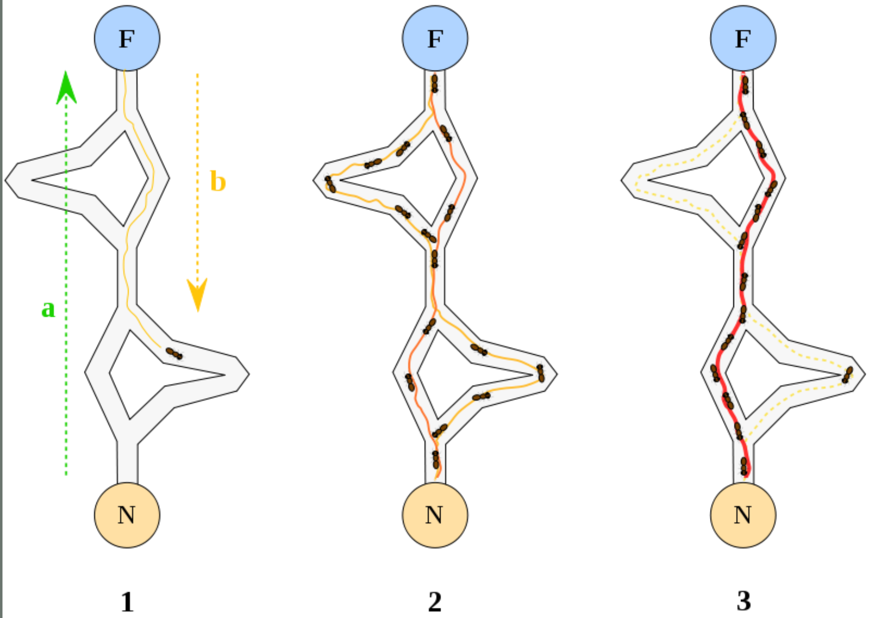
Ants.nlogo

Self-organisation

Local interactions (stigmergy) + **positive feedback** (reinforce pheromone) + **negative feedback** (evaporation)

MAS Modelling
The example of ants « highway »





Wikipedia

Emergence

Shortest path!

The *ant algorithm*

The example of ants « highway »
What have we done?

Focus on the environment of agents: the nest, the food sources

- The definition of **agents** and **interactions** are **driven** by the environment

- **Environment-driven** Model Design

There are others!

- **Agent-driven** model design
- **Interaction-design** model design
- **Hybrid** model design

Model design approaches on a case study

Case study

Anna Dornhaus, Franziska Klügl, Christoph Oechslein et al., 2006
Benefits of recruitment in honey bees; effects of ecology and colony size in an individual-based model

Scientific question:

Why do some social insects have sophisticated recruitment systems while other species do not communicate at all on the location of food sources?

Building an agent-based model

- Foraging bees (*Apis mellifera*)
- Quantify the benefits of recruitment by:
  - different food sources (location, quality, variability)
  - different colony sizes

Case study
Agent-driven model design

Focus on the agents, their behavior and their decision making mechanisms

- **Agent's point of view!**
- Interactions and environment are secondary, and added when needed

➢ **Basic strategy**

1. Agent observation and coarse behavior description: observation, literature, domain experts
2. Categorize agents: where is the heterogeneity? how many classes or types of agents are necessary?
3. Decide about agent architecture
4. Formalize and implement agent's behavior and goals
5. Add interactions and environmental aspects **when needed**
6. Test (simulation) whether necessary macro-phenomena are sufficiently reproduced

Case study
Agent-driven model design

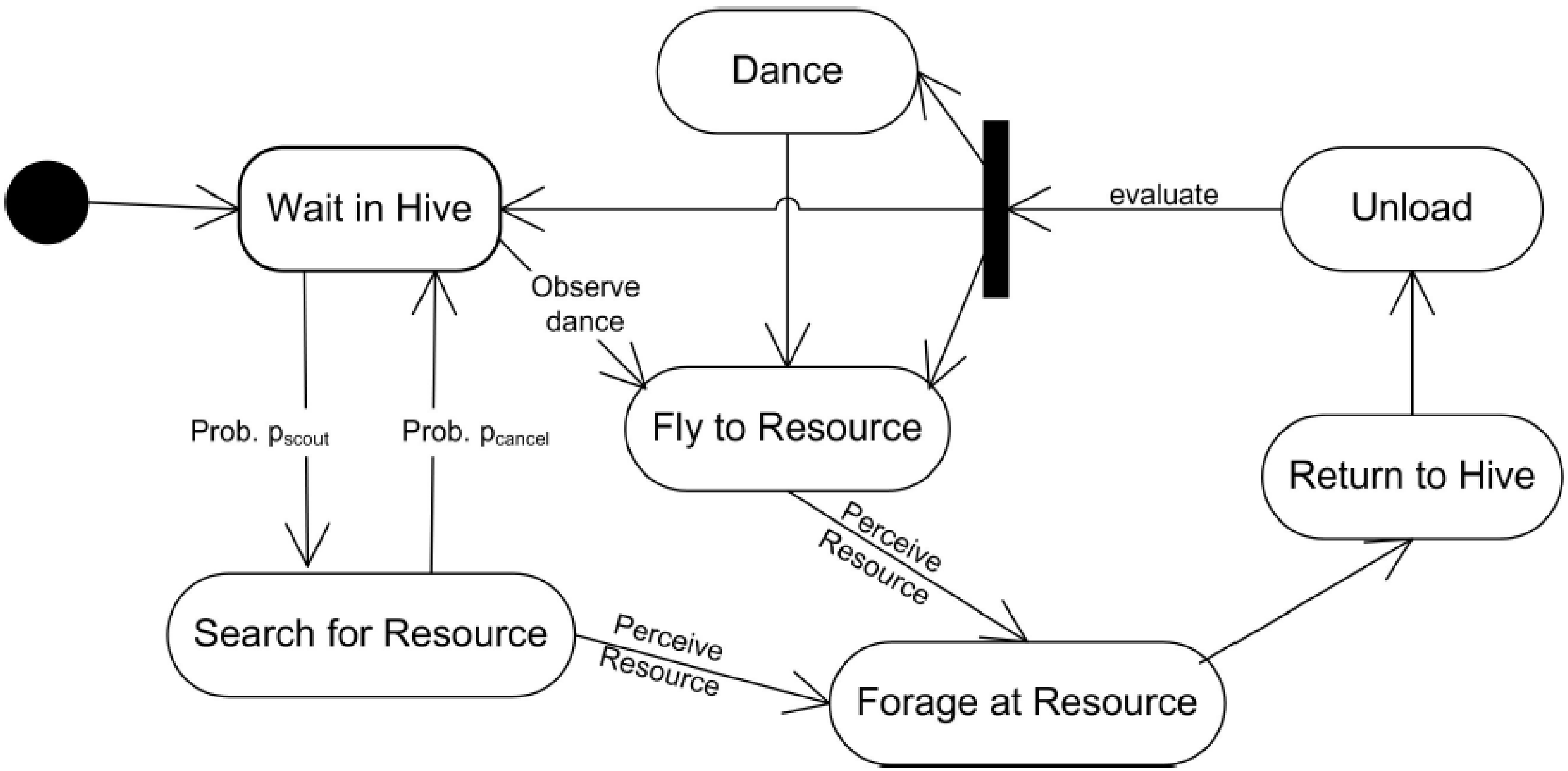
1. Agent's behaviour: different tasks
   Exploration, foraging, recruitment and looking for dances

2. Only one type of agent: the foraging bee

3. Simple architecture, reactive agents

4. Formalizing and implementing: activity diagrams

Case study
Agent-driven model design



Environment and interactions?

Case study
Agent-driven model design

1. Agent’s behaviour: different tasks
   Exploration, foraging, recruitment and looking for dances

2. Only one type of agent: the foraging bee

3. Simple architecture, reactive agents

4. Formalizing and implementing: activity diagrams

5. Environment and interactions:
   - A 2D map for scouting and discovering resources where agents may move
   - Resources that provide a nectar of certain quality
   - The hive: abstract, just a place to unload and to dance for recruitment of others
   - Dances: an agent recruits other (waiting) agents and communicates some info

6. Simulation!

Agent-driven model design
Discussion

- Agent's point of view
  - Behaviouralist
  - Role-playing games to validate, discover...
- Intuitive
- Complicated when validation is not achieved: Trials / errors
- Sometimes non-trivial to find the agent's behavior -> intentionality?
- Level of detail / appropriate granularity? The simplest model possible?

Case study
Interaction-driven model design

Focus on interactions, at the micro **and** macro levels

- **View from above** (bird perspective)
- Agents seen as black boxes producing messages -> Actors / Entities

➢ **Basic strategy**

1. Identify actors/entities and interactions among them
2. Coarse description of protocols (actors + interactions) and their conditions, constraints etc.
3. Add environment entities and derive the agent’s behavior for producing the atomic interaction elements
4. Implement agent’s behavior and test (simulation)

Case study
Interaction-driven model design

1. Identify actors/entities and interactions among them

Interactions	Bees	Resources	Nectar Storage
Bees	Recruitment	Harvest	Unloading
Resources	Localization	-	-
Nectar Storage	Status Information	-	-

Table 8.1: Interaction table when choosing bees as agents.

