Multi-Agent Paradigm and **G**raphics **P**rocessing **U**nit Simulation

Fabien Michel

FMICHEL@LIRMM.FR

LIRMM
University of Montpellier - CNRS

RED seminar
October 17^{th} 2024



Academic backgrounds

Academic position

- PhD in **computer science** (2004)
- **Maître de conférences**, Reims 2005 → **Montpellier** 2008 -
- **LIRMM / SMILE research team**

Research topics

- **Multi-Agent Based Simulation, MABS**
  - generic **models** (e.g. IRM4S)
  - generic **tools** (e.g. TurtleKit)

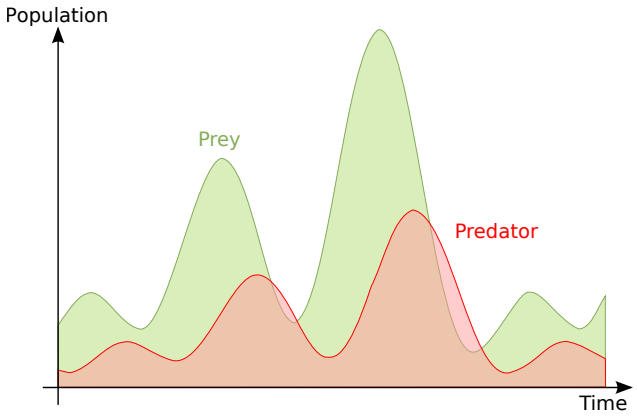
- **Agent Oriented Software Engineering, AOSE**
  - **abstractions and models** (e.g. E4MAS)
  - generic **tools** (e.g. MaDKit)

Equation-Based Modeling limitations

$$\frac{dx}{dt} = \alpha x - \beta xy$$

$$\frac{dy}{dt} = \delta xy - \gamma y$$

[Lotka-1925]-[Volterra 1926]
Prey Predator model



Limitations of using only a **macro-level perspective**:

- Integrating micro-level considerations:
  - e.g. existence of refuges for prey [Gause, 1934]
  - spatial repartition of the population
  - diversity among individuals
- Integrating relationships between micro and macro levels:
  - Does the global state influence individuals? [Orcutt, 1957]
  - Study emergent phenomena coming from the micro-level

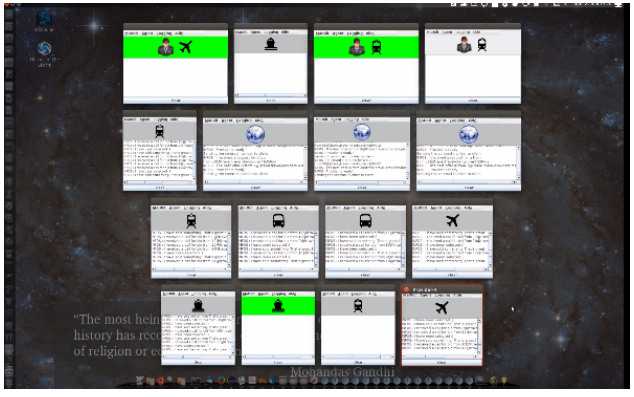
Motivations for Multi-Agent Based Simulation?

MABS is a complementary approach addressing EBM limitations

Modeling at the micro-level:

- Explicitly model the **individuals**, their **states** and **behaviors**
- Explicitly model their **environment** and its **micro-level dynamics**
- Explicitly model the **interactions** among individuals and with the environment

Multi-Agent Systems



DAI-based software systems
e.g. Contract Net

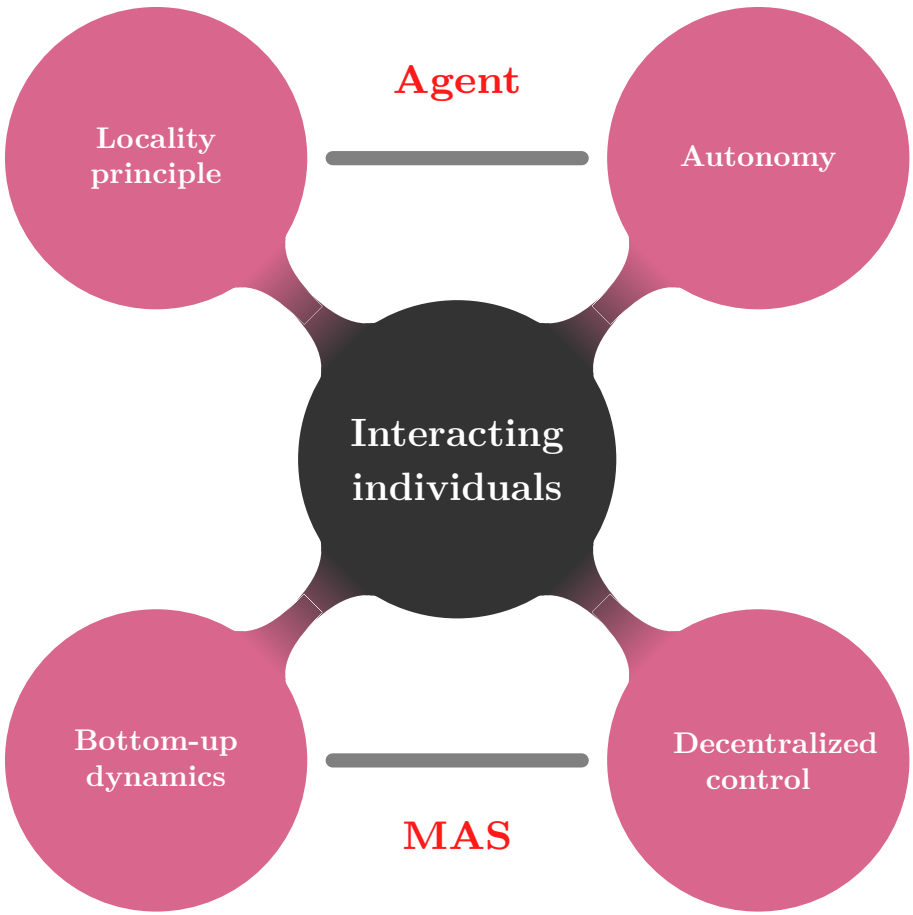
Collective robotics
e.g. RoboCup.org

Interacting individuals

Natural systems
e.g. ant colony

Human systems
e.g. traffic and crowd

Multi-Agent Systems



No global view

Local effects

Proactivity

No direct intervention

Unpredictability

Emergence

Self-organization

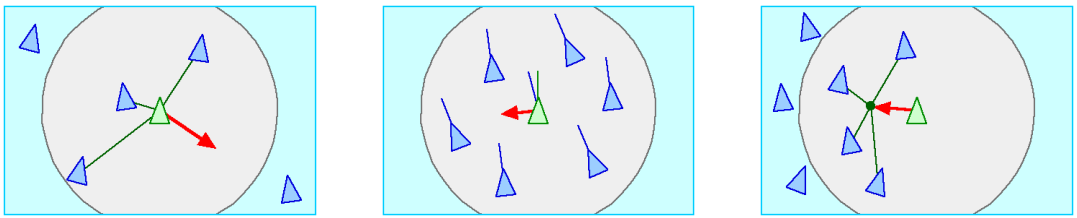
Resilience

The ***boids*** model [Reynolds 1987]

