Agent and Multi-Agent Systems: architectures and reasoning

Multi-agent Based Simulation

28.09.2021

CentraleSupélec- SAFRAN AI Training

Organization of the course

Organization

Dates	
27/09/2021	Course 1: Agent and MAS: Concepts (Course+PW)
28/09/2021	Course 2 : Multi-Agent based Simulation (Course+PW)
29/09/2021	Course 3 : Interaction mechanisms: models and Implementation (Course+PW)

Before to start !

`www.wooclap.com/MEHCLI`

Table of contents

Simulation in General

What is a simulation ?

"... The process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies (within the limits imposed by a criterion or a set of criteria) for the operation of the system." [10]

$\rightarrow$ Study a real system through a model in order to understand how does it work and / or to predict its evolution under certain conditions.

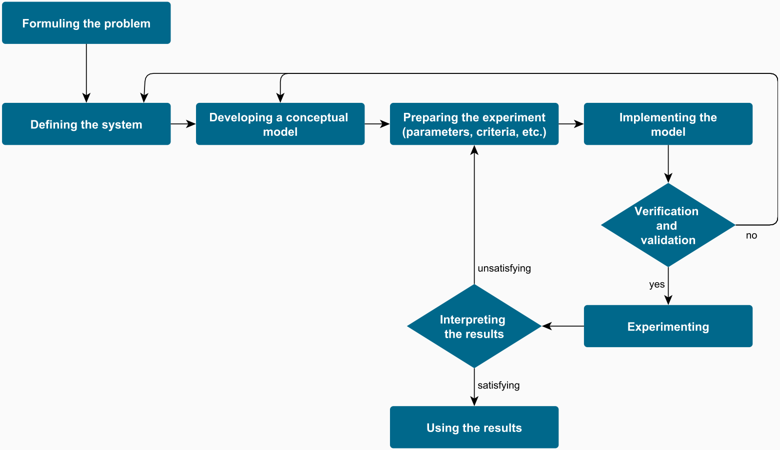
What is a computer simulation ?

"Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output." [3]

Computer simulation is a **non-linear iterative process** composed of three strongly interdependent fundamental tasks:

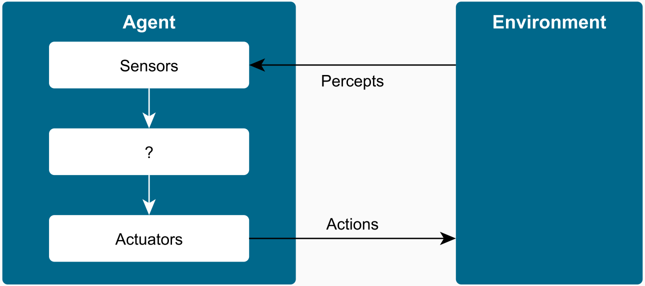
1. Developing the model.
2. Executing the model on computer.
3. Analysing the execution of the model and the obtained results.

Computer simulation: An experimental process



→ Computer simulation is inseparable from the experimental process which is linked to its objective [11].

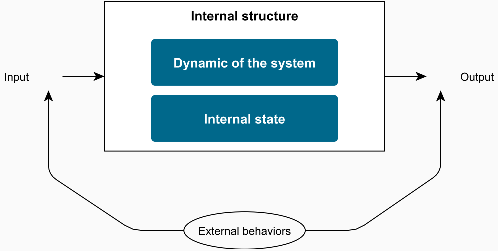
Agent as a dynamic system



Introducing dynamic systems

The vast majority of the systems we want to simulate are described as dynamic. Dynamic systems are characterized by:

- The **external behavior** of the system: The observable reactions of the system from the outside.
- The **internal structure** of the system: Its internal state and its dynamics.



Classic representation of a dynamic system: The black box

Dynamic systems

The internal structure of a dynamic system is defined by three parameters:

- **Internal state**: Define the state of the system (*represented by state variables*).
- **State transition function**: Define how state variables evolve (*from inputs or autonomously*).
- **Production mechanism**: Define how the system produces an output result (*based on its internal state*).

Time in Simulation

Time representation in simulation

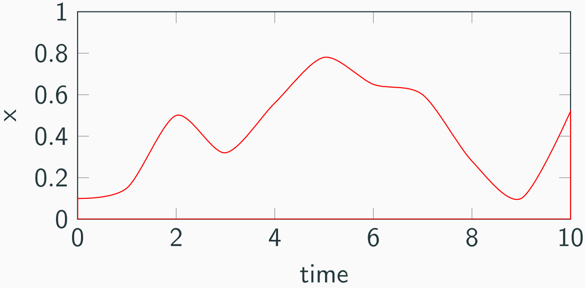
A dynamic system is defined by how it evolves over time. So, one of the most important features of a model which represent a dynamic system is **how time is represented**.

