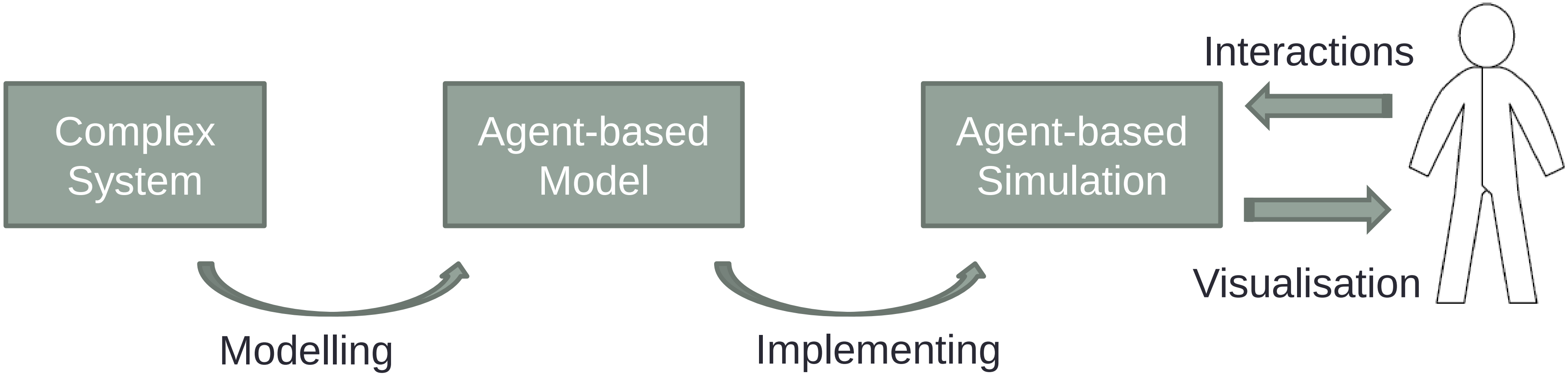
MULTI-AGENTS SYSTEMS AND INTERACTIVE SIMULATION

Complex systems
& Agent-based modelling and simulation

Jérémy Rivière
jeremy.riviere@univ-brest.fr

Département d’Informatique
INUIT Team, Lab-STICC

Introduction et Plan



Complex System

Agent-based Model

Agent-based Simulation

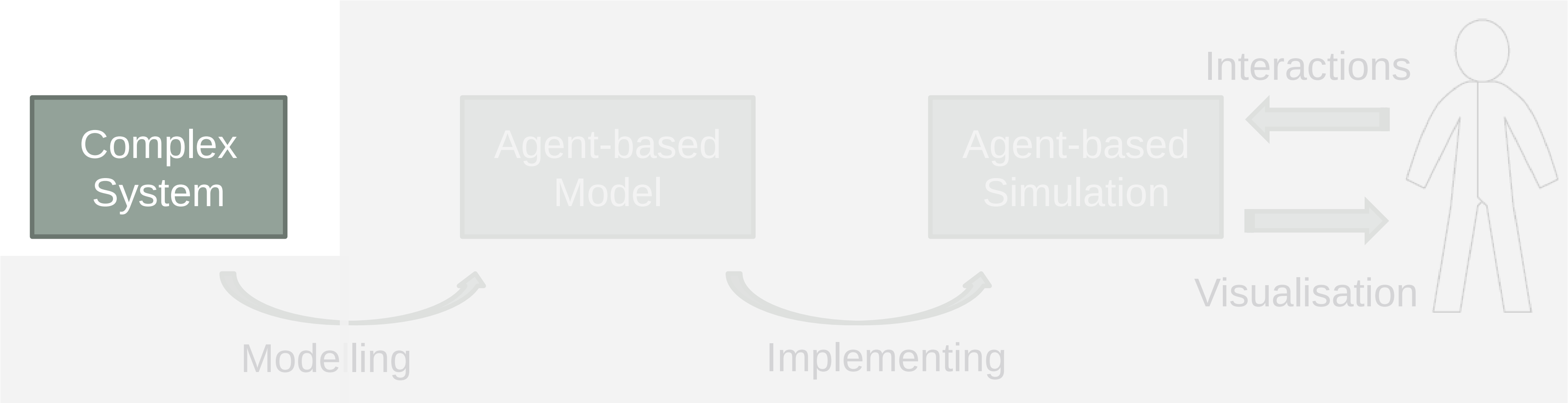
Modelling

Implementing

Interactions

Visualisation

Plan


Complex System

Agent-based Model

Agent-based Simulation

Interactions

Visualisation

Modelling

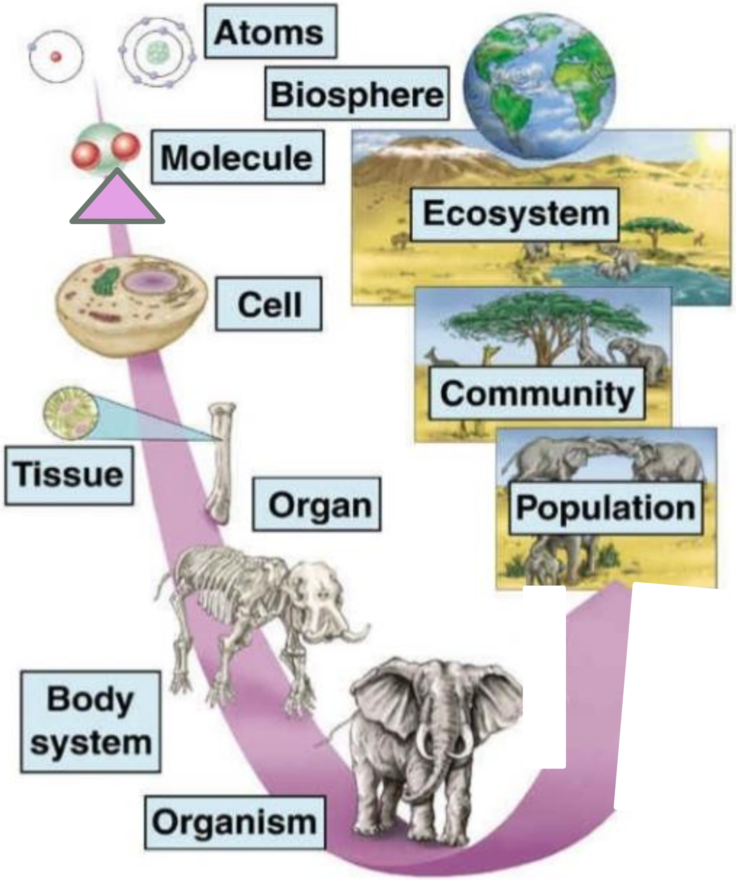
Implementing

What’s complexity?

Example:
the biologic complexity

- How to comprehend it?

Reductionism since 300 years



What's complexity?

Example:
the biologic complexity

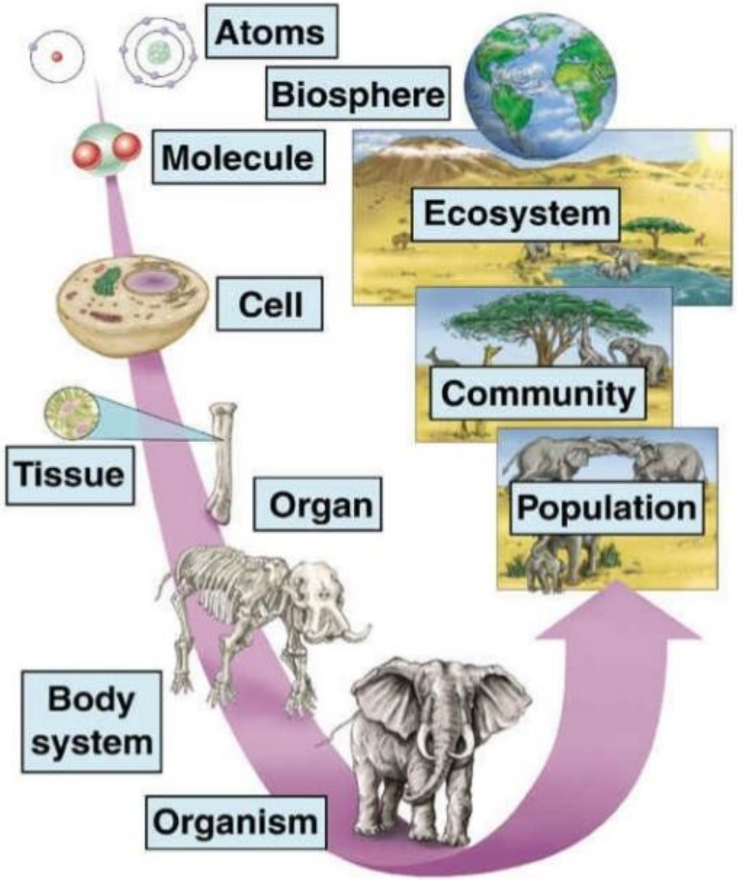
- How to comprehend it?

Reductionism since 300 years

- OK, but how do we go back up?

- It's **complex**:

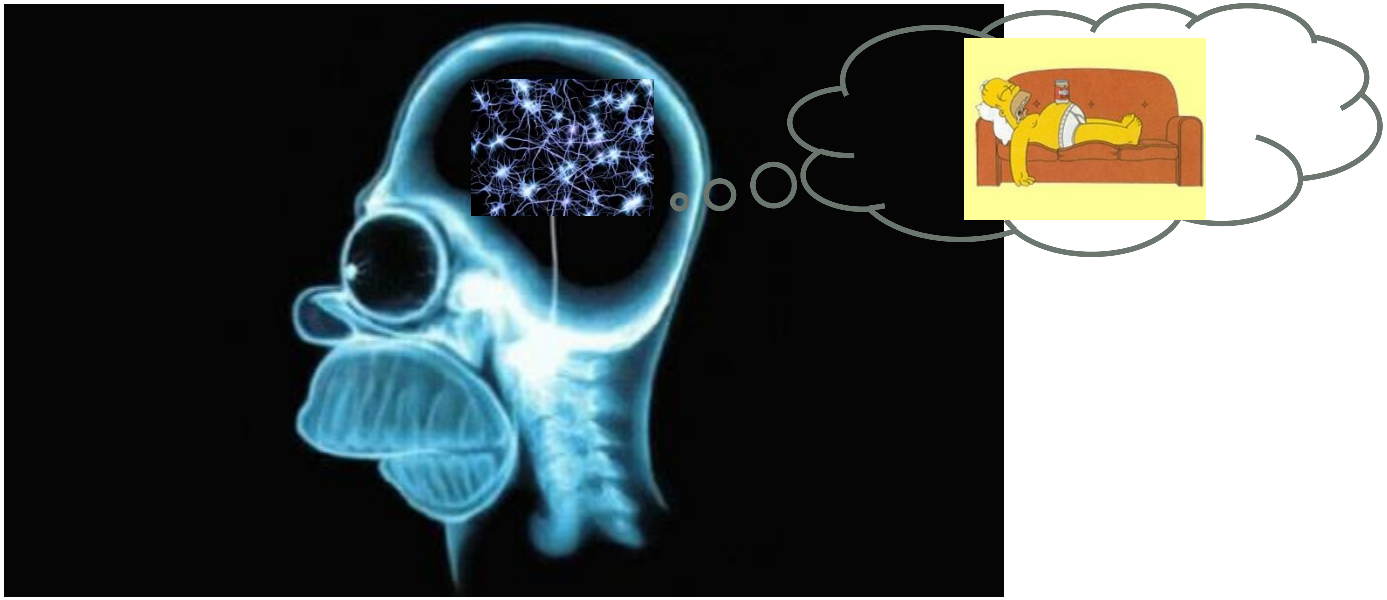
self-organisation, interdependence, positive/negative feedbacks, *emergent* properties



What’s complexity?

Another example: the **brain**

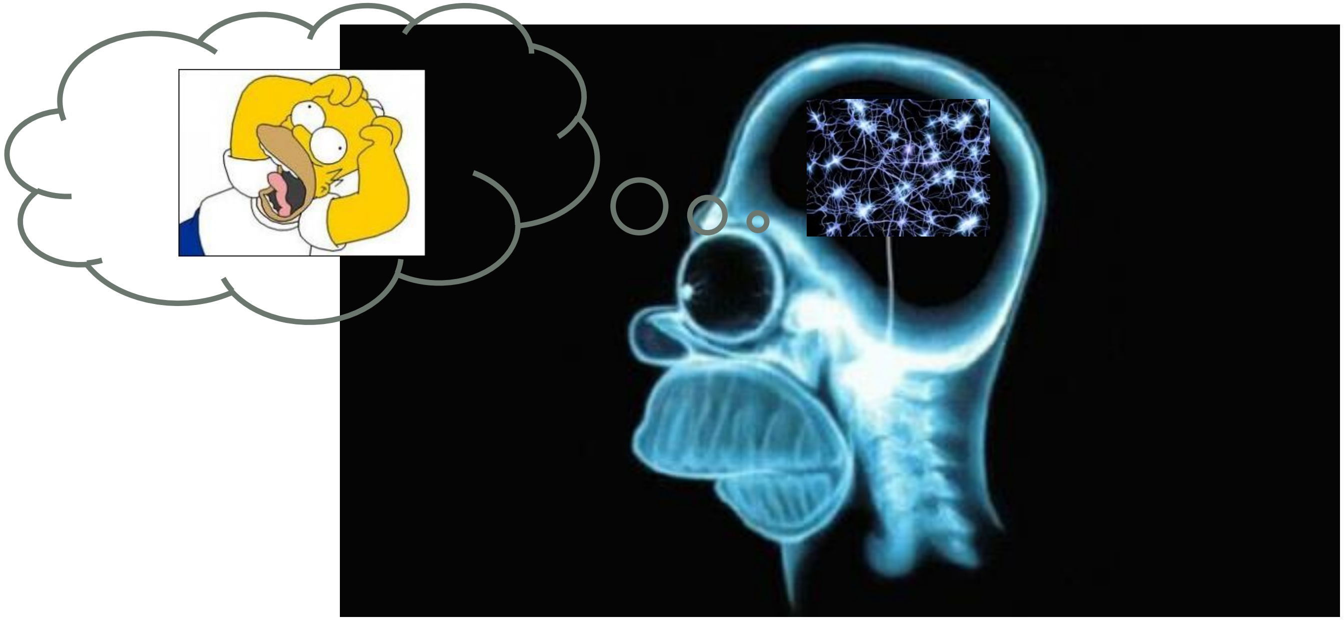
- about 10^{11} neurons
- interconnected by synapses, communicating by electrical and chemical impulses



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 and its interactions

How? (and how to verifiy it?)

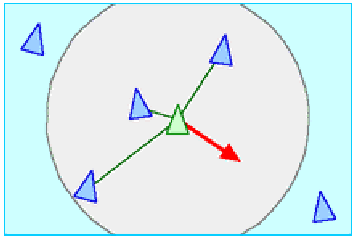
We know that 1) there is no leader and 2) no global knowledge

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

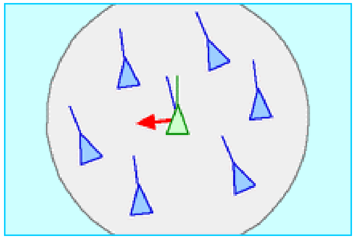
Only 3 rules to define the **behaviour** of each bird

(in order of importance)

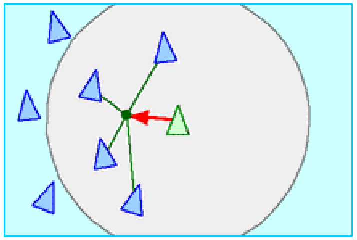
1. Separation: steer to avoid crowding local flockmates.



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.

And then, what?

- Concrete implementation: **virtual forces** or **FSM / rule-based architecture**
- **Simulation**, with for ex. the **NetLogo** platform*(http://netlogoweb.org/)*

⇒ **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).

The Heroes and Cowards Model

Example of an **artificial complex system** created from **a very simple model**

Initialisation

1. Each individual chooses someone else in the room to be their perceived friend, and someone to be their perceived enemy
2. They don't tell anyone who they have chosen

Behavior loop : they all move to position themselves either such that

  a. they are between their friend and their enemy (Brave)
  b. or such that they are behind their friend relative to their enemy (Cowardly)

What happens if

A. they all act as brave?
B. they all act cowardly?
C. half act as brave and the other half act cowardly?

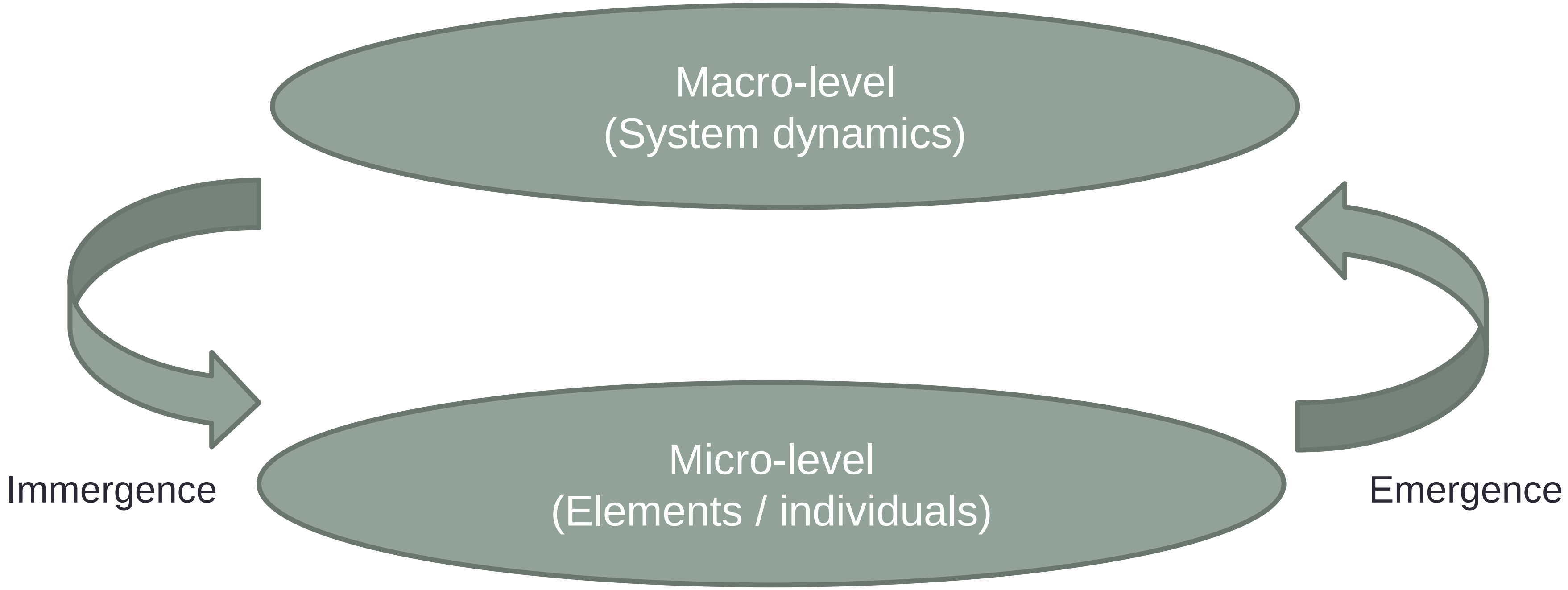
Fundamental characteristics

- **<u>Emergence</u> = the process of deriving some new and coherent structures, patterns and properties**
  - The system (macro) has properties that the elements (micro) do not have
  - Emergent phenomena are observable at a **macro-level**, even though they are generated by **micro-level** elements
  - Emergent phenomena occur due to the **non-linear and distributed interactions** between the elements of the system

Fundamental characteristics

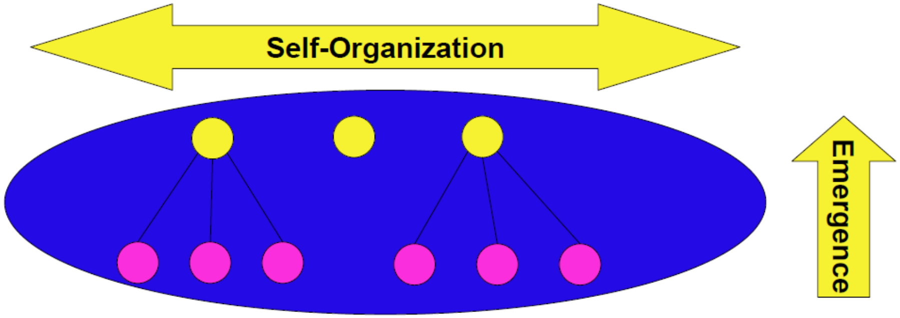
- **Emergence**

The emergent phenomena can also feedback/influence the elements
e.g. traffic jam influences drivers, stock market influences buyers…



Fundamental characteristics

- **Emergence**
- **<u>Self-organisation</u> = changes in the internal order or organisation of a system without guidance or management from an outside source**



