



Co-simulation (and multimodel)

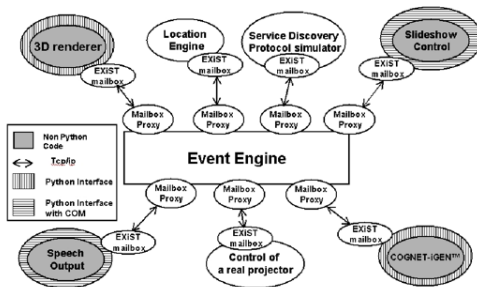
Evaluation tools, benchmarks for future networks and services
from simulation to real testbeds

Laurent Ciarletta (1), Julien Siebert (PhD thesis (3)), Tom Leclerc (PhD thesis (3)), Olivier Festor and Vincent Chevrier (2) (collab. Maia)

(1) UDL-ENSMN, (2) UDL-ESSTIN, (3) ANR Sarah + Regional funding

Starting point

Historically : Coupling simulators/emulators/real devices and application for Pervasive Computing assessment. The users and physical and digital environment are interacting. Multiple "scientific" domains (its experts, dedicated tools).



Multimodeling to manage complexity

Dynamic Networks, Pervasive, Ubiquitous Computing systems as Complex Systems

Foundation

From Pervasive Computing to Ubiquitous Networks
and Services towards Cyber Physical Systems.

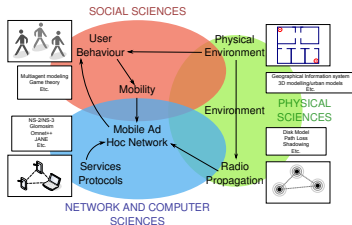
Context

Multi-domain (Social Science, Physics, Computer
Science (Networking, AI, Embedded Systems, ...)...)

- ▶ Human Centric Systems
- ▶ Context-awareness
- ▶ Interacting, interconnected autonomous elements

Objectives

- ▶ Domain : expertise and models are specific
- ▶ Separation of Concerns and Levels of Abstraction



Achievements

- ▶ Extension of a P2P simulator (PeerfactSim.KOM) with users' behavior
- ▶ Direct coupling a Network Simulator (Jane) : with a Robotic Simulator (Simbad 3D), with an Pedestrian simulator (MASDYNES)
- ▶ A conceptual framework : AA4MM (Agent and Artefact for Multiple Models coordination) with a specification of the coordination model

Co-simulation

Collaboration of simulators, emulators, software, real and modeled users

Context

- ▶ Domain specific tools : modeling tools, simulators/emulators in each domain
- ▶ Human in the loop : representing the behavior
- ▶ Hardware in the loop : representing the usage
- ▶ Environment / Context
- ▶ Vertical and horizontal scalability

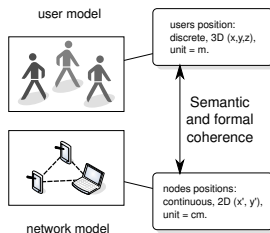
Objectives

- ▶ Collaboration of Simulators, Applications and Hardware
- ▶ Co-development and co-simulation
- ▶ Benchmarks and Evaluation of Solutions and

Technologies in individual or accross domains

An application framework AA4MM (Agent and Artefact for Multiple Models coordination)

applied to Ad Hoc networks and services evaluation (see new Mobility models)



Achievements

- ▶ Coupling multiple simulators
- ▶ Experimental Agent Platform (NetLogo)

Build a society of interacting models and simulators

Goal :

- ▶ Simulation of complex (collective) phenomena.
- ▶ Build a society of interacting models and simulators.

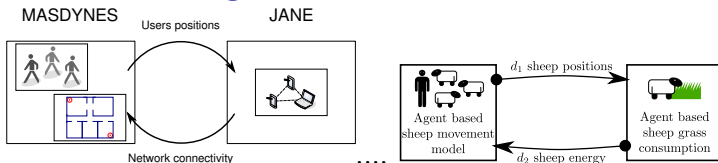
Hypothesis :

- ▶ Reuse existing models and simulators (and experts).
- ▶ One model corresponds to one simulator.

Constraints : simulators = autonomous entities (agents).