There are three types of time representations:

- **Continuous** time models.
- **Discrete** time models.
- **Discrete event** time models.

Continuous time models

In a finite time interval, the system state variables change in value infinitely often, ie *continuously*.



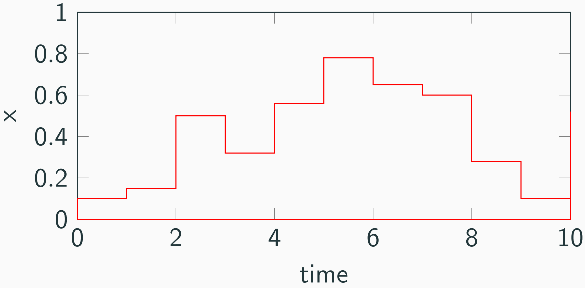
Simulation of continuous time models

The multi-agent approach is far from continuous models because **each change** in the behavior of an agent is always a **punctual event**, and therefore discrete.

$\rightarrow$ The simulation of continuous models raises many problems due to the nature of the computer: It is simply impossible to reproduce the continuity of the dynamics of a system because it evolves infinitely often while the computer simulation needs punctual computations.

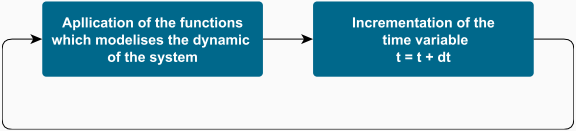
Discrete time models

The time axis is **discretized** according to a constant period of time called time step (dt). The evolution of the system state variables is done in a discrete way, ie *instantaneous*, from t to $t + dt$.



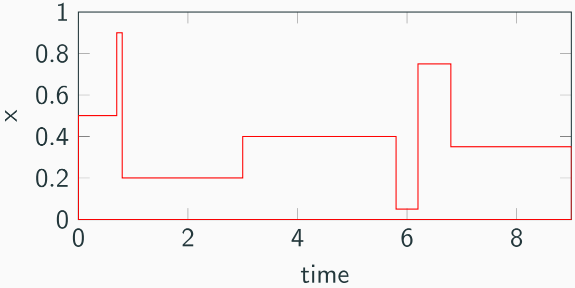
Simulation of discrete time models

When the functions which implement the dynamics of the system are clearly defined, it is only necessary to set up an algorithm that **applies these functions** and then **increments time**.



Discrete event time models

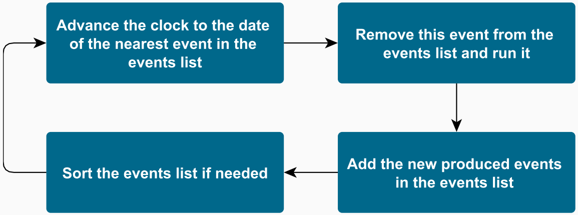
The time axis is generally **continuous**, ie *represented by a real number*. However, unlike continuous models, system state variables change discretely to specific times that are called **events**.



Simulation of discrete event time models

There are three ways to simulate discrete event time models:

- Event scheduling (see figure).
- Activity scanning.
- Interaction process.



Modeling & Simulation theory

M&S theory [13] aims to provide a general methodological basis for the design of a simulation.

M&S theory helps to:

- Identify the different entities that constitute a simulation experiment.
- Study the relations that exist between these different entities.

$\rightarrow$ This is to give precise definitions to the different concepts that are manipulated in the field of computer simulation.

Entities in the M&S theory

There are 6 entities clearly defined in the M&S theory:

- The **source system** and its behavioral database.
- The **experimental scope**.
- The **model**.
- The **simulator**.
- The **simulation relationship**.
- The **modeling relationship**.

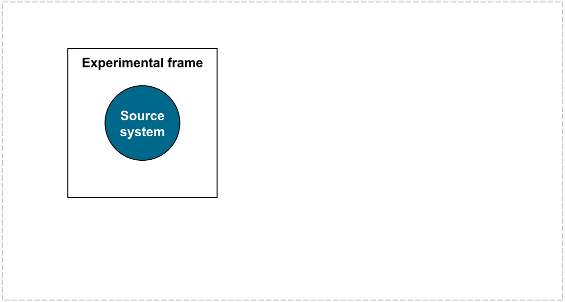
Entities in the M&S theory

The source system and its behavioral database

The source system corresponds to **the environment to be modeled**. It must be seen as a source of **observable data** that constitutes what is called the behavioral database.