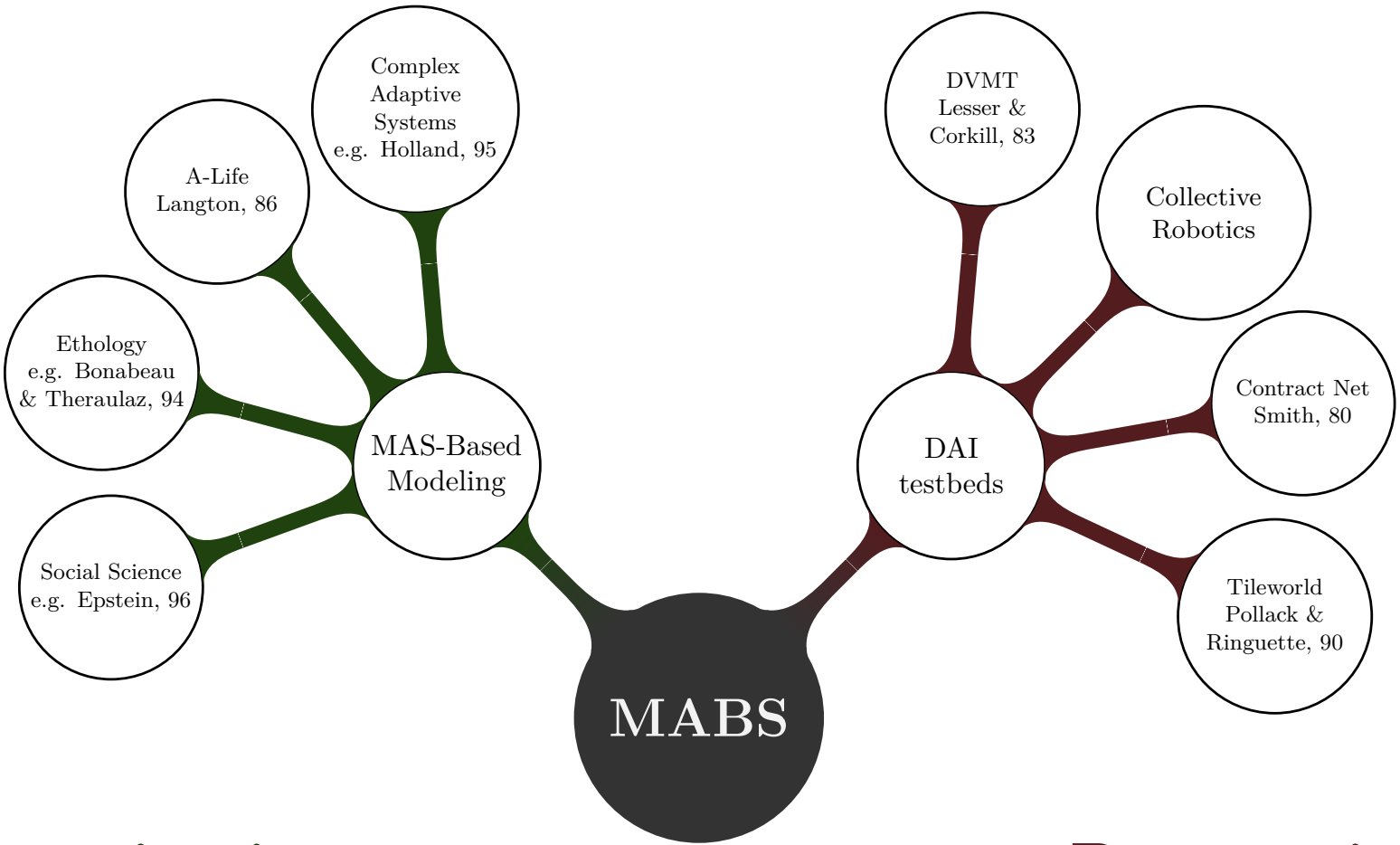
Bird flocks → MAS

as **three local rules → Agent**

separation *alignment* *cohesion*



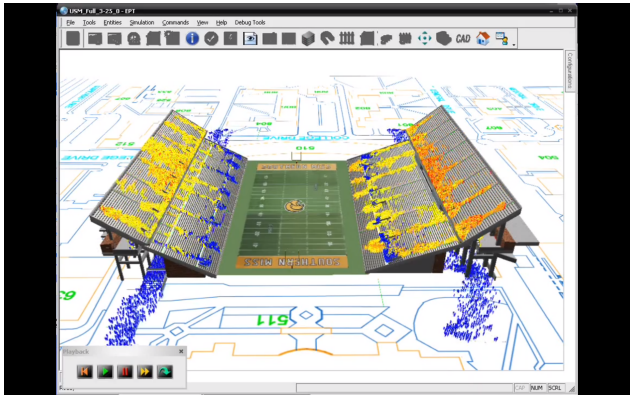
Multi-Agent Based Simulation



Investigating MAS

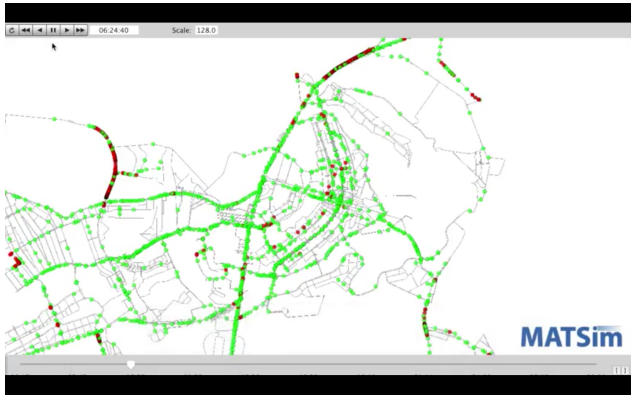
Prototyping MAS

MABS application domain examples



Evacuation scenario
e.g. REGAL Evac

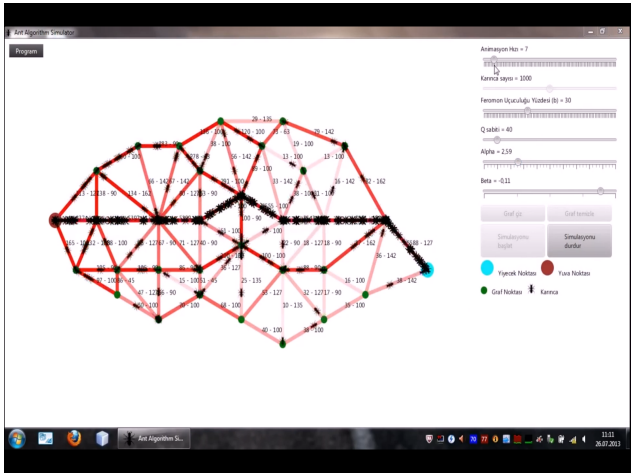
Investigating



Traffic simulation
e.g. MATSim

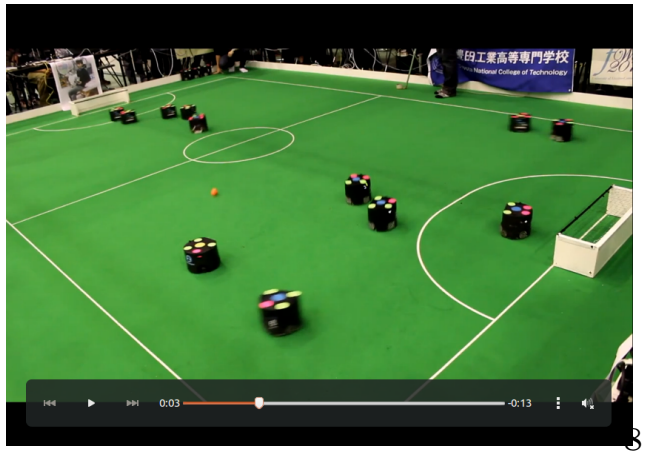
MABS

Ant Colony Optimization
e.g. TSP resolution

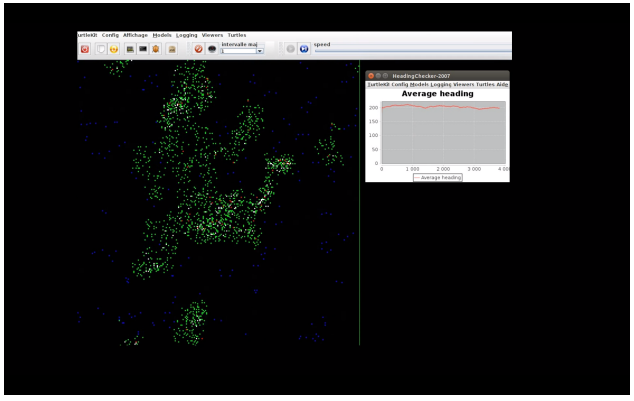


Prototyping

Collective robotics
e.g. Designing DAI strategies

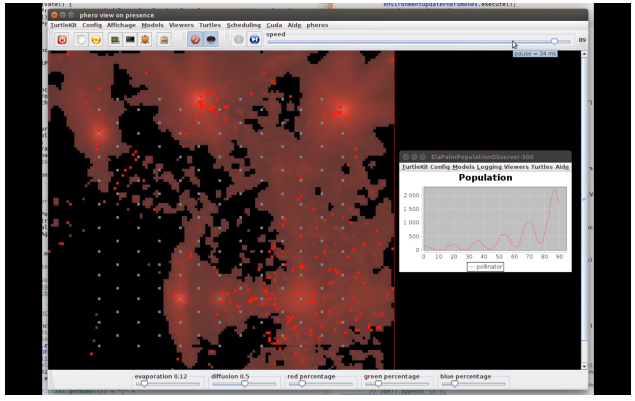


MABS application domain examples



Flocking [ECAL'17]

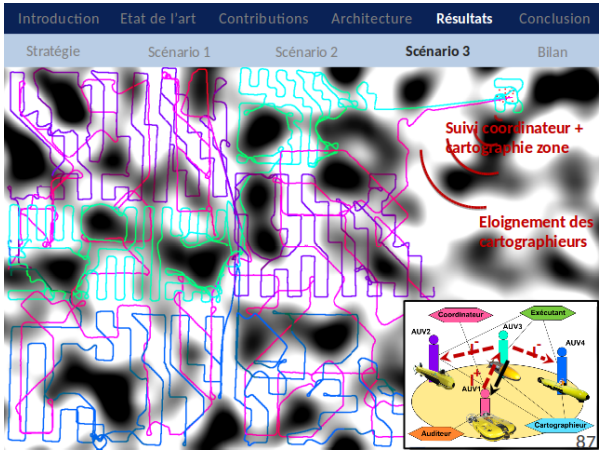
Investigating



Pollination dynamics [MMAS 2018] collaboration with CIRAD

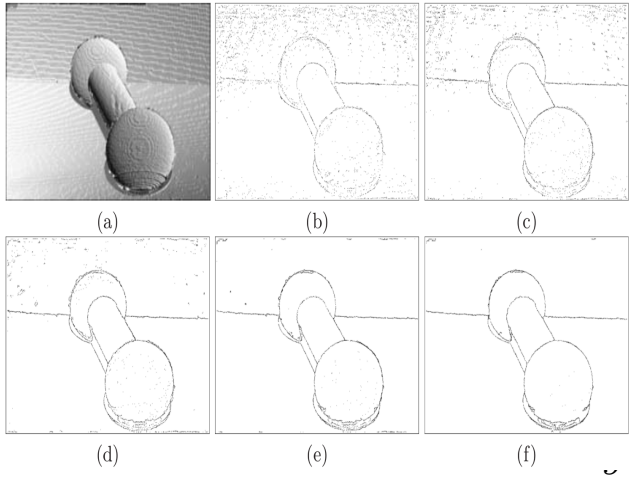
MABS

Underwater collective robotics Carlesi's PhD @ LIRMM



Prototyping

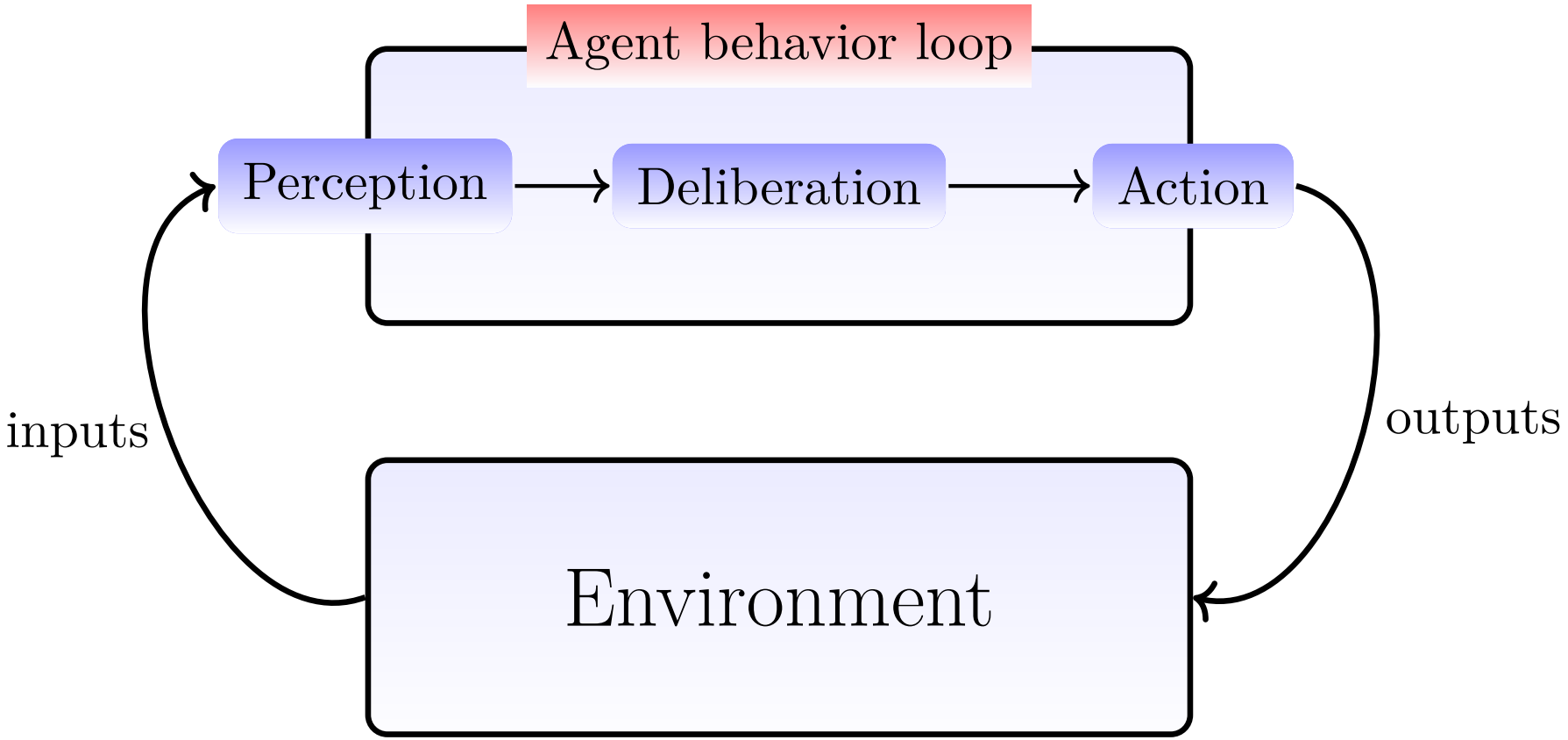
Image segmentation Mazouzi's PhD @ CReSTIC



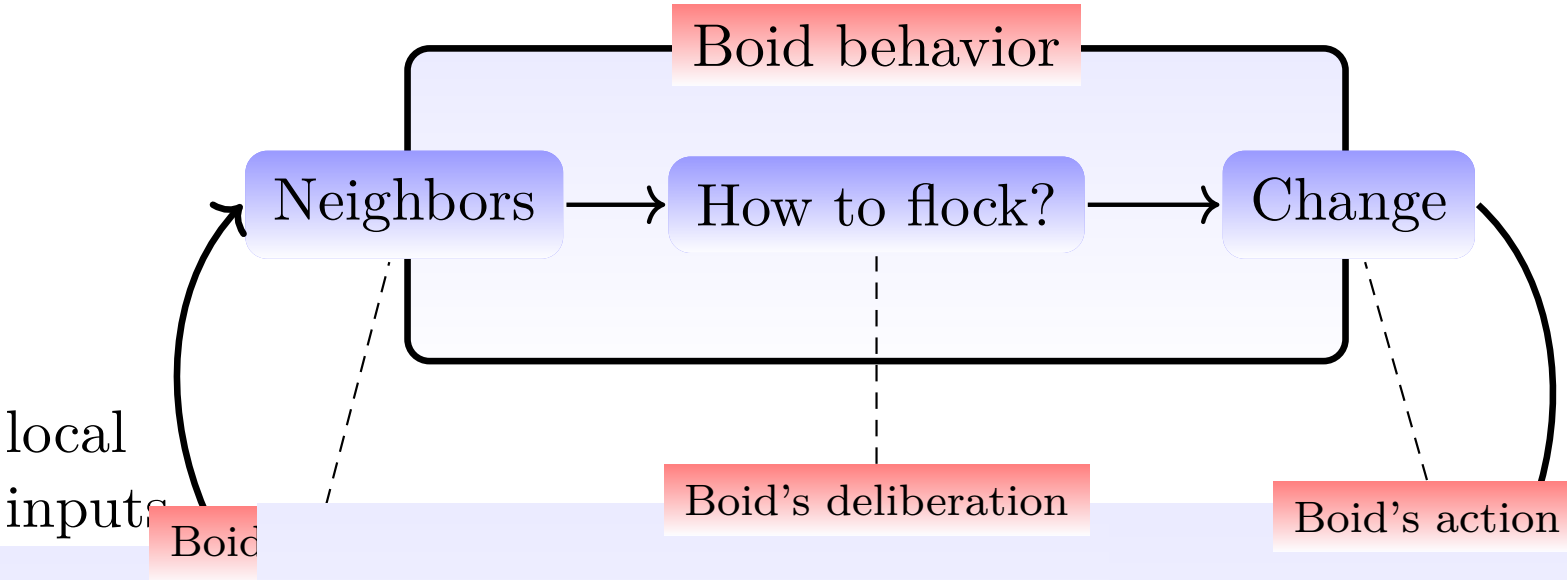
Do not get confused...

- Individual-Based Modeling **IBM**
- Multi-Agent Based Simulation **MABS**
- Agent-Based Modeling/Models **ABM**
- Agent-Based Simulation **ABS**
- Agent-Based Social Simulation **ABSS**
- ...

Modeling agent behaviors



Modeling *boid* behaviors

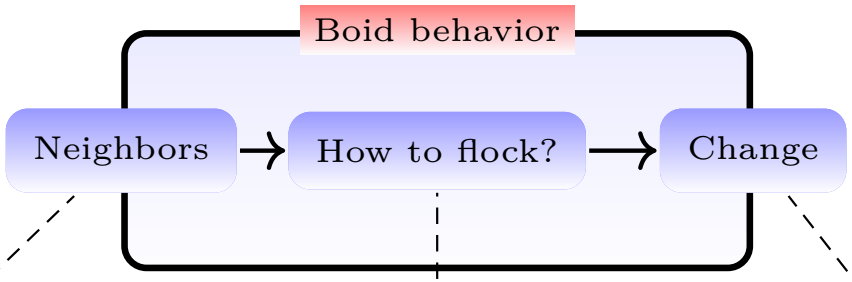


- others
- others

- computing average neighbors
- computing average neighbors
- choosing own speed
- choosing own orientation

- change speed
- change orientation

Modeling *boid* behaviors... Not so simple



Boid's percepts

- how many neighbors?
- front?
- back?