Interactions	Scout	Forager	Reserve	Ressource	Nectar Storage
Scout	-	-	Recruitment (*)	Discovery (*)	-
Forager	-	-	-	Harvest	Unload
Reserve	Observe Dance (*)	-	-	-	-
Resource	Localization Information	Nectar	-	-	-
Nectar Storage	Status Information	Status Information	Status Information	-	-

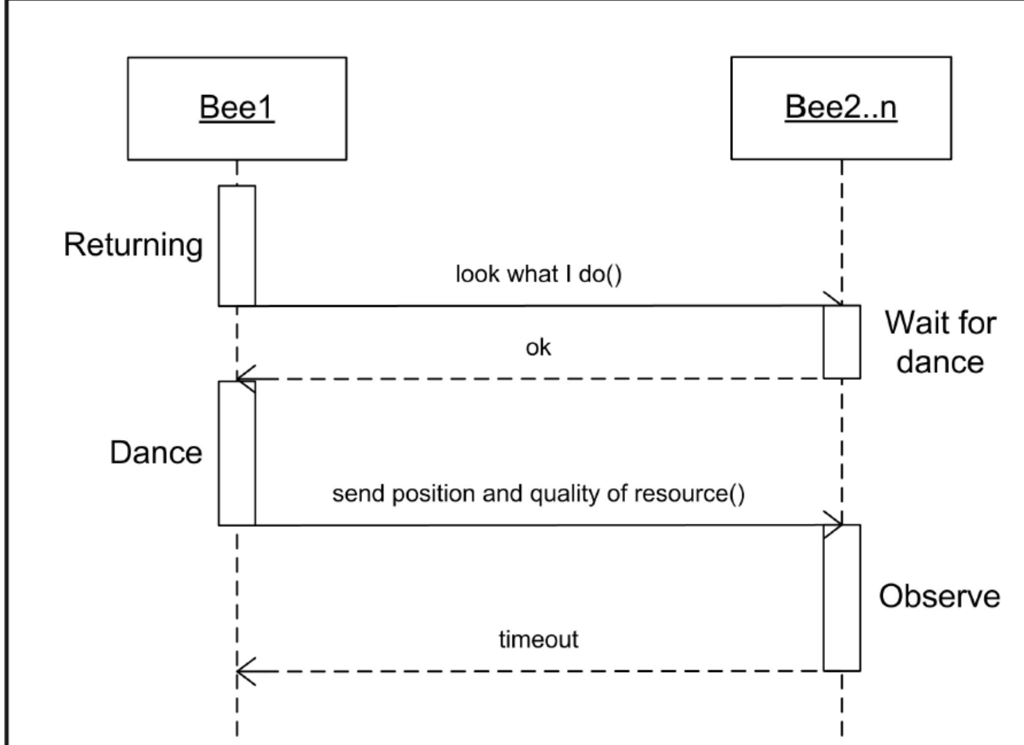
Case study
Interaction-driven model design

1. Identify actors/entities and interactions among them

2. Protocols: recruitment example, with the context and the different actors

Context

This interaction takes place after a scout (*bee1*) returned and has evaluated the distance and quality of its newly discovered food source as good enough for recruiting others to this source. The bee moves to the dance place and starts its waagle dance. There are one or more bees (*bee2..n*) there that are waiting for information/recruitment. At the end of the interaction, the previously waiting bees have the information about the location of the food source and may become recruited to it.



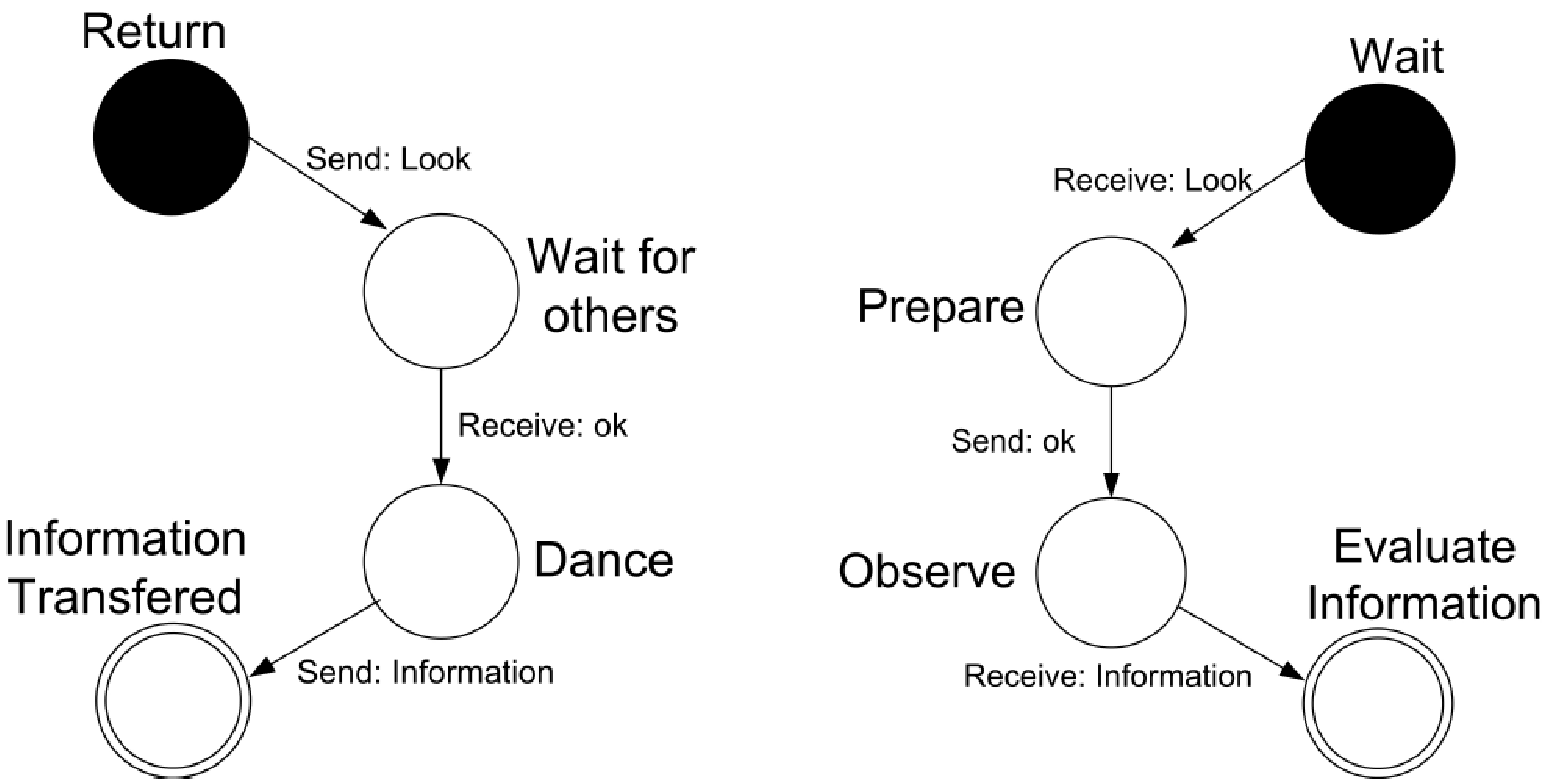
Case study
Interaction-driven model design

1. Identify actors/entities and interactions among them

2. Protocols: recruitment example, with the context and the different actors

3. Derive the agent's behaviour
   - from each interaction protocol
   - then unify to produce only one behaviour
   - using, for example, finite-state machines

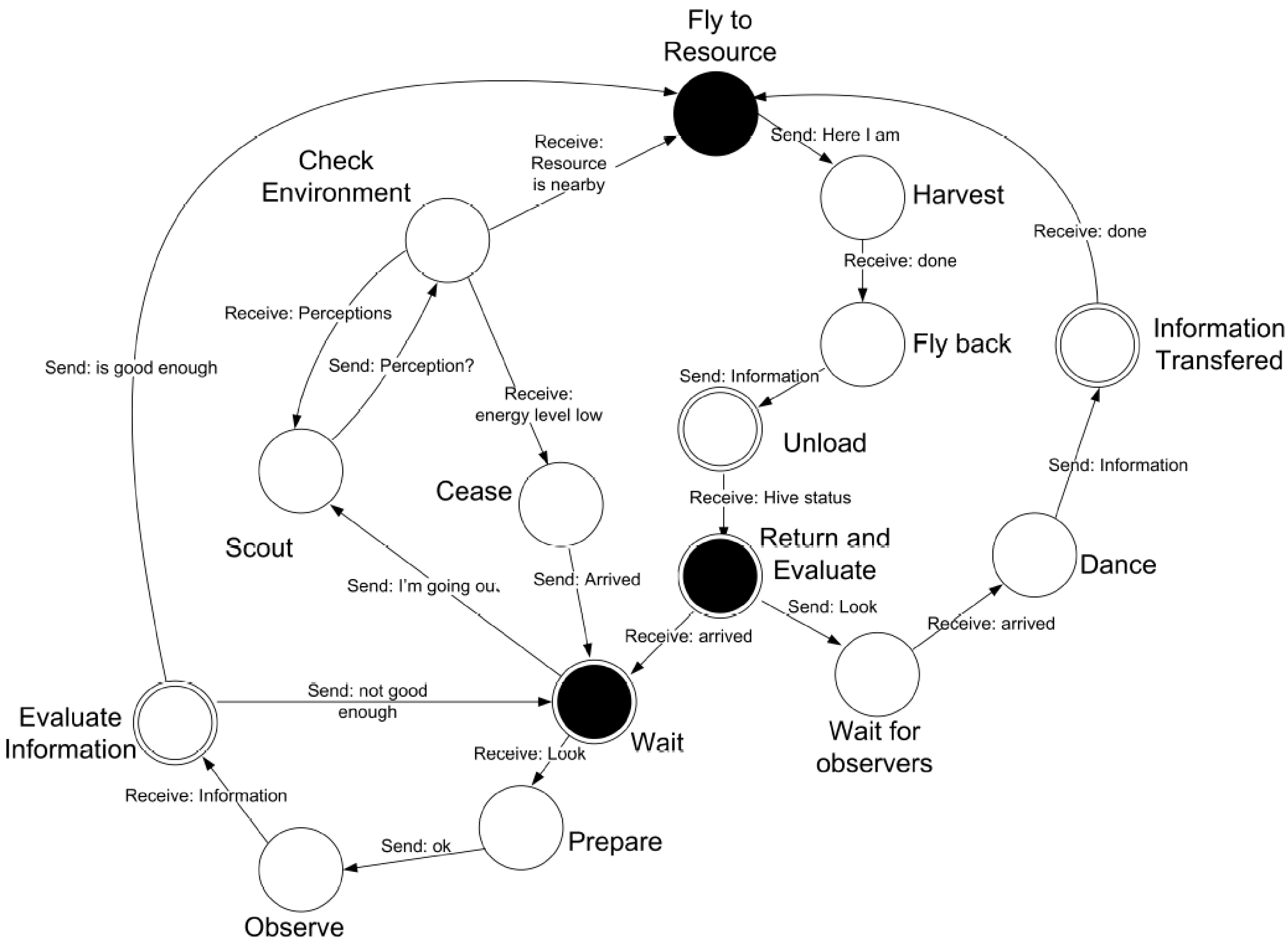
Case study
Interaction-driven model design



(a) Recruiter

(b) Recruitee

Case study
Interaction-driven model design



Interaction-driven model design
Discussion

- Dependencies / effects of interactions on agent behaviour are explicitly represented
- Problem with proactive behaviours, not triggered by external messages
- Everything is an agent!
- Stigmergy? Representation of pheromones?
- Organizational-oriented model-design
  - Abstraction of the organization, definition of roles for agents, and interactions according to roles...
  - When, how do we change roles?