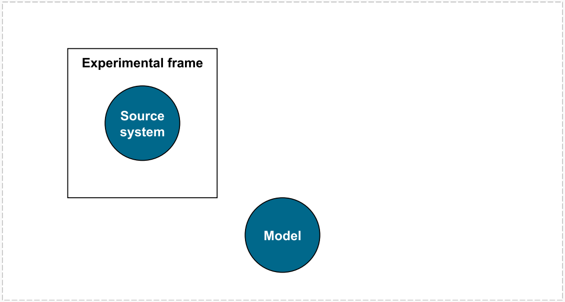
The experimental scope

The experimental frame is a specification of (1) the **observing conditions** of the system and of (2) the **objectives of the simulation** project.



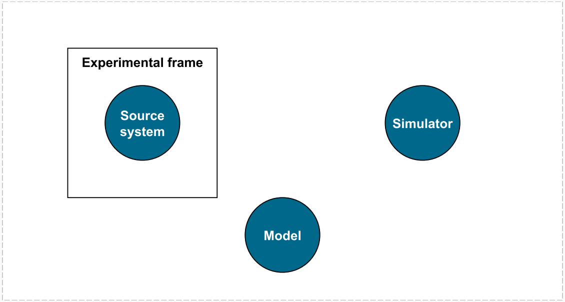
The model

The model refer to the specification of all the instructions used to generate the **behavior of the system**.



The simulator

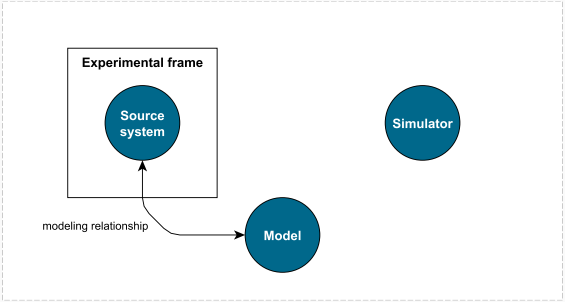
The simulator refer to any **computing system** capable of **executing the model** and generate its behavior. By separating a model from its simulator, a model can be run by different simulators which increases its portability.



Entities in the M&S theory

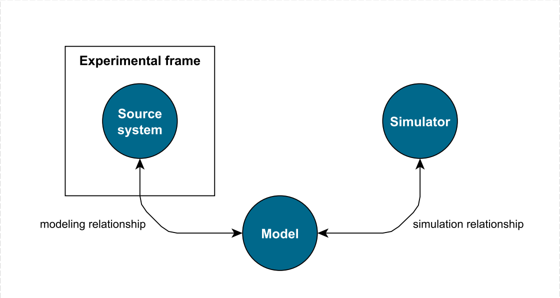
The modeling relationship

The modeling relationship defines the notion of **validity of the model**. *Does the modeling which is made of the system is an acceptable simplification of this one according to the qualitative criteria chosen and the objectives of the experimentation ?*



The simulation relationship

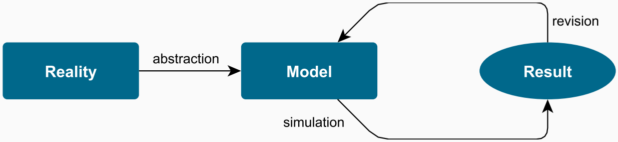
The simulation relationship defines the notion of **validity of the simulator**.
Does the simulator correctly generates the behavior of the model ?
Does the simulator reproduces the mechanisms defined in the model (without introducing errors) ?



Multi-Agent Based Simulation

Simulation advantages

- Ability to quickly test an hypothesis.
- Highlights the emerging aspects of a phenomenon according to individual choices.
- Highlights the existence of stable situations, resilience capabilities, etc.



Simulation of Equation Based Model (EBM) [7]

- EBM are build on an interrelation of a set of equations that **captures the variability of a system over time**.
- EBM represents **the whole system** and does not support an explicit representation of components (*top-down*).
- EBM is most naturally used to model central systems.