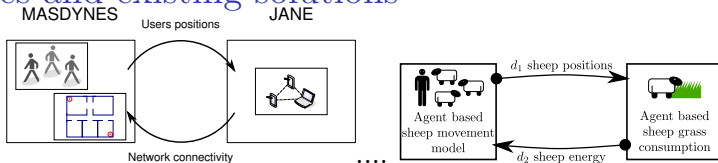
- ▶ Do not add external simulation events (\neq parallel event driven simulations [Fuj01]).
- ▶ Local control (one simulator cannot control another one).

Challenges and existing solutions



Challenges	Existing approaches
Communication / inter-operability : <ul style="list-style-type: none"> ► Coherence ► Compatibility 	Conversion operations [BRC07], functions [RAF ⁺ 04]
Add/remove reuse simulators	HLA [RAF ⁺ 04] : expensive overhead
Models execution	Global centralized scheduler [BRC07]

Challenges and existing solutions



Challenges	Our proposition
Communication / inter-operability : <ul style="list-style-type: none"> ► Coherence ► Compatibility 	Conversion operations [BRC07], functions [RAF ⁺ 04]
Add/remove reuse simulators	Simulators interfaces
Models execution	Interaction protocol : <ul style="list-style-type: none"> ► Distributed execution ► Objective coordination

Agents and Artefacts (A&A)




Artefacts (from [ORV08])

[...] passive components of the systems such as resources and media that are intentionally constructed, shared, manipulated and used by agents to support their activities, either cooperatively or competitively




- ▶ In a multiagent system :
 - ▶ Artefacts are part of the environment.
 - ▶ Artefacts reify the concept of interaction.
 - ▶ Artefacts \sim tools for agents.

Agents and artefacts : a paradigm to solve the previous challenges.

Agents and artefacts for multiple models co-evolution (AA4MM)

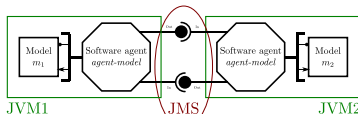
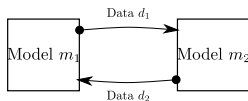
Challenges	A&A entities	Solutions
Communication / Interoperability	<i>coupling-art.</i> 	Conversion operations [BRC07, RAF ⁺ 04].
Modularity	<i>model-artefact</i> 	Simulators interfaces
Simulators execution	<i>model-agent</i> 	Algorithm for decentralized execution [SRC ⁺ 09]

A&A concepts as a paradigm to build a society of simulators

Challenges	Symbol	A&A concepts
Communication	 <i>coupling-artefact</i>	Links two simulators (point-to-point / multipoint) Used to specify and implement conversion operations and functions on data flows
Add/remove a simulator	 <i>model-artefact</i>	Allows the agents to interact on the simulator (interface).
Distributed execution	 <i>software-agent</i>	Manages simulation process : (executes the model) Manage I/O data flows

Implementation

- ▶ Open source (JAVA).
- ▶ Implementation based upon Java Messaging System (JMS)¹.
- ▶ Available toy examples (Netlogo). Used for proof of concepts.
- ▶ Real applicative case study (ANR SARAH project).



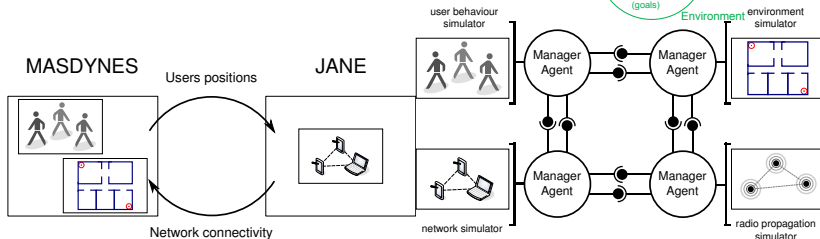
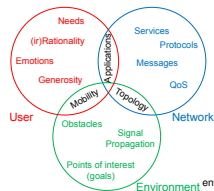
A&A entity	Functions	Role
	<i>post()</i>	Post data to exchange
	<i>read()</i>	Return exchanged data
	<i>operate()</i>	Conversion operation
	<i>init()</i>	Initialize m_i
	<i>run()</i>	Run one simulation event, step or time interval
	<i>getOutputData()</i>	Return output d_i from m_i
	<i>setInputData()</i>	Send input d_j data to m_i
	<i>getCurrentTime()</i>	Return current simulation time
	<i>stop()</i>	Manage the end of the simulation (storing data, etc.)