Case study
Environment-driven model design

Focus on the environment in which agents operate

- Interactions and agents are secondary, and added when needed

➢ **Basic strategy**

1. Identify relevant aspects (global status, global dynamics/ local entities) of the part of the model that represent the agents' environment
2. **Determine the primitive actions** of the agent and the **reaction** of the environmental entities
3. Determine what information from the environment must be given to the agent for its decision making
4. Decide on an agent architecture that is apt to connect perceptions and actions of the agent appropriately for actually producing the agents behavior
5. Test whether the usage of a **learning mechanism** (reward based) for determining the actual agent behavior is feasible and advisable or not
6. Implement the environmental model and the agent's behaviour
7. Test and analyze the overall simulation results and individual trajectories

Case study
Environment-driven model design

1. Environment model :
   - a 2D map (Cartesian coordinates)
   - a hive, for storage
   - resources distributed on the map, containing a certain amount of nectar

2. Initial environmental configuration:
   - hive at the center of the map
   - sources initialized to random positions
   - normal nectar distribution

3. Perceptions and actions: resource existence (from a certain distance), its position and capacity (if nearby), hive storage (if nearby), etc. ; Fly towards perceived resource, towards the hive, load / unload nectar, etc.

4. Rule-based approach for the agent's behaviour

Case study
Environment-driven model design

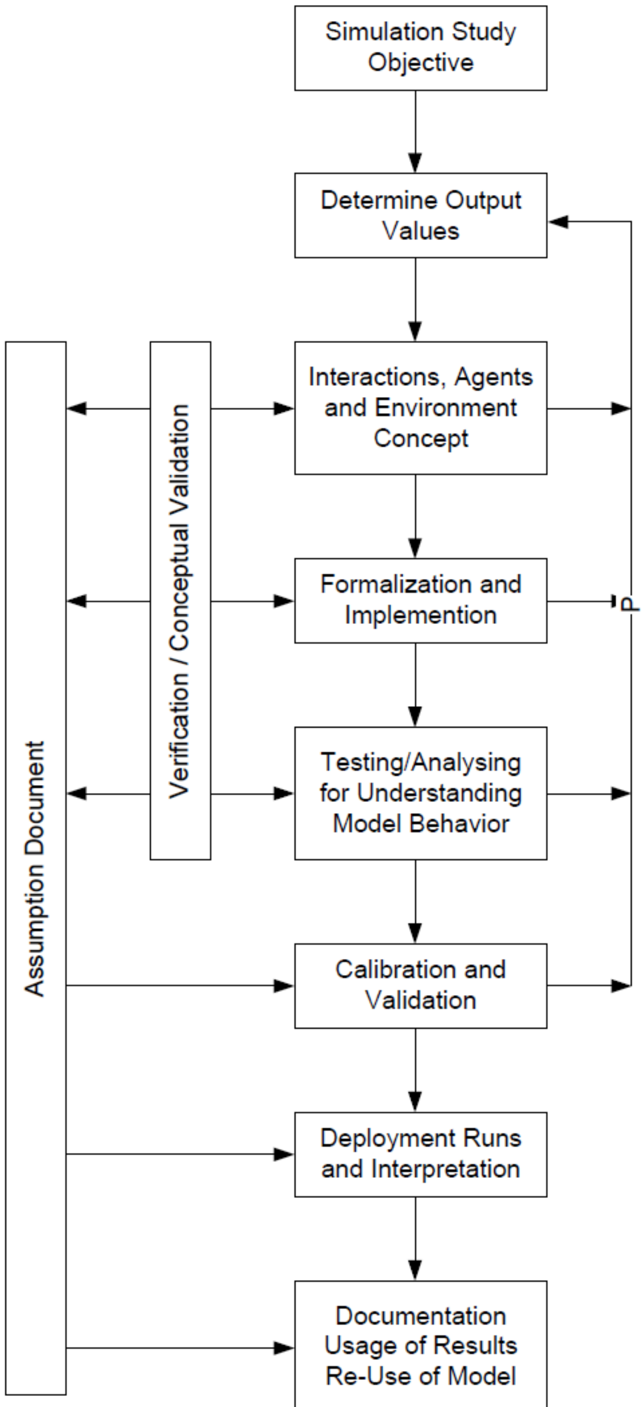
4. Rule-based approach for the agent's behaviour

    1. **if** hive-storage < A **then** perform random search (with probability pA)
    2. **if** not at hive and not perception of resource **then** perform random search
    3. **if** perception of resource **then** fly towards perceived resource
    4. **if** at resource **then** memorize resource information
    5. **if** at resource **then** load nectar with rate load
    6. **if** nectar load > B **then** fly towards hive
    7. **if** at hive and nectar load > B **then** unload nectar with rate unload
    8. **if** at hive and resource information memorized **then** display resource information
    9. **if** not at hive and not perception of resource **then** fly to hive (with probability pC)

Environment-driven model design
Discussion

- As with the interaction-oriented approach, problem with proactive behaviours, not triggered by external stimuli
- How to choose the level of detail of the environment?
- Complexity of rules, sometimes -> another agent architecture

MAS Modelling - To sum up

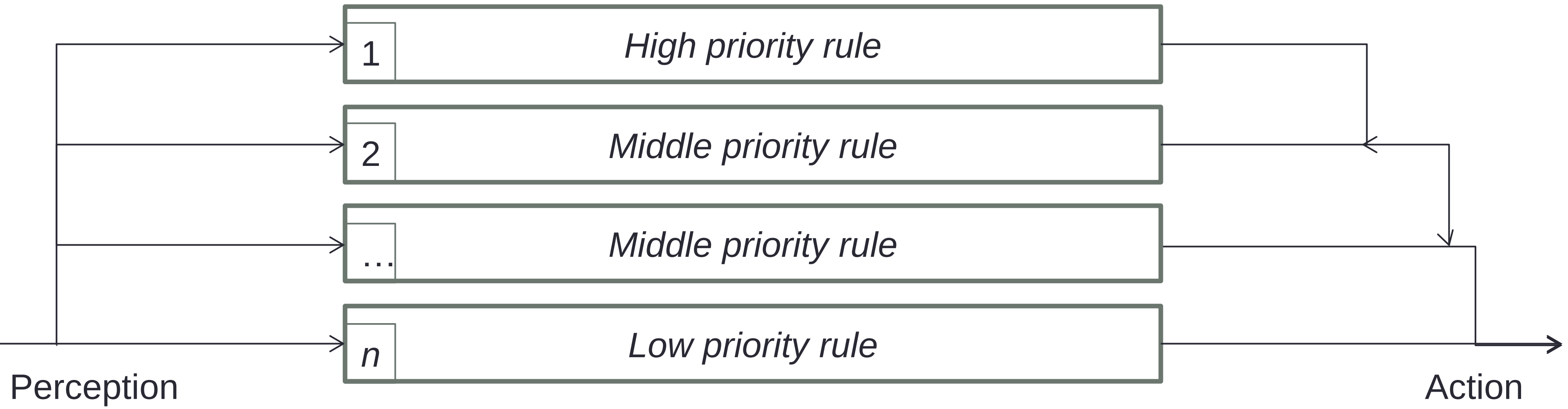


MAS Modelling - To sum up Formalizing the agent behavior

1. Perception-based architecture: **subsumption architecture**

- All information is perceptible in the agent's immediate environment
- No memorization of the environment
- Its actions in the environment (located) are driven by its perceptions and internal state
- Rules of action : **If <internal state> and <perceived state> then <action>**

  Only one rule applied per decision taken!