`http://systems-sciences.uni-graz.at/etextbook/sysmod/ebm_vs_abm.html`

Prey - Predator [12]

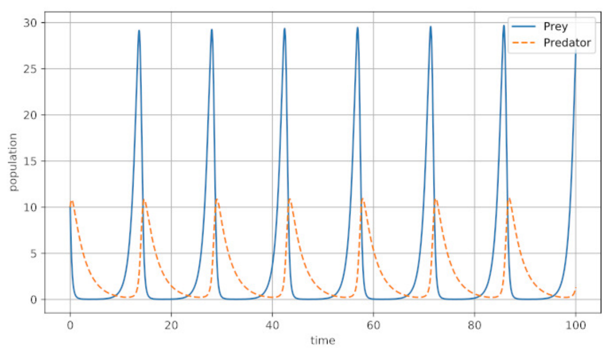
Used to describe the **dynamics of biological systems** in which two species interact, one as a predator and the other as prey.

$$\frac{dx(t)}{dt} = a * x(t) - b * x(t) * y(t)$$

$$\frac{dy(t)}{dt} = c * x(t) * y(t) - d * y(t)$$

$$\begin{aligned}
x(t) &= \text{number of preys over time} \\
y(t) &= \text{number of predators over time} \\
a &= \text{preys reproduction rate} \\
b &= \text{preys death rate} \\
c &= \text{predators reproduction rate} \\
d &= \text{predators death rate}
\end{aligned}$$

Prey - Predator [12]



Source: `https://commons.wikimedia.org/w/index.php?curid=75212926`

$\rightarrow$ Focus on the global variation of the prey and predator populations.

Limits of the EBM simulation

- **Large number of parameters** sometimes hard to understand.
- Difficulty to move from macro to micro level.
- Does not represent behaviors but **behaviors results**.
- Difficulty to represent behaviors.
- Does **not represent interactions and organizations**.

Idea to overcome EBM limits

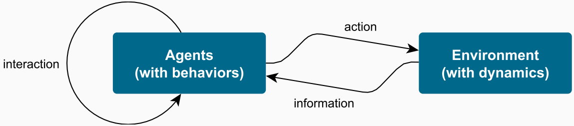
To overcome EBM limitations:

- Use models that **focus on entities and their interactions** (*bottom-up*).
- Consider that **the dynamics** of the system **come from the interactions** between these entities.

$\rightarrow$ Use Multi-Agent Based Simulation (MABS): First works in the early 1990s, has grown since the 2000s.

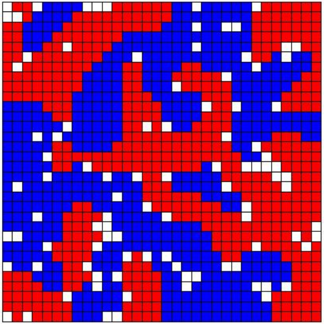
Multi-Agent Based Simulation (MABS)

- Creating an **artificial world** made up of **interacting agents** (*Agent-Based Models*).
- Each agent is described as an **autonomous entity**.
- The behaviour of agents is the consequence of their observations, internal trends, beliefs and interactions with the environment and other agents.
- Agents act and change the state of the environment through their actions.



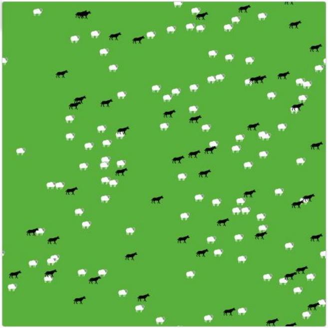
Schelling's Model of Segregation

[9]



Prey Predator

[12]

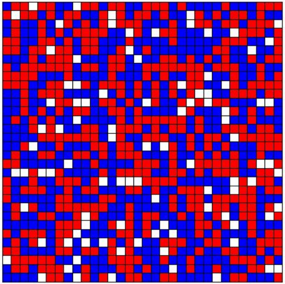


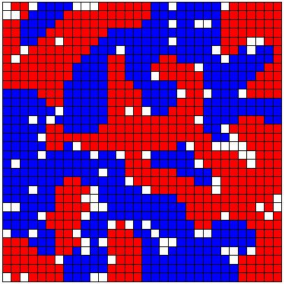
Schelling's Model of Segregation [9]

Study of racial segregation: Shows that segregation comes quickly if individuals want to be with people "like them".

Each entity is represented by an agent and has two behaviors:

- **Stay** (*happy*) if the number of similar neighbors $> x$.
- **Move** (*unhappy*) if the number of similar neighbors $< x$.