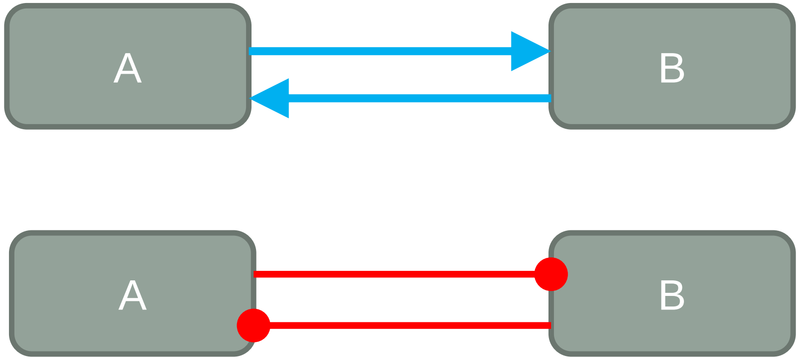
  - Neural networks, flocking behaviour, natural selection, division of labor …

Fundamental characteristics

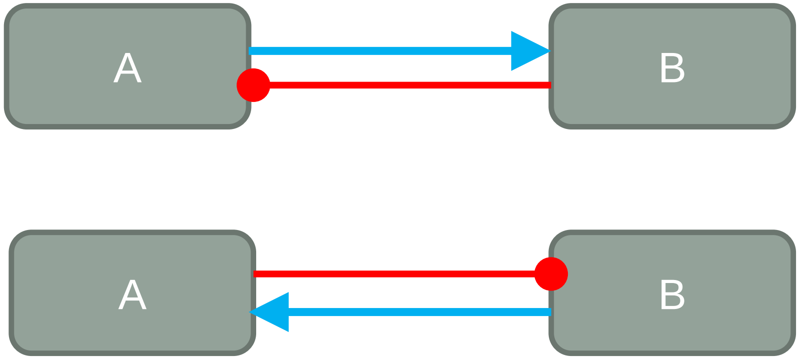
- **Emergence**
- **<u>Self-organisation</u>**

- The result of **an equilibrium** between **feedback loops** that occur due to **local interactions**

Positive feedbacks



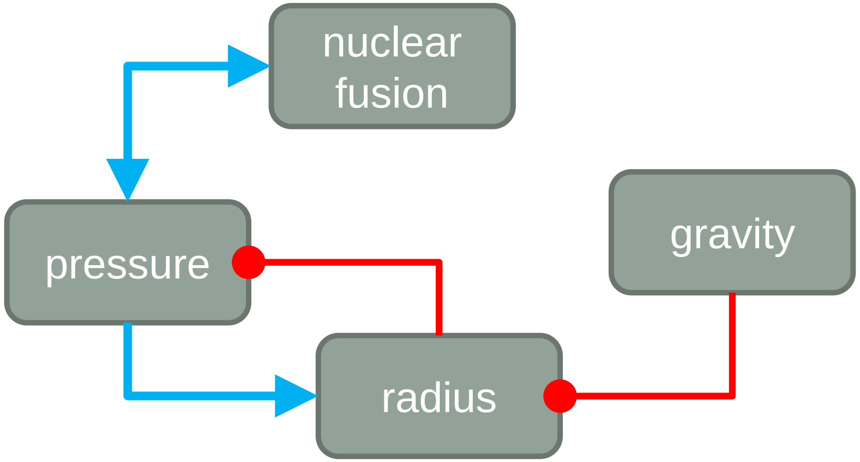
Negative feedbacks



Fundamental characteristics

- **Emergence**
- <u>**Self-organisation**</u>

- The result of **an equilibrium** between **feedback loops** that occur due to **local interactions**

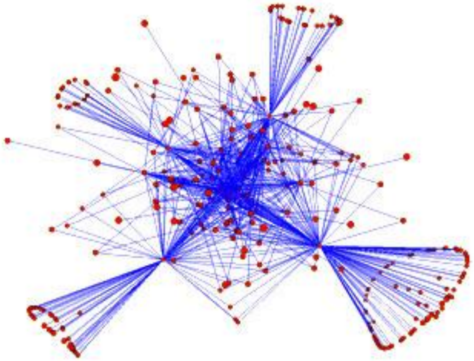


Example : hydrostatic equilibrium

Fundamental characteristics

- **Emergence**
- **Self-organisation**
- **Self-adaptation**
- **Decentralisation**
- **Non-linearity** (random, path-dependance)
- « **Butterfly effect** », **threshold effects** : small actions, big consequences

Sea otter and kelp forests



Fundamental characteristics

Example: Ant nests and "highways"

- **Emergence** of structures and global behaviour: shortest path in foraging
- **Decentralization**
- Ant **self-organization**: division of labor (roles), foraging, nest building...
- **Self-adaptation** of the system in case of disruption (attack, famine...)
- **Non-linearity** (e.g. in random food search)
- **Threshold effects**: an "highway" needs a minimum number of ants to appear

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 in computer science?
- 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

Complex systems
Models and simulations

Jacques Tisseau

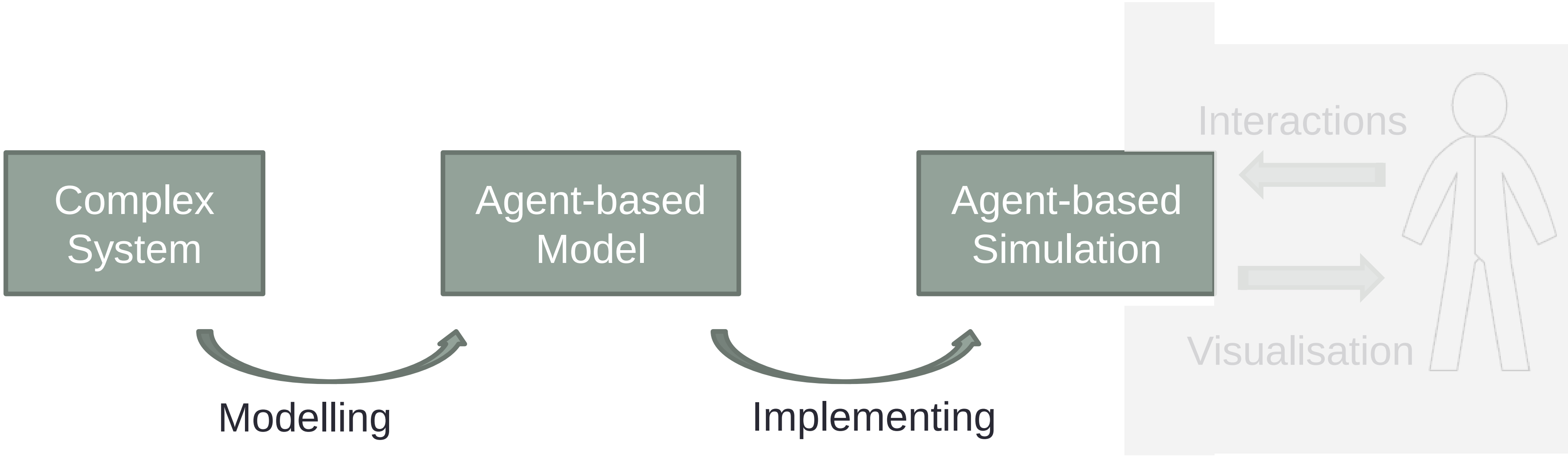
« Describing, explaining, predicting and simulating the behavior of natural or artificial complex systems is, for scientists, one of the major challenges of the 21st century »

- Modeling and simulating complex systems
  - **How?**

- Visualizing complexity, to understand or to supervise these systems
  - **How?**

- ***in virtuo* simulation**: immersive environments to **visualise** the model’s behaviour and **interact** whit the simulation

Plan



Complex System

Agent-based Model

Agent-based Simulation

Modelling

Implementing

1. What’s a model? Why modelling and simulating?
2. Modelling engineering: process and strategies
3. Focus on Agent-Based Models: case studies
4. Implementing ABM

What's a model?

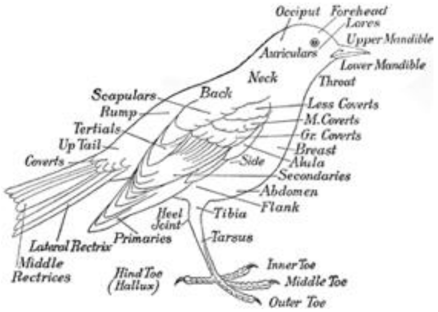
- A model is a simplified representation of reality, a tool for reflection
- One (or several) aspect(s) of the real system
- Exemples : maquette, drawing, robot, equation …

Real bird

MODELLING

Flying maquette

Digesting robot



Schematic description

" Modelling" and "simulating"

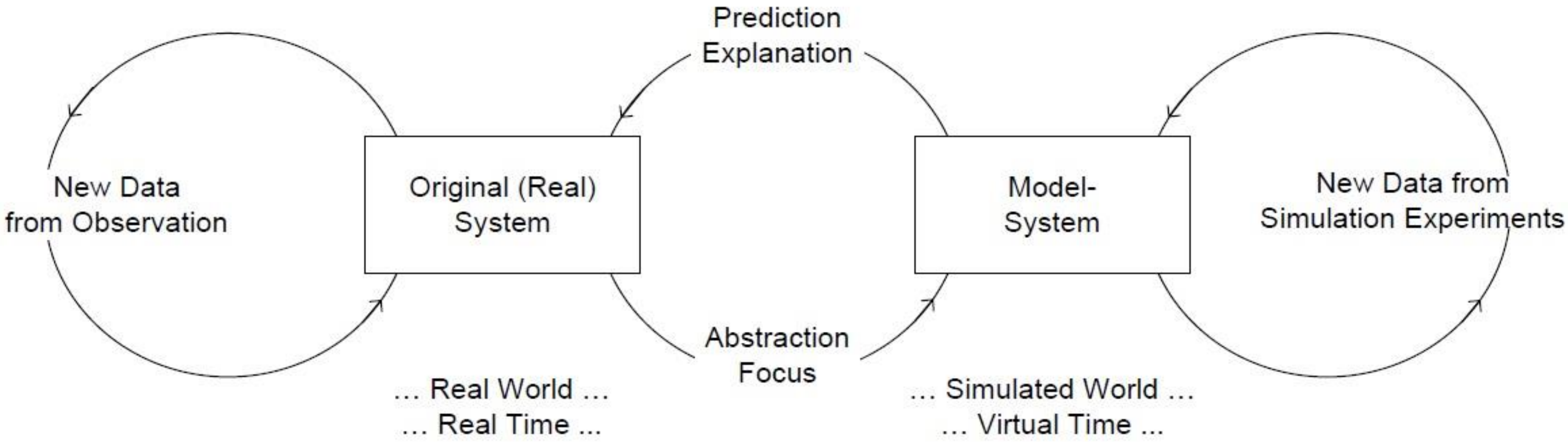
- Modelling is

- 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

" Modelling" and "simulating"



From [F. Klügl 2016]

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

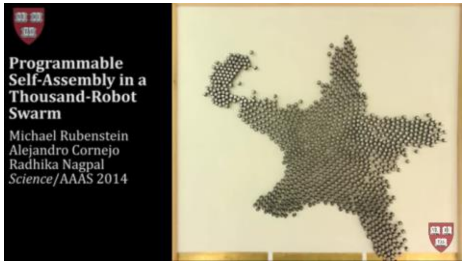
Why model and simulate…

… <u>complex systems</u>?

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

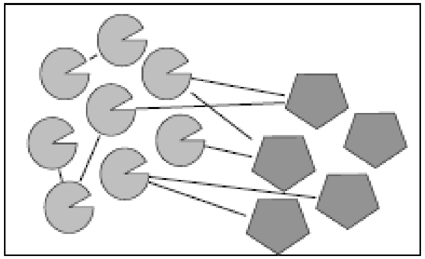
« *I can't understand what I can't create* » Richard Feynman

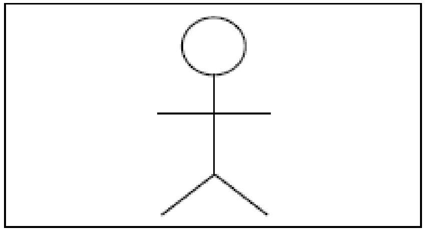
- Create artificial systems that include swarm algorithms, self-organising properties (e.g. task allocation) …

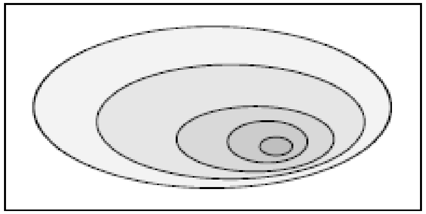


- … and **the human in interaction** with the system

Concretely
Modelling paradigms







$$y = f(x)$$

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

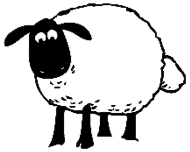
Models	Global behavior of the system
Agent-Based Models (ABM, IBM) *"bottom-up"*	Results of micro level dynamics (interactions and agents' behavior)
Participatory models played by several experts	Given by the experts
Expanded models Known configurations based on dataset	Given by the data
Population models 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 when fed
  - starve to death



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

How evolve the two populations?

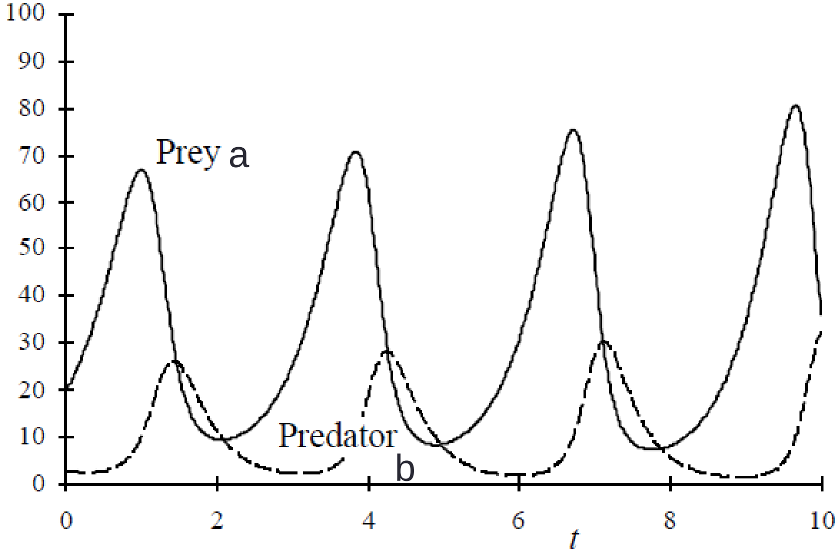
Concretely
top-down vs. bottom-up

Example: predator-prey dynamics

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

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

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



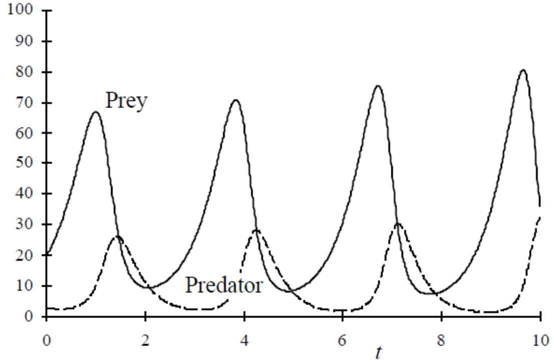
Concretely
top-down vs. bottom-up

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:

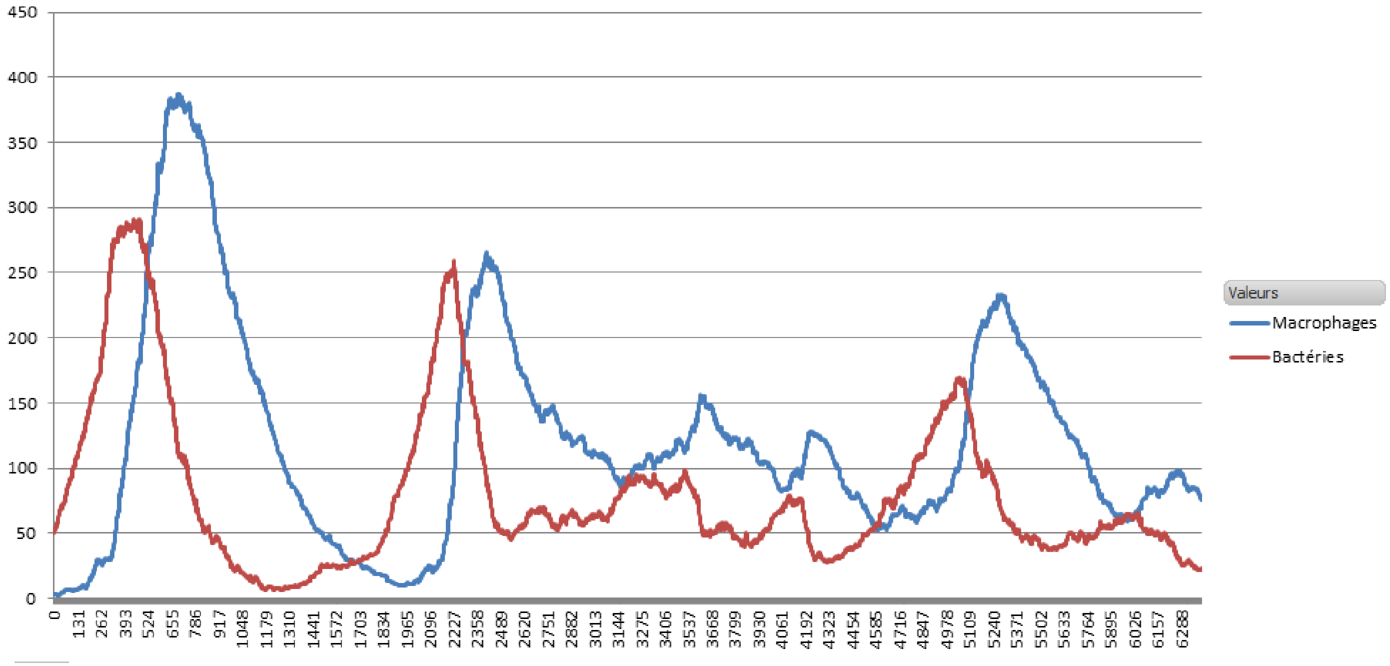
- The space/environment (location of prey, grass, fences, dry periods...)
- The diversity of individuals
- The non-deterministic nature of real life

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!

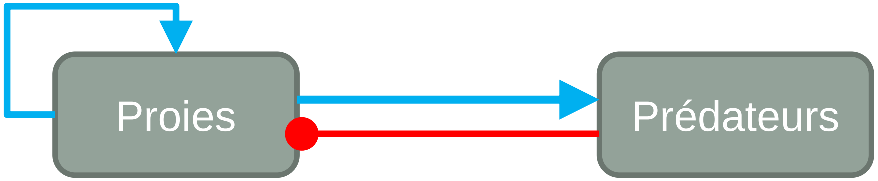


Concretely
top-down vs. bottom-up

Example: predator-prey dynamics

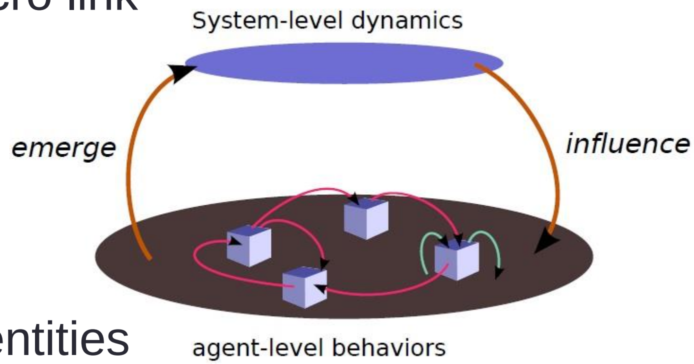
- « ***Bottom-up*** » approach: **ABM**

- **Self-organisation of the 2 populations?**



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

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 understand / 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 / understand

→ 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.

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

3. **A modelling strategy**

Modelling Strategy
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.