New mobility models

Couple existing simulators and models

- ▶ In a joint work ...
- ▶ Couple existing simulators
- ▶ From different domains

⇒ AA4MM[SCC10b]



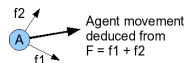
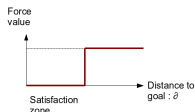
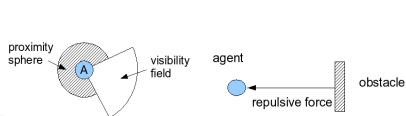
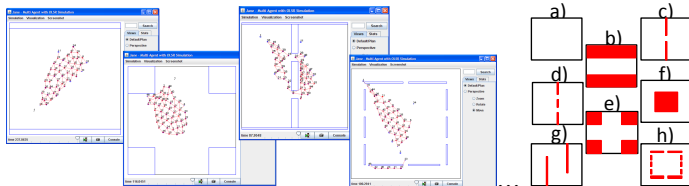
New kind of experiments[LSC⁺10]

Go further than "classic" simulations :

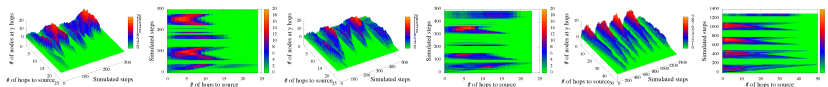
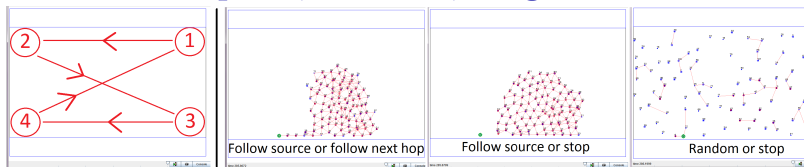
Evaluation, prototyping of network protocols and applications

What if :

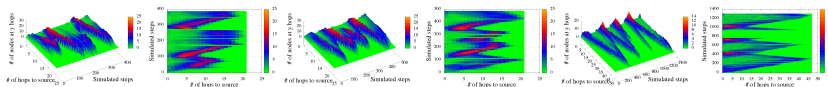
- ▶ Users react to network (QoS) and environment (obstacles)?
- ▶ Behaviors evolve (over time and space)?



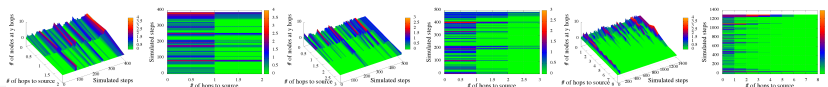
New kind of experiments[LSC⁺10], mobility in different environments : square, corridor, long corridor



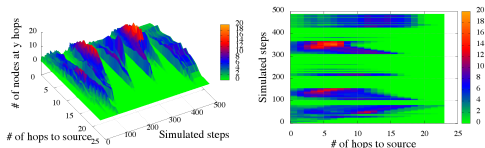
(b) Follow source or stop.



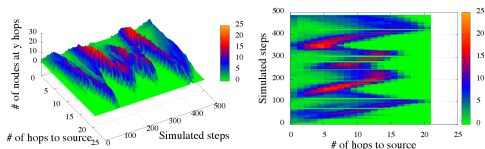
(c) Follow source or follow next hop.



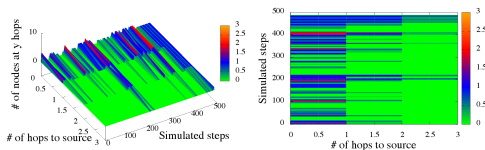
New kind of experiments[LSC⁺10], mobility in a street



(e) Follow source or stop.



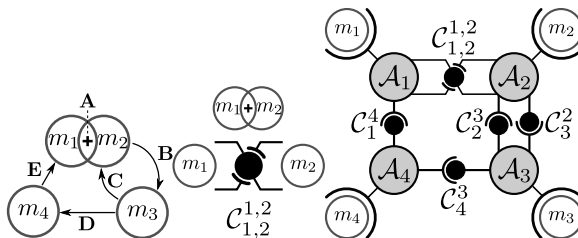
(f) Follow source or follow next hop.



Conclusion (AA4MM)

- ▶ Summary :
 - ▶ Goal : to build a society of interacting simulators (Multiagent system).
 - ▶ Proposition : AA4MM approach.
 - ▶ Agents and artefacts approach. Modular and decentralized.
 - ▶ Respects synchronization and causality [SRC⁺09].
 - ▶ Allows different abstraction levels (scales, subsystems). Separation of concerns.
 - ▶ Agent and artefact paradigm on modelling, simulation and implementation levels.
- ▶ First implementation :
 - ▶ Available toy examples (Netlogo). Used for proof of concepts.
 - ▶ Real applicative case study (ANR SARAH project → mobile ad hoc networks, Mobiquitous extended).