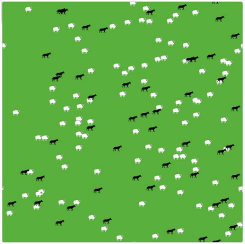




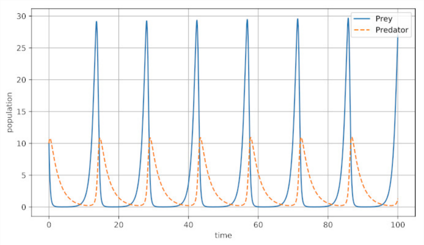
Each entity (prey and predator) is represented by an agent and has behaviors.

- **Prey**: move, eat, reproduce, flee, die.
- **Predator**: move, eat, reproduce, hunt, die.

Each agent activate one of its behavior according to its perception of the environment.



Micro level



Macro level

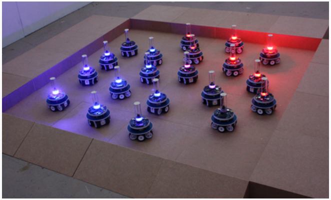
Actual use of MABS

Special effects in movies

Massive software for crowd simulation

Video games

Glassbox engine for SimCity



Collective behaviors for cooperative actions

Collective robotics

Flock of birds simulation

Simulation of complex systems

Actual use of MABS

Actual use of MABS

- Economy
- Crowd movement, emergency and security
- Renewable resource management
- Space Management, development of territory
- Etc.

Equation Based Model vs Agent Based Model

Equation Based Model	**Agent Based Model**
Macrospecifications reveal microstructures (*top down*)	Microspecifications generate macrostructures (*bottom up*)
Externally observable phenomenon (*equations*)	Autonomous decision making entities (*agents*)
Simplicity in modeling inputs, state and outputs	Simplicity in modeling rules / behaviors
Internal behavior is unknow	Emerging behaviors
Easy to test	Difficult to validate

Visualizing MABS

Visualizing Multi-Agent Based Simulation

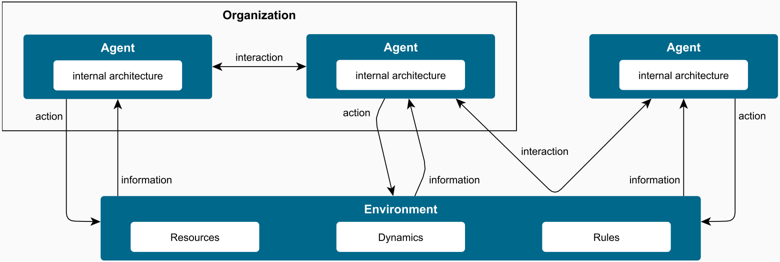
Open MABS_2021_Student.ipynb

The visualization of the simulation is very important in the context of Multi-Agent based Simulation because the simple and localized rules that allow to reproduce complex and global behaviors of the system are also responsible of the emergence of many phenomena.

Implementing MABS

Architecture of a Multi-Agent Base Simulation

- Define **architecture of autonomous entities** that act in the system (*Agent*)
- Define **interactions** between entities (*Interaction*)
- Define **shared resources** and **processes** between agents (*Environment*)
- Define **coordination** and **cooperation** between agents (*Organization*)



Overview of tools

Specialized tools for Multi-Agent Based Simulation:

- *ATOM, ARCHISIM, MATSim, MITSIMlab*

\+ Allow to reuse some parts of models.
\- Models need adaptation to be used in a new platform.

Generic tools for Multi-Agent Based Simulation:

- *AnyLogic, GAMA, MASON, SWARM, CORMAS, TurtleKit, Repast, Netlogo, JASMIN*

\+ Possible to use the same tool for different models.
\- Need to create a single operational model for each simulation.

Generic tools for multi-agent softwares:

- *Jason, JADE/TAPAS, JADE/PlaSMA, MADKIT*

\+ General knowledge of the tool (easy to use).
\- Need adaptation to implement simulation models.