MAS Modelling - To sum up Formalizing the agent behavior

2. Activity-based architecture: **finite-state machine**

- **State**: an agent activity
- **Event**: something that happens in the outside world (or inside the agent) that can be perceived. Serve as a trigger for an activity.
- **Action**: something that the agent does that will change the world situation and produce other events. The action is directly linked to the activity

Current State (Activity)	Action	Condition (Event)	Next State
Fleeing	Flee the enemies	Safe	Patrolling
Patrolling	Patrol the area	Too many (strong) enemies	Fleeing
		A few (weak) enemies	Attacking
Attacking	Fight the enemies	Enemies are defeated	Patrolling
		Enemies are stronger	Fleeing

MAS Modelling - To sum up
Formalizing the interactions

Interaction table, communication protocols (if needed)

Interactions	Scout	Forager	Reserve	Ressource	Nectar Storage
Scout	-	-	Recruitment (*)	Discovery (*)	-
Forager	-	-	-	Harvest	Unload
Reserve	Observe Dance (*)	-	-	-	-
Resource	Localization Information	Nectar	-	-	-
Nectar Storage	Status Information	Status Information	Status Information	-	-

MAS Modelling - To sum up Formalizing the environment

Properties description [Russel&Norvig, 2003]

- partially known to the agent (*vs.* fully accessible)
- stochastic (*vs.* deterministic)
- dynamic (*vs.* static)
- continuous (*vs.* discrete)
- open (*vs.* closed)

+ Topology, how the dynamic works …

Modelling time in simulations

- To simulate is to immerse the model into time

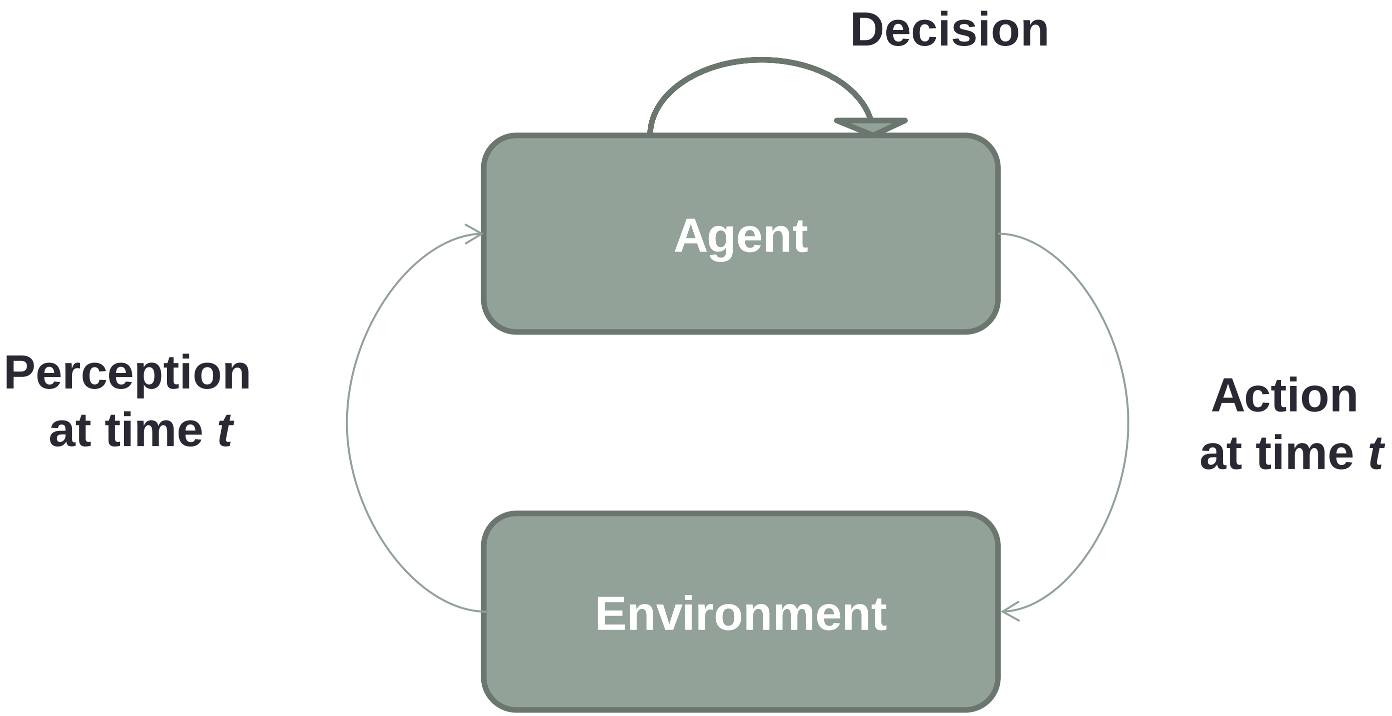
- **But what is time?**
  - For us?
  - For our environment?
  - For simulated agents? (sheep, bees, humans …)

Agents, like us, are supposed to act and interact **concurrently** (principle of **causality**)

- **Modelling time is**
  1. Modelling the **behavior time** of an agent
  2. Modelling the **endogenous dynamics of the environment**
  3. **Coupling** the agents and the environment -> **Scheduling (*ordonnancement*)**

Modelling time in simulations

1. Modelling the **behavior time** of an agent



Modelling time in simulations

2. Modelling the **endogenous dynamics of the environment**

In many systems, the environment **not only reacts** to the agents inputs, but **also evolves according to its own dynamic**

e.g. Robocup simulations (rolling ball), prey-predator (grass), beehive (pheromones)

3. **Coupling** the agents and the environment -> **Scheduling**

Continuous or **discrete** time?

- Continuous time (e.g. in EBM): the interval between 2 "actions" is arbitrarily small

e.g. $$\frac{da}{dt} = \lambda_a a - \gamma ab$$

- **Discrete time in agent-based models**: event-based or time-stepped

Scheduling
Discrete time: time step

- Time evolves discretely with respect to constant time intervals
  - For example, **1 time step = 1ms of simulated (virtual) time**

- **At each time step**, the environment evolves and the agents “live” (perception, decision, action)

- Correspondence between virtual time and real time (rule of three)
  - For example, 1ms of simulated (virtual) time = 1s of real time

- **The “real” time represents the speed of the simulation**
  -> depends on computing resources … and modifiable by the user!
  - Slow down to better observe
  - Accelerate to predict the future state of the system
  - "In real time" to integrate the human and allow interactions!

Climate Change.nlogo

Scheduling
Discrete time: time step

- ***At each time step** dt, the environment evolves and the agents “live” (perception, decision, action)*

$$sigma(t + dt) = Evolution(\uplus(A_n(t), E_n(t)), \sigma(t))$$

  - How? Concurrently? In which order?

- Scheduling solution in discrete time simulations

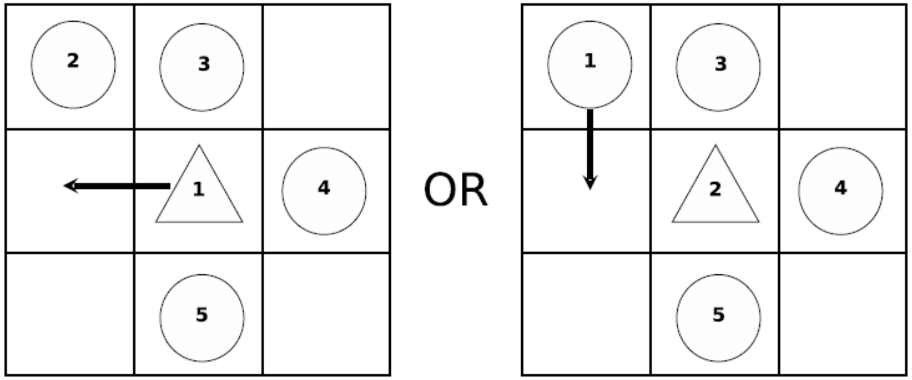
```
while(globalVirtualTime != endOfSimulation){
        for (SimulatedAgent agent : AllTheAgents)
                agent.act(); //perception, deliberation, action
        virtualEnvironment.evolve(); // for a dynamic environment
        globalVirtualTime++;
}
```

- What’s the issue?

Scheduling
Discrete time: time step

```
while(globalVirtualTime != endOfSimulation){
        for (SimulatedAgent agent : AllTheAgents)
                agent.act(); //perception, deliberation, action
        virtualEnvironment.evolve(); // for a dynamic environment
        globalVirtualTime++;
}
```

- Bias problem with the activation list



The prey could be dead or alive depending on its rank in the activation list.

- Solution?

Scheduling
Discrete time: time step

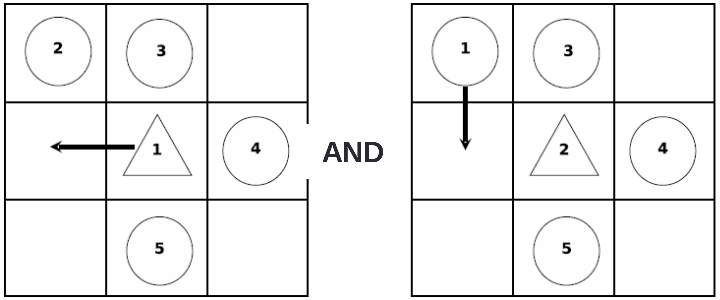
- Solution 1 : **shuffle** the agents list every time step

shuffle(AllTheAgents)

```
while(globalVirtualTime != endOfSimulation){
        for (SimulatedAgent agent : AllTheAgents)
                agent.act(); //perception, deliberation, action
        virtualEnvironment.evolve(); // for a dynamic environment
        globalVirtualTime++;
}
```

- Solution 2 : **simulate concurrency**
  - all the agents operate on temporary variables
  - the perceived environment is the same for all the agents
  - once done, the next system state is computed