Conclusion (AA4MM)



- ▶ Current and future works :
 - ▶ Dependencies graph, structural coupling
 - ▶ Mobile ad hoc networks simulations : impact of mobility on network performances (Experimental work), definition of benchmarks (behavior + environment).
 - ▶ Integration of several other simulators (NS3, omnet++, Cooja, Repast, Samovar etc.)

What else ?

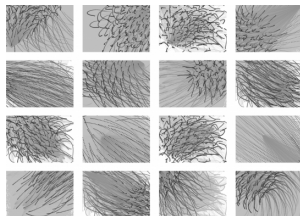
Protocol stack for Advanced Service Discovery in adhoc networks (SLSF / SLSR / Zeronconf)

Ongoing, future work

- ▶ Novel symbiotic systems for networks : (basic protocol + basic behavior + known environment) vs. (super-duper protocol with no so good behavior in unknow environment), case study : UAV + adhoc procotols (AETOURNOS (STARLING) - Loria inter-department project)
- ▶ Control of Pervasive Computing systems by Faster Than Real-Time co-simulations, from Smart rooms to digital cities ... Urban mobility project (URBAN(H)OM consortium)
- ▶ ARTEM Research : the digital workplace ?

AETOURNOS :

Airborne
Embedded auTonomOUs Robust
Network of Objects and Sensors
a Loria
platform for Cyber Physical System



for the development and the assessment **of** 2D/3Dflight
formation/flocking of autonomous elements **using** cross-layer
hybrid protocols and algorithms in noisy, uncontrolled,
challenging environments

for the development and the assessment **by** 2D/3Dflight
formation/flocking of autonomous elements **of (cross-layer
hybrid)** protocols and algorithms in noisy, uncontrolled,
challenging environments

Flock.i.N.G.² : Flocking + next generation interconnections

"The final system should demonstrate a symbiosis, i.e. the whole set of more or less simple solutions contributes to a globally more robust and performing system than their individual sum"

Networking / communications :

Between UAVs / With the UAVs from other systems (ground control, high altitude relays)

Testing protocols, behaviors, models (Standards, New ...)

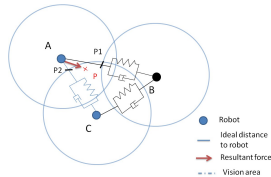
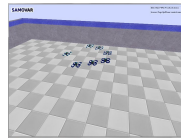
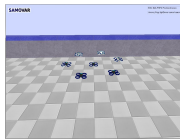
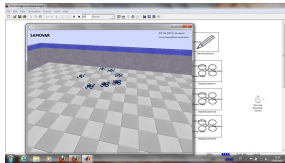
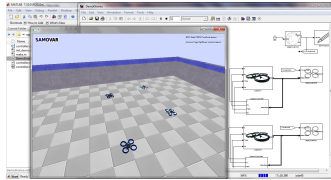
Can (and will) embed :

Wireless networks Wireless sensors and actuators network (Contiki ... + 6LoWPan + RPL) Full IPv6, adhoc stack

Flocking ++, Steering Behavior ++ :

Loop with net data (number of neighbors, signal strength, centrality) Prey and predator Radio/physical interferences, failures Roles (leader, manager, etc.)

Status



Thank you for your attention

Questions ?



Stéphane Bonneaud, Pascal Redou, and Pierre Chevaillier.
Pattern oriented agent-based multi-modeling of exploited ecosystems.

In 6th EUROSIM congress on modelling and simulation, september 9-13 2007.



Richard M. Fujimoto.

Parallel simulation : parallel and distributed simulation systems.

In Wsc '01 : proceedings of the 33rd conference on winter simulation, pages 147–157, Washington, DC, USA, 2001.
IEEE Computer Society.



Tom Leclerc, Julien Siebert, Vincent Chevrier, Laurent Ciarletta, and Olivier Festor.

Multi-modeling and co-simulation-based mobile ubiquitous protocols and services development and assessment.

In 7th International ICST Conference on Mobile and Ubiquitous Systems - Mobiquitous 2010, Sydney Australia, 12 2010.