Modelling Strategy
KIDS

"Keep It Descriptive, Stupid"

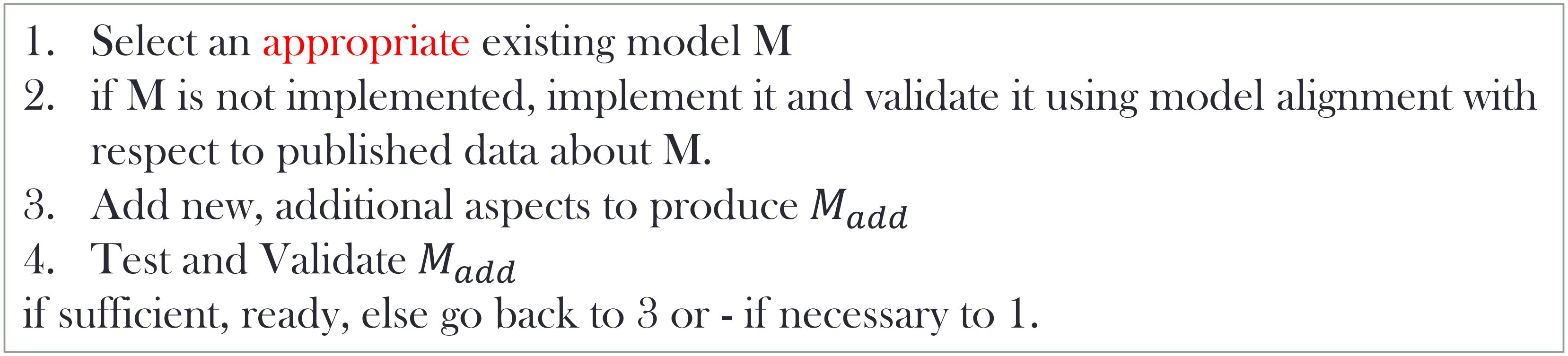
- « The point is that it is simply not appropriate to make simplifications **before** one knows what is relevant and what not ».
- Models of complex systems must sometimes contain a certain level of detail to be creditable.
- What to do if the model is not sufficiently valid in terms of aspects that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

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

3. A modelling strategy

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

Modelling and Simulating
Global process

Step 3: Results analysis

1. **Which representations** 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

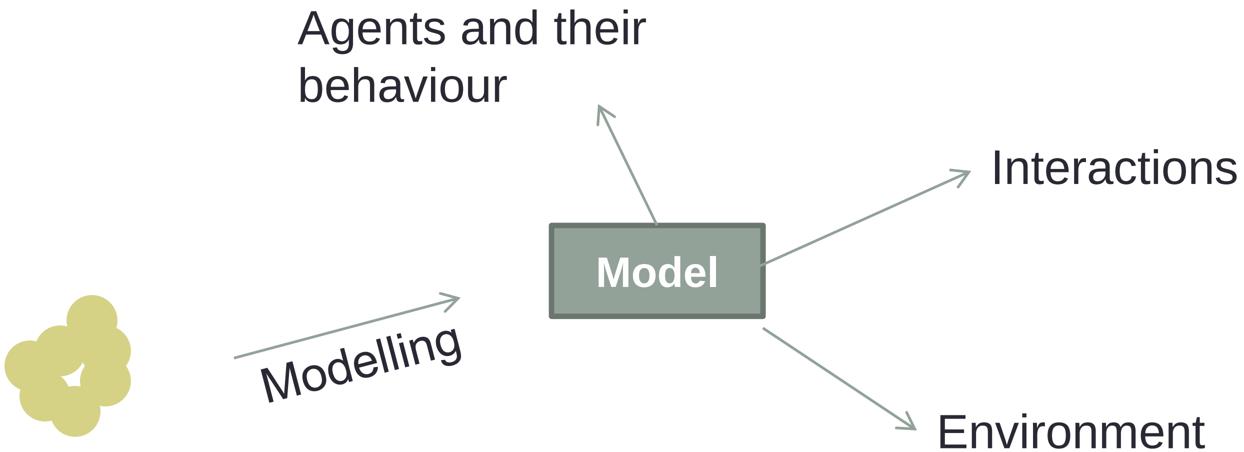
5. **Enrich the theoretical framework** with new hypotheses

6. **Publish** a scientific article!

   → Model description

Focus on Agent-Based Models
Create a model

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



Agents and their behaviour

Interactions

Model

Modelling

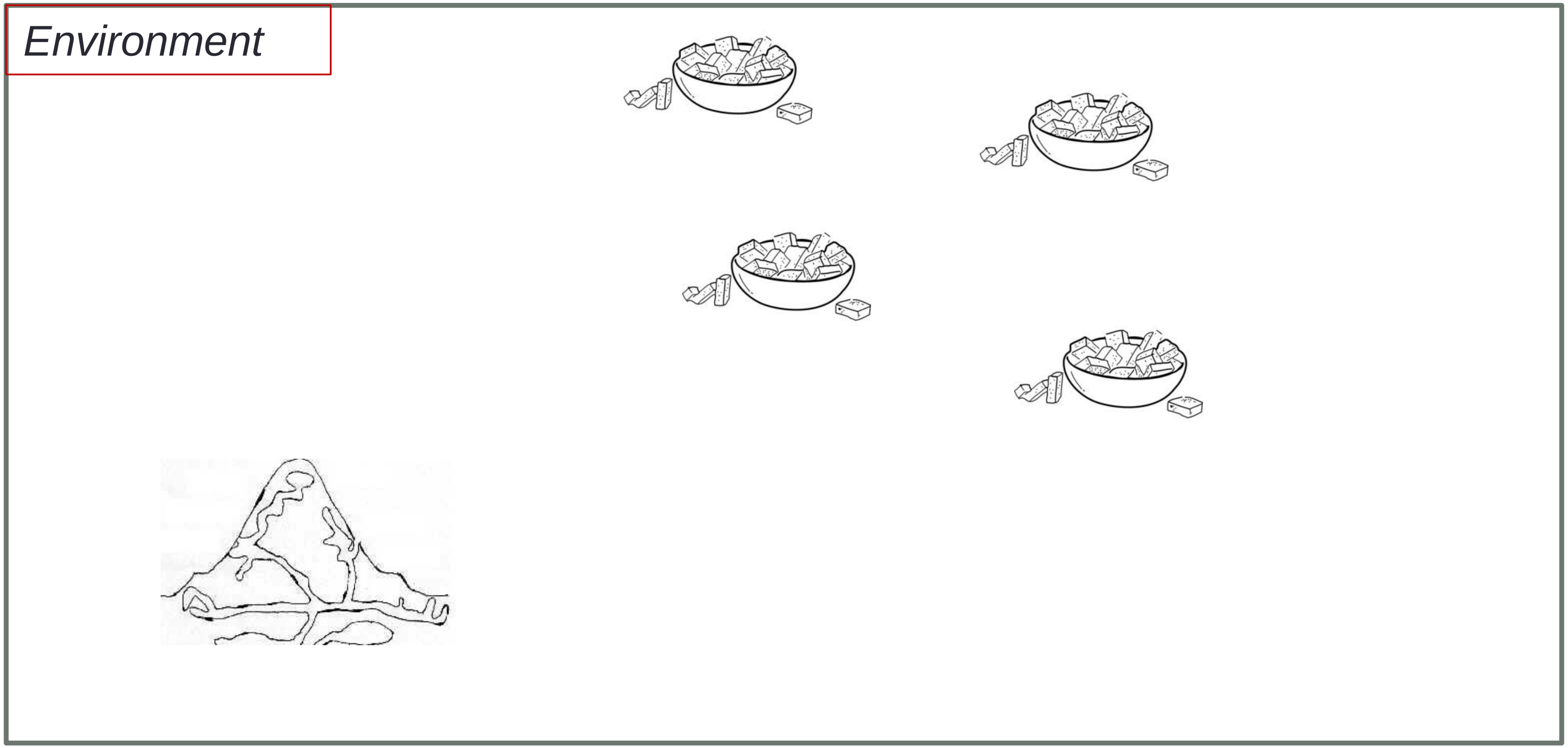
Environment

Focus on Agent-Based Models
The example of ants « highway »

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

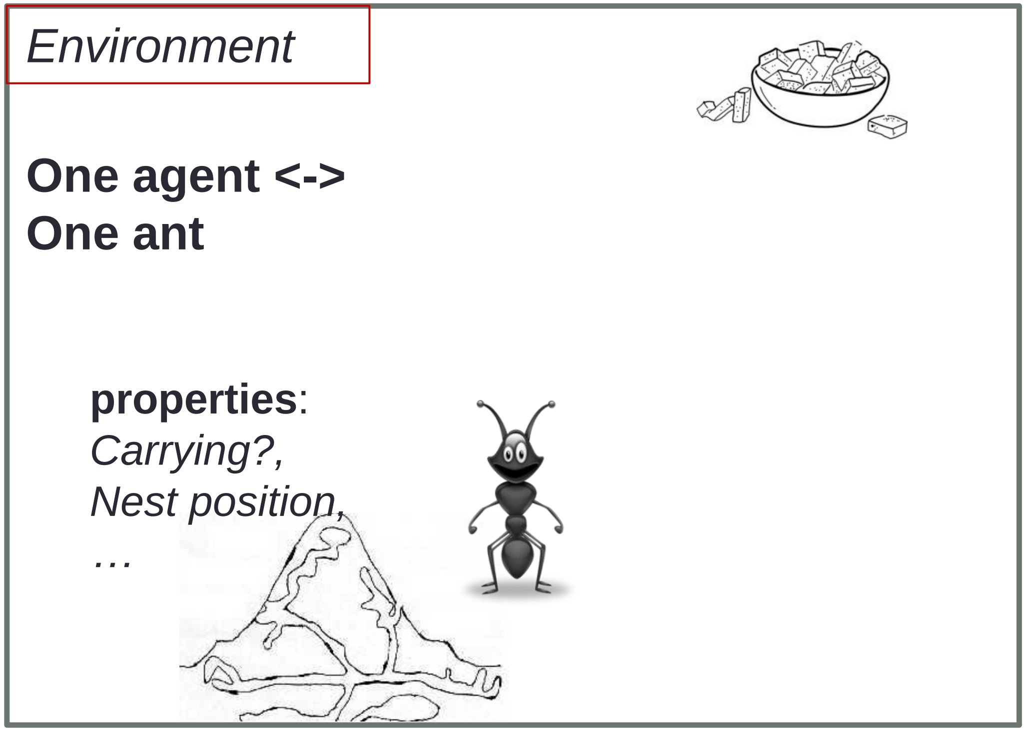
- Biology data and observations
  - Observed behaviour of ants
  - Perception and action skills
  - Indirect and simple communication by pheromones
  - Evaporation of pheromones over time

Focus on Agent-Based Models
The example of ants « highway »



Environment

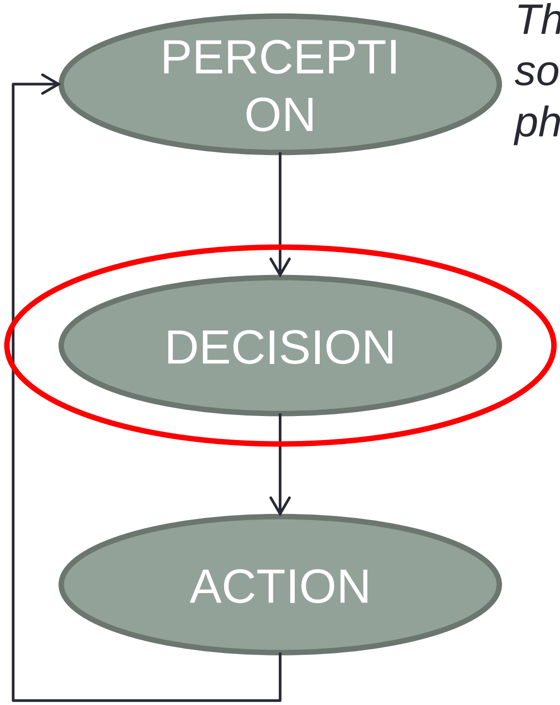
Focus on Agent-Based Models
The example of ants « highway »



Environment

One agent <-> One ant

properties:
Carrying?,
Nest position,
…

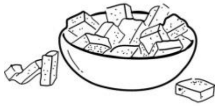


The nest, food sources, pheromone

Moving, Collecting / Depositing food, Depositing pheromone

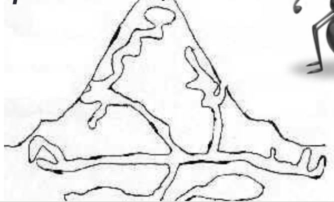
Focus on Agent-Based Models
The example of ants « highway »

Environment



One agent <->
One ant

properties:
Carrying?,
Nest position,
…

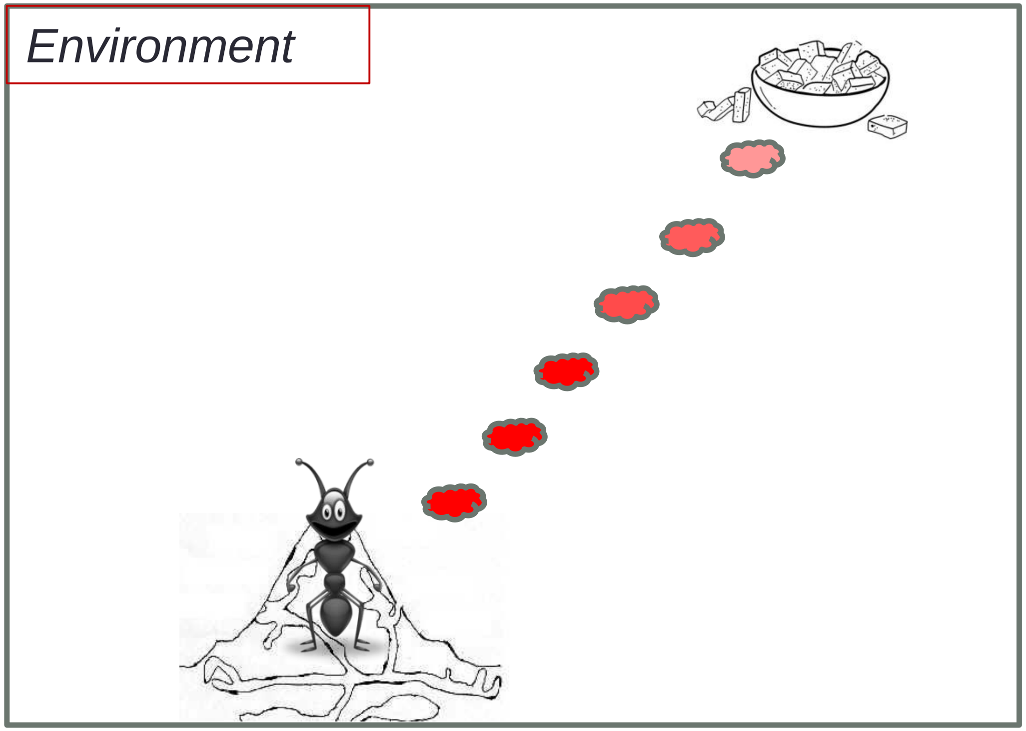


Objectives

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

-> Decision rules

Focus on Agent-Based Models
The example of ants « highway »



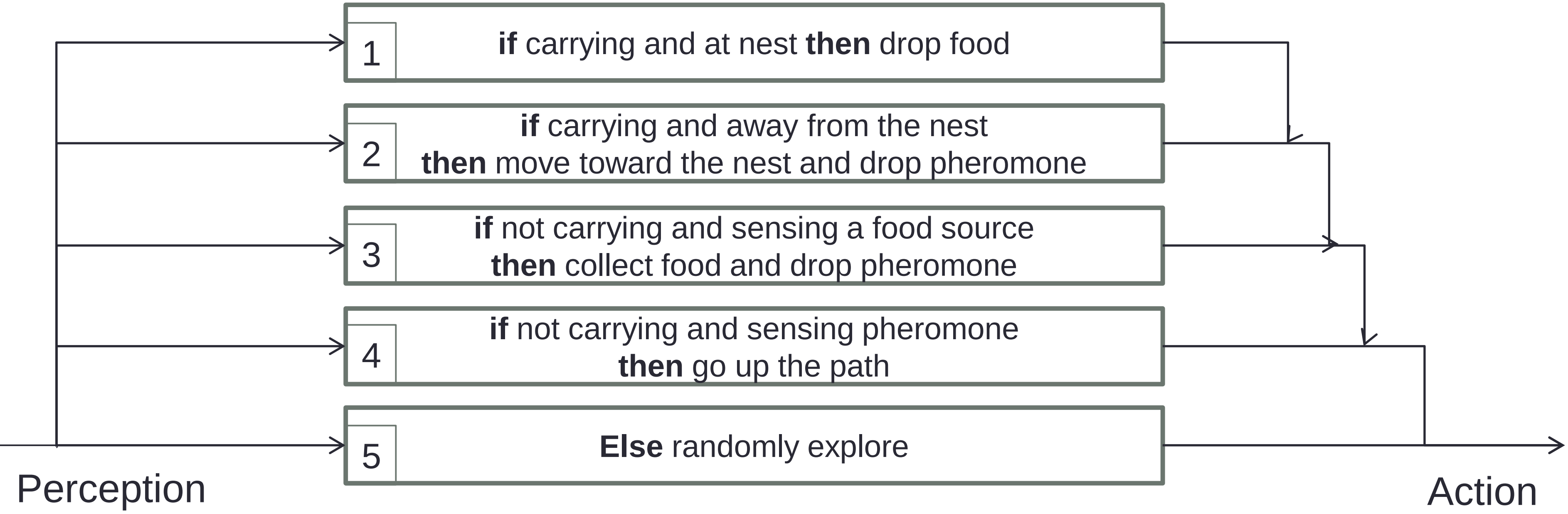
Rules for the agent's decision:

- 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

Focus on Agent-Based Models
The example of ants « highway »

Organising the decision rules

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



1 **if** carrying and at nest **then** drop food

2 **if** carrying and away from the nest **then** move toward the nest and drop pheromone

3 **if** not carrying and sensing a food source **then** collect food and drop pheromone

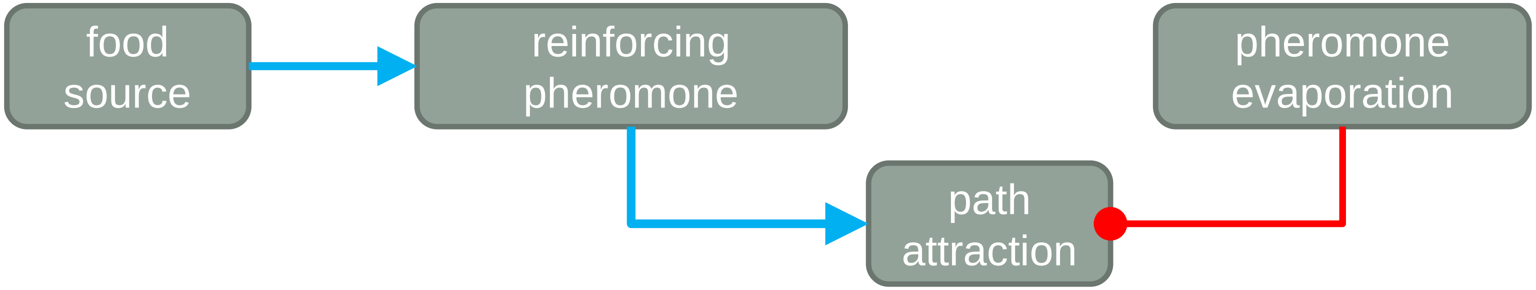
4 **if** not carrying and sensing pheromone **then** go up the path

5 **Else** randomly explore

Focus on Agent-Based Models
The example of ants « highway »

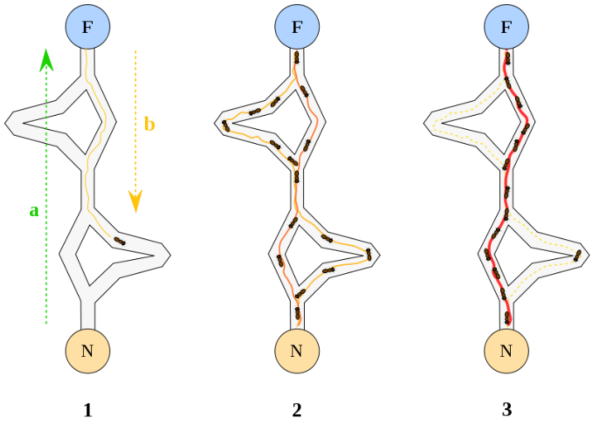
Self-organisation

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


food source
reinforcing pheromone
pheromone evaporation
path attraction

Emergence

- **Shortest path!** The *ant algorithm*



Wikipedia

Focus on Agent-Based Models
The example of ants « highway »

What have we done?