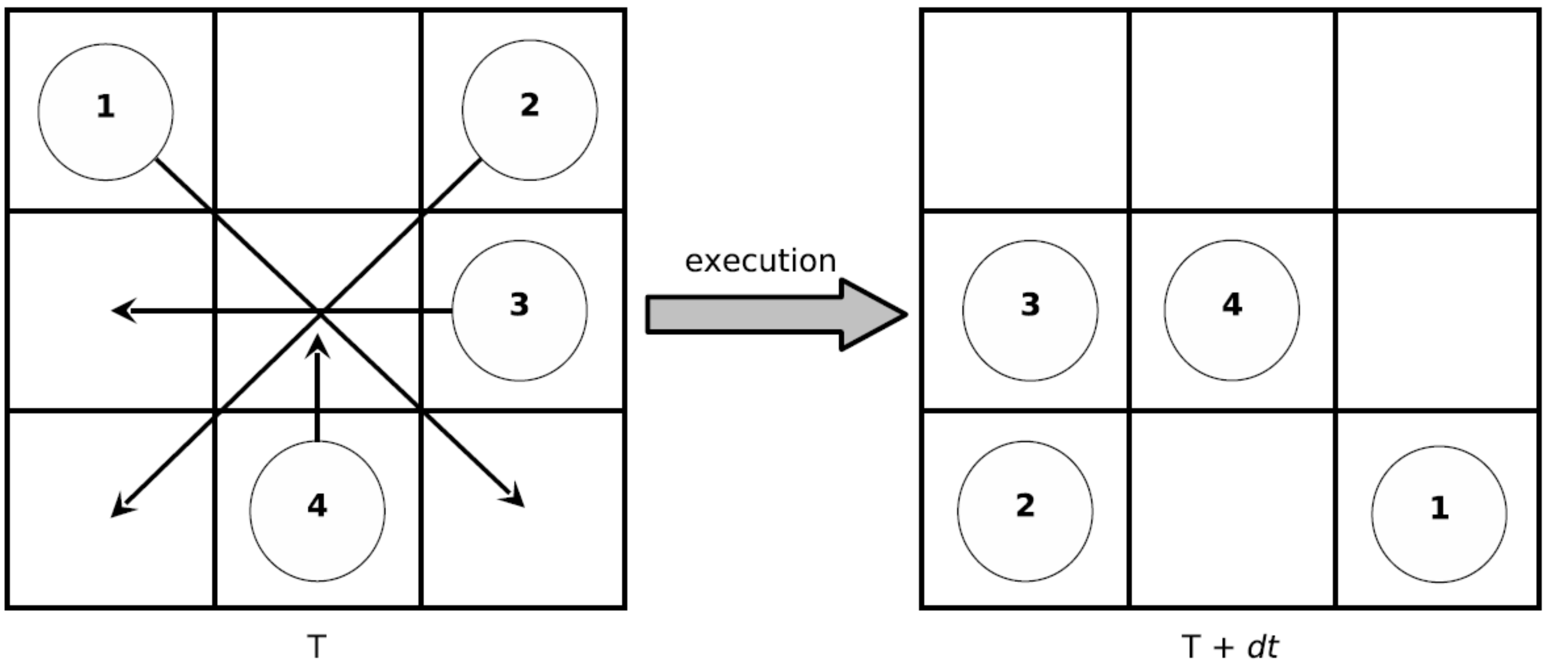
BUT we have to solve the conflicts



Conflict

Scheduling
Discrete time: time step

- Defining the **temporal granularity of actions**
  - "What can an agent do in 1 time step?"
    or "How much can an agent do in 1 time step?"



Plan

ABMs implementation

- **Object-oriented** programming languages are particularly suitable for agents: Java, C+++..

- Also parallelism-oriented languages and distributed computing (OpenCL)

To be managed:

- **Scheduling mechanisms**
- **Physical distribution**, if necessary! (Large number of agents / interactions)
- **Communication mechanisms** between agents (language format, mailboxes, etc.)
- **Visualization** of the system's behaviour (messages exchanged, "emergence", self-organisation …)

ABMs implementation

- Many platforms exist!
  - Generic: JADE, Jason, MadKit ...
  - Dedicaded for simulation: NetLogo, RePast, Gama, Flame, NetBioDyn...
  - For 3D animation (Massive), financial markets (ATOM)...
  - Model-dependent (IODA - JEDI : interaction-oriented modeling) …

- What's a platform for?
  - Management of agent life cycles, scheduling
  - Communication mechanisms
  - Distributed architecture
  - Visualisation -> micro (agent's behaviour, life cycle) and macro (emergent patterns)
  - Agent programming languages

Agents and Object-oriented programming

- **Agent**: an extension of the **object** concept in object-oriented programming
  - **Properties -> private attributes**

```
public class AntAgent {

    private boolean carrying;
    private Environment env;
    private int x;
    private int y;

    public AntAgent(Environment e) {
        env=e;
        carrying=false;
        x=env.getNestX();
        y=env.getNestY();
    }
```

Agents and Object-oriented programming

- **Agent**: an extension of the object concept in object-oriented programming
  - Properties -> private attributes
  - **Perception and action abilities -> private methods**

```
 * Randomly choose the next position of the agent
private void randomlyExplore() {

 * @param p list of detected pheromones
private void goUpPath(List<Pheromone> p) {

 * Compute the next position in direction of the nest
private void moveTowardNest() {

 * Movement primitive of the agent, according to the temporal granularity of action
private void moveToward(int tx, int ty) {

private void dropFood() {

private void pickUpFood() {

private void dropPheromone() {
```

Scheduling

- One public method *act()* within the AntAgent class

Solution 1

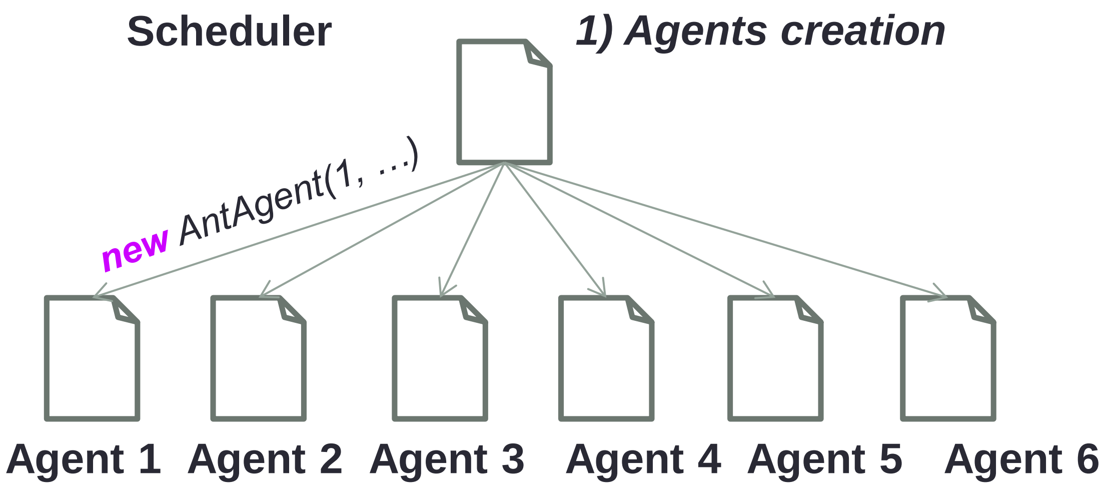
A Scheduler class to make the agent *act*, at each time step, *each time in a different order*

```
public void act() {
    // Perception
    List<Pheromone> around=env.pheromoneAround(x,y);
    boolean atNest=env.atNest(x,y);
    boolean foodAround=env.foodAround(x,y);

    // Decision and action according to ranked rules
    if(carrying && atNest) {
        dropFood();
        carrying=false;
    }
    else if(carrying && !atNest) {
        dropPheromone();
        moveTowardNest();
    }
    else if(!carrying && around != null ) {
        goUpPath(around);
    }
    else if(!carrying && foodAround) {
        pickUpFood();
        carrying=true;
        dropPheromone();
    }
    else {
        randomlyExplore();
    }
}
```