Andrea Omicini, Alessandro Ricci, and Mirko Viroli.
Artifacts in the A&A meta-model for multi-agent systems.
Autonomous Agents and Multi-Agent Systems,
17(3) :432–456, December 2008.
Special Issue on Foundations, Advanced Topics and
Industrial Perspectives of Multi-Agent Systems.



George F. Riley, Mostafa H. Ammar, Richard M. Fujimoto,
Alfred Park, Kalyan Perumalla, and Donghua Xu.
A federated approach to distributed network simulation.
ACM Trans. Model. Comput. Simul., 14(2) :116–148, 2004.



Julien Siebert, Laurent Ciarletta, and Vincent Chevrier.
Agents & artefacts for multiple models coordination.
Objective and decentralized coordination of simulators.
In 25th Symposium On Applied Computing, 2010.



Julien Siebert, Laurent Ciarletta, and Vincent Chevrier.
Agents and Artefacts for Multiple Models coordination.
Objective and decentralized coordination of simulators.

In SAC 2010 25th Symposium on Applied Computing,
Lausanne Suisse, 2010. ACM.



Julien Siebert, Joris Rehm, Vincent Chevrier, Laurent Ciarletta, and Dominique Mery.

AA4MM coordination model : event-b specification,
RR-7081.

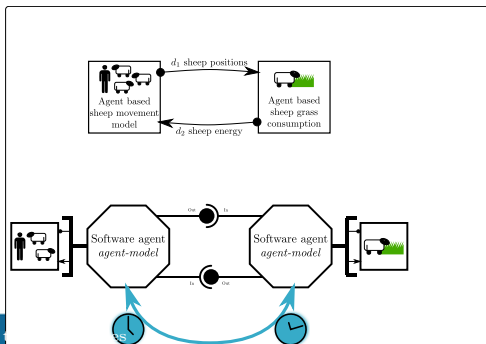
Technical report, INRIA, 2009.

Focus on *model-agent*

Manages the execution of a simulator and its interactions with the others.

- ▶ Each *model-agent* has its own simulation clock.
- ▶ Synchronization happens via the exchanged data (objective coordination).
- ▶ Causality constraints are respected.

Model-agent behaviour specification [SRC⁺09, SCC10a] enables fully decentralized execution without deadlock (formal proof).



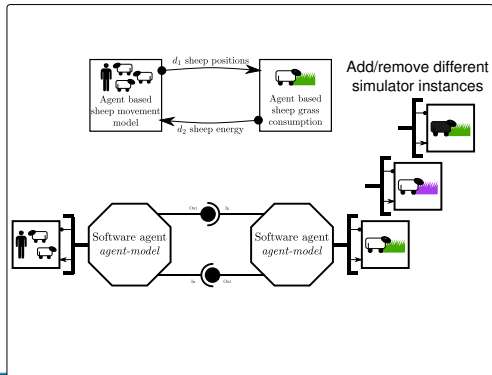
Focus on *model-artefact*

- ▶ Role : interface with a simulator.
- ▶ Needs 5 functions to implement once for a simulator.

Ease the reuse, addition of existing simulators. API between one *model-agent* and a simulator in order to manage the simulator execution.

Implementation

Use simulator API (if exists).
Build the interface from scratch (specification).



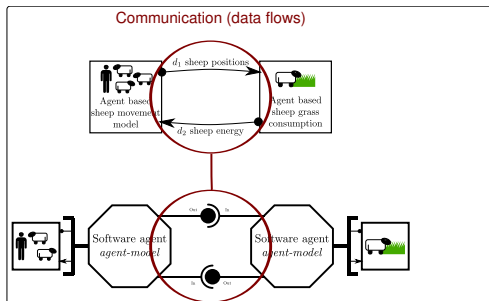
Focus on *coupling-artefact*

Implements the simulators interactions as *model-agents* interactions.

- ▶ Role : specify, implement simulators interactions.
- ▶ Operations [BRC07] (reuse it ! compose them ! artefact strength)
- ▶ Links two or more simulators together
- ▶ Environment Memory [SRC⁺09]

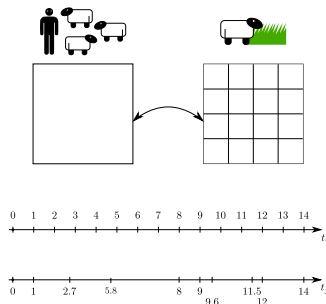
Implementation

Based upon message oriented middleware (MOM) :
memory, concurrency, message passing.