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

- **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

Focus on Agent-Based Models
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 behavior description: observation, literature, domain experts

Different tasks: exploration, foraging, recruitment and looking for dances

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 behavior description
   Different tasks: exploration, foraging, recruitment and looking for dances

2. Categorize agents: how many classes or types of agents are necessary? where is the heterogeneity?

only one type of simple (reactive) agent, the foraging bee

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 behavior description
   Different tasks: exploration, foraging, recruitment and looking for dances

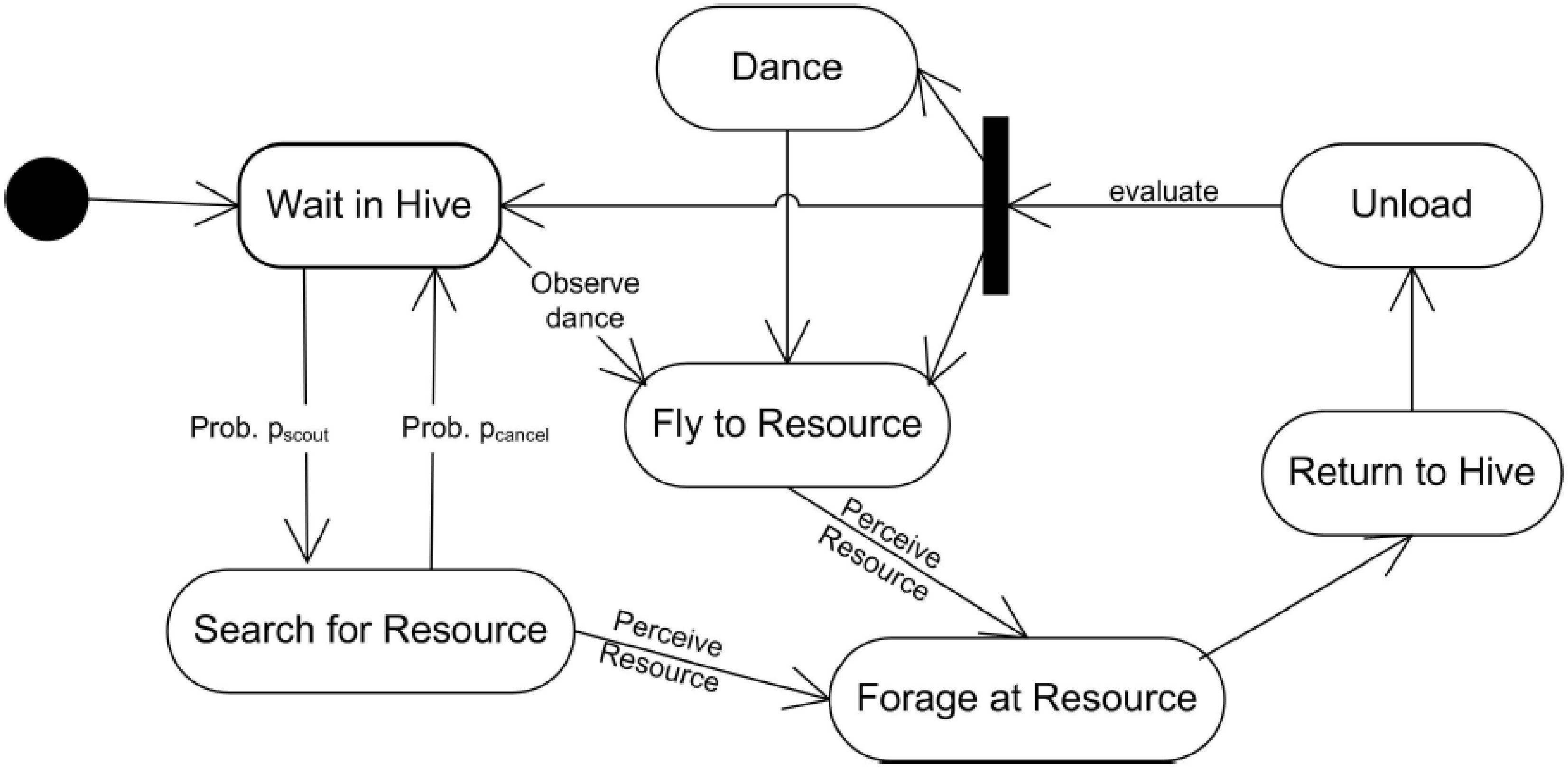
2. Categorize agents: only one type of simple (reactive) agent, the foraging bee

3. Formalize and implement agent’s behavior and goals

→ Activity diagrams

Case study

Agent-driven model design



Case study
Agent-driven model design

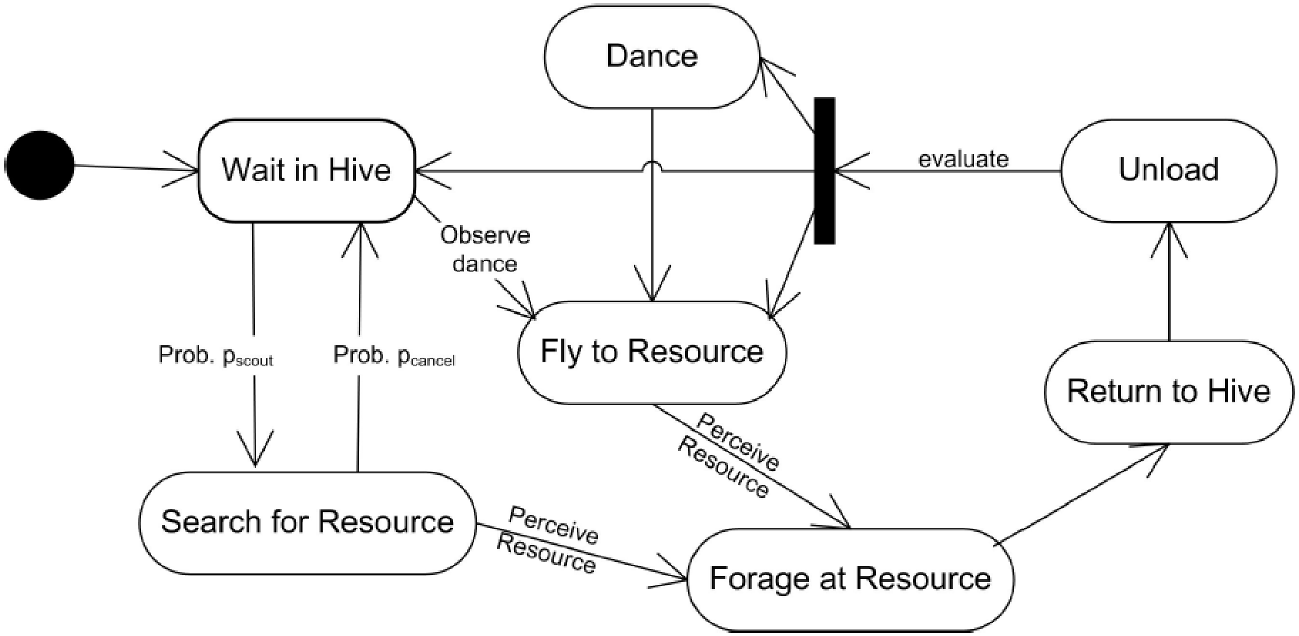
Focus on the agents, their behavior and their decision making mechanisms

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

➢ **Basic strategy**

1. Agent observation and behavior description
   Different tasks: exploration, foraging, recruitment and looking for dances
2. Categorize agents: only one type of simple (reactive) agent, the foraging bee
3. Formalize and implement agent’s behavior and goals → Activity diagrams
4. Add interactions and environmental aspects **when needed**

Case study
Agent-driven model design



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

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 behavior description
     Different tasks: exploration, foraging, recruitment and looking for dances
  2. Categorize agents: only one type of simple (reactive) agent, the foraging bee
  3. Formalize and implement agent's behavior and goals → Activity diagrams
  4. Add interactions and environmental aspects **when needed**
  5. Test (simulation) whether necessary macro-phenomena are sufficiently reproduced

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

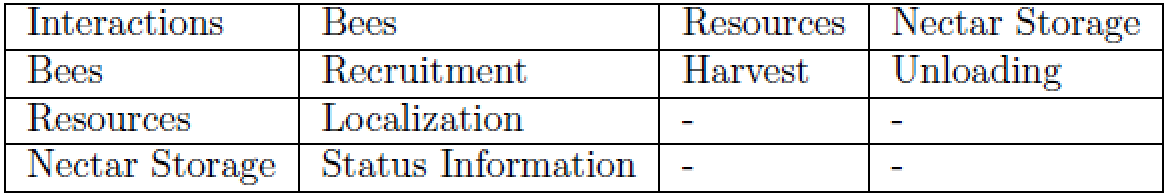
Focus on interactions, at the micro **and** macro levels

- **View from above** (bird perspective)
- Agents seen as black boxes producing messages → Actors / Entities
- Everything is an agent!

Case study
Interaction-driven model design

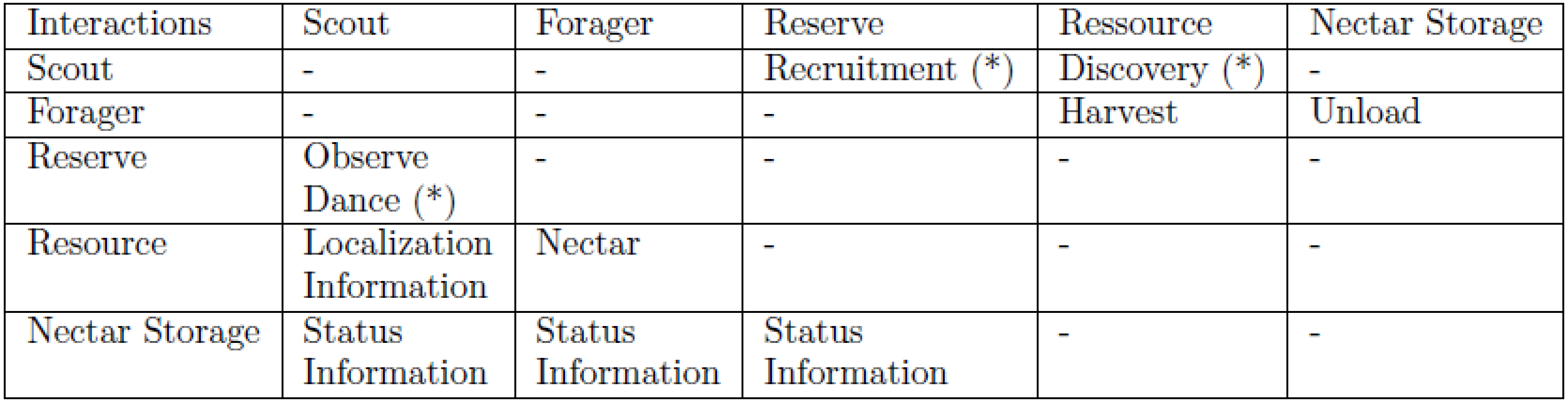
- **Basic strategy**

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	-	-

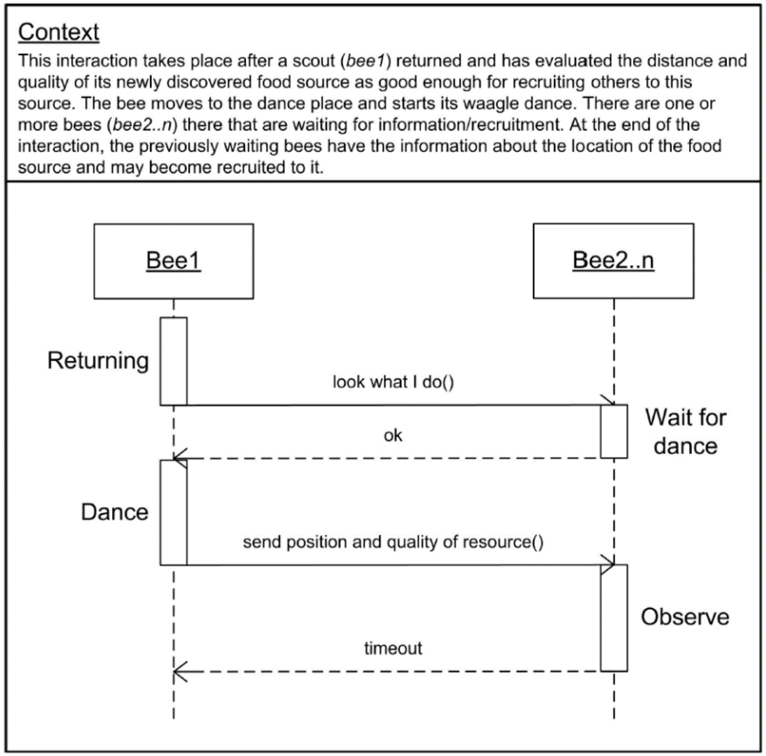
Case study
Interaction-driven model design

- **Basic strategy**

1. Identify actors/entities and interactions among them → Interaction table

2. Coarse description of **protocols** (actors + interactions) and their conditions, constraints etc.

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.

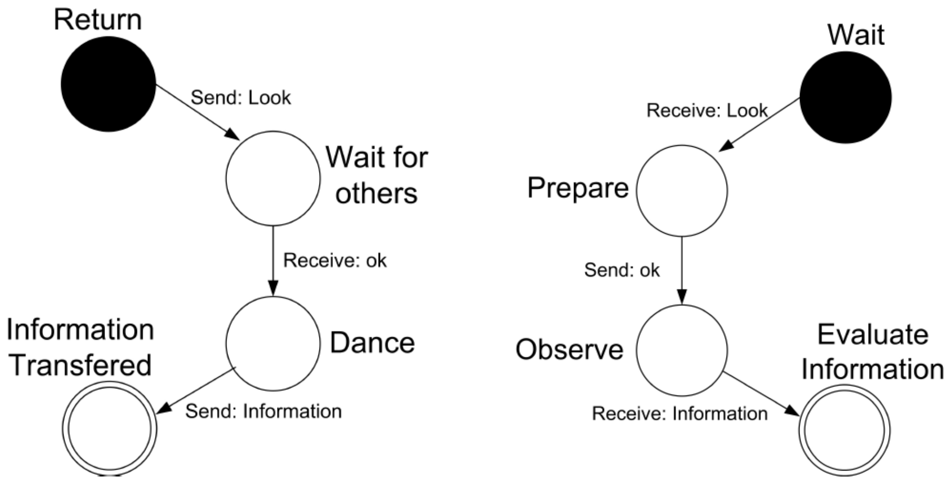
Case study
Interaction-driven model design

➢ **Basic strategy**

1. Identify actors/entities and interactions among them → Interaction table

2. Coarse description of **protocols** (actors + interactions) and their conditions, constraints etc.

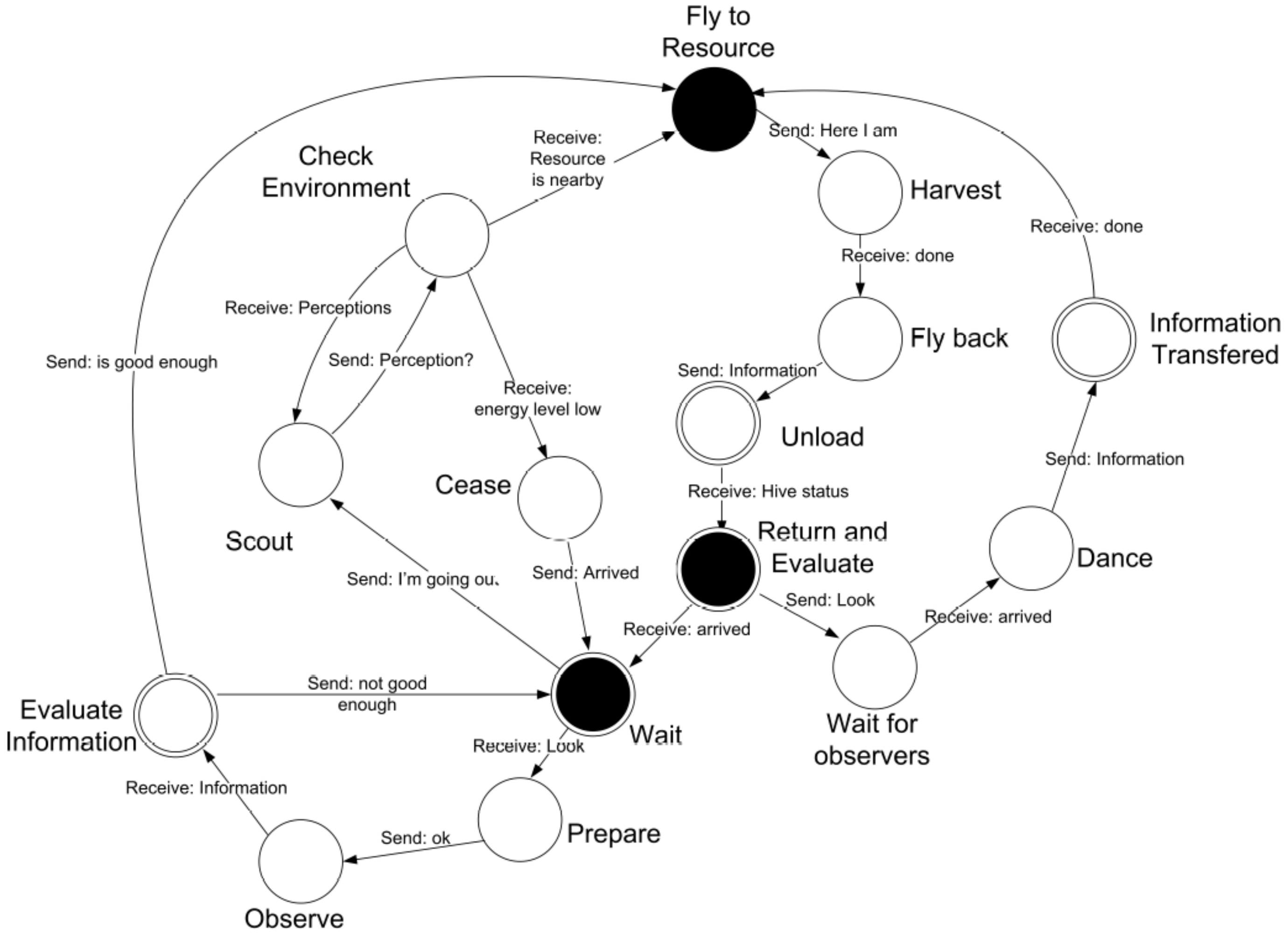
3. Add environment entities and derive the agent's behavior for producing the atomic interaction elements
   - from each interaction protocol
   - then unify to produce only one behaviour
   - using, for example, finite-state machines



(a) Recruiter (b) Recruitee

Case study

Interaction-driven model design



Case study
Interaction-driven model design

- **Basic strategy**

1. Identify actors/entities and interactions among them → Interaction table
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)

Interaction-driven model design Discussion

- Dependencies / effects of interactions on agent behaviour are explicitly represented

- Problem with proactive behaviours, not triggered by external messages

- 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 environment

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

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

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 environment
2. **Determine a) the primitive actions** of the agent and the **reaction** of the environmental entities and b) what information from the environment must be given to the agent for its decision making

→ **Perceptions** and **actions**

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 environment
  2. **Determine a) the primitive actions** of the agent and the **reaction** of the environmental entities and b) what information from the environment must be given to the agent for its decision making

Perceptions: *resource existence (from a certain distance), its position and capacity (if nearby), hive storage (if nearby), etc.*

Actions: *Fly towards perceived resource, towards the hive, load / unload nectar*, etc.