Boid's deliberation

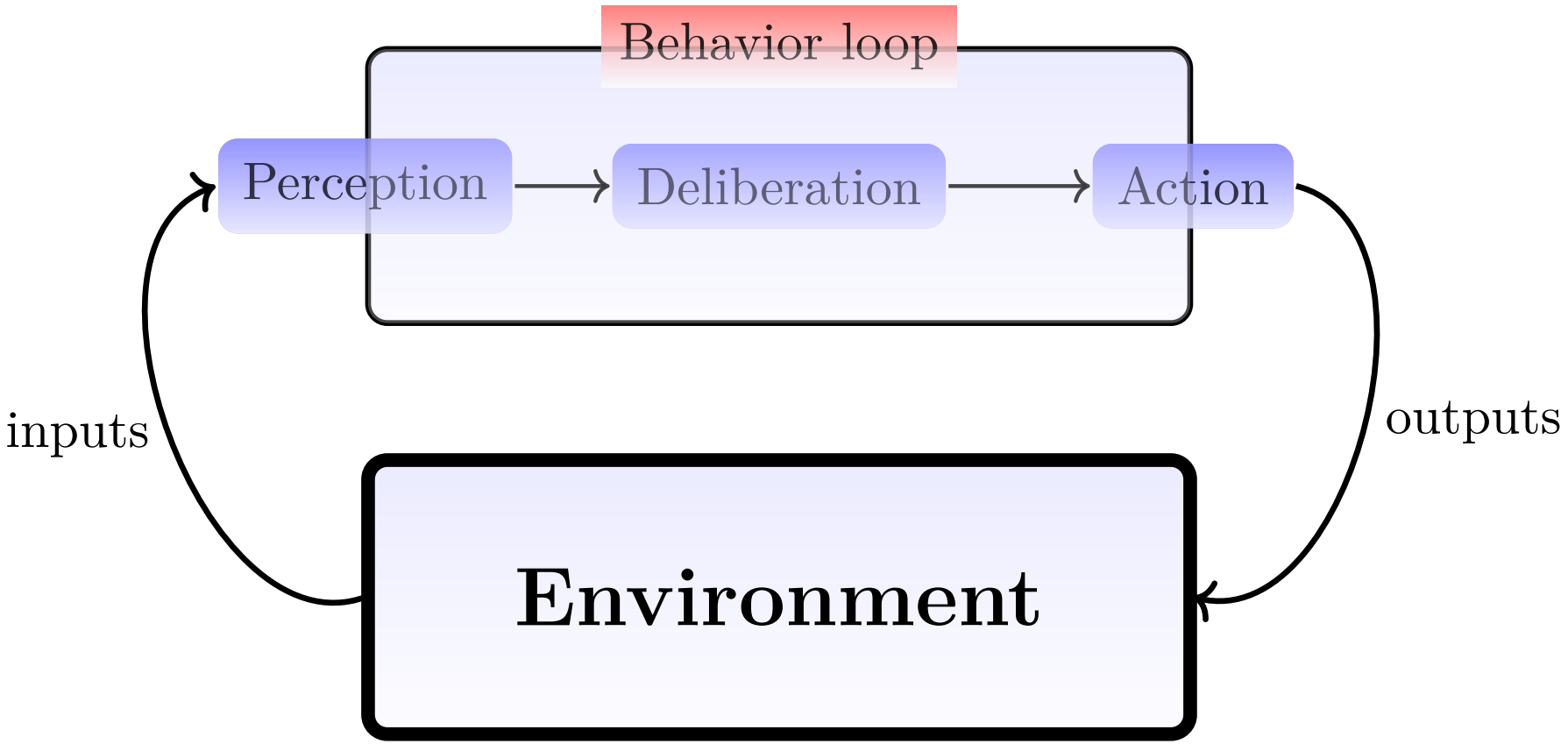
- which type of flocking?
- starlings murmuration?
- formation flying?

Boid's action

- acceleration?
- flying abilities?

- stability?
- does the size of the flock matter?

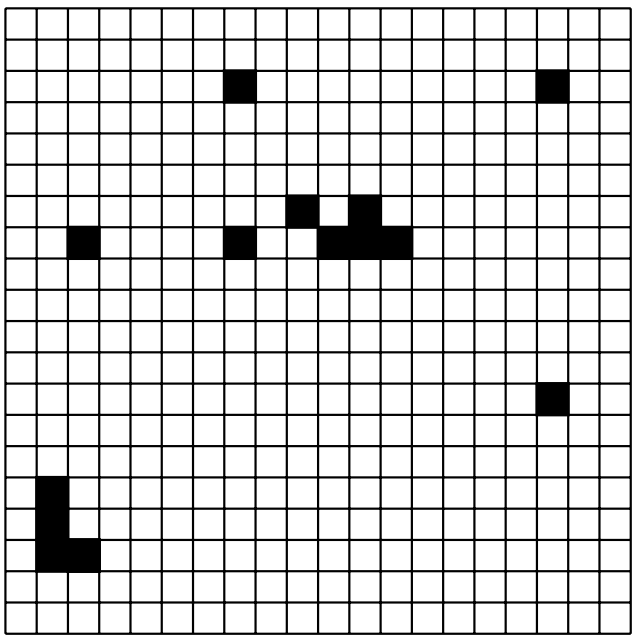
Modeling agent environments

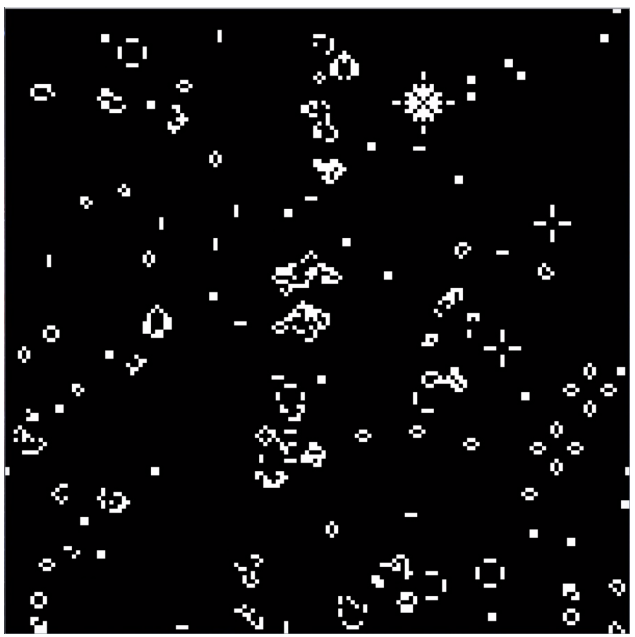


Conway's Game of Life (1970)

A **cell** is either **dead** or **alive** and follows these rules:

- A **live cell** with
  - 2 or 3 live neighbors stays alive
  - more than 3 live neighbors dies (underpopulation)
  - with less than 2 live neighbors dies (overpopulation)
- A dead cell with 3 live neighbors becomes alive (reproduction)

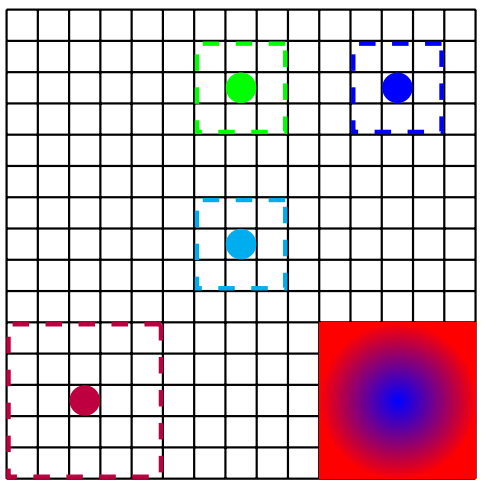




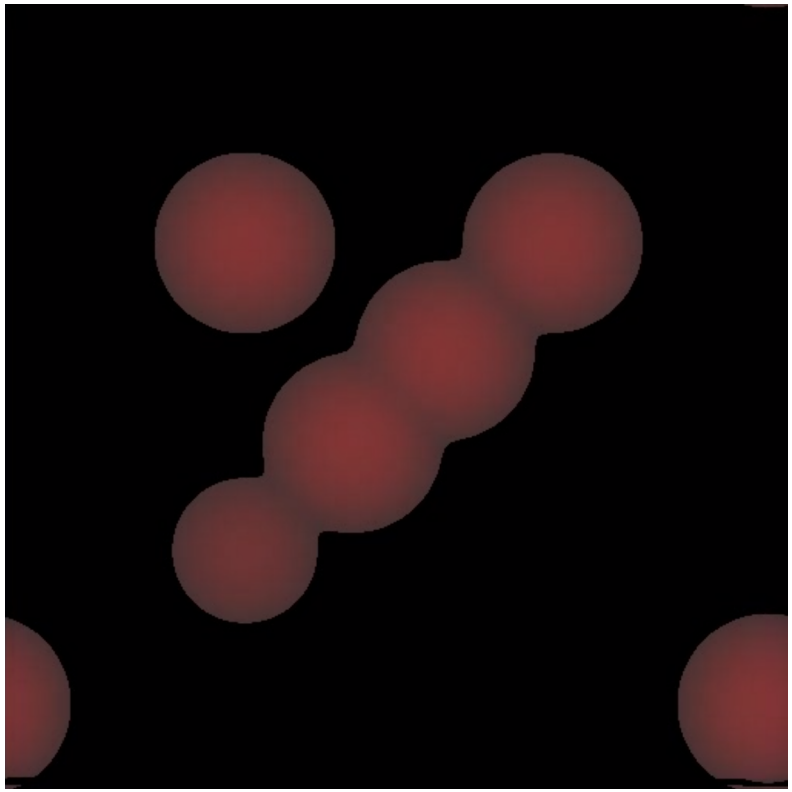
Agent environments as (2D) cell grids

Cell grids are widely used as environments in MABS:
Providing a simple **discretization of space**

- Naturally defines locality of agents percepts and actions
- Easily defines environments with no boundaries
- Allow the modeling of powerful environmental dynamics!

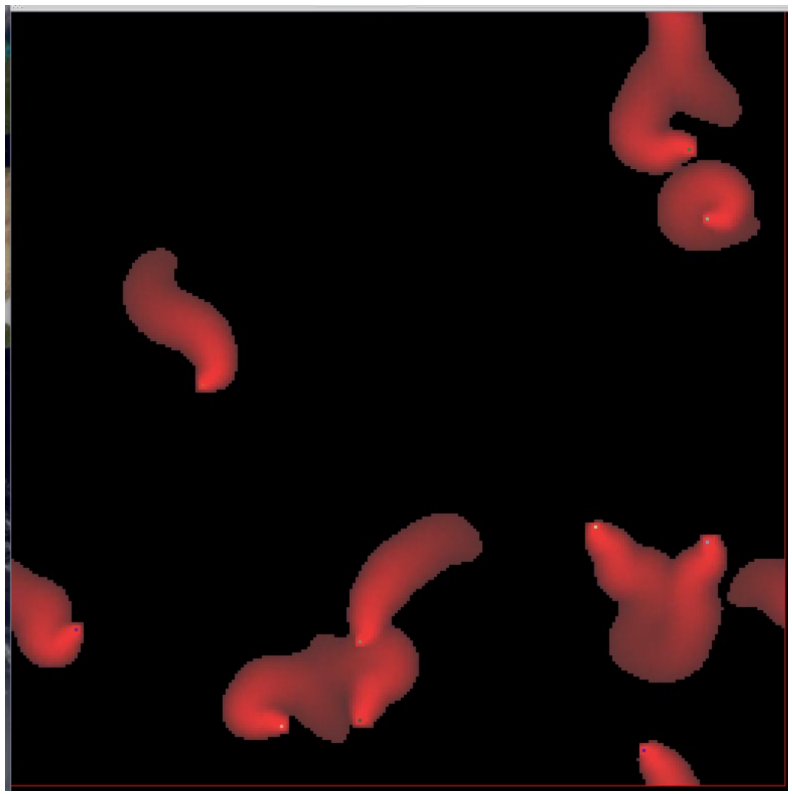


Modeling digital pheromones



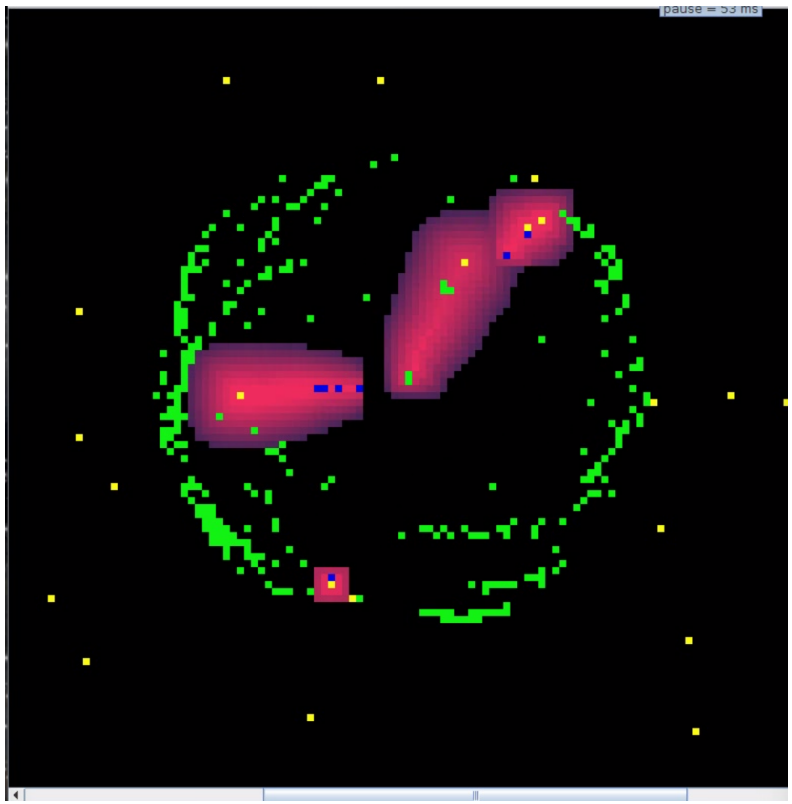
Digital pheromones model chemical substances that **diffuse** and **evaporate**

Mixing agents and pheromones



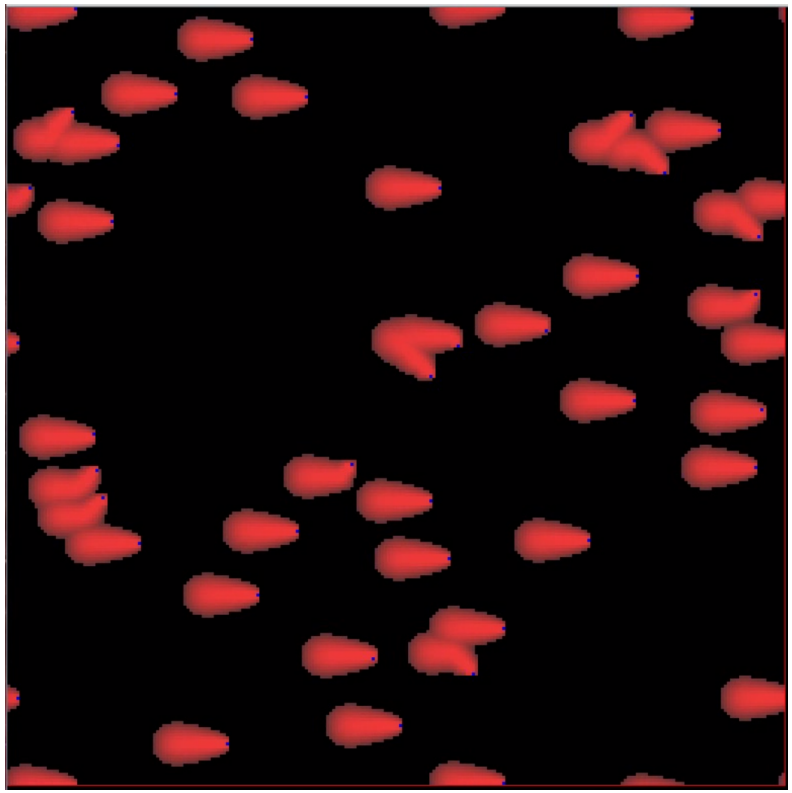
Agents (e.g. ants) can emit pheromones

Modeling foraging ants



Ants search for food, take some, and then return to the nest leaving pheromone trails