Dealing with different scales

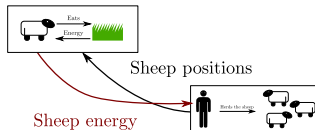
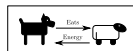
- ▶ Different scales or dimensions.
 - ▶ Specific coupling operations [BRC07] :
 - ▶ Discretization,
 - ▶ Interpolation,
 - ▶ Ratio,
 - ▶ Mean...
 - ▶ Specify and implement into the *coupling-artefact*.
- ▶ Different execution policies.
 - ▶ No change !
 - ▶ Already managed by the simulation algorithm [SCC10a].



Dealing with a new model / simulator ?

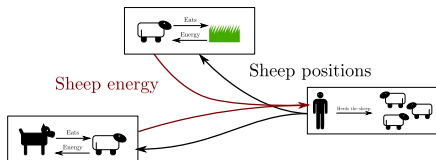
- ▶ Add a model of sheep wolf predation ?
 - ▶ The *wolf* simulator exists.
 - ▶ Add it to the existing AA4MM simulation.

Add a new model ??



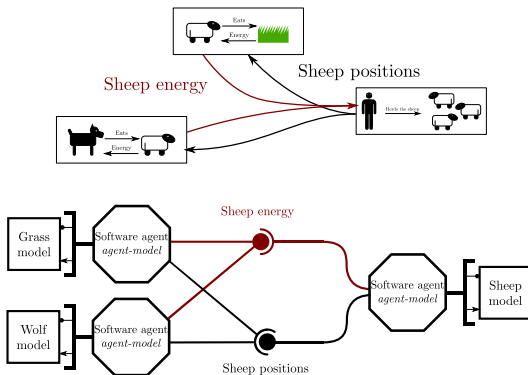
Dealing with a new model / simulator ?

- ▶ Add a model of sheep wolf predation ?
 - ▶ The *wolf* simulator exists.
 - ▶ Add it to the existing AA4MM simulation.
- ▶ Define the new links between simulator.



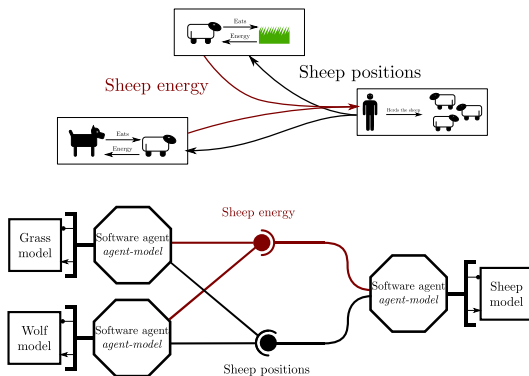
Dealing with a new model / simulator?

- ▶ Add a model of sheep wolf predation?
 - ▶ The *wolf* simulator exists.
 - ▶ Add it to the existing AA4MM simulation.
- ▶ Define the new links between simulator.
- ▶ Create corresponding AA4MM components or plug the existing ones.



Dealing with a new model / simulator ?

- ▶ Add a model of sheep wolf predation ?
 - ▶ The *wolf* simulator exists.
 - ▶ Add it to the existing AA4MM simulation.
- ▶ Define the new links between simulator.
- ▶ Create corresponding AA4MM components or plug the existing ones.
- ▶ No need to worry about execution



Software-agent behaviour

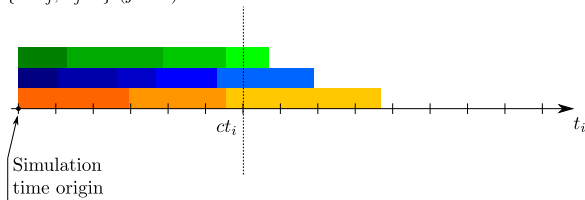
- ▶ Coordination questions :
 - ▶ How do the *software-agents* stay synchronized ?
 - ▶ How do the *software-agents* respect causality constraints ?
- ▶ *Software-agent* behaviour (at simulation time t)
 - ▶ reads input data d_{in} from the environment (perception),
 - ▶ asks the simulator to execute the model and to increase local simulation time $t \leftarrow t + \Delta t$ (decision),
 - ▶ posts output data d_{out} to the environment.
- ▶ Our proposition :
 - ▶ To use a validity interval $\Gamma =]ct, nt] =]t, t + \Delta t]^2$
 - ▶ To associate d_{out} with Γ