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 environment
2. **Determine a) the primitive actions** of the agent and the **reaction** of the environmental entities and b) what information from the environment must be given to the agent for its decision making
3. Decide on an agent architecture that is apt to **connect perceptions and actions** of the agent appropriately for actually producing the agents behavior

→ **Rule-based approach** for the agent’s behaviour

Case study

Environment-driven model design

Focus on the environment in which agents operate

Hierarchical rules of decision

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)

Case study
Environment-driven model design

Focus on the environment in which agents operate

- Interactions and agents are secondary, and added when needed

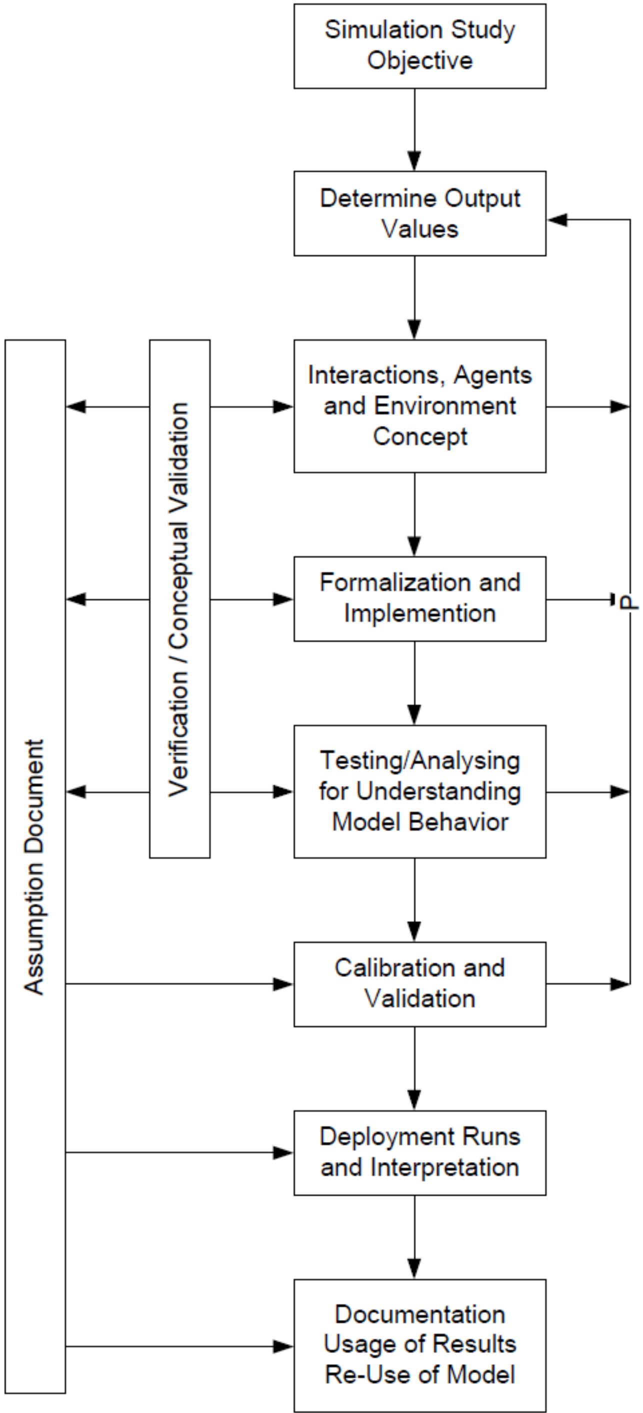
➢ **Basic strategy**

1. Identify relevant aspects (global status, global dynamics/ local entities) of the environment
2. **Determine a) the primitive actions** of the agent and the **reaction** of the environmental entities and b) what information from the environment must be given to the agent for its decision making
3. Decide on an agent architecture that is apt to **connect perceptions and actions** of the agent appropriately for actually producing the agents behavior
4. Implement the environmental model and the agent's behaviour
5. Test and analyze the overall simulation results and individual trajectories

Environment-driven model design
Discussion

- Possibility to add a **learning mechanism** (reward based) to determine the actual agent behavior
- How to choose the level of detail of the environment?
- As with the interaction-oriented approach, problem with proactive behaviours, not triggered by external stimuli
- Complexity of rules, sometimes → another agent architecture (FSM …)

MAS Modelling
- To sum up



Simulation Study Objective

Determine Output Values

Interactions, Agents and Environment Concept

Formalization and Implemention

Testing/Analysing for Understanding Model Behavior

Calibration and Validation

Deployment Runs and Interpretation

Documentation
Usage of Results
Re-Use of Model

Verification / Conceptual Validation

Assumption Document

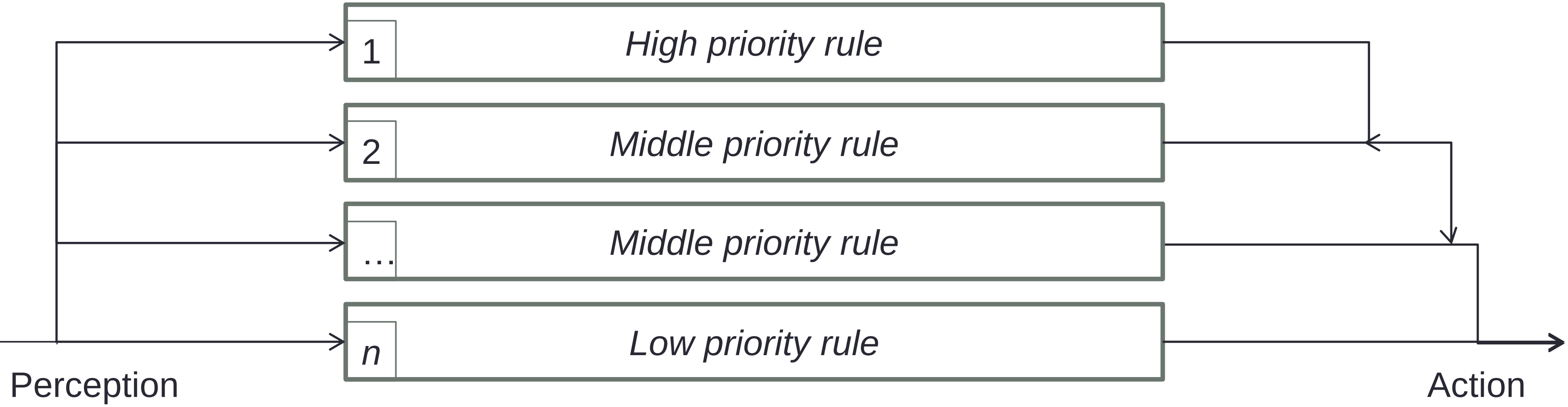
P

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>**

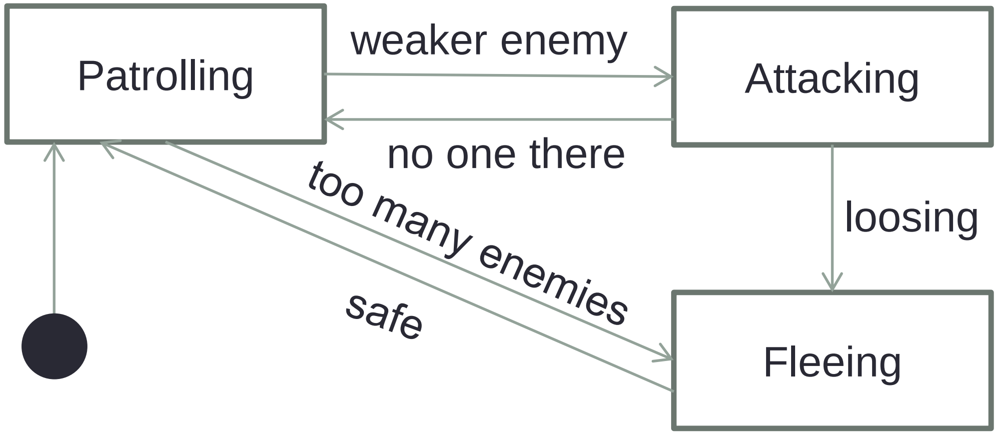
Only one rule applied per decision taken!



MAS Modelling - To sum up
Formalizing the agent behavior

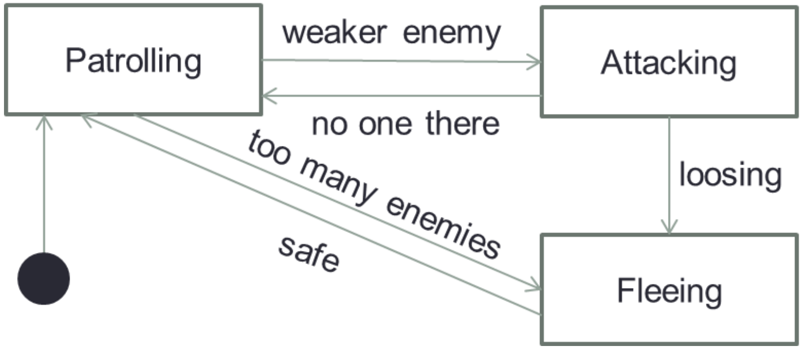
2. Activity-based architecture: **finite-state machine**

- **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



MAS Modelling - To sum up
Formalizing the agent behavior

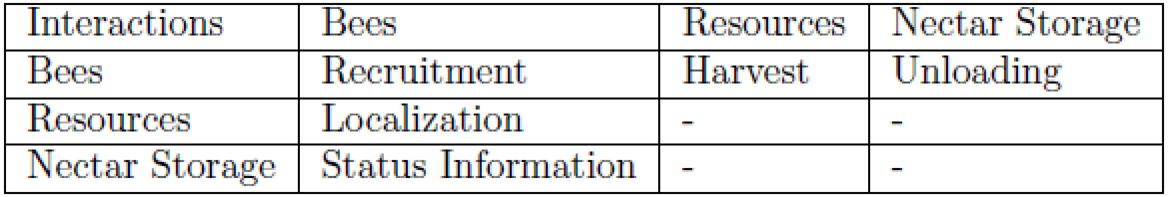
2. Activity-based architecture: **finite-state machine**



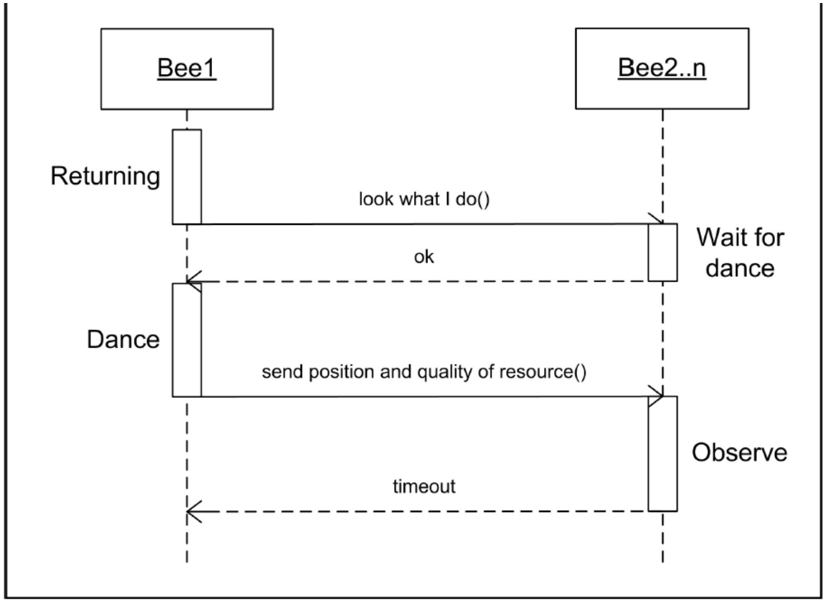
Current State (Activity)	Action	Condition (Event)	Next State
Fleeing	Flee the enemies	Safe	Patrolling
Patrolling	Patrol the area	Too many (strong) enemies	Fleeing
		A few (weak) enemies	Attacking
Attacking	Fight the enemies	Enemies are defeated	Patrolling
		Enemies are stronger	Fleeing

MAS Modelling - To sum up
Formalizing the interactions

Interaction table, communication protocols



Interactions	Bees	Resources	Nectar Storage
Bees	Recruitment	Harvest	Unloading
Resources	Localization	-	-
Nectar Storage	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, number of dimensions, how the dynamic works …

Your turn

The following behaviour has been observed in termites:
Each termite begins to randomly search for a wood chip. When a termite encounters a wood chip (isolated or surrounded by others), it picks it up, moves away and then randomly looks for another chip. If the termite encounters another chip, it deposits the one it is carrying in a free place next to it, then moves away again and starts over.

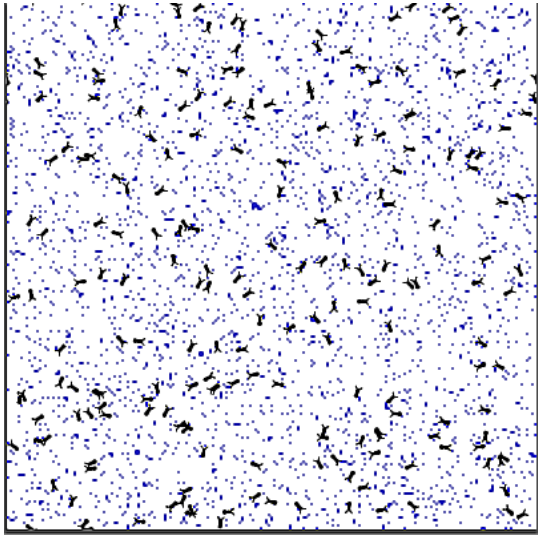


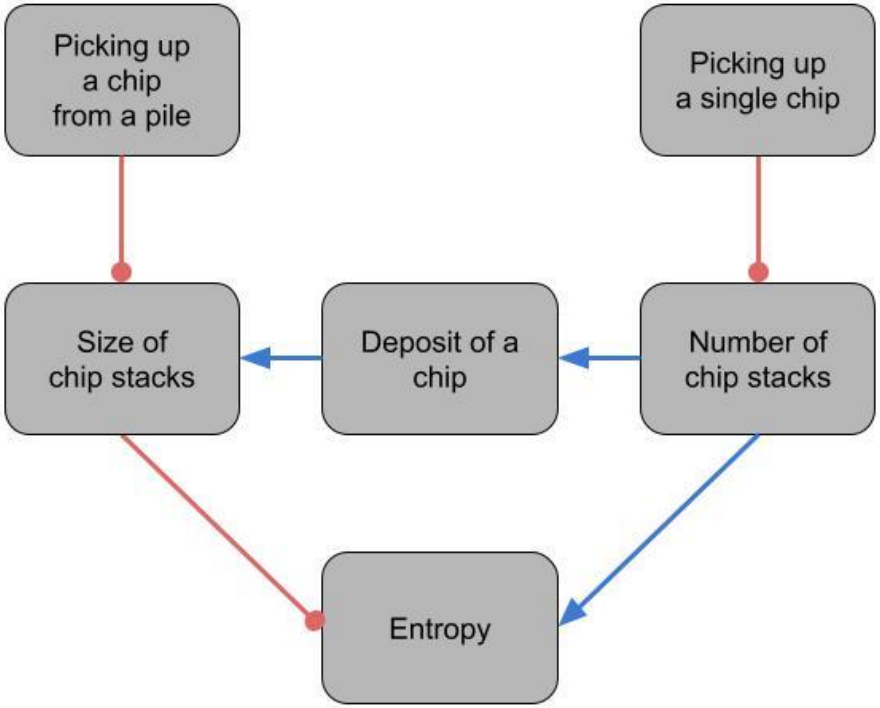
Figure 1: Simple 2D environment with termites and wood chips

Environment-driven model design

1. Identify relevant aspects of the environment
2. Give the action and perception capacities of a termite agent in this environment.
3. Use a FSM or a subsomption architecture to describe the behaviour of the termite agent.
4. If Figure 1 is the initial state of a simulation, try to imagine and <u>draw</u> an example of the final state, justifying your answer. How do the wood chips end up ?

5. Try to draw the equilibrium between feedback loops that occur due to local interactions → self-organisation

5. Try to draw the equilibrium between feedback loops that occur due to local interactions → self-organisation



Implementing ABM
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 …)

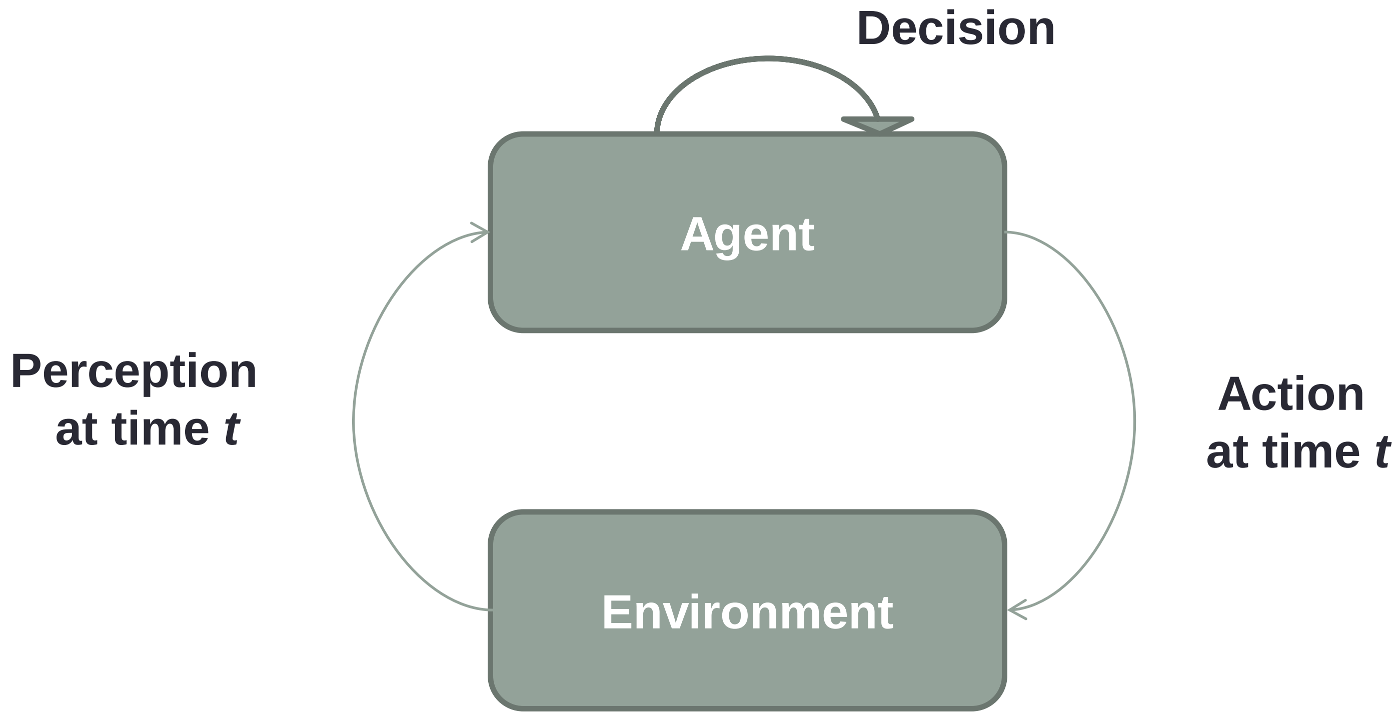
Agents, like us, are supposed to act and interact **concurrently** (principle of **causality**)

- **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

Problematics

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



Decision

Agent

Perception at time *t*

Action at time *t*

Environment

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 each ms (real time)**

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

- Correspondence between real time and virtual time (rule of three)
  - For example, 1 time step = 1 ms (real time) = 1 second of simulated virtual time
  - To simulate 100 time steps **takes** 100 ms and **represent** 100 seconds of virtual 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!

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?

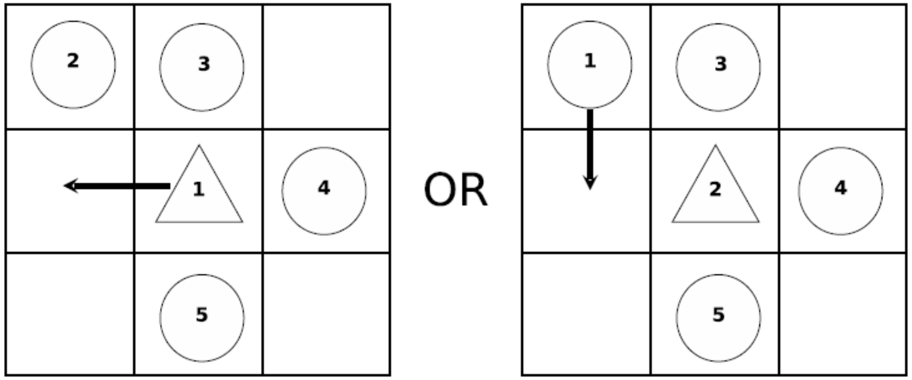
- Scheduling solution in discrete time simulations

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

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

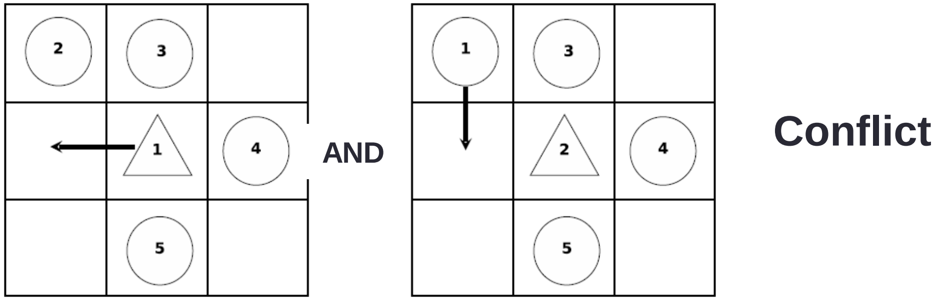
- Solution 1 : **shuffle** the agents list every time step

shuffle(AllTheAgents)

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

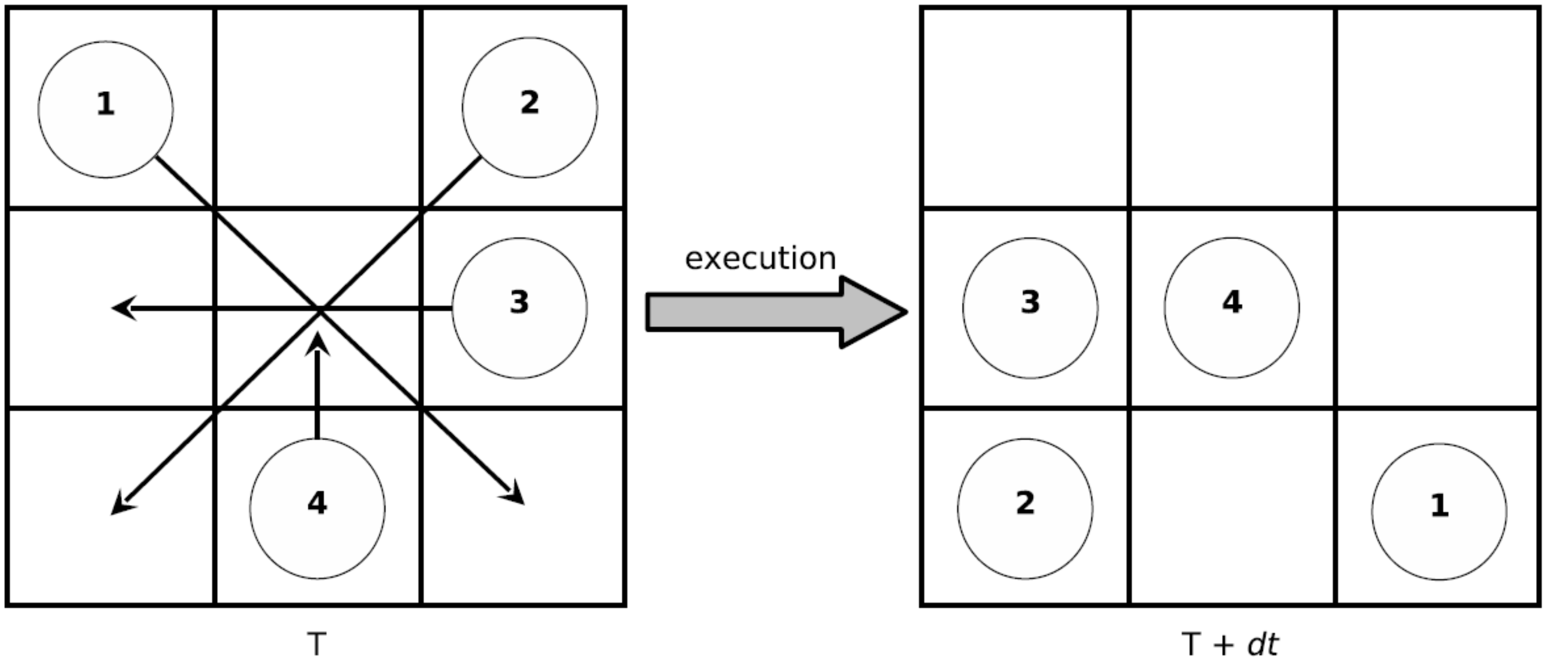
- Solution 2 : **simulate concurrency**
    - all the agents operate on temporary variables
    - the perceived environment is the same for all the agents
    - once done, the next system state is computed

BUT we have to solve the conflicts



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?”