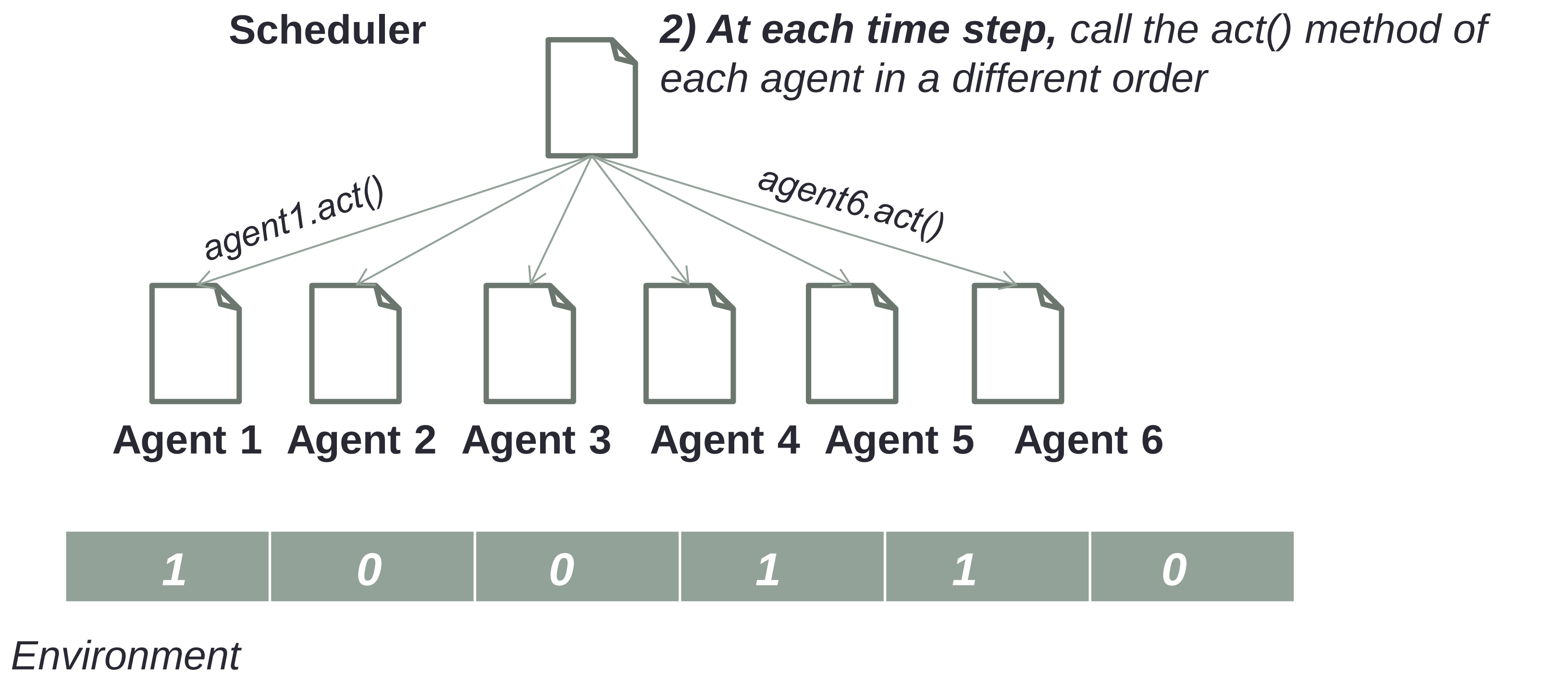
Scheduling

Solution 1



Scheduling

Solution 1



Scheduling

- One public method *act()* within the AntAgent class

Solution 2

The Scheduler class to make the agent *act*, at each time step

act() **return the Action** decided by the agent

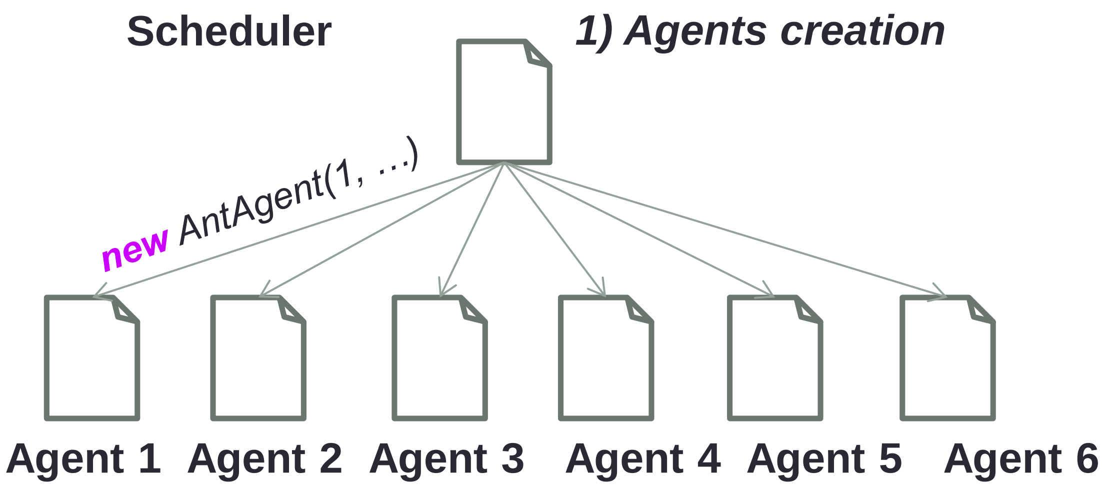
The Scheduler **retrieve all the Actions**, solve the conflicts, and applied their effects

```
public Action act() {
    // Perception
    List<Pheromone> around=env.pheromoneAround(x,y);
    boolean atNest=env.atNest(x,y);
    boolean foodAround=env.foodAround(x,y);

    // Decision and action according to ranked rules
    Action decided;
    if(carrying && atNest) {
        decided = new DropFoodAction(x, y);
    }
    else if(carrying && !atNest) {
        decided =new ComeBackCarrying(x, y);
    }
    else if(!carrying && around != null ) {
        decided= new GoUpPath(x, y);
    }
    else if(!carrying && foodAround) {
        decided=new PickUpFood(x, y);
    }
    else {
        decided=new Explore(x, y);
    }
    return decided;
}
```

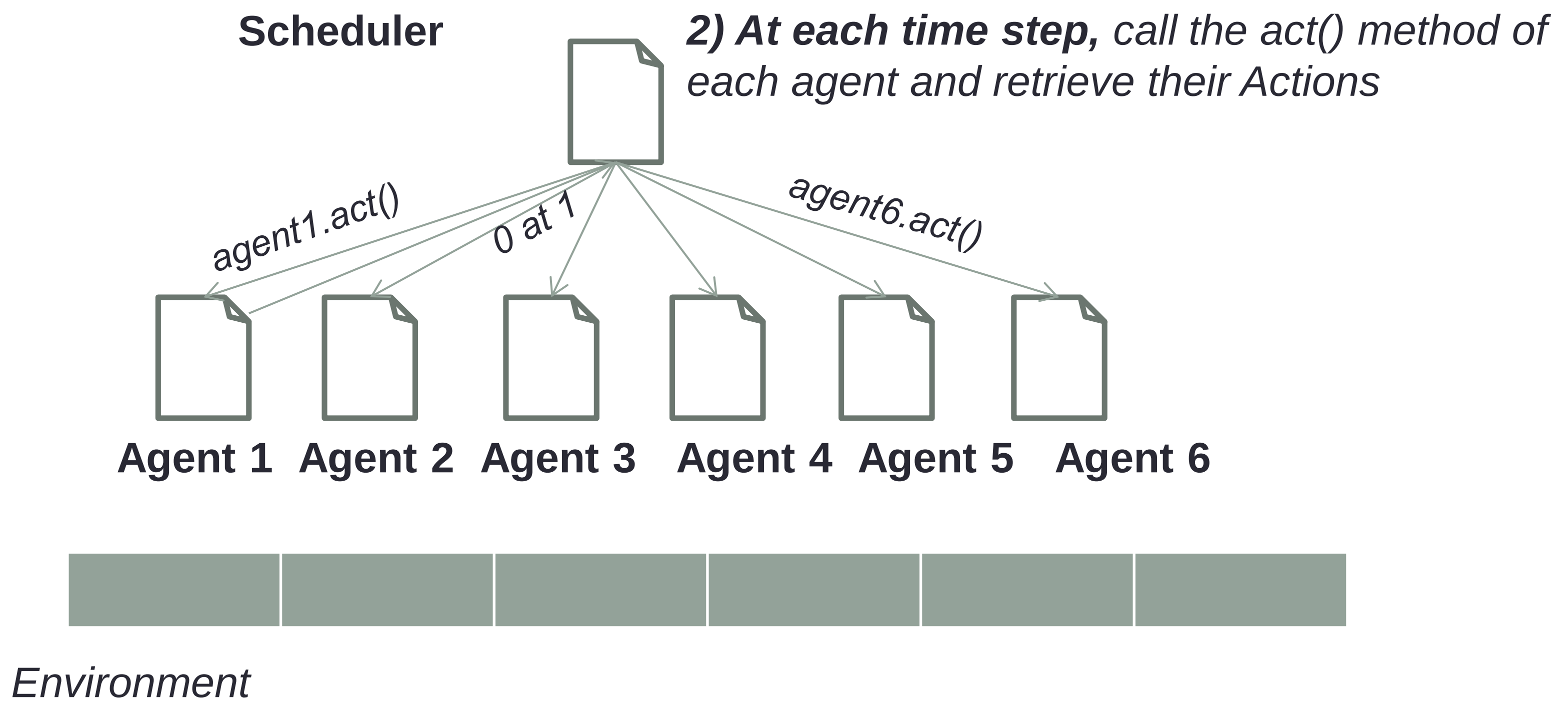
Scheduling

Solution 2



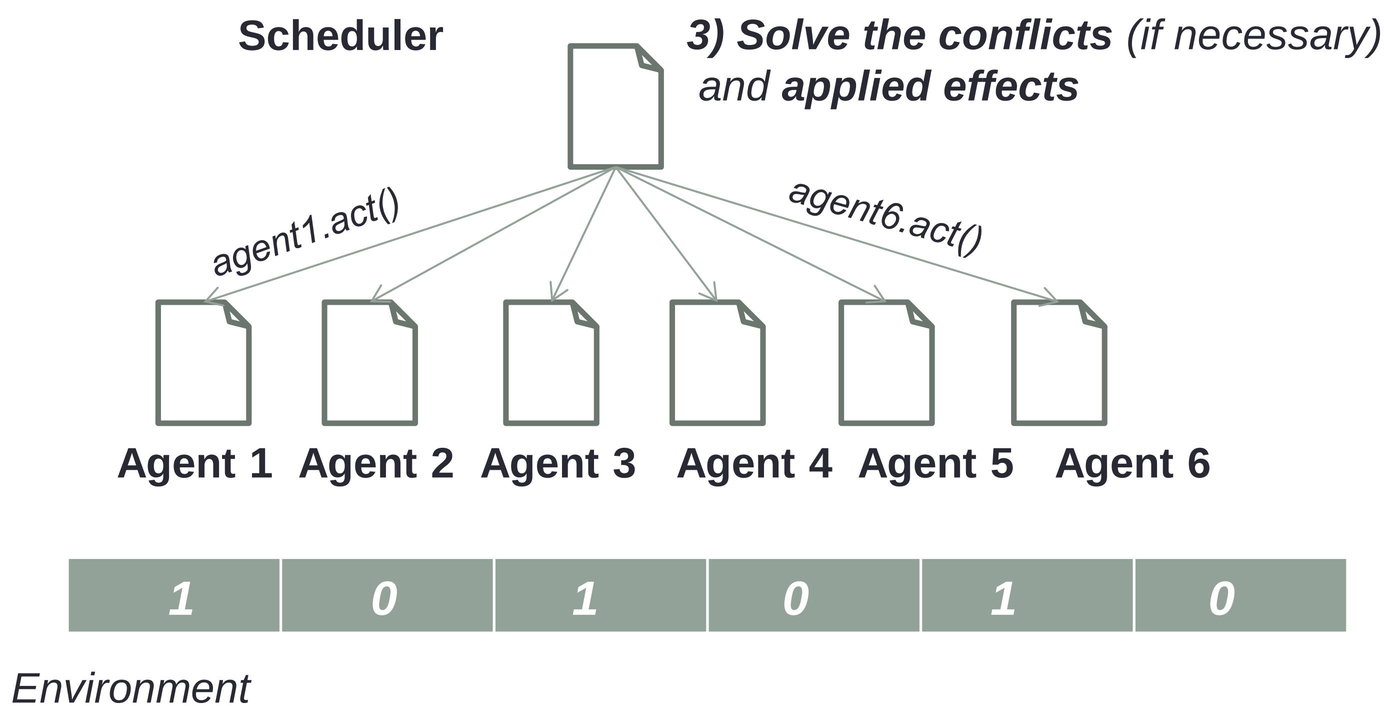
Agents and Object-oriented programming
Scheduling

Solution 2



Scheduling

Solution 2



Plan

1. System complex: an Introduction

2. Modelling and Simulation

3. Multi-Agent Systems for complex systems modelling and simulation

4. Implementing ABM

5. Visualisation and Interactions with an Agent-Based simulation
   1. User OUT : Platforms HMI - simulation, manipulation and visualisation
   2. User IN : the role of Virtual Reality in EVAH and simulation of Human-swarm interactions