Execution algorithm

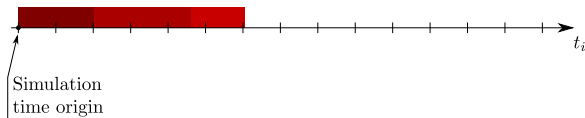
```
(1)    $ct_i \leftarrow \text{CurrentSimulationTime}(s_i)$ 
(2)   foreach input data  $(d_j, \Gamma_j)$  in  $I$ 
(3)     if  $ct_i \in \Gamma_j$ 
(4)       set  $d_j$  to the right input port  $x_i \in X_i$  of  $m_i$ 
(5)     then
(6)       wait for a new input data  $(d_j, \Gamma_j)$ 
(7)     end if
(8)   end for
(9)    $s_i$  executes  $m_i$ 
(10)   $s_i$  increases its local simulation time
(11)   $nt_i \leftarrow \text{CurrentSimulationTime}(s_i)$ 
(12)   $\Gamma_i \leftarrow ]ct_i, nt_i]$ 
(13)  foreach output data  $d_i$  in  $Y_i$ 
(14)     $O \leftarrow \text{add}(O, (d_i, \Gamma_i))$ 
(15)  end for
(16)  return  $O = \{(d_i, \Gamma_i)\}$ 
```

Execution example

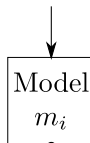
Input data sets
 $\{ \langle d_j, \Gamma_j \rangle \} \ (j \in S)$