Implementation of agents and environment issues

The implementation of multi-agent models requires taking into account constraints:

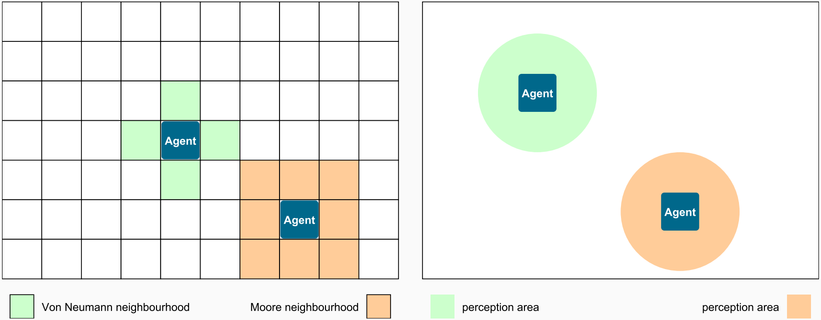
1. **Genericity and modularity** constraint
2. **Locality** constraint
3. **Environmental integrity** constraint

Genericity and modularity constraint:
Individual and autonomous entities are modeled and implemented. We want to be able to vary their implementation without questioning the entire source code.

Implementation of agents and environment issues

Locality constraint:
The environment is **not fully accessible** to an agent. An agent has only a **local perception of the world** and his actions/perceptions are limited in their scope.

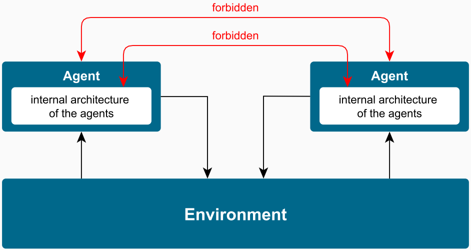
Discrete environment *Continuous environment*



Implementation of agents and environment issues

Environmental integrity constraint:

- **No direct relationship** between the internal architectures of two autonomous agents
- An agent must **not be able to directly change environment state** variables



Time management in MABS

Need to manage the evolution of time

Multi-Agent Based Simulation is based on the idea that it is **possible to represent a set of autonomous entities operating in a common environment**.

$\rightarrow$ All agents must be subject to the same temporal law in order to respect the principle of causality [2].

Agent Behaviour: A Discrete Process

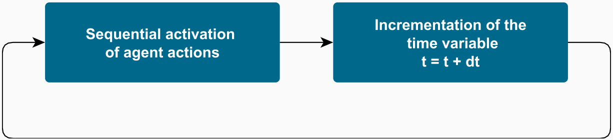
The behaviour of an agent is inherently discrete. The two main implementation principles for handling discrete events:

- **Synchronous approach**: Regular discretization of time.
- **Asynchronous approach**: Event simulation principle.

$\rightarrow$ All parameters that shape an agent's internal architecture and model its behaviour (desires, beliefs, world representation, etc.) change instantaneously.

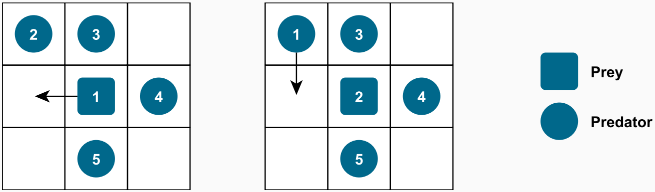
Synchronous approach

Very simple to use because it only need to program a very **simple function that activates** all the agents behaviors.



→ The activation order of the agents has a direct impact on the evolution of the world.

Synchronous approach: Order impact

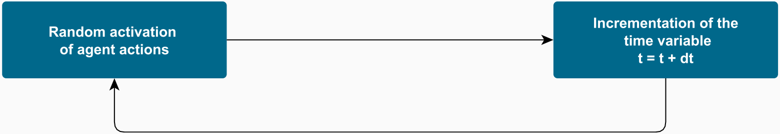


A prey is captured when it is surrounded by four predators following the vicinity of Von Neumann.

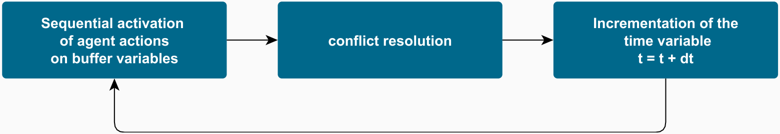
$\rightarrow$ From the same state of the world, the same behaviors can produce different states of the world according to the order of activation of the agents.

Synchronous approach: Random / Buffer memory

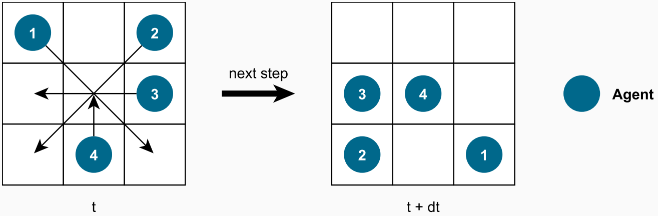
Use random activation process [1]



Use buffer variables



Temporal granularity of actions



An agent can move on many cells in one time step. Several agents may have paths that intersect without ever colliding (*speed not directly related to time*).