Evap: Very simple behaviors with emergent features



Inspired by the *Evap model* [Chu et al., 2007]

```
private String behavior() {
        emitPheromone();
        followPheromoneMinGradient();
        moveForward();
        return SAME_BEHAVIOR;
}
```

Another great simple model relying on pheromones [Beurier's PhD, 2007]

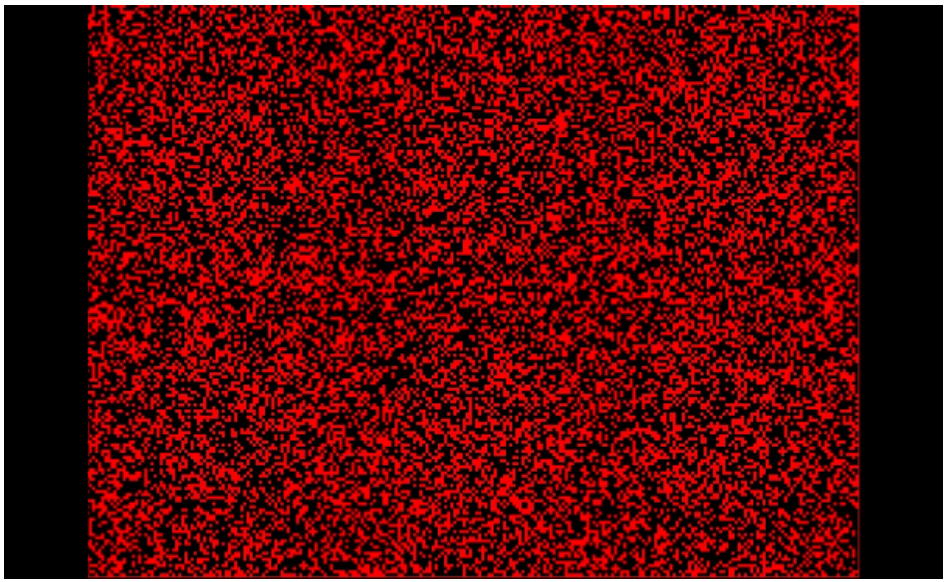
Agents are defined by a single property, i.e. their *level*: n

- All agents begin with level $n = 1$

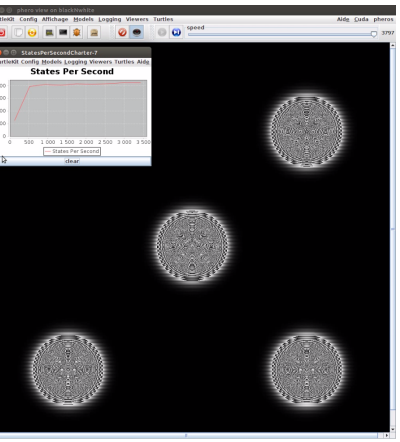
One single agent behavior

- Agents emit 3 pheromones according to their level n:
  - (1) ***attraction-n***, (2) ***repulsion-n***, (3) ***presence-n***
- Agents perceive 3 pheromones and act accordingly :
  - ***attraction-(n+1)***: Go toward *level-(n+1)* agents
  - ***repulsion-(n+1)***: Not too close! move around the gradient
  - ***presence-n***: Mutate to *level-(n+1)* if the place is crowded

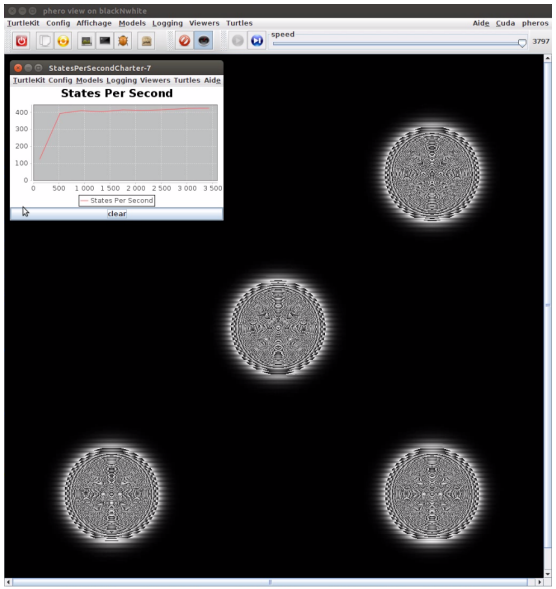
Let us run this model...



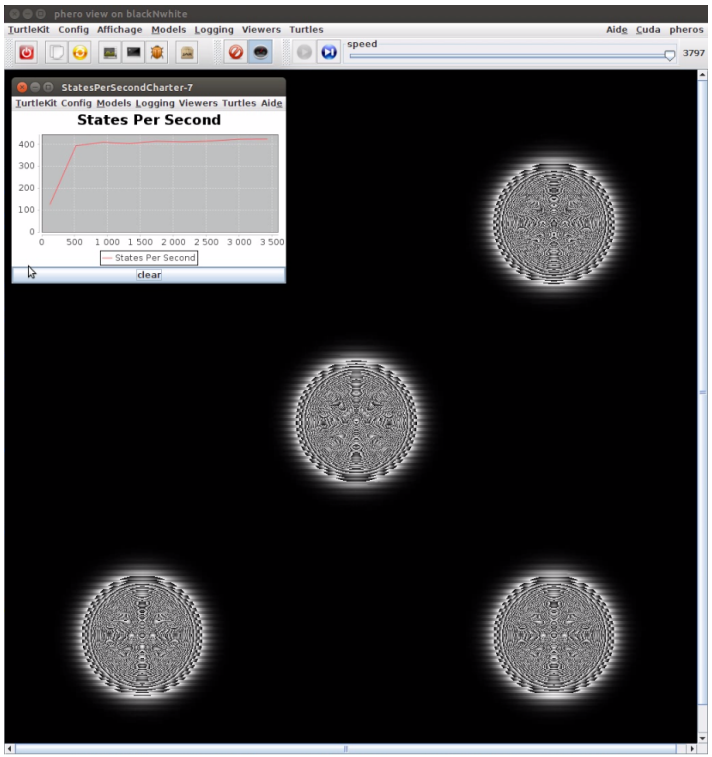
Pheromone dynamics cost is very high...



256 x 256



512 x 512



1024 x 1024

Computation time is a major issue for MABS

Each part of a MABS is a potential bottleneck

- Number of agents
- Behaviors complexity
- Environment size and dynamics (e.g. pheromones)

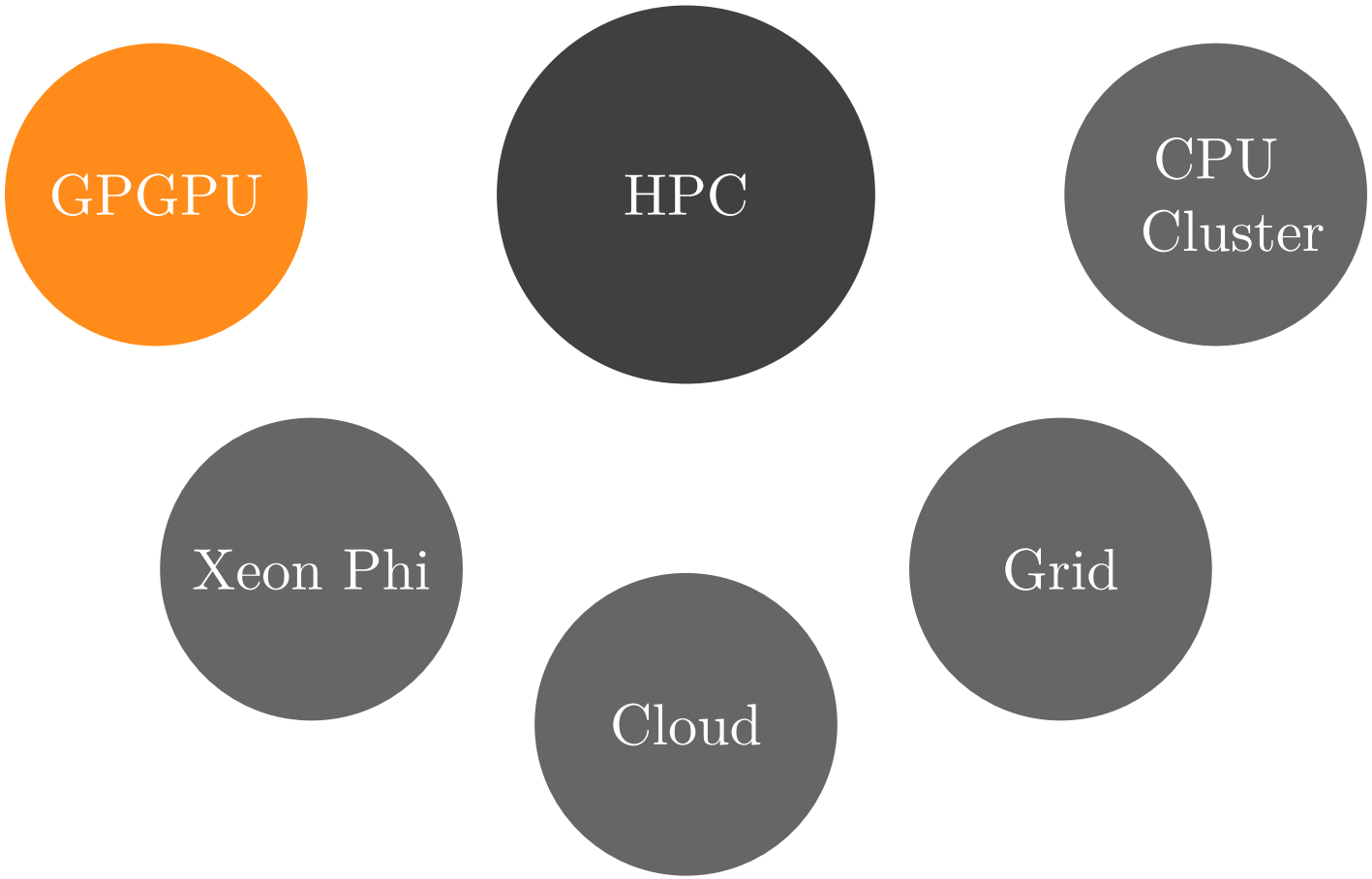
You can get stuck while prototyping your model!

Solutions for dealing with 'large scale' MABS

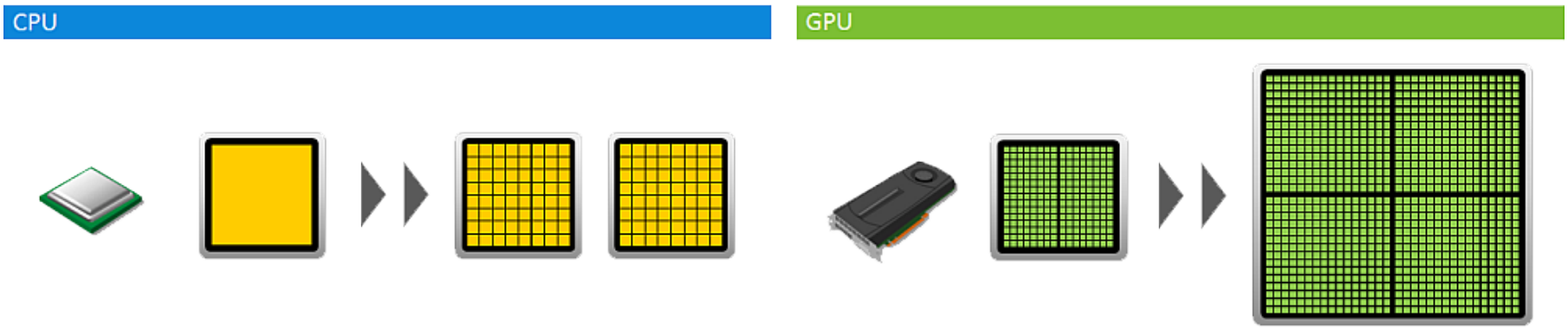
Some "solutions" to the speed issue: (see *e.g.* [Parry and Bithell, 2012])

- Buy the **latest gamer desktop**?!
  - $\Rightarrow$ Not very relevant, only few gains can be expected...
- **Forget scalability**!
  - Limit the number of agents, downsize environment, ...
  - Downgrade behavior complexity / environment dynamics
- **Change your modeling**!
  - Model *Super individuals*
  - Use a fractal modeling approach
- **Forget about agents**! and get depressed...
  - Revert to an equation-based approach for modeling some parts of the system, when possible...