ABMs implementation

- **Object-oriented** programming languages are particularly suitable for agents: Java, C++, C# (Unity), Python (Pygame) ...

- 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 …)

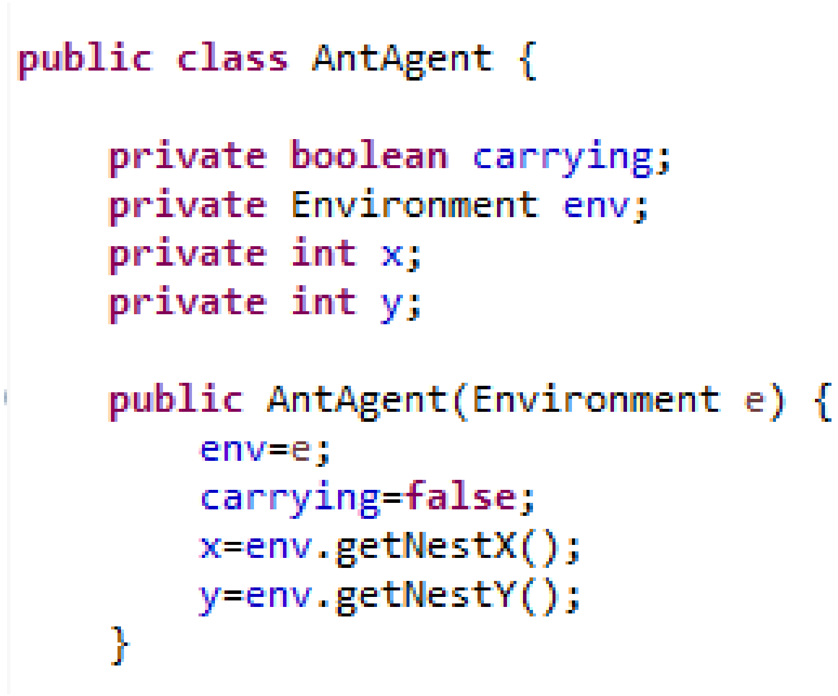
ABMs implementation

- Many platforms exist!
  - Generic: JADE, Jason, MadKit ...
  - **Dedicaded for simulation**: NetLogo, **Gama**, RePast,, Flame, Centyllion...
  - 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)
  - Some interaction means
  - Agent programming languages

Agents and Object-oriented programming

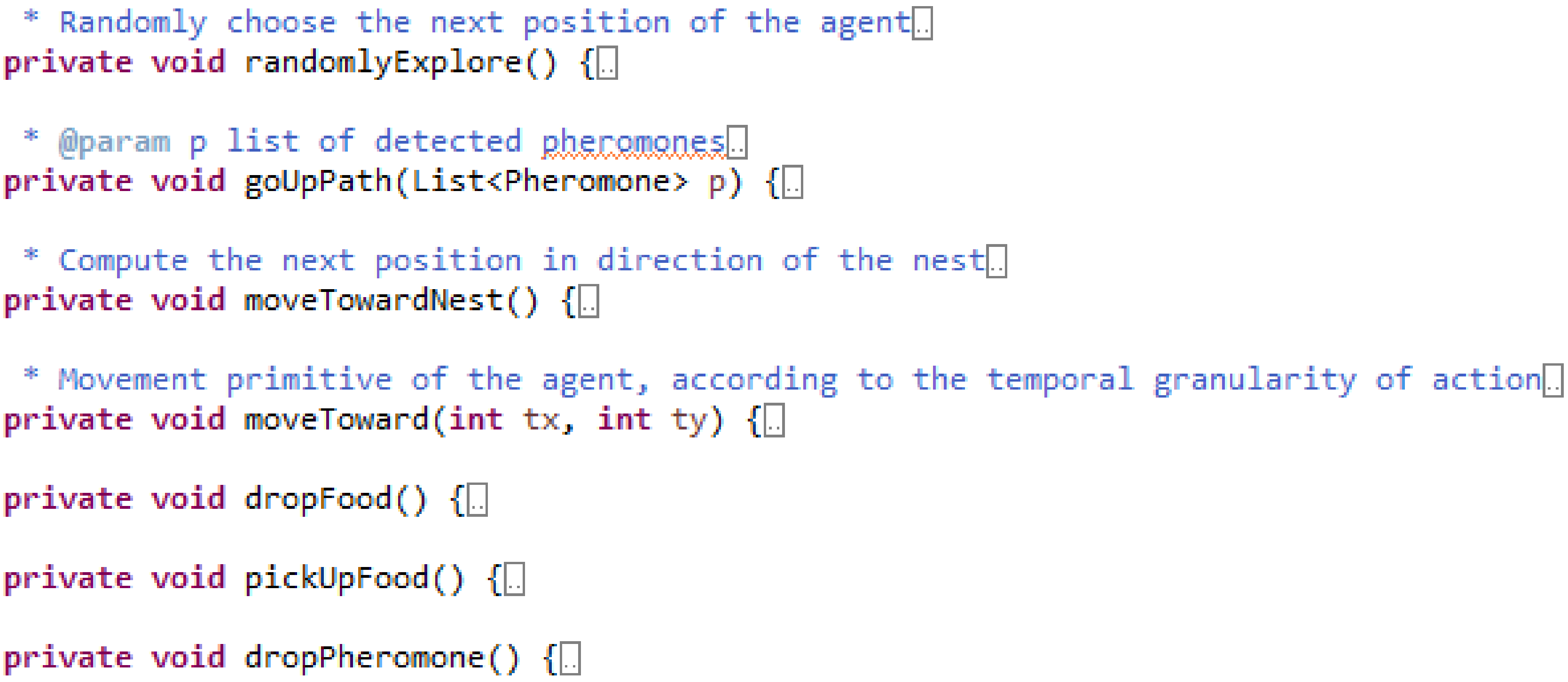
- **Agent**: an extension of the **object** concept in object-oriented programming
  - **Properties -> private attributes**



```
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**



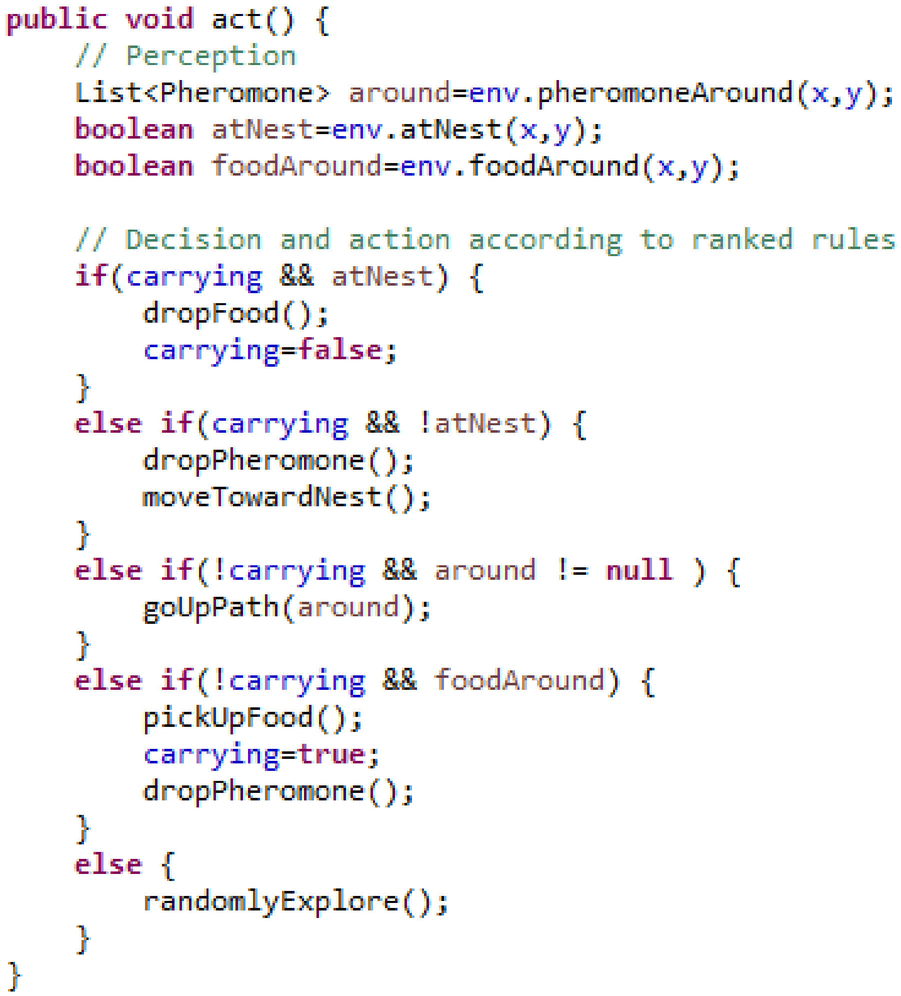
```
 * Randomly choose the next position of the agent[..]
private void randomlyExplore() {[..]

 * @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 implementation

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

<u>Solution 1</u>:
shuffle the agents list every time step

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

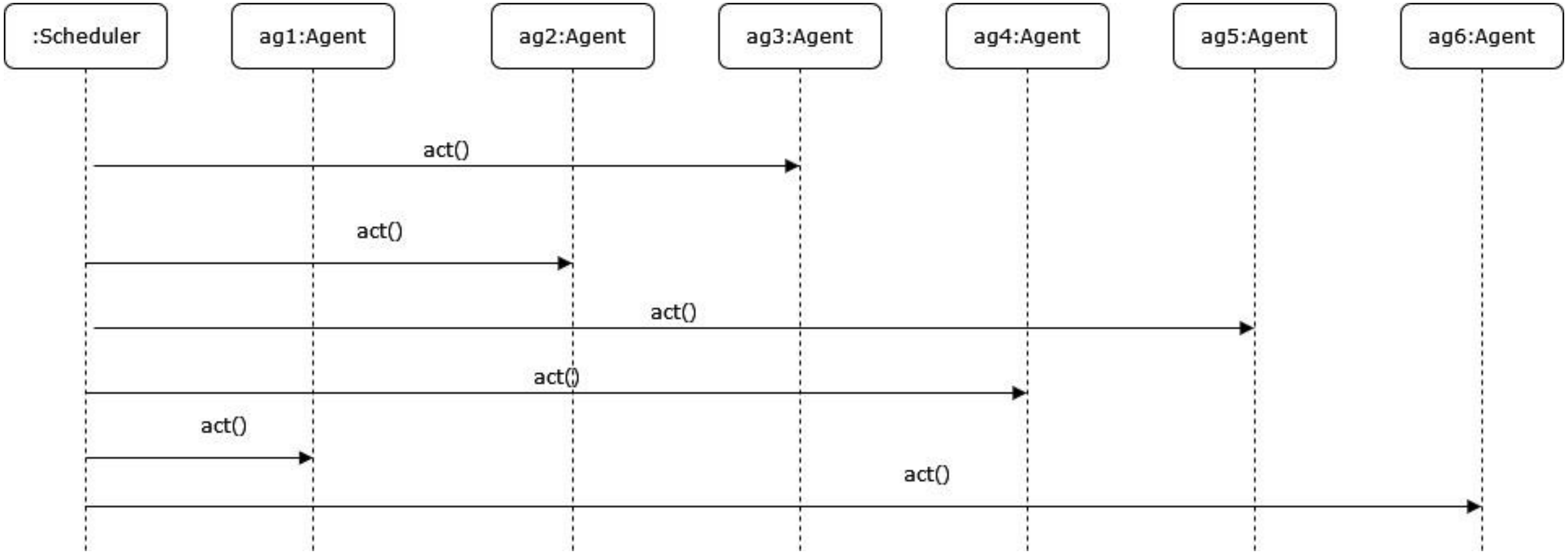


```
public void act() {
    // Perception
    List<Pheromone> around=env.pheromoneAround(x,y);
    boolean atNest=env.atNest(x,y);
    boolean foodAround=env.foodAround(x,y);

    // 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 implementation

Solution 1: shuffle the agents list every time step

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



1	*0*	*0*	*1*	*1*	*0*

Environment

Scheduling implementation

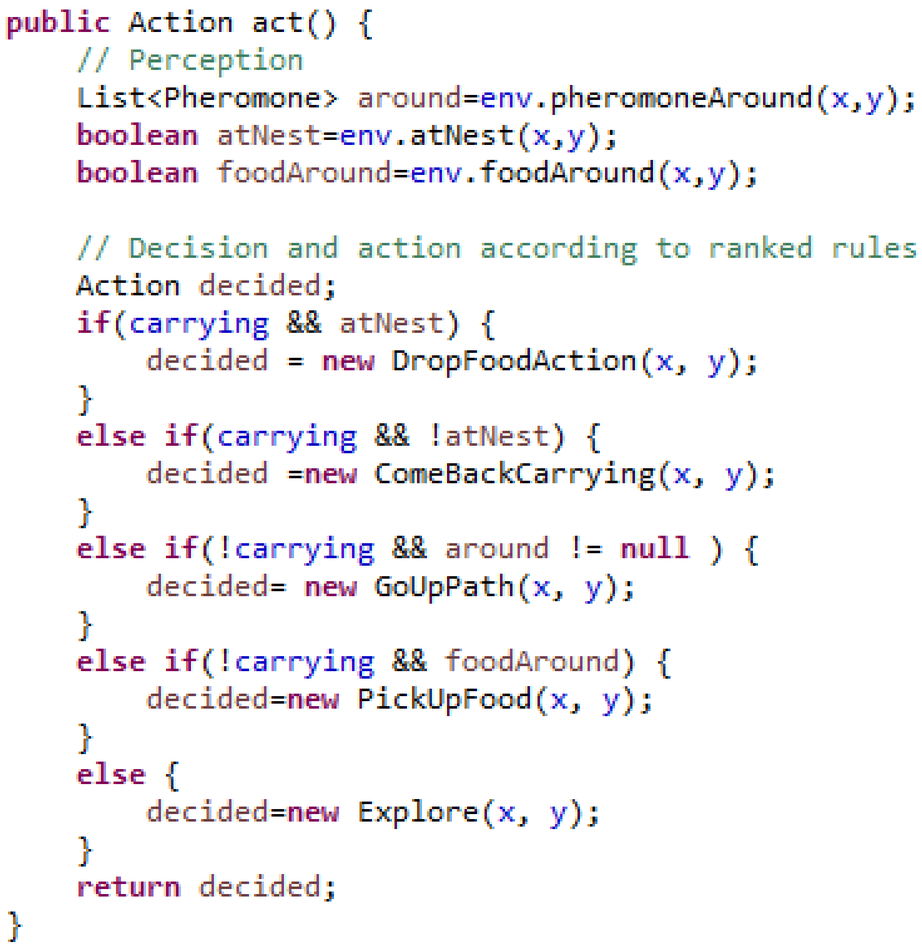
- One public method *act()* within the AntAgent class

Solution 2: **simulate concurrency**

The Scheduler class makes
the agents *act*, at each time step

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

The Scheduler **collects all the Actions**
solves the conflicts, and applies
their effects

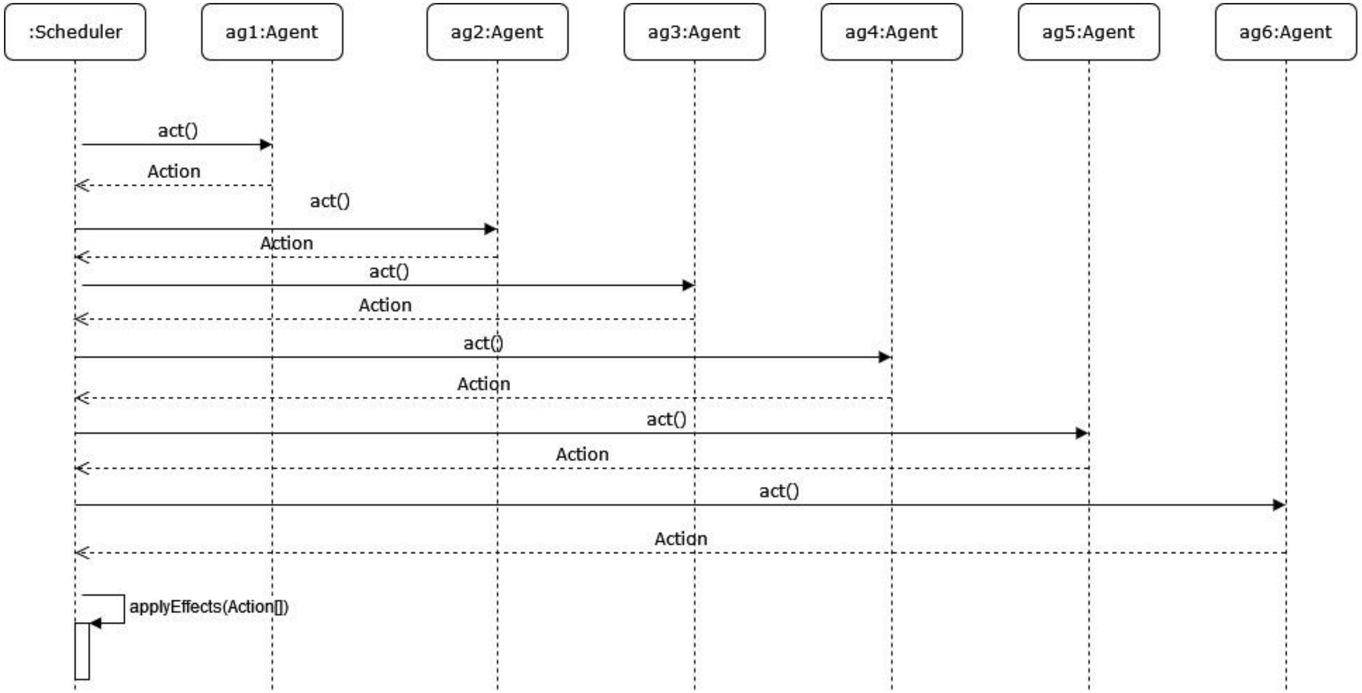


```
public Action act() {
    // Perception
    List<Pheromone> around=env.pheromoneAround(x,y);
    boolean atNest=env.atNest(x,y);
    boolean foodAround=env.foodAround(x,y);

    // 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 implementation

Solution 2: simulate concurrency

The Scheduler **collects all the Actions**, solves the conflicts (if necessary), and applies their effects