Output data sets
 $\{ \langle d_i, \Gamma_i \rangle \}$



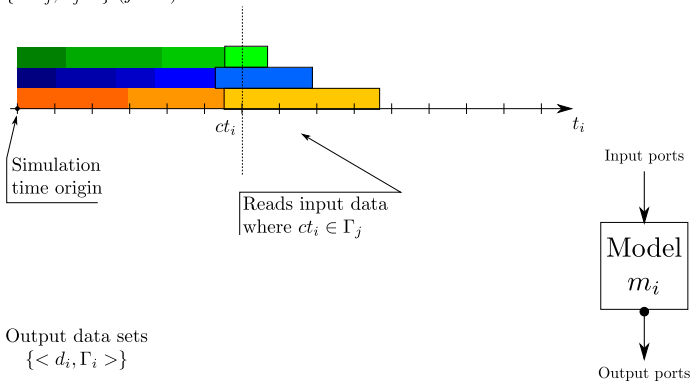
Input ports



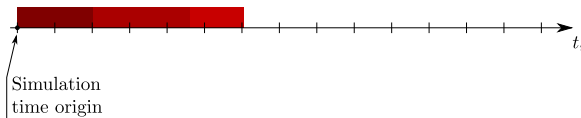
Output ports

Execution example

Input data sets
 $\{ \langle d_j, \Gamma_j \rangle \} \ (j \in S)$

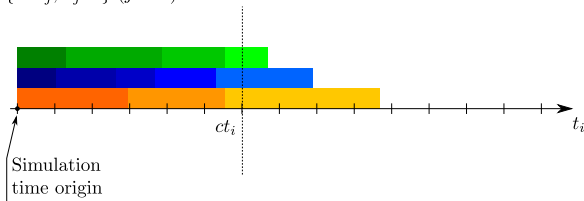


Output data sets
 $\{ \langle d_i, \Gamma_i \rangle \}$



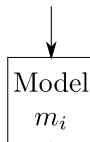
Execution example

Input data sets
 $\{ \langle d_j, \Gamma_j \rangle \} \ (j \in S)$



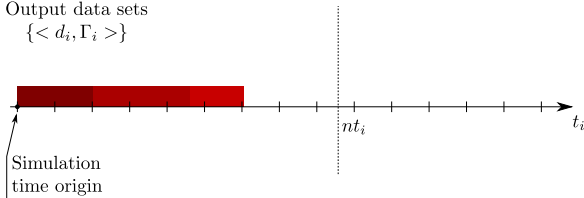
Executes model m_i

Input ports



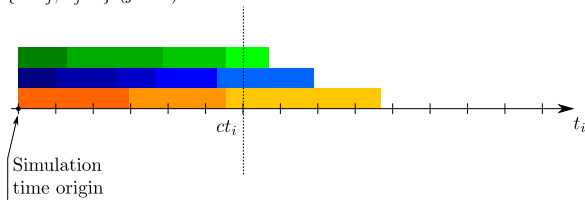
Output ports

Output data sets
 $\{ \langle d_i, \Gamma_i \rangle \}$

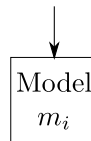


Execution example

Input data sets
 $\{ \langle d_j, \Gamma_j \rangle \} \ (j \in S)$



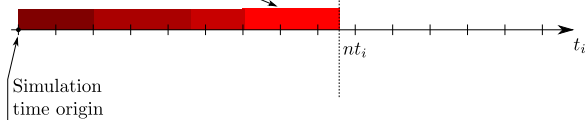
Input ports



Output ports

Output data sets
 $\{ \langle d_i, \Gamma_i \rangle \}$

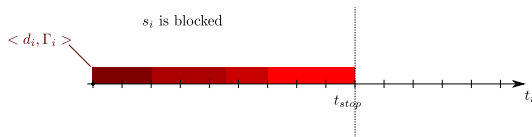
Posts output data
 $\langle d_i, \Gamma_i \rangle = [ct_i, nt_i]$



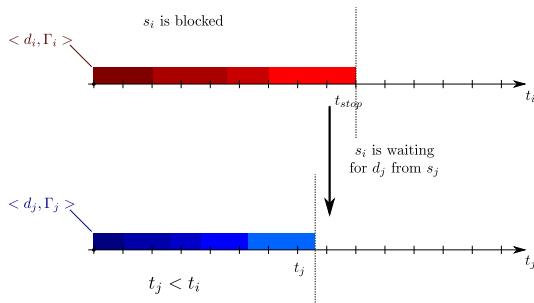
Coordination model : no deadlock (sketch of proof)

Formal specification (Event-B) [SRC⁺09]²

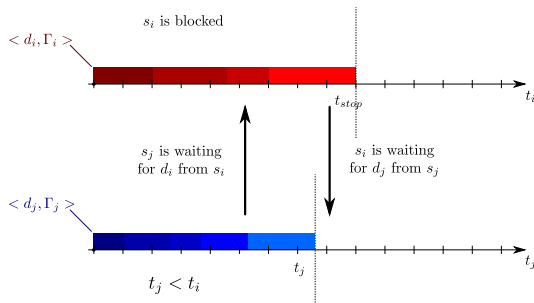
Coordination model : no deadlock (sketch of proof)



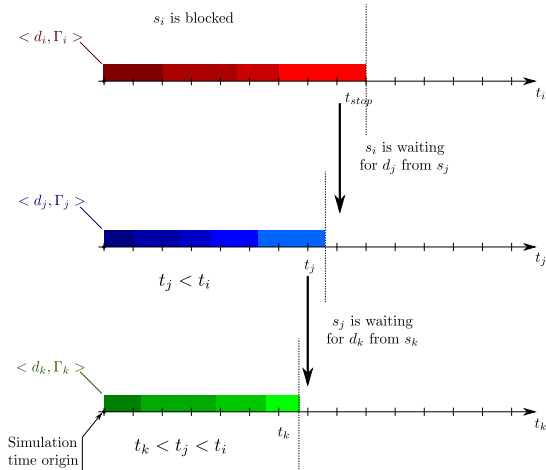
Coordination model : no deadlock (sketch of proof)



Coordination model : no deadlock (sketch of proof)

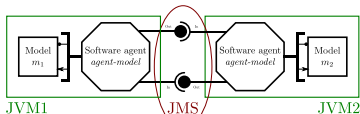
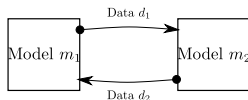


Coordination model : no deadlock (sketch of proof)



Implementation

- ▶ Open source (JAVA).
- ▶ Implementation based upon Java Messaging System (JMS)².
- ▶ Available toy examples (Netlogo). Used for proof of concepts.
- ▶ Real applicative case study (ANR SARAH project).



A&A entity	Functions	Role
	<i>post()</i>	Post data to exchange
	<i>read()</i>	Return exchanged data
	<i>operate()</i>	Conversion operation
	<i>init()</i>	Initialize m_i
	<i>run()</i>	Run one simulation every step or time interval
	<i>getOutputData()</i>	Return output d_i from m_i
	<i>setInputData()</i>	Send input d_j data to m_j
	<i>getCurrentTime()</i>	Return current simulation time
	<i>stop()</i>	Manage the end of the simulation (storing data etc.)