6. Issues and challenges in Agent-Based Models

Visualisation and Interactions with an Agent-Based simulation

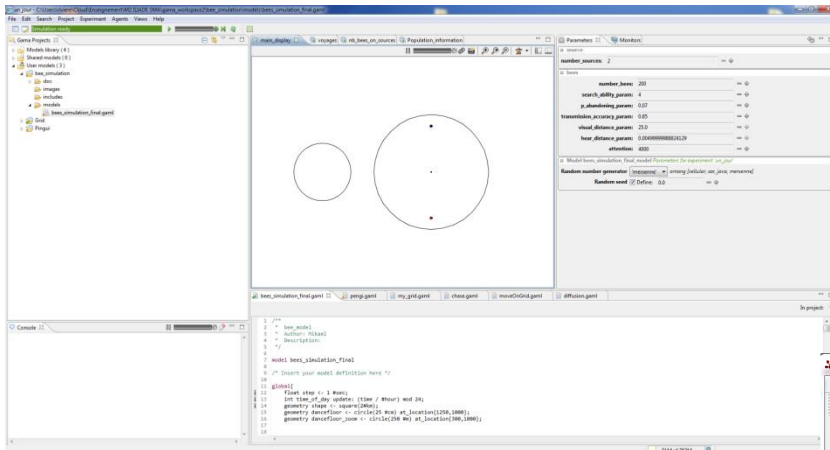
(Reminder of Pierre Chevaillier's course)

AEIO .. U !

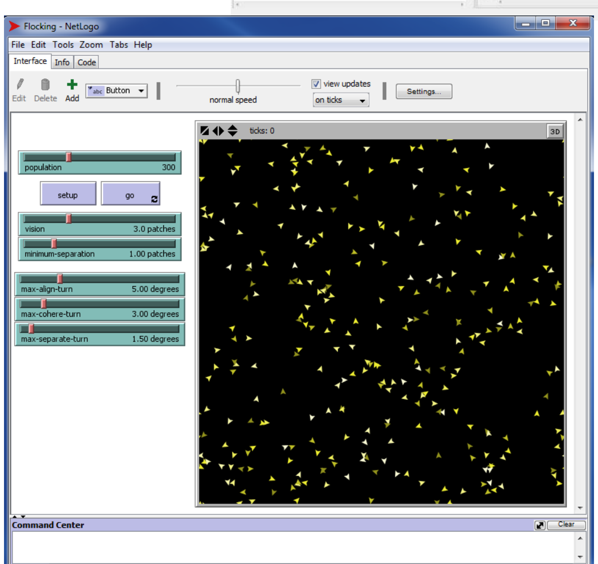
- The User as a Spectator / Actor / Creator for the MAS
- IN or OUT ? Agent of the system, or classic Human (in HMI)
- Immersed User
  - How s/he is perceived by the Agents?
  - How can s/he interact with the Agents?
  - Is there a mediator agent b/w U and Agents?

User OUT : Platforms HMI

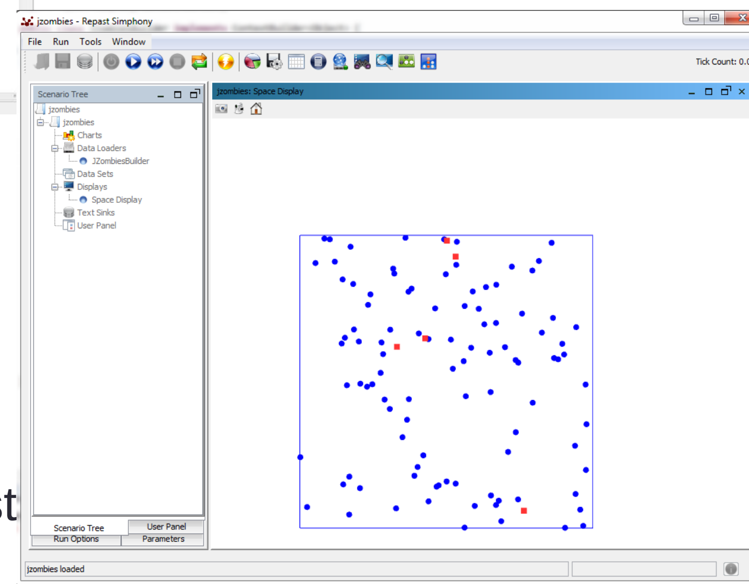
Various **HMIs**, depending on the platform, to verify and test the models



Gama



NetLogo



Repast

User OUT : Platforms HMI

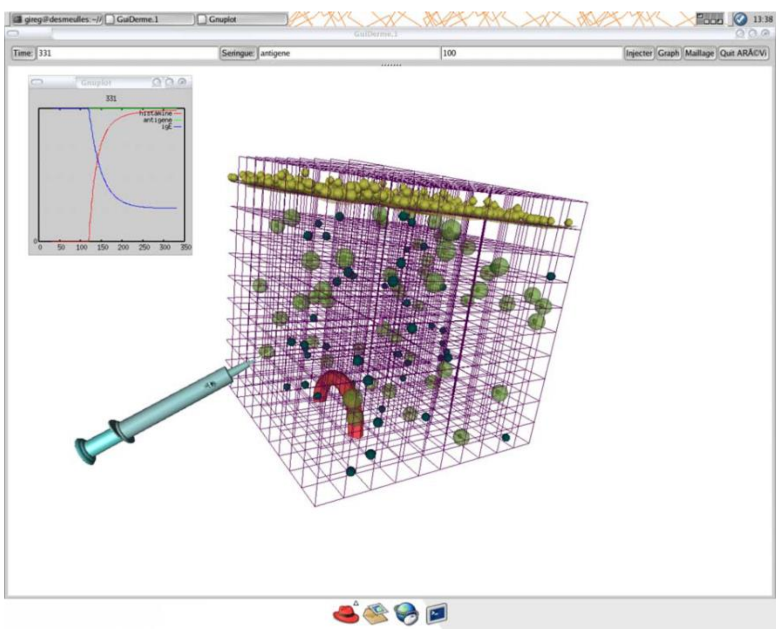
Visualisation	Interactions
2D / 3D depending on the platorm and the model	**With the simulation** • Start / Stop the simulation • Control the simulation **speed** • Modify the environment (at the initial state, or during the simulation)
At the micro level: **Agents** behaviour • Movements • Decision making • Interactions (messages …)	
At the macro level **System** behaviour • Curves (population evolution …) • Pattern, emergent phenomena • Self-organisation	**With the agents** • Adding / Deleting / Moving agents -> e.g. "*Point and click*" • Send messages

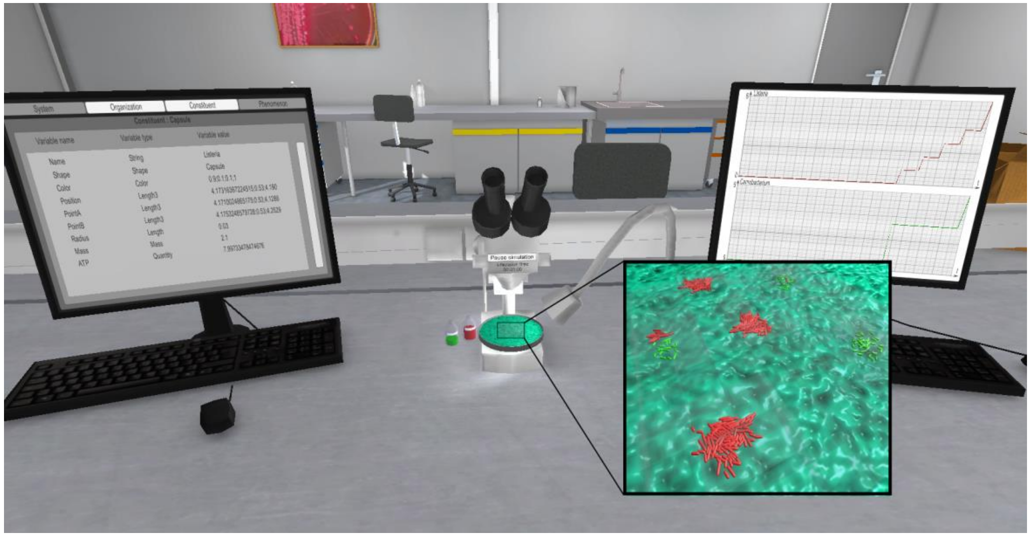
User OUT : Platforms HMI
The natural interaction metaphor

It could be hard for non-expert in computer science (and ABM) to manipulate the simulation

-> HMI with "natural interactions"

- Usually 3D / VR-like, e.g. *The virtual laboratory [Desmeulles, 2006]*





- Also tangible interface, motion capture in VR and AR, …

User IN : the role of Virtual Reality

Virtual reality as a vector for interaction and visualisation with the simulation

- VR allows
  - the User to be immersed in the simulation
  - natural interactions means

- The User plays the role of an agent (not always of the same type) with interaction and perceptions capacities, and can be seen by the other agents

- The User interacts directly interacts with the agents (or the environment), and not with the whole system

Participatory Simulations

Multi-agent participatory simulations [Guyot, 2006]

A simulation with which humans interact directly by controlling one of the system's agents -> e.g. in social simulations

- Experiments conducted in laboratories or via the Internet,
- With human participants and who are part of a multi-agent approach

Goals

- Learning and training
- Model building and validation
- Support for decision-making

The means of interaction are the same as those of the system's agents

In immersion -> EVAH and **Virtual Reality**

Participatory Simulations
Learning, EVAH (Intelligent Tutoring Systems)

EVAH: « Environnements Virtuels pour l’Apprentissage Humain »

Role: put the learner in a learning situation in a virtual reality environment.