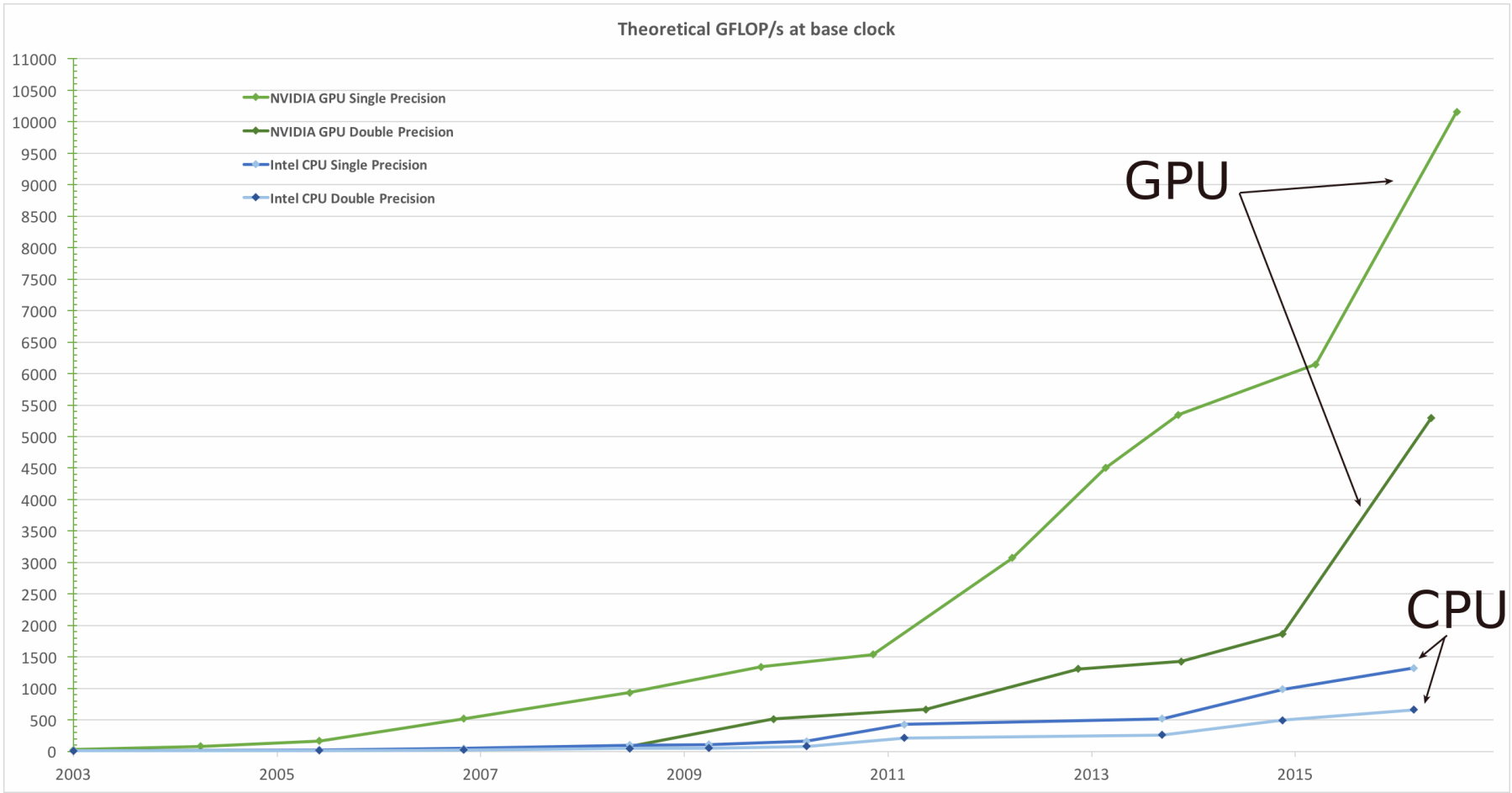
High Performance Computing



CPU vs GPU architectures



CPU vs GPU theoretical performances



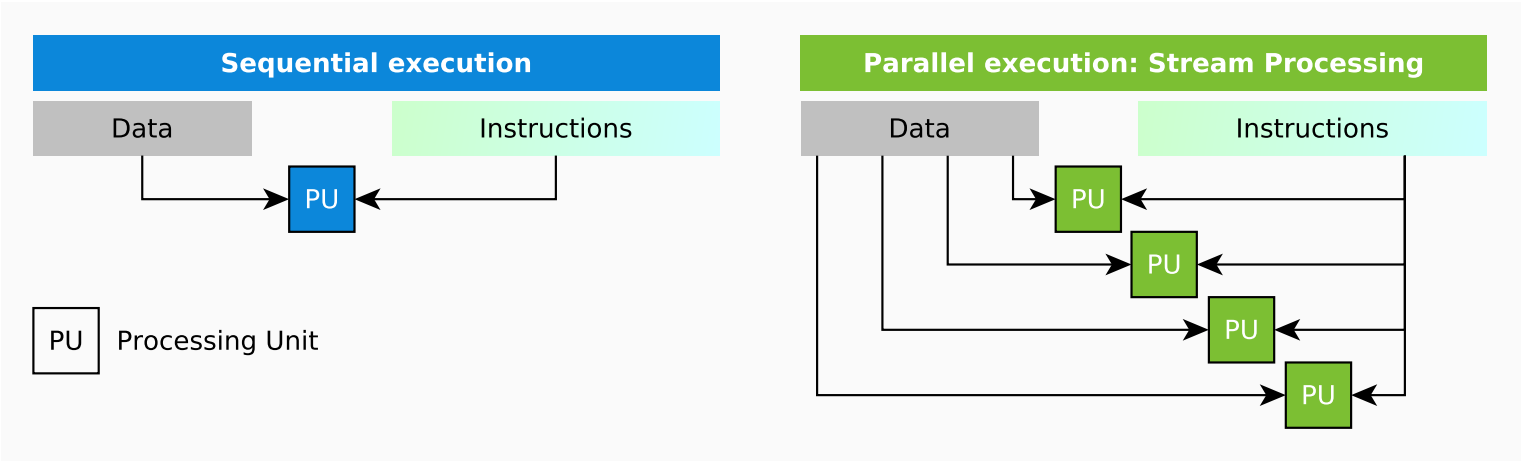
What is **GPGPU ?**

General-**P**urpose computing on **G**raphics **P**rocessing **U**nits:

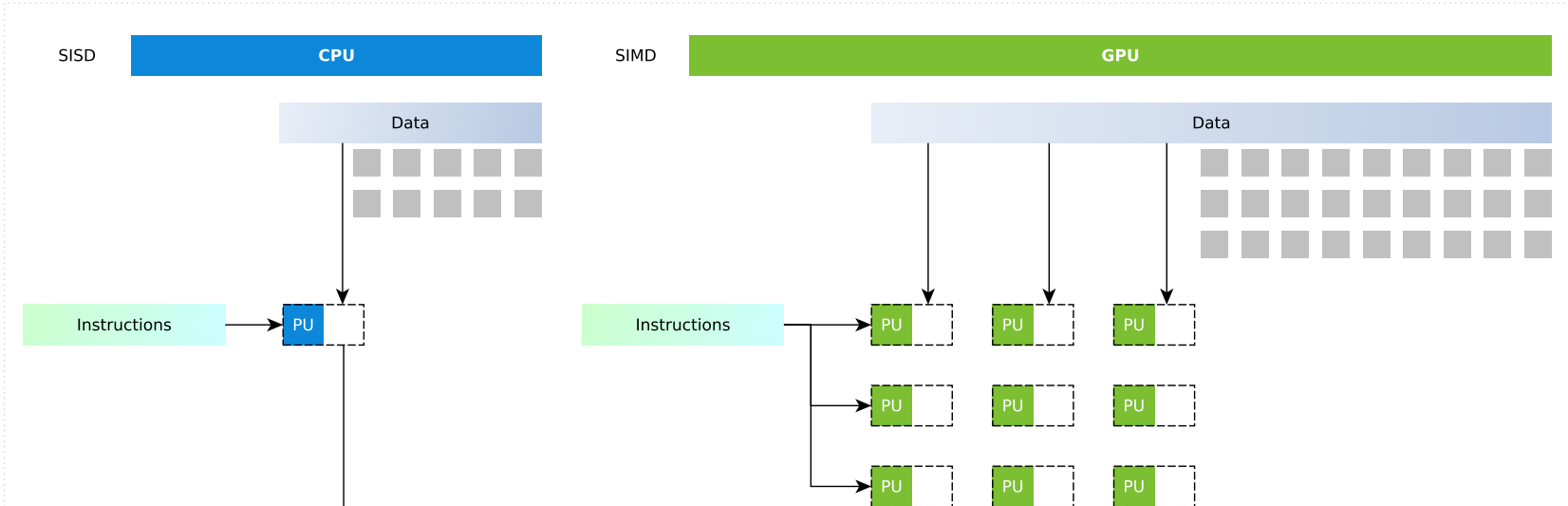
Using the **massively parallel architecture**
of usual PC **graphics cards** to do
General-Purpose Computing

GPGPU Principle

***S**imple **I**nstructions **M**ultiple **D**ata* (stream processing)



Consists in executing simultaneously a **kernel** (a series of computations) on a **data set** (the flow / stream)



Why using GPGPU for MABS?

As a **H**igh **P**erformance **C**omputing solution, GPGPU is:

- **Very cheap and common hardware**:
  Your laptop probably has one GPGPU compliant device.

- **Vs. cluster of CPUs:**
  - Distributing MABS has proved very challenging
  - Parallelizing multiple runs doesn't help prototyping MABS

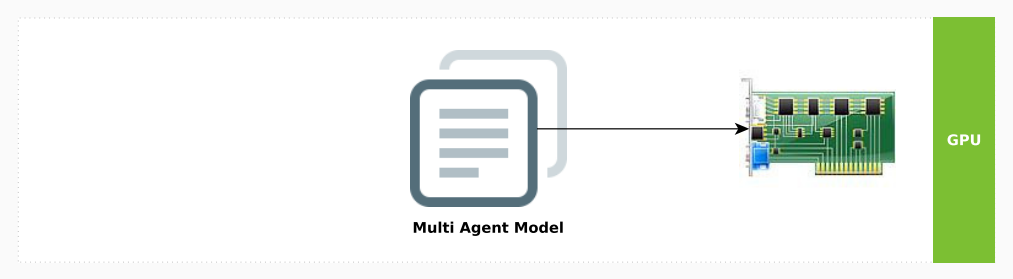
BUT **there is a catch!**

Programing the *massively parallel way*!

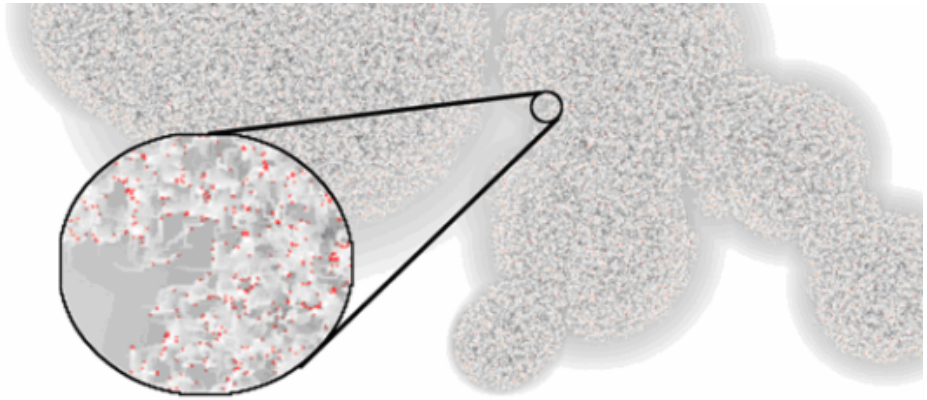
***S**ingle **I**nstructions **M**ultiple **D**ata*
stream processing

GPGPU 4 MABS History

All-in-GPU



[D'Souza et al., 2007]Sugarscape with 2 millions of agents, 2560x1024, 196 cores



Very promising performances but **unmaintainable** and **hardly reusable**

2007 – *SugarScape on steroids* [D'Souza et al., 2007]

[Perumalla and Aaby 2008]

First study on using GPGPU for MABS

Purpose	Evaluate the advantages and drawbacks of GPGPU
Experimentation	Compare CPU and GPU implementation of several MABS
Conclusion	Performances are only obtained at the expense of **programmability**, **modularity** and **reusability**

Release of CUDA (2007)

2007 – *SugarScape on steroids* [D'Souza et al., 2007]

2008 – *First GPU framework* [Lysenko and D'Souza, 2008]

– *ABGPU* [Richmond and Romano, 2008]

– **First flocking simulation** [Passos et al., 2008]

2009 – **Flame GPU** [Richmond et al., 2010]

– **First crowd and traffic simulation** [Strippgen and Nagel, 2009]

2010

2011 – **Pedestrian framework** [Richmond and Romano, 2011]

2012

–

2013

2014

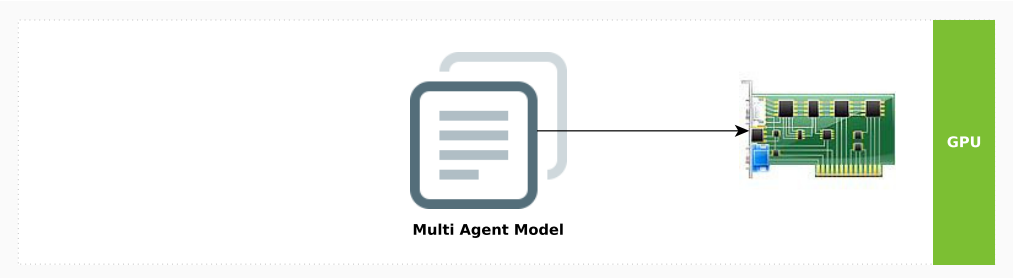
2015

–

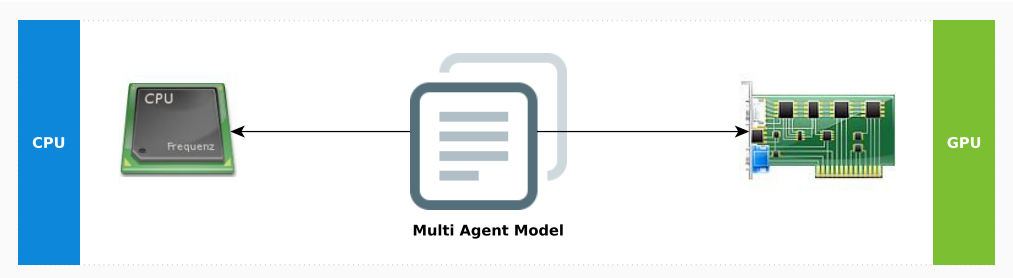
2016

Evolution: All-in-GPU $\longrightarrow$ Hybrid

All-in-GPU



Hybrid



Hybrid Approaches

2007 – *SugarScape on steroids* [D'Souza et al., 2007]

2008 – *First GPU framework* [Lysenko and D'Souza, 2008]

- – *ABGPU* [Richmond and Romano, 2008]

- – First flocking simulation [Passos et al., 2008]

2009 – Flame GPU [Richmond et al., 2010]

- – First crowd and traffic simulation [Strippgen and Nagel, 2009]

2010 – First hybrid simulation [Vigueras et al., 2010]

2011 – Pedestrian framework [Richmond and Romano, 2011]