1	***0***	***0***	***1***	***1***	***0***

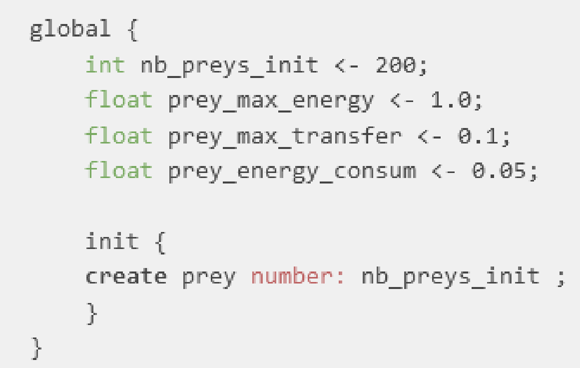
Environment

Scheduling implementation in GAMA and Unity

	GAMA

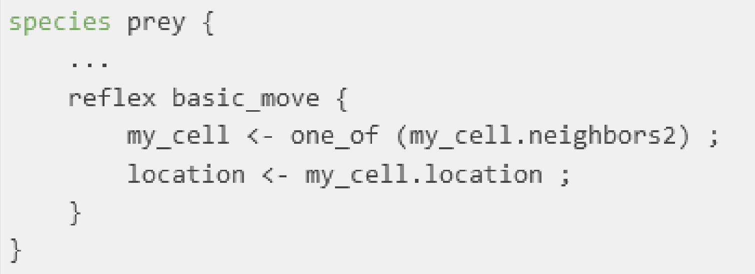
Agents Initialisation

Global parameters:



```
global {
    int nb_preys_init <- 200;
    float prey_max_energy <- 1.0;
    float prey_max_transfer <- 0.1;
    float prey_energy_consum <- 0.05;

    init {
    create prey number: nb_preys_init ;
    }
}
```

Agents specification and behaviour:



```
species prey {
    ...
    reflex basic_move {
        my_cell <- one_of (my_cell.neighbors2) ;
        location <- my_cell.location ;
    }
}
```

Scheduling

By default, the agents are activated in the same order (order of creation)

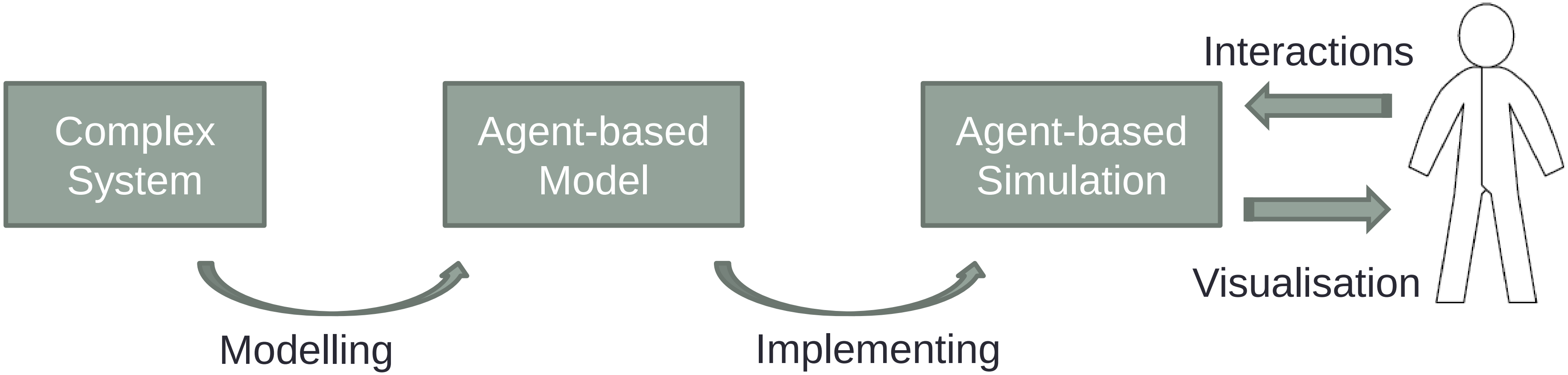
The Solution 1 is possible

```
species A schedules: shuffle(A) {...}
```

Scheduling implementation in GAMA and Unity

	Unity
Agents Initialisation	Start function Each agent is a GameObject Its behaviour can be defined in the Update, LateUpdate or FixedUpdate functions
Scheduling	With Update, the agents are activated in the same order (order of creation) The same with FixedUpdate, but the real time is defined by the frame per second rate The LateUpdate function implements the Solution 2

Plan



Complex System

Agent-based Model

Agent-based Simulation

Interactions

Visualisation

Modelling

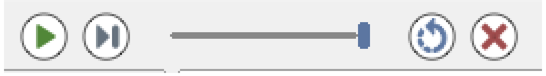
Implementing

Interactions with Agent-Based simulations

Interactions as expert modeler

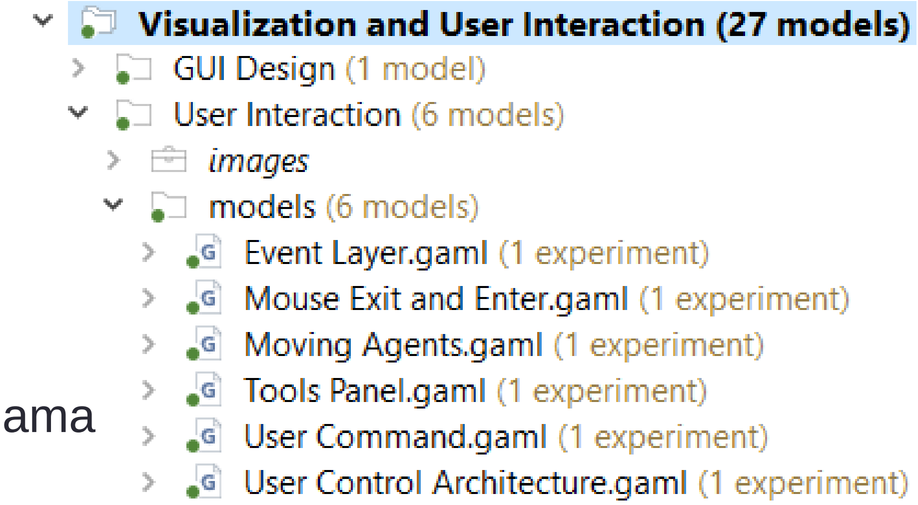
- By default in many simulation platforms

1. With the simulation itself
   - Start / Stop the simulation
   - Control the simulation speed
   - Modify the environment (at the initial state, or during the simulation)



2. With agents
   - Adding / Deleting / Moving agents → e.g. "Point and click"
   - Send messages
   - …

Examples in Gama



- **Visualization and User Interaction (27 models)**
  - GUI Design (1 model)
  - User Interaction (6 models)
    - *images*
    - models (6 models)
      - Event Layer.gaml (1 experiment)
      - Mouse Exit and Enter.gaml (1 experiment)
      - Moving Agents.gaml (1 experiment)
      - Tools Panel.gaml (1 experiment)
      - User Command.gaml (1 experiment)
      - User Control Architecture.gaml (1 experiment)

Interactions with Agent-Based simulations

Interactions as control methods: 4 types

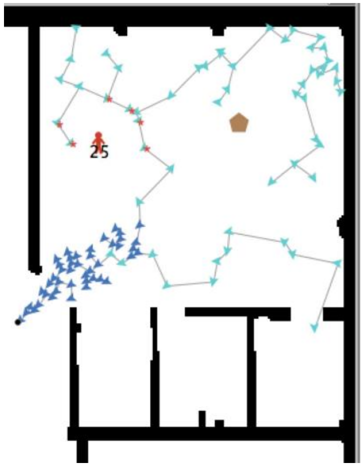
→ The objective is to control / influence the global behaviour

1. Indirect control via environmental influences

   Virtual pheromones, beacons, signals

   Attractor / Repellent, Behavior changer

   - Not easy to manipulate



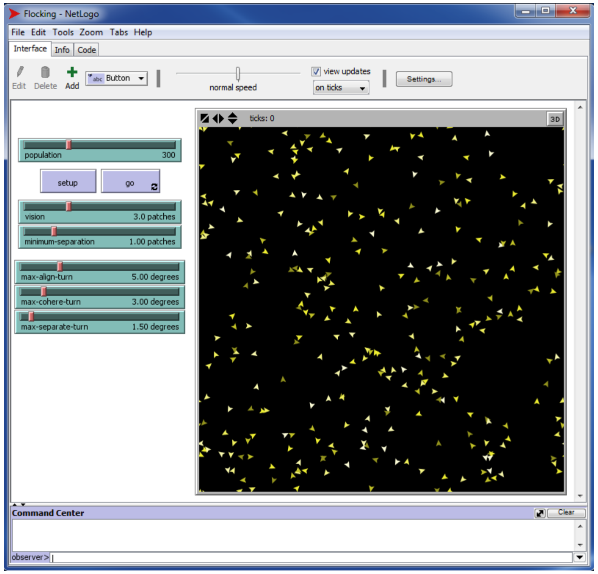
[Kolling et al., 2012]

Interactions with Agent-Based simulations

Interactions as control methods: 4 types

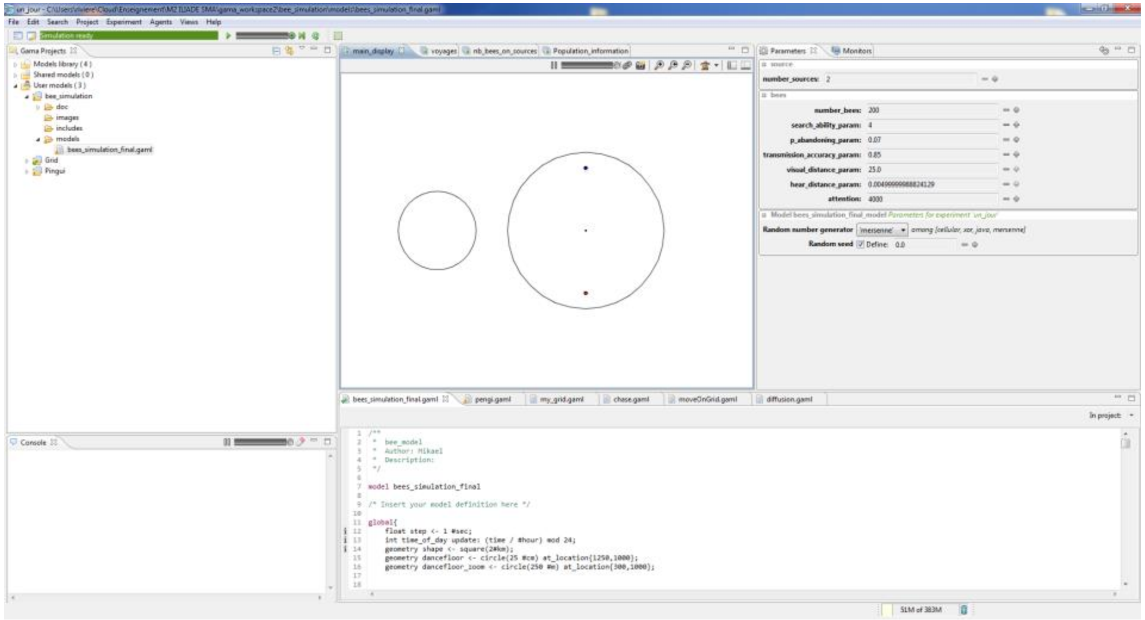
1. Indirect control via environmental influences

2. Changing behaviour parameters
   - Typically in platforms



population 300
vision 3.0 patches
minimum-separation 1.00 patches
max-align-turn 5.00 degrees
max-cohere-turn 3.00 degrees
max-separate-turn 1.50 degrees

NetLogo



source
number_sources: 2

bees
number_bees: 200
search_ability_param: 4
p_abandoning_param: 0.07
transmission_accuracy_param: 0.85
visual_distance_param: 25.0
hear_distance_param: 0.00499999988824129
attention: 4000

Model bees_simulation_final_model *Parameters for experiment 'un_jour'*
Random number generator 'mersenne' *among [cellular, xor, java, mersenne]*
Random seed ☑ Define: 0.0

```
/**
 *  bee_model
 *  Author: Mikael
 *  Description:
 */

model bees_simulation_final

/* Insert your model definition here */

global{
    float step <- 1 #sec;
    int time_of_day update: (time / #hour) mod 24;
    geometry shape <- square(2#km);
    geometry dancefloor <- circle(25 #cm) at_location{1250,1000};
    geometry dancefloor_zoom <- circle(250 #m) at_location{300,1000};
```

Gama

Interactions with Agent-Based simulations

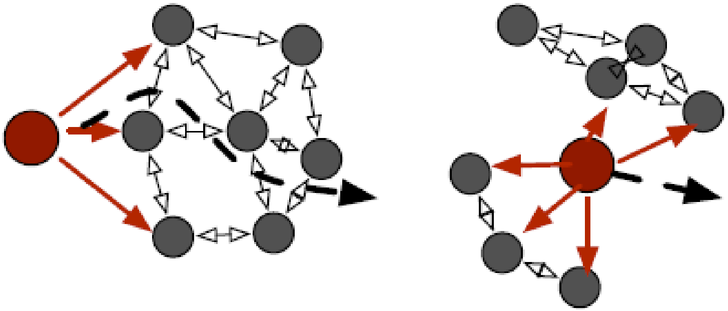
Interactions as control methods: 4 types

1. Indirect control via environmental influences
2. Changing behaviour parameters
3. Switching between behaviours
   - By selecting a set of agents and assigning them a new behaviour

Interactions with Agent-Based simulations

Interactions as control methods: 4 types

1. Indirect control via environmental influences
2. Changing behaviour parameters
3. Switching between behaviours
4. Control through selected agents
   - i.e. Teleoperation with leaders
   - Influence propagate through the selected agents



Control through selected agents 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 controls (plays the role of) an agent
  - interaction and perceptions capacities
  - can be seen by the other agents

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

Control through selected agents
The role of Virtual Reality

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**

Control through selected agents
The role of Virtual Reality

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

Control through selected agents
The role of Virtual Reality

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:

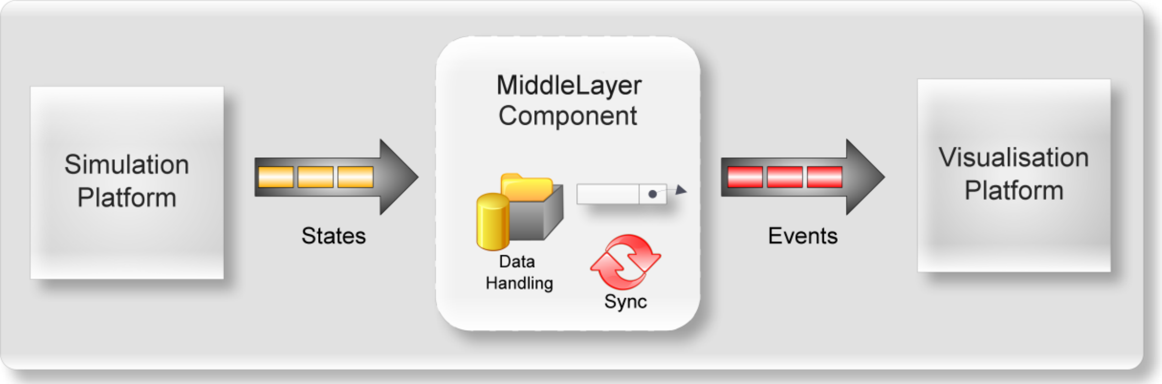
- Implement **credible** ECA
- Believability of the simulation itself
- Challenges!

Control through selected agents
The role of Virtual Reality

Virtual Reality and Agent-Based Simulations: 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]

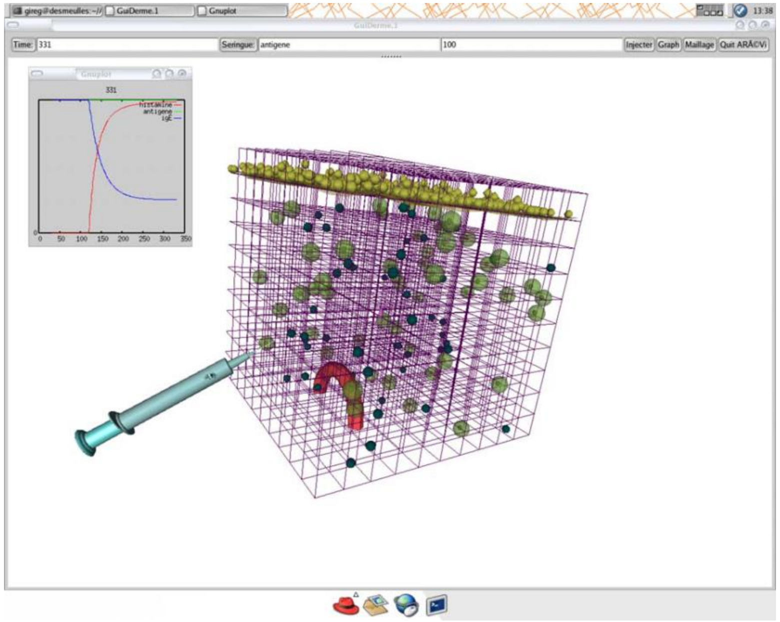


Interactions with Agent-Based simulations

« Natural » interactions

It could be hard for non-expert in computer science (and ABM) to manipulate the simulation → The natural interaction metaphor

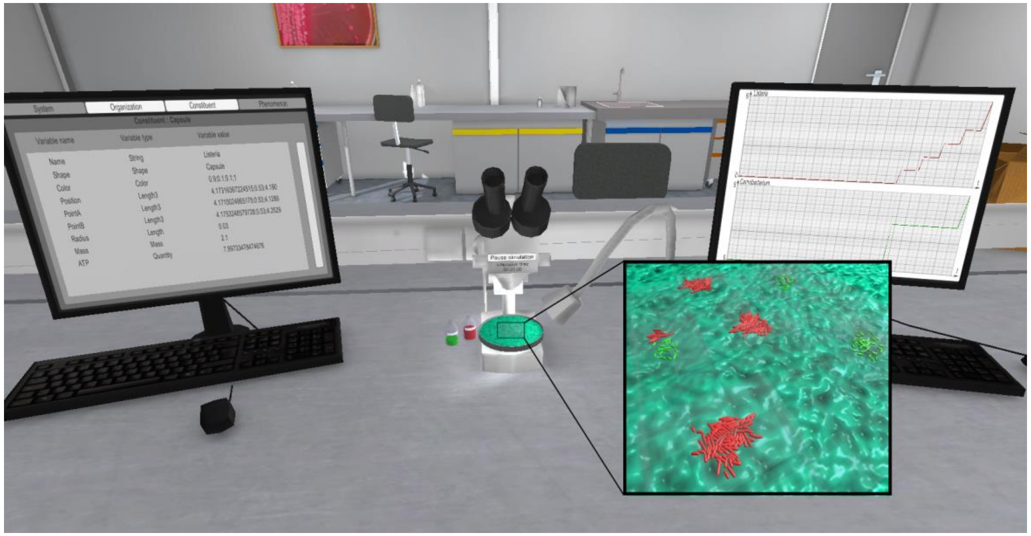
- Usually 3D / VR-like, e.g. *The virtual laboratory [Desmeulles, 2006]*



Time: 331

Seringue: antigene

100



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

Visualisation in Agent-Based simulations

Visualisation

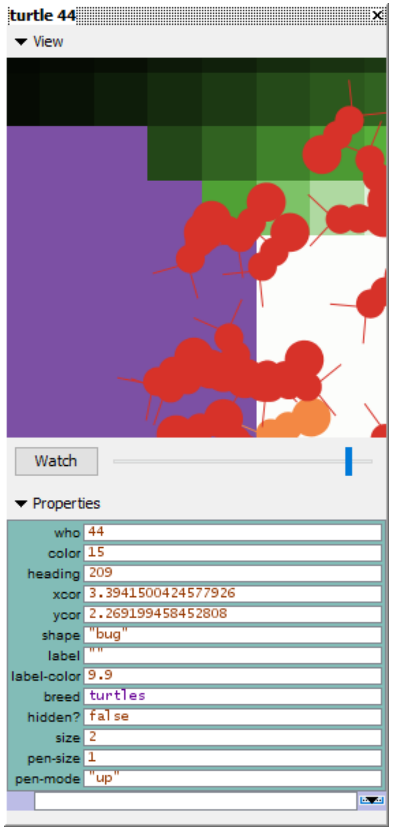
1. **Micro** and **macro** views

   By default in many simulation platforms

Micro

Follow an agent, its internal state,
its decision making, its interaction …

With Virtual Reality:
Other agents behaviour, movements,
voice, gestures, messages …



who	44
color	15
heading	209
xcor	3.3941500424577926
ycor	2.269199458452808
shape	"bug"
label	""
label-color	9.9
breed	turtles
hidden?	false
size	2
pen-size	1
pen-mode	"up"

Examples in NetLogo

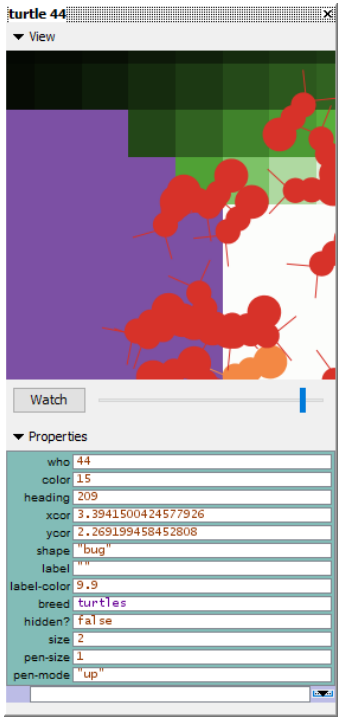
Visualisation in Agent-Based simulations