→ A consistency relationship between the time granularity of the actions and the time step must exist.

Asynchronous approach

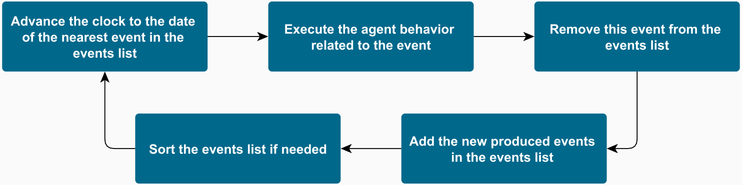
It is hard to picture a reality where a set of entities would be updated simultaneously by a global clock [4].

$\rightarrow$ It is more natural that each autonomous entity evolves at its own pace and generates events whose date of realization is personal to it. The agent activation is asynchronous.

Asynchronous approach

There are two ways to implement asynchronous approach:

- **Events are determined at the beginning** and the simulation consists in executing the list of events
- **Events are determined during the simulation** to take into account the effects of previous events



Asynchronous approach: Conflicts

Event conflict: Two events can occur at the same simulation time step.

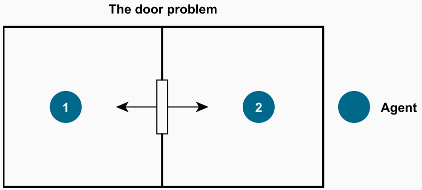
$\rightarrow$ The techniques seen above must be used to resolve the conflicts.

Competition conflict: Multi-Agent Based Simulation often used discrete time representation with events time representation. In that case, all agents do not act at each time step even if they are competing.

$\rightarrow$ Conflicts must be resolved by making a random choice to determine the scheduling in which the actions of the agents are validated.

Simultaneity issue

Whether the model uses a discrete or event time model, the real issue is the **representation of the simultaneity** of actions: The possibility of obtaining **several different system states** at $t + dt$ from depending on the scheduling of events at t.



$\rightarrow$ Multi-agent models remain in serialization or conflict resolution.

Simultaneity issue: Representation of actions

Multi-agent models use **the same representation of the actions**: The action of an agent is an event of the type *modify the state variable A with the value x*.

This representation of the action confuses **what is produced by the agents with what actually occurs**.

> *"They mix the gesture and the result of the gesture"*

→ Representation of the action that is inadequate in a concurrent context: Move towards new approaches like Influence/Reaction.

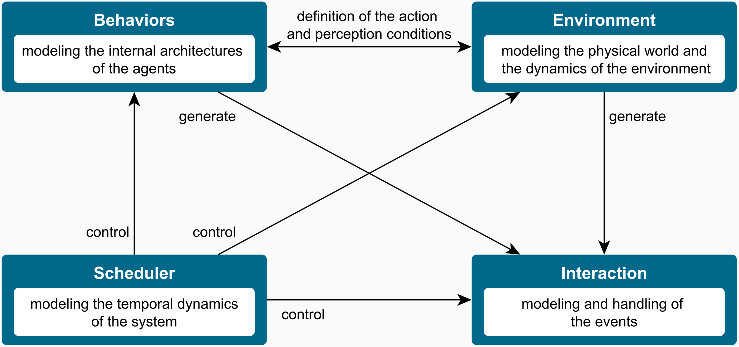
Resume

The four aspects of a multi-agent simulation model

To model and simulate an Agent Based Model, it is necessary to consider four fundamental aspects: Internal agent architecture, environmental model, time management and interaction management.

- **Behavior**: Define the modelling of agents' deliberation processes.
- **Environment**: Define the different physical objects of the world and the dynamics of the environment.
- **Scheduling**: Define how time evolves and the scheduling used.
- **Interaction**: Define the modeling of the result of the actions and the interactions that they produce.

The four aspects of a multi-agent simulation model



Implementing Agent Based Model with Mesa library

Open the document Self-organization of robots in a hostile environment

References

[1] Joshua M. Epstein and Robert L. Axtell.
Growing Artificial Societies: Social Science from the Bottom Up (Complex Adaptive Systems). The MIT Press, 1st printing edition, 1996. ISBN 0262550253.