2012 – **Sworm GPU [Laville et al., 2012]**

- – **TurtleKit GPU [Michel, 2013]**

2013 – **[Pavlov and Müller, 2013]**

2014 – **MCMAS [Laville et al., 2014]**

2015 – **SIMILAR [Abouaissa et al., 2015]**

- – **MASON with multiple GPU [Hoetal.,2015]**

2016 –

Overview of GPGPU for MABS

Reference	Approach	Implementation	Agent
[D'Souza et al., 2007]	*All-in-GPU*	Graphic functions	Reactive
[Lysenko and D'Souza, 2008]	*All-in-GPU*	Graphic functions	Reactive
[Richmond and Romano, 2008]	*All-in-GPU*	Graphic functions	Reactive
[Perumalla and Aaby, 2008]	*All-in-GPU*	Graphic functions	Reactive
[Erra et al., 2009]	*All-in-GPU*	CUDA	Reactive
[Richmond et al., 2010]	*All-in-GPU*	CUDA	Reactive and deliberative
[Husselmann and Hawick, 2011]	*All-in-GPU*	CUDA	Reactive and heterogeneous
[Richmond and Romano, 2011]	*All-in-GPU*	CUDA	Reactive and deliberative
[Laville et al., 2012]	*Hybrid*	C + OpenCL	Reactive and deliberative
[Pavlov and Müller, 2013]	*Hybrid*	C + CUDA	Reactive and deliberative
[Michel, 2013]	*Hybrid*	Java + CUDA	Reactive and deliberative
[Laville et al., 2014]	*Hybrid*	Java + OpenCL	Reactive and deliberative
[Ho et al., 2015]	*Hybrid*	CUDA	Reactive and deliberative
[Shekh et al., 2015]	*Hybrid*	C + CUDA and OpenCL	Reactive and deliberative

[Hermellin and Michel, RIA 2015]

All-in-GPU vs. Hybrid

All-in-GPU	**Hybrid**
⇓	⇓
Works that are only interested in performance gains	Works that ease the use of GPGPU
⇓	⇓
Direct use of GPGPU	**Transparent use of GPGPU**

Hiding the use of GPGPU

Not generic enough because of the wide variety of MABS models

The GPU Delegation Approach

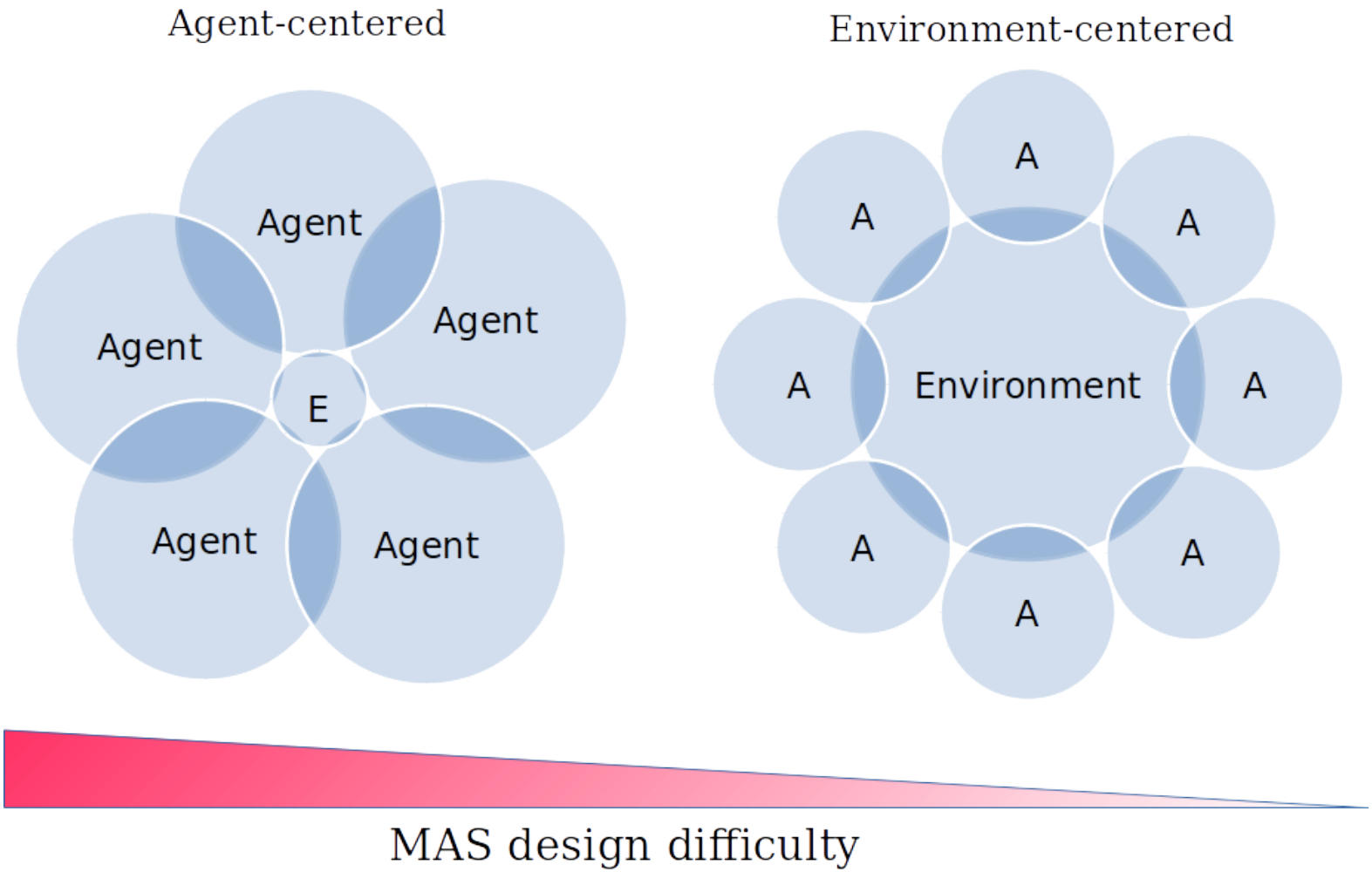
Our objectives

Provide means for actually **easing GPU programming for MABS**

The E4MAS perspective [Weyns et al. 2004]

"Moving complexity from agents to environment"

e.g. [Michel, 2015] HDR



Environment 4 Multi-Agent Systems

Agent-centered view: Not engineering the environment

- The whole MAS application logic relies on the agents (*e.g.* interaction protocols)
- But they are constrained by locality and autonomy
- $\Longrightarrow$ promotes complex behaviors

Environment-centered view: Engineering the environment

- Not restrained to autonomy or locality
- It mediates/constrains interactions as required
- $\Longrightarrow$ free agents from managing interactions (*e.g.* ants)
- $\Longrightarrow$ simplify the design of behaviors

The GPU Delegation Approach

Our objectives

Provide means for actually **easing GPU programming for MABS**

Our approach ← E4MAS trend

Using a **clear separation** between

agent behaviors managed by the **CPU**

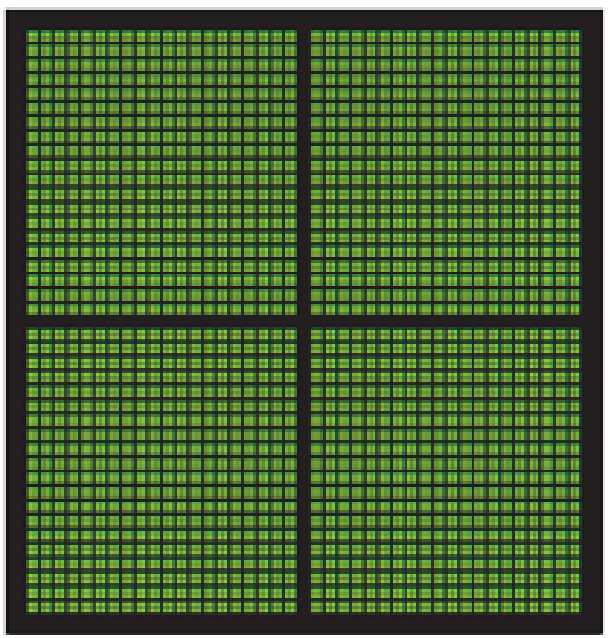
vs.

environmental dynamics managed by the **GPU**

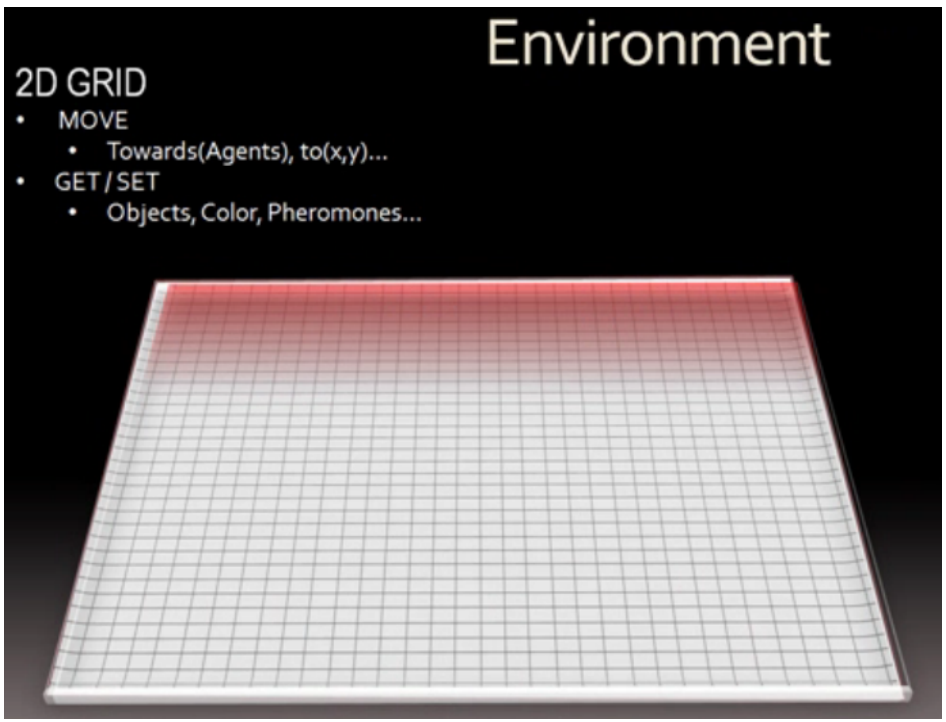
Using an E4MAS Perspective

Focus on the environment, not the agents

GPU architecture



Grid-based environment model



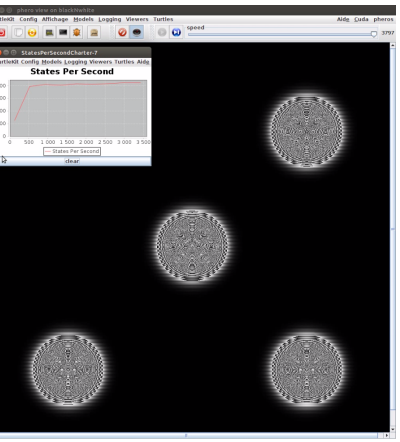
Pheromone dynamics: Data diffusion and evaporation

Pheromone evaporation sequential implementation

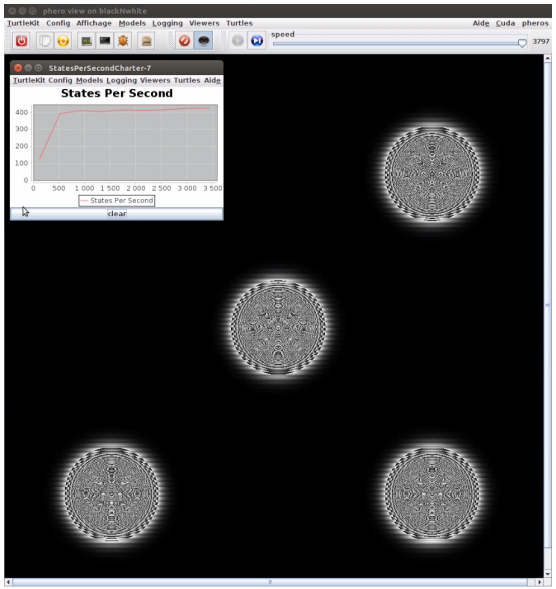
Algorithm 1 evaporation($cells, width, height, evapCoef$)

for $i = 0$ to $gridWidth$ **do**
  for $j = 0$ to $gridHeight$ **do**
    $cells[i][j] \leftarrow cells[i][j] * evapCoef$
  end for
end for

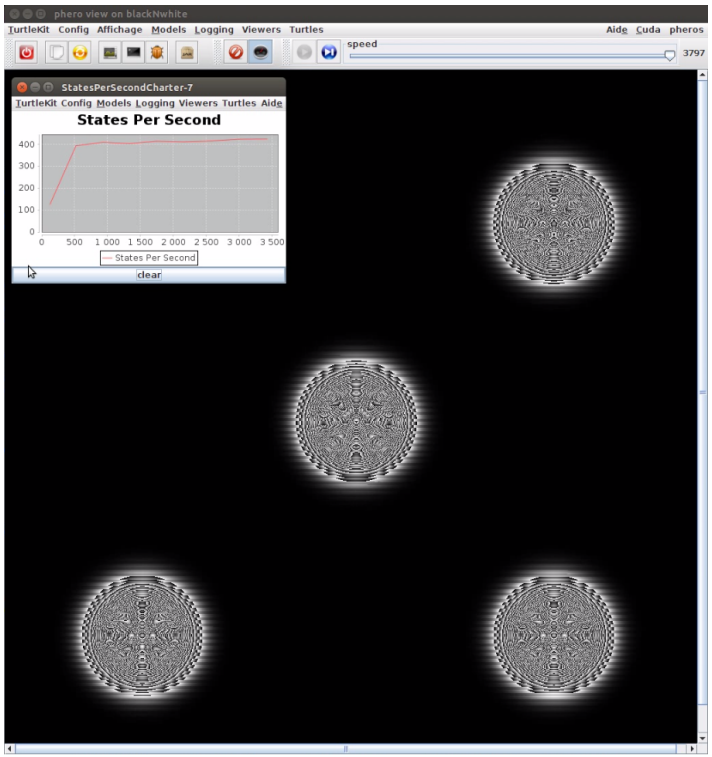
Pheromone dynamics cost is very high...