Visualisation

1. **Micro** and **macro** views

By default in many simulation platforms

Micro

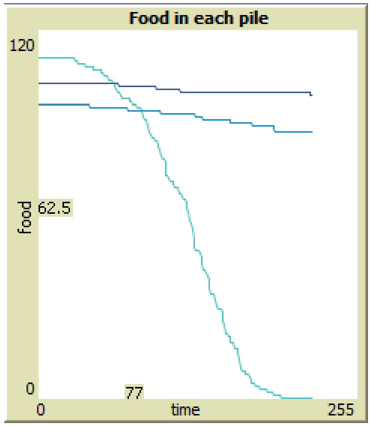


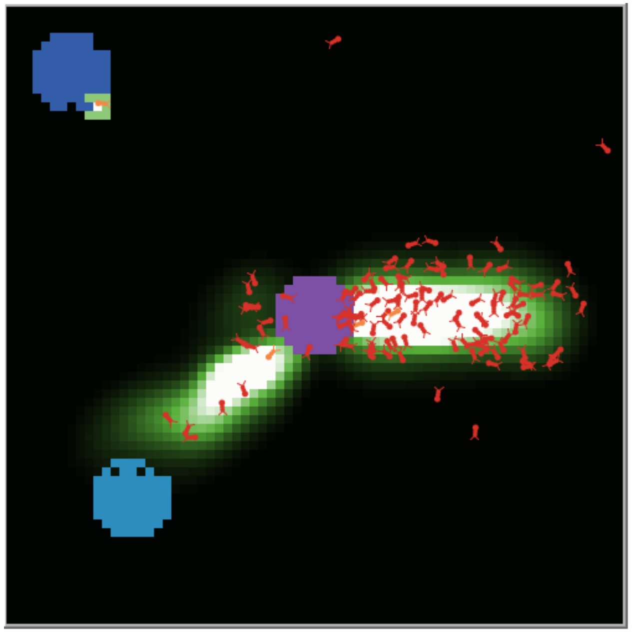
who 44
color 15
heading 209
xcor 3.3941500424577926
ycor 2.269199458452808
shape "bug"
label ""
label-color 9.9
breed turtles
hidden? false
size 2
pen-size 1
pen-mode "up"

Macro

View of agents
from a specific perspective

/ View of some properties of
the global system

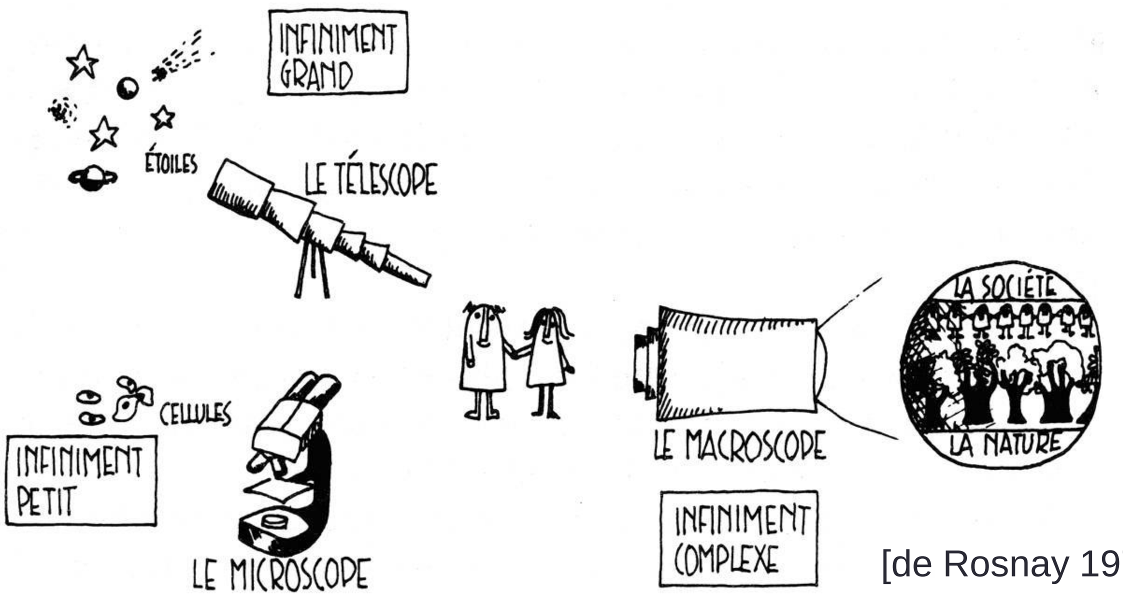




Visualisation in Agent-Based simulations

<u>Visualisation</u>

1. **Micro** and **macro** views
2. What about **complexity**? <u>How can we visualise "how it goes back up"?</u>



[de Rosnay 1975]

The "Macroscope" filters the details, amplifies what connects, underlines what brings together

Visualisation in Agent-Based simulations

Visualisation

1. **Micro** and **macro** views
2. What about **complexity**? How can we visualise “how it goes back up”?

Use interaction and visualisation to help the user understand the mechanisms
→ build himself a cognitive model of the system (mental representation)

Desirable properties [Hutzler, 2000]

- Diversity of forms / levels of representation
- Variety of information sources
- Modularity
- Structuration
- Interactions

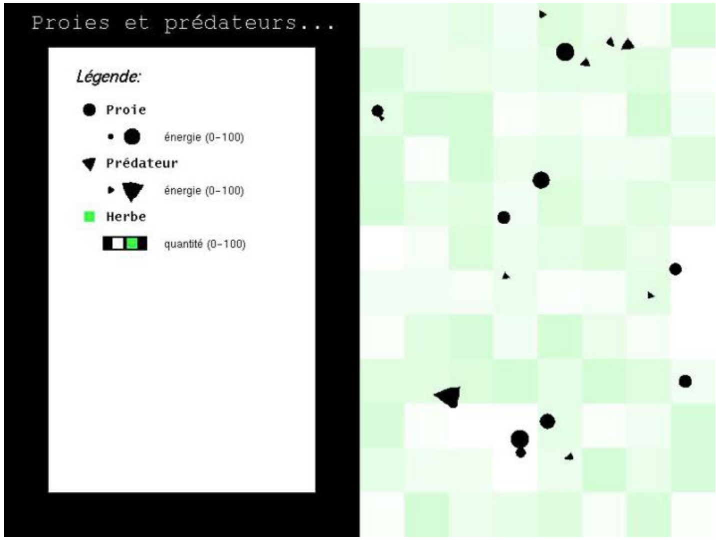
Visualisation in Agent-Based simulations

Visualisation

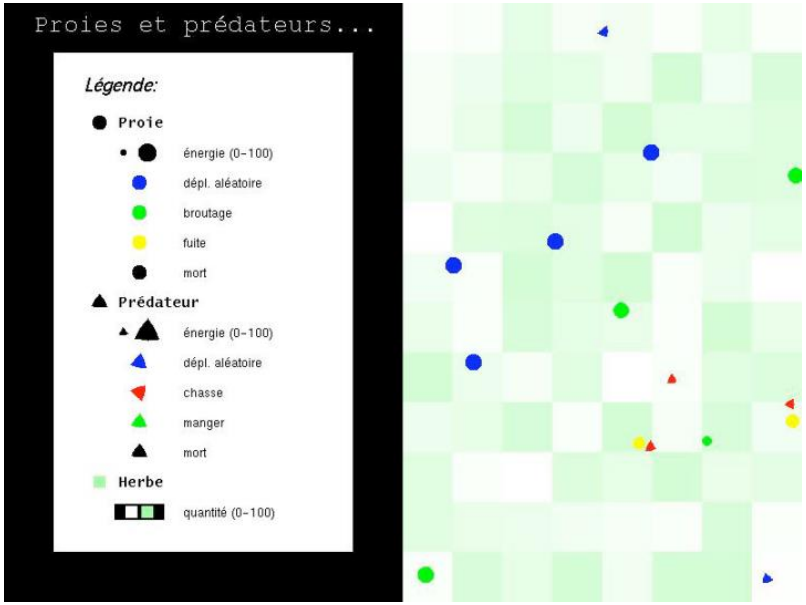
1. **Micro** and **macro** views
2. What about **complexity**? How can we visualise “how it goes back up”?

Several propositions, still an ungoing work

Predator - prey example [Hutzler, 2000]



Lien entre la taille d’un agent et sa quantité d’énergie



Visualisation du comportement activé

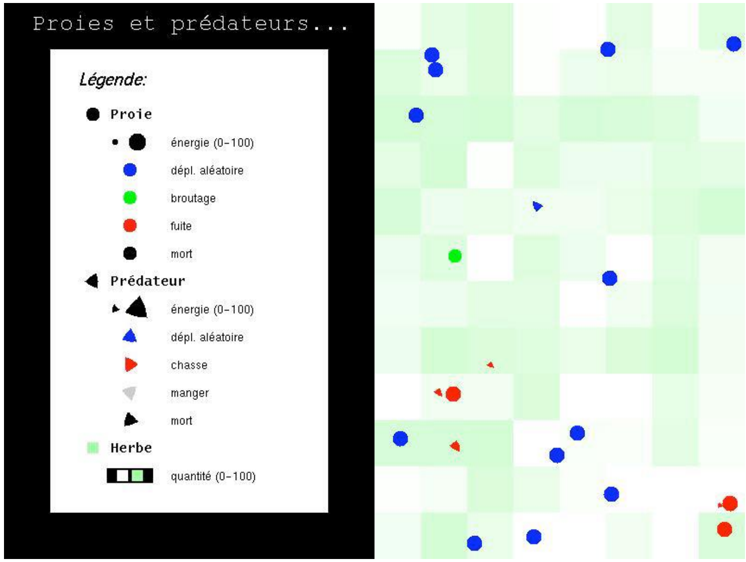
Visualisation in Agent-Based simulations

Visualisation

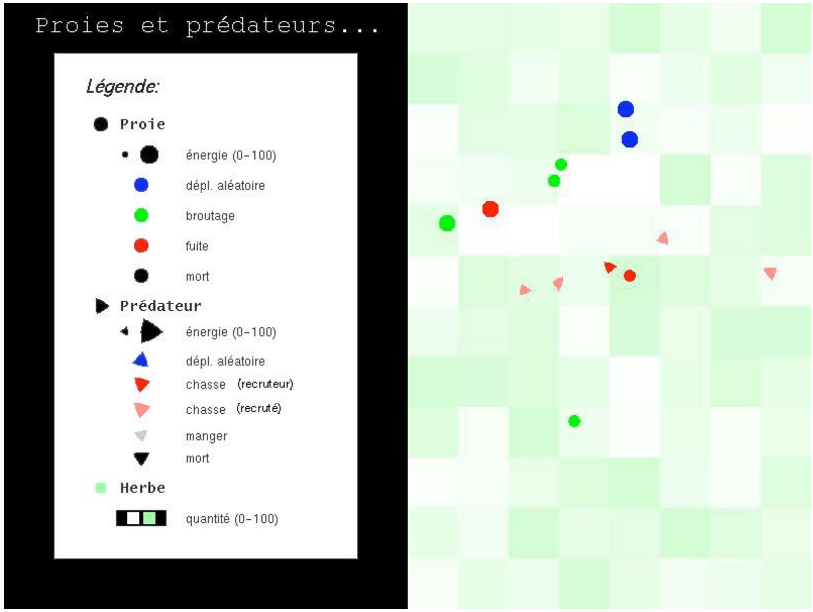
1. **Micro** and **macro** views
2. What about **complexity**? How can we visualise “how it goes back up”?

Several propositions, still an ungoing work

Predator - prey example [Hutzler, 2000]



Visualisation des interactions entre proies et prédateurs



Visualisation de la constitution de groupes d’agents prédateurs

Visualisation in Agent-Based simulations

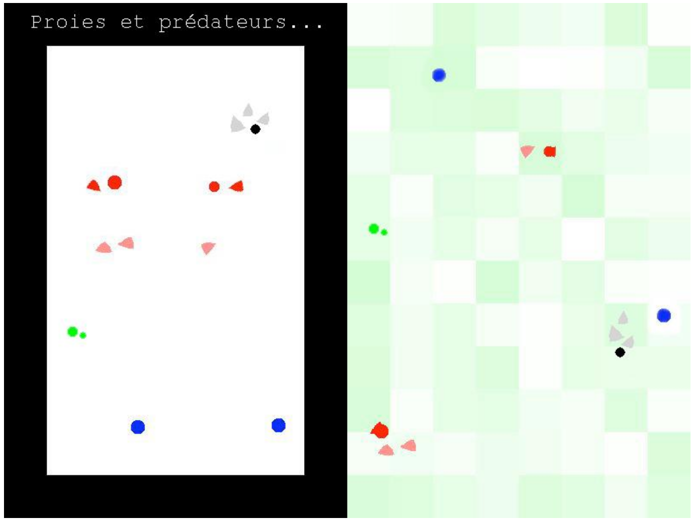
Visualisation

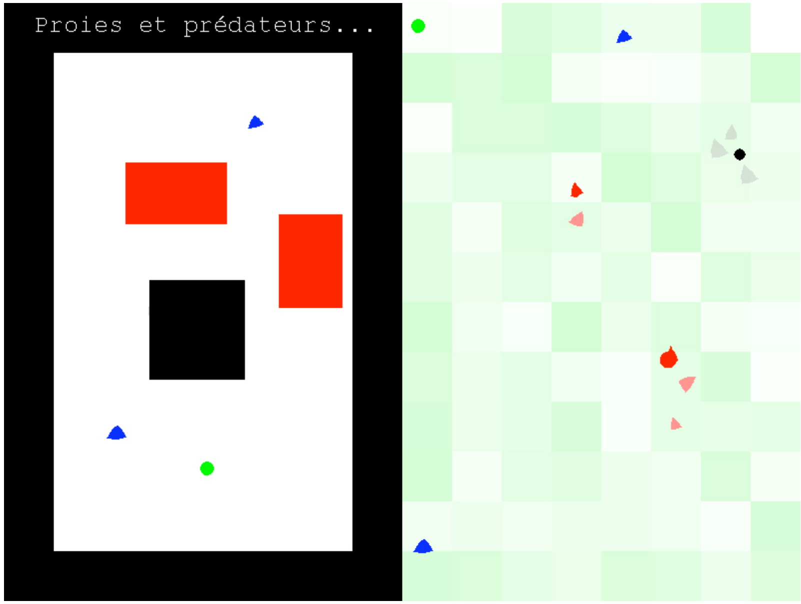
1. **Micro** and **macro** views
2. What about **complexity**? How can we visualise “how it goes back up”?

Several propositions, still an ungoing work

Predator - prey example [Hutzler, 2000]

Hiérarchie de groupes : des carrés synthétisent la présence d’un groupe prédateur/proie ou recruteur/recruté





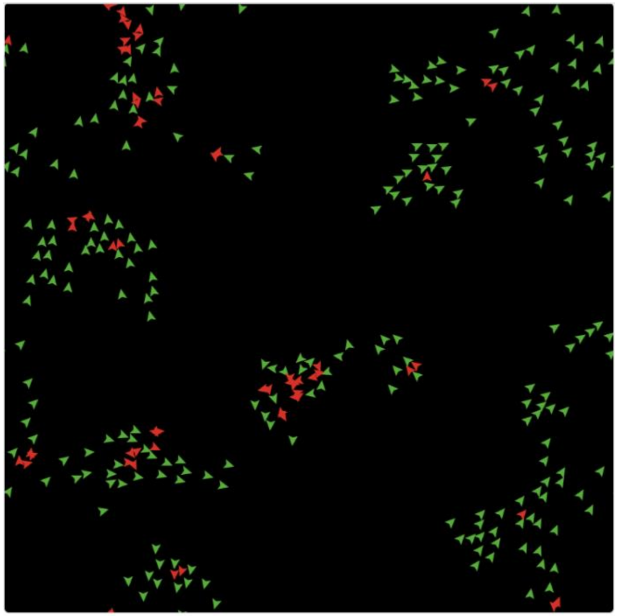
Hiérarchie de comportement : les agents accomplissant les actions les plus prioritaires se retrouvent dans le haut de l’image de gauche

Visualisation in Agent-Based simulations

Visualisation

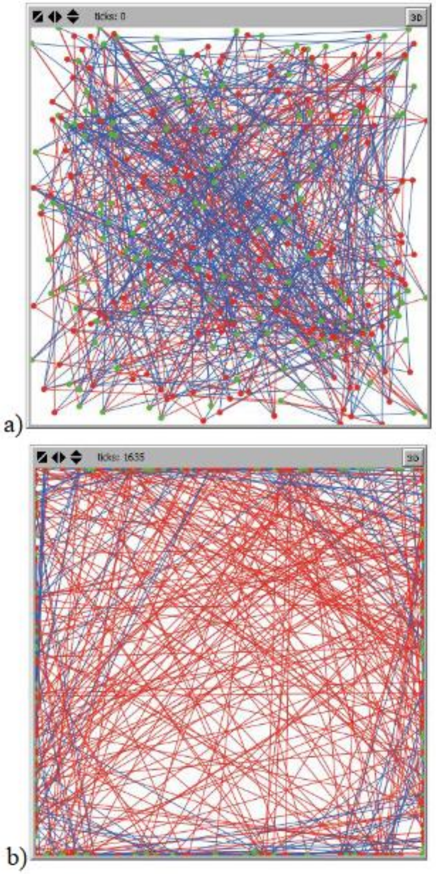
1. **Micro** and **macro** views
2. What about **complexity**? How can we visualise “how it goes back up”?

Several propositions, still an ungoing work



Flocking NetLogo
(Alternative Visualization)

Heroes and Cowards
[Friesen et al., 2017]

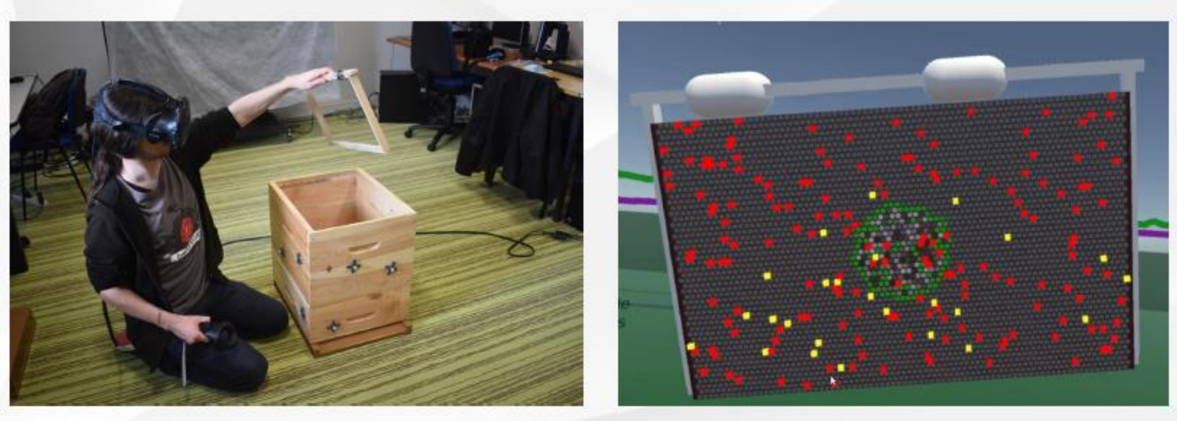


Visualisation in Agent-Based simulations

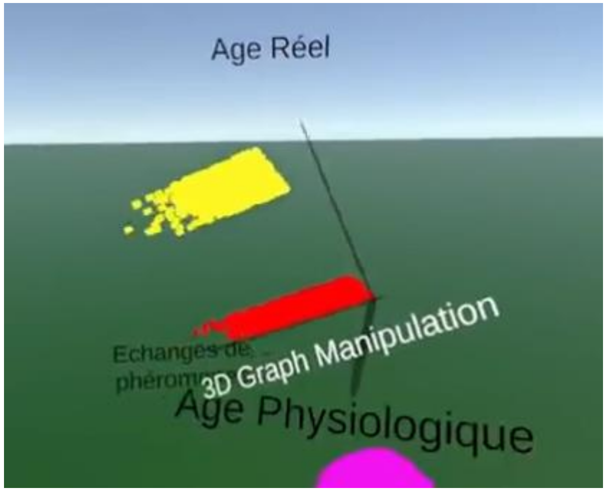
Visualisation

1. **Micro** and **macro** views
2. What about **complexity**? How can we visualise “how it goes back up”?

Several propositions, still an ungoing work



[Alves, 2021]



Conclusion
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 (2016). "***Human Interaction With Robot Swarms: A Survey***" in IEEE Transactions on Human-Machine Systems, vol. 46, no. 1, pp. 9-26