[2] Edem Fianyo, Jean-Pierre Treuil, Edith Perrier, and Yves Demazeau. Multi-Agent Systems and Agent-Based Simulation: First International Workshop, MABS '98, Paris, France, July 4-6, 1998. Proceedings, chapter Multi-agent Architecture Integrating Heterogeneous Models of Dynamical Processes: The Representation of Time, pages 226–236. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-49246-7. doi: 10.1007/10692956_16. URL http://dx.doi.org/10.1007/10692956_16.

[3] Paul A. Fishwick. Computer simulation: Growth through extension. Trans. Soc. Comput. Simul. Int., 14(1):13–23, March 1997. ISSN 0740-6797. URL http://dl.acm.org/citation.cfm?id=264096.264103.

[4] Natalie S. Glance and Bernardo A. Huberman. Organizational fluidity and sustainable cooperation. In Cristiano Castelfranchi and Jean-Pierre Müller, editors, From Reaction to Cognition, 5th European Workshop on Modelling Autonomous Agents, MAAMAW '93, Neuchatel, Switzerland, August 25-27, 1993, Selected Papers, volume 957 of Lecture Notes in Computer Science, pages 89–103. Springer, 1993. ISBN 3-540-60155-4. doi: 10.1007/BFb0027058. URL https://doi.org/10.1007/BFb0027058.

[5] Yutao Guo, Jörg P. Müller, and Bernhard Bauer. A multiagent approach for logistics performance prediction using historical and context information. In 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 19-23 August 2004, New York, NY, USA, pages 1164–1171. IEEE Computer Society, 2004. ISBN 1-58113-864-4. doi: 10.1109/AAMAS.2004.10287. URL http://doi.ieeecomputersociety.org/10.1109/AAMAS.2004.10287.

[6] Michael Luck, Ronald Ashri, and Mark d'Inverno. Agent-Based Software Development. Artech House, 2004. ISBN 978-1-58053-605-9.

[7] H. Dyke Parunak, Robert Savit, and RickL. Riolo. Agent-based modeling vs. equation-based modeling: A case study and users' guide. In JaimeSimão Sichman, Rosaria Conte, and Nigel Gilbert, editors, Multi-Agent Systems and Agent-Based Simulation, volume 1534 of Lecture Notes in Computer Science, pages 10–25. Springer Berlin Heidelberg, 1998. ISBN 978-3-540-65476-6. doi: 10.1007/10692956_2. URL http://dx.doi.org/10.1007/10692956_2.

[8] Juan Pavón, Jorge J. Gómez-Sanz, and Rubén Fuentes. Model driven development of multi-agent systems. In Arend Rensink and Jos Warmer, editors, Model Driven Architecture - Foundations and Applications, Second European Conference, ECMDA-FA 2006, Bilbao, Spain, July 10-13, 2006, Proceedings, volume 4066 of Lecture Notes in Computer Science, pages 284–298. Springer, 2006. ISBN 3-540-35909-5. doi: 10.1007/11787044_22. URL https://doi.org/10.1007/11787044_22.

[9] Thomas C. Schelling. Micromotives and Macrobehavior. W. W. Norton, revised edition, September 1978. ISBN 0393329461. URL http://www.worldcat.org/isbn/0393329461.

[10] Robert E. Shannon. Simulation modeling and methodology. SIGSIM Simul. Dig., 8(3):33–38, April 1977. ISSN 0163-6103. doi: 10.1145/1102766.1102770. URL http://doi.acm.org/10.1145/1102766.1102770.

[11] Robert E. Shannon. Introduction to the art and science of simulation. In Proceedings of the 30th Conference on Winter Simulation, WSC '98, pages 7–14, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. ISBN 0-7803-5134-7. URL http://dl.acm.org/citation.cfm?id=293172.293175.

[12] Vito Volterra. Fluctuations in the abundance of a species considered mathematically. Nature, 118:558–560, 1926.

[13] Bernard P. Zeigler, Tag Gon Kim, and Herbert Praehofer. Theory of Modeling and Simulation. Academic Press, Inc., Orlando, FL, USA, 2nd edition, 2000. ISBN 0127784551.

Some References

- Robert E. Shannon (1977), Simulation modeling and methodology, SIGSIM Simulation Digital.
- Robert E. Shannon (1998), Introduction to the art and science of simulation, IEEE Computer Society Press.
- Bernard P. Zeigler (2000), Theory of Modeling and Simulation, Academic Press, Inc.