256 x 256



512 x 512



1024 x 1024

. . . but can be easily ported on GPU

Evaporation GPU implementation $\Longrightarrow$ GPU module

Algorithm 2 GPU_evap($cells, width, height, evapCoef$)

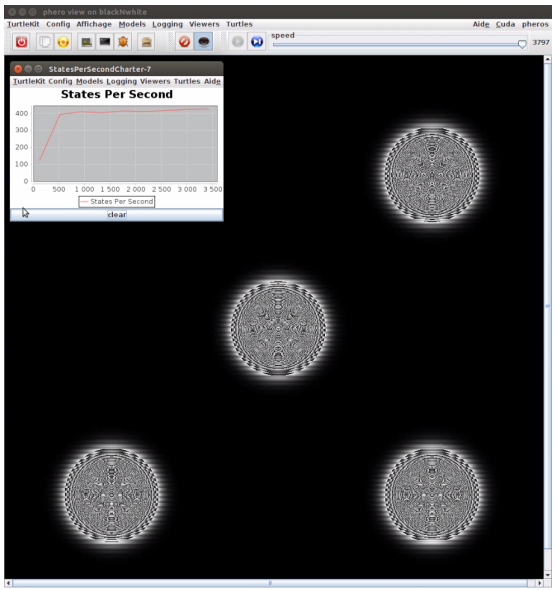
$i \leftarrow blockIdx.x * blockDim.x + threadIdx.x$; // 1st GPGPU idiom
$j \leftarrow blockIdx.y * blockDim.y + threadIdx.y$; // thread's location

//2nd GPGPU idiom: Check that the thread is inside data boundaries
if $(i < gridWidth$ **and** $j < gridHeight)$ **then**
  $cells[i][j] \leftarrow cells[i][j] * evapCoef$ // the kernel is a one-liner
end if

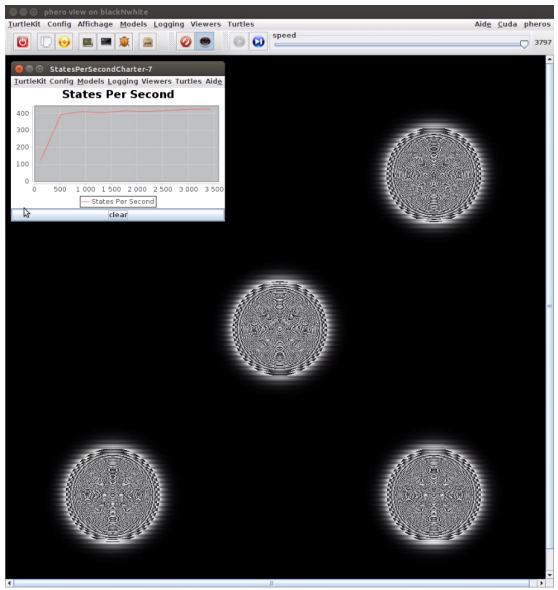
No loops!

The code is in the structure

Pheromones: CPU vs. GPU



1024 x 1024 CPU

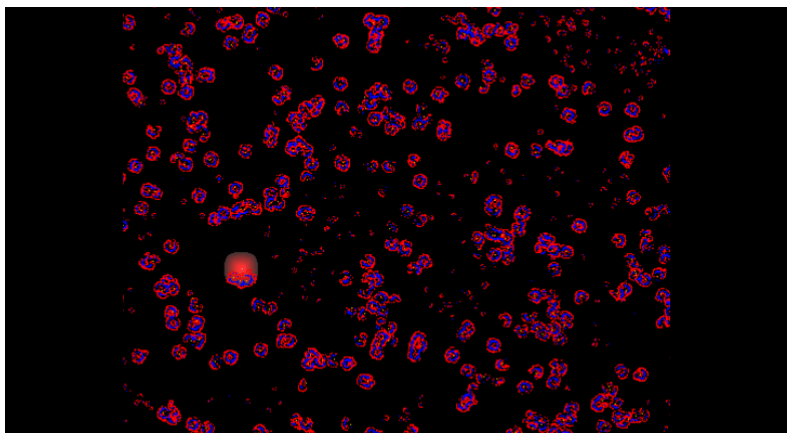


1024 x 1024 GPU

Benefits

- **performance** and **scalability**
- agent API untouched $\Longrightarrow$ **reusability; genericity**

[Beurier's PhD, 2007] was about modeling...



Level-1
Level-2
Level-3
Level-4

Applying GPU Delegation on Agents

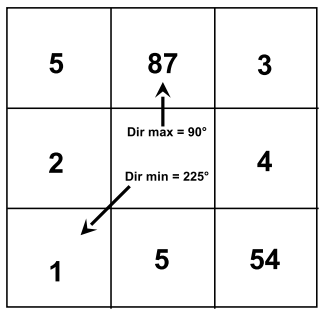
Main idea: E4MAS

Identify **agent computations** which can be **transformed into environmental dynamics** and thus **performed by GPU modules**.

Next Bottleneck: Behavioral complexity of agents

Agents move according to pheromone fields

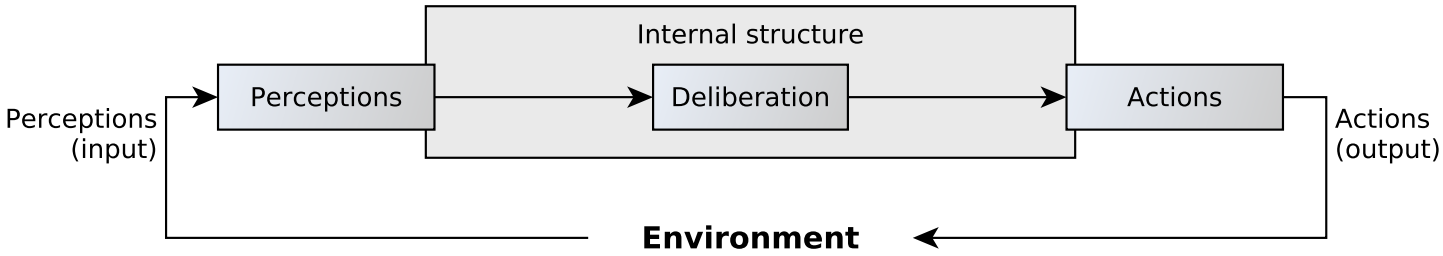
- Behaviors (CPU) consume a lot of computing resources
  - *getMaxDirection(attractionField)*
  - *getMinDirection(repulsionField)*



These computations are independent from agents' state

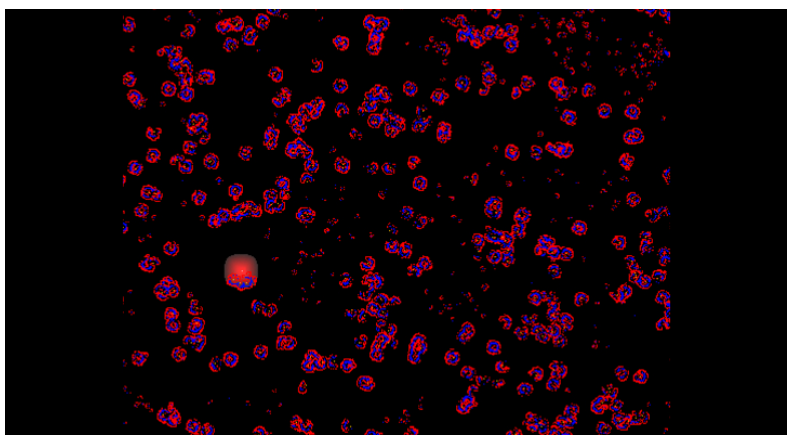
- **These are perceptions**
- $\Longrightarrow$ Compute them in the environment (GPU module)

Using the E4MAS Perspective



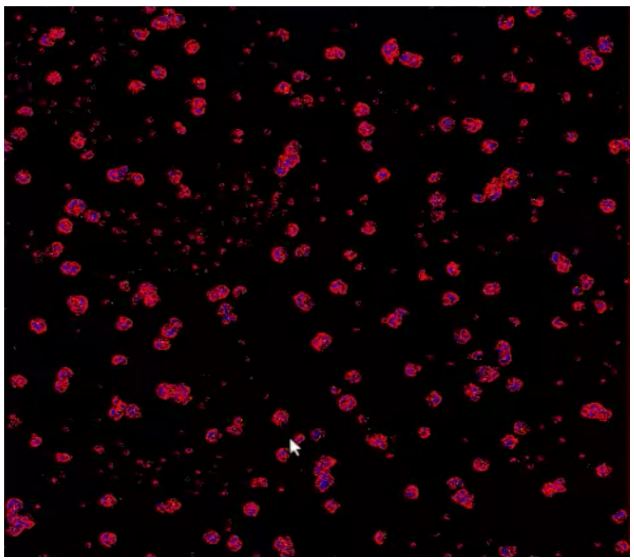
Without GPU delegation

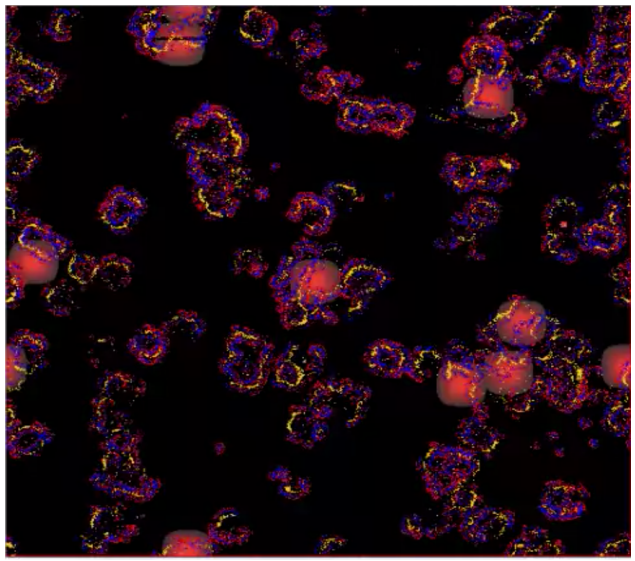
MLE with level-4 structures

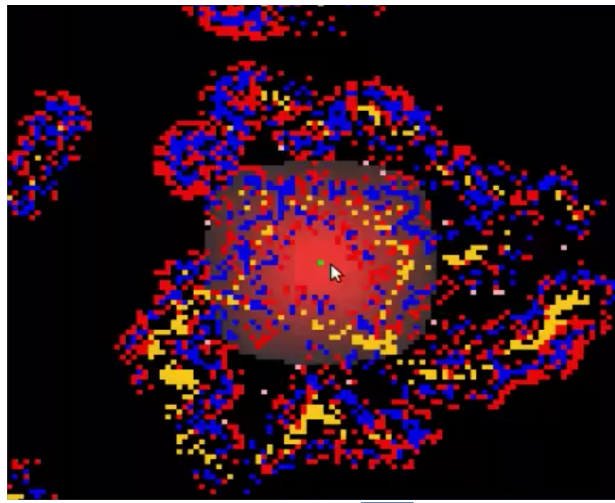


Level-1
Level-2
Level-3
Level-4

MLE avec beaucoup d'agents dans dans un environnement large

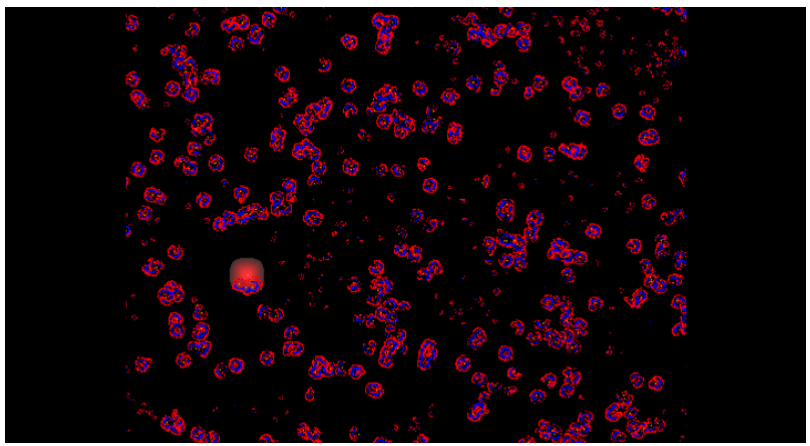






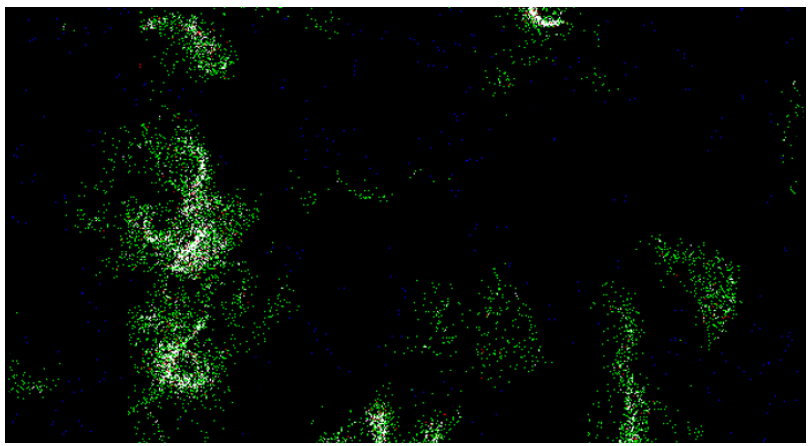
GPU delegation case studies

Multi-Level Emergence



[Michel, 2013]

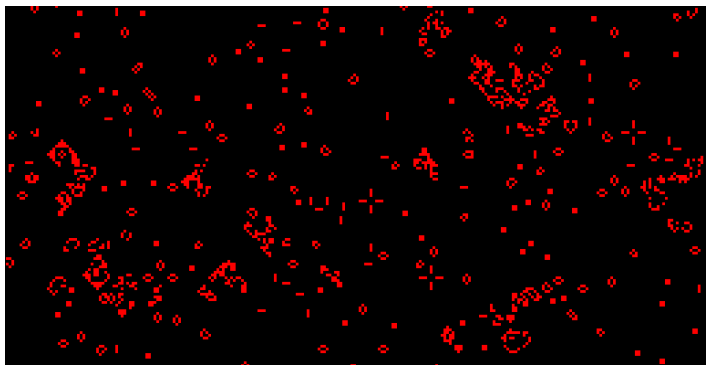
Reynolds's Boids



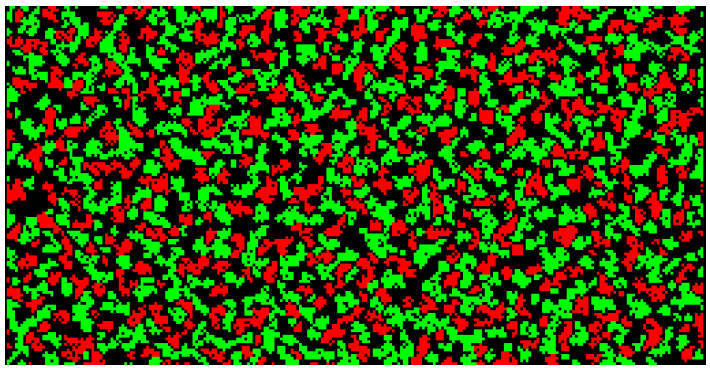
[Hermellin and Michel, 2017]