MASCARET [Querrec, 2011]

The user embodies an agent within the simulation, with limited interaction and perception capacities

Participatory Simulations
Learning, EVAH (Intelligent Tutoring Systems)

EVAH: « Environnements Virtuels pour l'Apprentissage Humain »

Advantages:

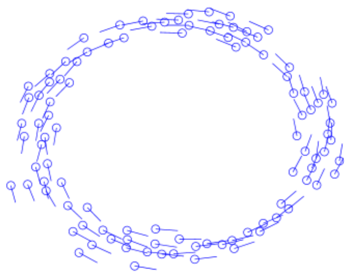
- Create many different pedagogical scenarios to train the user
- Keep a digital record of the user actions and decisions
- Create situations hard to reproduce in reality (huge fire, …)
- …

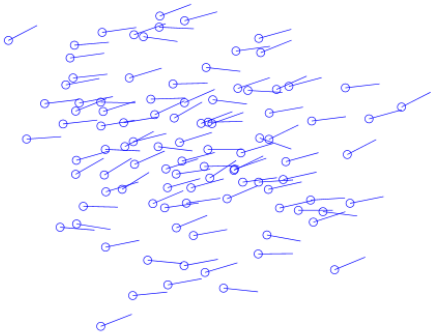
Drawbacks:

- Implement **credible** ECA
- Believability of the simulation itself

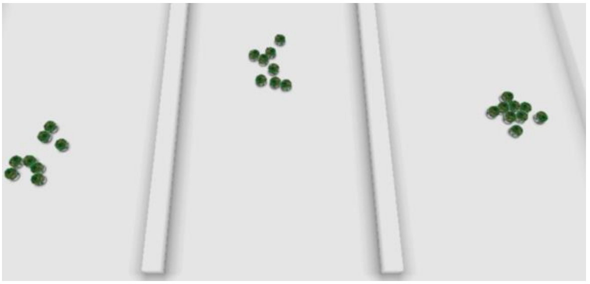
Human-Swarm (simulation) Interactions

- Robot swarms: multiple robots that coordinate autonomously via local control laws, based on the robot’s current state and environment





- Role of simulation: to test the robots behavior



How to allow a human to interact with the swarm?

Human-Swarm (simulation) Interactions

How to allow a human to interact with the (simulated) swarm?

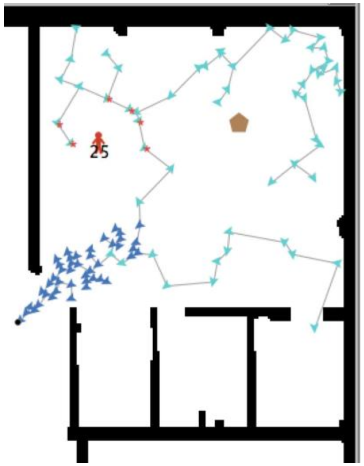
2 examples

1) Indirect control via environmental influences

Virtual pheromones, beacons, signals

Attractor / Repellent, Behavior changer

- Not easy to manipulate



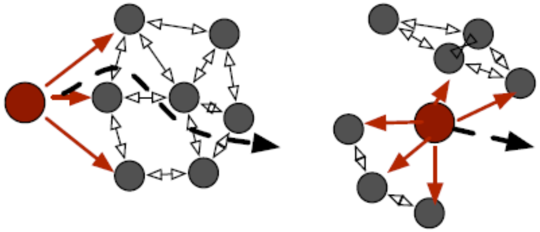
[Kolling et al., 2012]

2) Control through selected swarm leaders

i.e. Teleoperation

Influence propagate through the selected robots

- How this disturbs the autonomous coordination of the swarm?



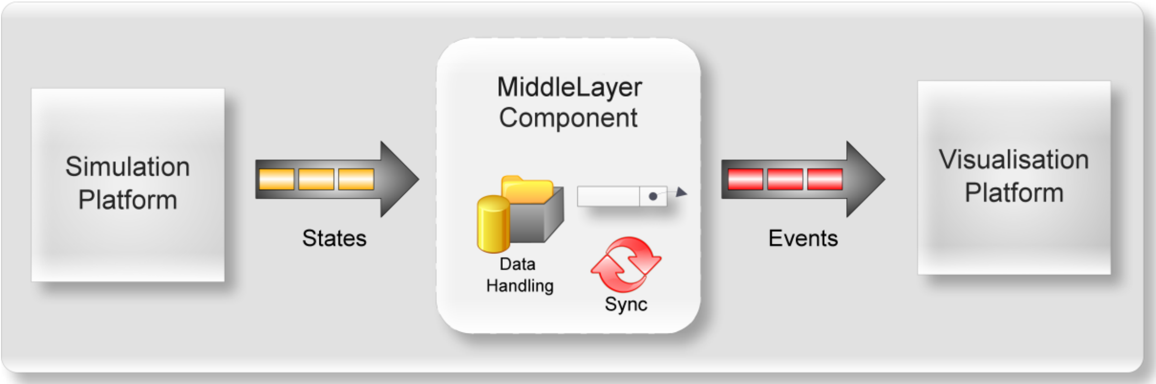
User IN : the role of Virtual Reality

Visualisation	Interactions
3D, virtual reality, tangible, haptic interface	**With the simulation** Limited
At the micro level: **Agents** behaviour • Movements • Interactions (voice, gestures, messages …)	
At the macro level • Indirect effects of micro actions	**With the agents** • The human user is an avatar perceived by agents • Intuitive / natural interactions: natural language, gestures • Other interactions: beacon, pheromones, signals, selection …

Virtual Reality and Natural Interactions Challenges

- Correspondence Simulation ⟷ RV
  - Agents ⟶ 3D Objects
  - Behaviours / Agent states ⟶ Animations
- Environmental dynamics (temperature, fog, signals, etc.)
- Interactions between agents
- Synchronization between the simulator time and the real (lived) time
- Interpolation of behaviours
- Configuring the virtual world
- Human – simulation interactions !

[Louloudi and Klügl, 2012]



Plan

1. System complex: an Introduction
2. Modelling and Simulation
3. Multi-Agent Systems for complex systems modelling and simulation
4. Implementing ABM
5. Visualisation and Interactions with an Agent-Based simulation
6. Issues and challenges in Agent-Based Models

Requirements for “good” Agent-Based Models

- **Documentation**!
  - Assumption document: scientific hypothesis and statements, parameters and their values + sources

- Formalization: Subsumption architecture, finite-state machines, interaction tables …

- Validity and Reproducibility
  - The **reproducibility** of results is a major prerequisite in science
  - A non-reproducible model cannot be reused for scientific or industrial purposes
  - A model goes with all its documentation, data, implementation details etc.

- Simplicity, Comprehensibleness, Flexibility and the Ability for Exploration

- Maintainability and the Ability for Extension

Issues and challenges in Agent-Based Models

- **Micro-Macro Link**

  The agents' behaviours (micro) generate the system global behaviour (macro)
  - The connection between both is not always clear or even not existing
  - Often one cannot exactly predict in advance what the macro level behavior of the model will be until it is simulated

- **Emergence** and **Non-Linearity**

  Worst case of macro-micro connection: no explicit connection
  - Non-linearity of micro behaviours
  - Feedback loops micro-macro (emergence and immergence)

  -> Complex systems!

- **Brittleness** and **Sensitivity**
  - Subtle differences result in major changes in the outcome of the model.

    parameter values, behavior definition or even technical aspects

Life Turtle-Based.nlogo

Issues and challenges in Agent-Based Models

- **Tuning Micro Rules and Falsification**

  The aim of ABM is to identify micro-level behaviours that produce some macro pattern or behaviour.

  - Freedom in modeling these behaviours: calibrating, tuning, modifying parameters

  -> Different behaviours (e.g. from different theories)
  -> Non-credible behaviours
  } can produce the aimed pattern! *No unique solution*

- **Level of Detail and Number of Assumptions**
  - Every detail incorporated into a model means increasing the number of assumptions that have to be justified and explained
  - What is necessary? Where to stop?

- **Size and Scalability**
  - For many phenomena, a minimum agent number is necessary
  - Scalability of the model **and** scalability of the simulation

Bibliography / References

Complex systems

- Melanie Mitchell, **Complexity : a guided tour**, 2009.
- **Complexity: Life at the Edge of Chaos.** Roger Lewin. Dent
- **Les Systèmes complexes.** Hervé P. Zwirn. Odile Jacob
- Institut des systèmes complexes https://iscpif.fr/
- Santa Fe Institute: http://www.santafe.edu/

Multi-agents systems

- M. Wooldridge. ***An Introduction to MultiAgent Systems***, Wiley, 2002
- J. Ferber. ***Les systèmes Multi-Agents**,* InterEditions, 1995
- F. Michel, J. Ferber, A. Drogoul. ***Multi-Agent Systems and Simulation : a Survey From the Agents Community's Perspective***, 2001
- Kravari, Kalliopi and Bassiliades, Nick. ***A Survey of Agent Platforms***, Journal of Artificial Societies and Social Simulation 18 (1) 11, 2015.
- Macal, Charles, and Michael North, 2005**, *Tutorial on Agent-based Modeling and Simulation***,Proc. 2005 Winter Simulation Conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds., Orlando, FL, Dec. 4-7, pp. 2-15,

Bibliography / References

Visualisation and interactions with agent-based simulation

- Ronan Querrec. ***Apprentissage de procédures en environnements virtuels***. HdR, 2011
- Athanasia Louloudi, Franziska Klügl-Frohnmeyer. ***A New Framework For Coupling Agent-Based Simulation And Immersive Visualisation***. In ECMS, 2012.
- Gireg Desmeulles et al. ***The virtual reality applied to biology understanding: The in virtuo experimentation*.** In Expert Systems with Applications, 2006.
- Paul Guyot. ***Simulations multi-agents participatives: Faire interagir agents et humains pour modéliser, explorer et reproduire les comportements collectifs*.** Thèse, 2006
- A. Kolling, P. Walker, N. Chakraborty, K. Sycara and M. Lewis, ***Human Interaction With Robot Swarms: A Survey.*** In IEEE Transactions on Human-Machine Systems, vol. 46, no. 1, pp. 9-26, Feb. 2016.