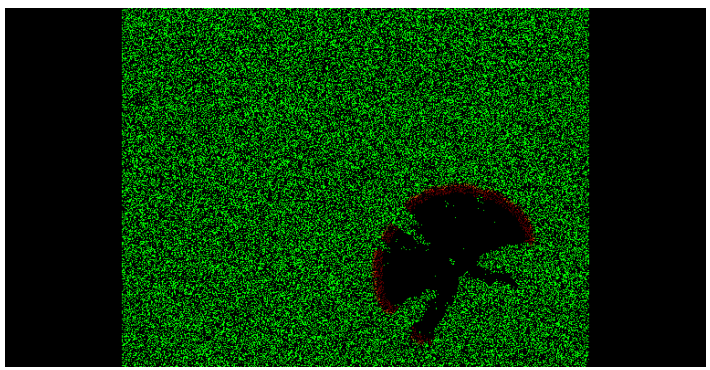
GPU delegation case studies [Hermellin and Michel, 2016]

Game of Life

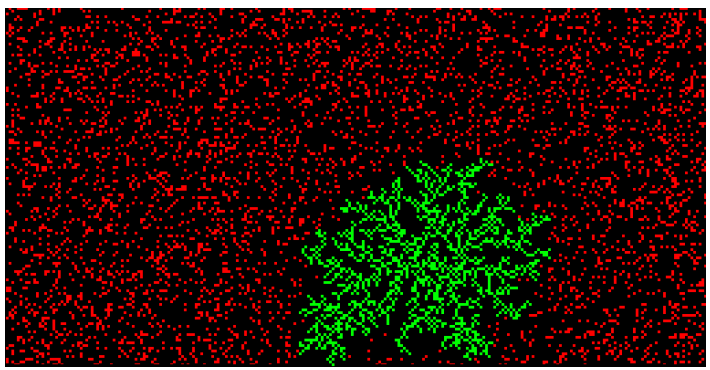


Schelling's Segregation





Fire



DLA

Long term goal: **Renewing MAS modeling**

1. Agent modeling

2. MAS modeling

2. MAS modeling

Agent behavior modeling

Issue: Behavior implementations are hard to reuse/grasp

- Engineering behaviors relies on an iterative process
- Final specifications overrun initial ones
  $\Longrightarrow$ Hard to reuse/understand
  $\Longrightarrow$ Need means to simplify the behavior

GPU Delegation: Extracting the essence of the behavior

- $\Longrightarrow$ Criterion for **a posteriori analysis**
- $\Longrightarrow$ Extending GPU Delegation to other criteria
- *e.g.* IRM4S mind vs body: Distinguishing agent's states
  $\Longrightarrow$ Emmanuel's PhD [MABS'15]

Long term goal: **Renewing MAS modeling**

1. Agent modeling

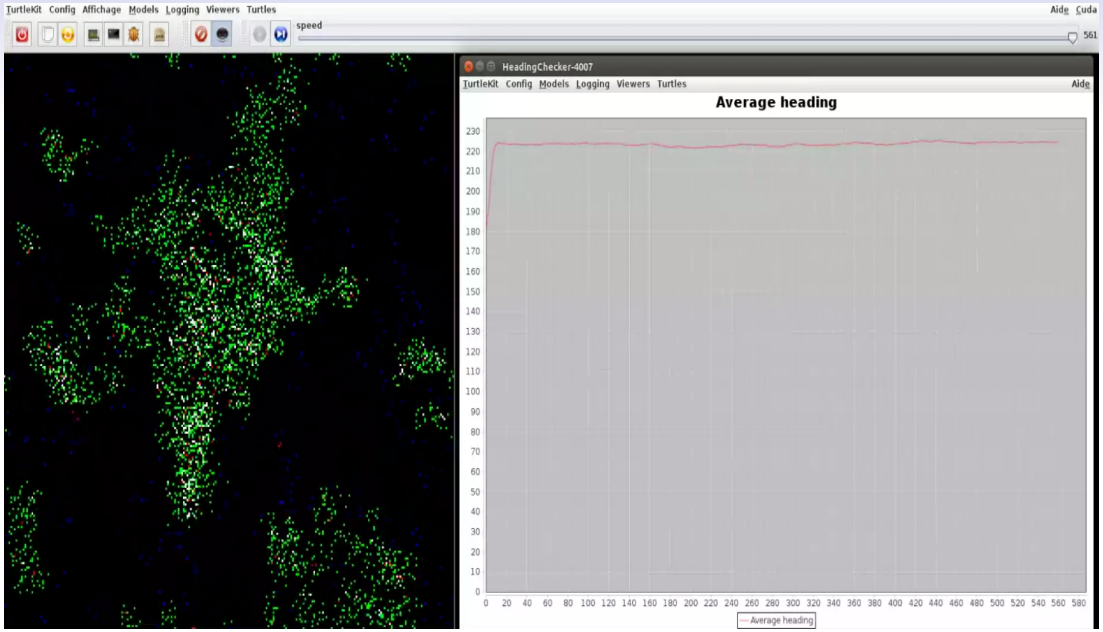
2. MAS modeling

GPGPU is more than speed

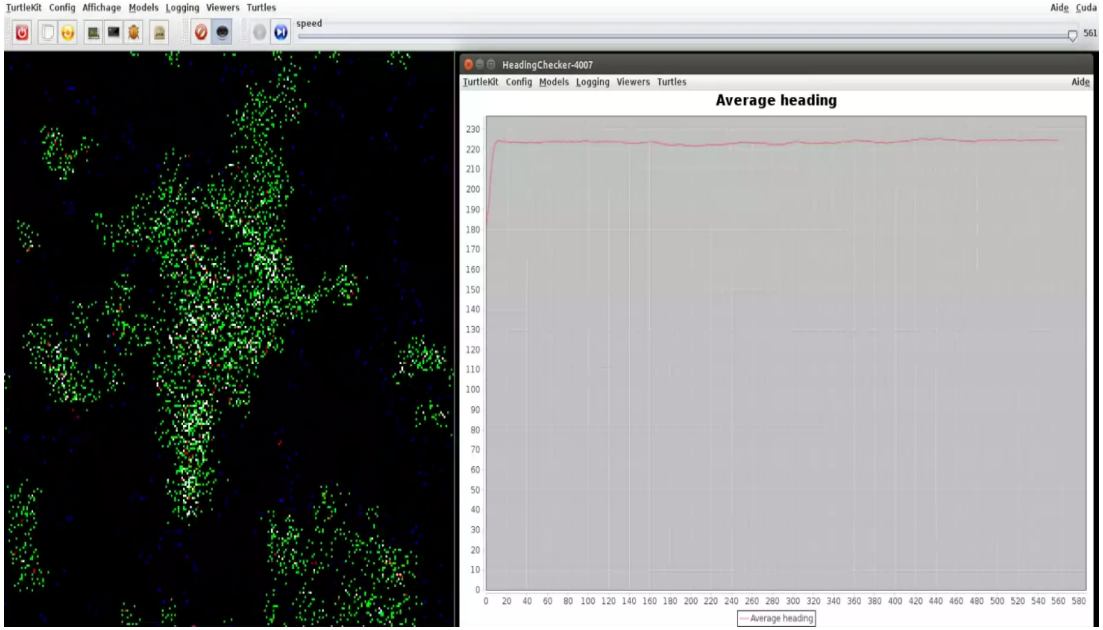
How much is GPGPU 4 MABS important?

- Everything done with a GPU can be done with a CPU
- But way more slowly
- So what?
- So modelers (humans) will not do it!

Using **GPGPU for MABS**
as a **paradigm shift**



[ECAL 2017]: *Complex flocking dynamics without global stimulus*



Conclusions

MABS is a useful **experimental tool**

- In-silico laboratory for experimentally testing hypotheses
  - Investigating complex dynamics in a *what-if* mode
  - Behaviors, interactions, environment, parameters ...
- Can produce complex dynamics that other modeling cannot
  - Multi-Level Emergence, complex flocking, ants...
- There are a lot of success stories using MABS
  - Ethology, ecology, social sciences, biology, systems engineering, chemistry...

Conclusions

But the full potential of MABS remains to be unleashed...

- Many interaction models remain to be explored
  - The modeling of perceptions, deliberation, and action is still a hot topic
  - The same holds for the environment: e.g. the potential of pheromone dynamics is underestimated
- The speed issue contributed to restrain MABS expressiveness
  - New HPC technologies will help discovering new dynamics
  - New ways of doing MABS are about to arise

References I

Chu, H.-N., Glad, A., Simonin, O., Sempe, F., Drogoul, A., and Charpillet, F. (2007).

Swarm Approaches for the Patrolling Problem, Information Propagation vs. Pheromone Evaporation.

In *19th IEEE International Conference on Tools with Artificial Intelligence - ICTAI 2007*, Patras, Greece. IEEE.

The conference proceedings will be published by the IEEE Computer Society Press in hard copy.

D'Souza, R. M., Lysenko, M., and Rahmani, K. (2007).

SugarScape on steroids: simulating over a million agents at interactive rates.

Proceedings of Agent 2007 conference.

Erra, U., Frola, B., Scarano, V., and Couzin, I. (2009).

An Efficient GPU Implementation for Large Scale Individual-Based Simulation of Collective Behavior.

In *High Performance Computational Systems Biology, 2009. HIBI '09. International Workshop on*, pages 51–58.

Gause, G. F. (1934).

The struggle for existence.

Williams & Wilkins, Baltimore.

Hermellin, E. and Michel, F. (2016).

Gpu delegation: Toward a generic approach for developping mabs using gpu programming.

In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 1249–1258, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

References II

Hermellin, E. and Michel, F. (2017).
Complex flocking dynamics without global stimulus.
In Knibbe, C., Beslon, G., Parsons, D. P., Misevic, D., Rouzaud-Cornabas, J., Bredèche, N., Hassas, S., Simonin, O., and Soula, H., editors, *Proceedings of the Fourteenth European Conference Artificial Life, ECAL 2017, Lyon, France, September 4-8, 2017*, pages 513–520 [poster presentation]. MIT Press.

Ho, N., Thoai, N., and Wong, W. (2015).
Multi-agent simulation on multiple GPUs.
Simulation Modelling Practice and Theory, 57:118 – 132.

Husselmann, A. V. and Hawick, K. A. (2011).
Simulating Species Interactions and Complex Emergence in Multiple Flocks of Boids with GPUs.
In *International Conference on Parallel and Distributed Computing and Systems*, pages 100–107. IASTED.

Laville, G., Mazouzi, K., Lang, C., Marilleau, N., Herrmann, B., and Philippe, L. (2014).
MCMAS: A Toolkit to Benefit from Many-Core Architecure in Agent-Based Simulation.
In an Mey, D., Alexander, M., Bientinesi, P., Cannataro, M., Clauss, C., Costan, A., Kecskemeti, G., Morin, C., Ricci, L., Sahuquillo, J., Schulz, M., Scarano, V., Scott, S., and Weidendorfer, J., editors, *Euro-Par 2013: Parallel Processing Workshops*, volume 8374 of *Lecture Notes in Computer Science*, pages 544–554. Springer Berlin Heidelberg.

Laville, G., Mazouzi, K., Lang, C., Marilleau, N., and Philippe, L. (2012).
Using GPU for Multi-agent Multi-scale Simulations.
In *Distributed Computing and Artificial Intelligence*, volume 151 of *Advances in Intelligent and Soft Computing*, pages 197–204. Springer Berlin Heidelberg.

References III

Lysenko, M. and D'Souza, R. M. (2008).

A Framework for Megascale Agent Based Model Simulations on Graphics Processing Units.

Journal of Artificial Societies and Social Simulation, 11(4):10.

Michel, F. (2013).

Translating Agent Perception Computations into Environmental Processes in Multi-Agent-Based Simulations: A means for Integrating Graphics Processing Unit Programming within Usual Agent-Based Simulation Platforms.

Systems Research and Behavioral Science, 30(6):703–715.

Michel, F. (2015).

Approches environnement-centrées pour la simulation de systèmes multi-agents. Pour un déplacement de la complexité des agents vers l'environnement.

PhD thesis, Université de Montpellier.

Orcutt, G. H. (1957).

A new type of socio-economic system.

Review of economics and statistics, 39(2):116–123.

Parry, H. and Bithell, M. (2012).

Large scale agent-based modelling: A review and guidelines for model scaling.

In Heppenstall, A. J., Crooks, A. T., See, L. M., and Batty, M., editors, *Agent-Based Models of Geographical Systems*, pages 271–308. Springer Netherlands.

References IV

Pavlov, R. and Müller, J. (2013).

Multi-Agent Systems Meet GPU: Deploying Agent-Based Architectures on Graphics Processors.

In Camarinha-Matos, L., Tomic, S., and Graça, P., editors, *Technological Innovation for the Internet of Things*, volume 394 of *IFIP Advances in Information and Communication Technology*, pages 115–122. Springer Berlin Heidelberg.

Perumalla, K. S. and Aaby, B. G. (2008).

Data parallel execution challenges and runtime performance of agent simulations on GPUs.

Proceedings of the 2008 Spring simulation multiconference, pages 116–123.

Richmond, P. and Romano, D. M. (2008).

Agent based GPU, a real-time 3d simulation and interactive visualisation framework for massive agent based modelling on the GPU.

In *In Proceedings International Workshop on Super Visualisation (IWSV08).*

Richmond, P. and Romano, D. M. (2011).

A High Performance Framework For Agent Based Pedestrian Dynamics On GPU Hardware.

European Simulation and Modelling.

Richmond, P., Walker, D., Coakley, S., and Romano, D. M. (2010).

High performance cellular level agent-based simulation with FLAME for the GPU.

Briefings in bioinformatics, 11(3):334–47.

References V

Shekh, B., de Doncker, E., and Prieto, D. (2015).

Hybrid multi-threaded simulation of agent-based pandemic modeling using multiple GPUs.

In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 1478–1485.