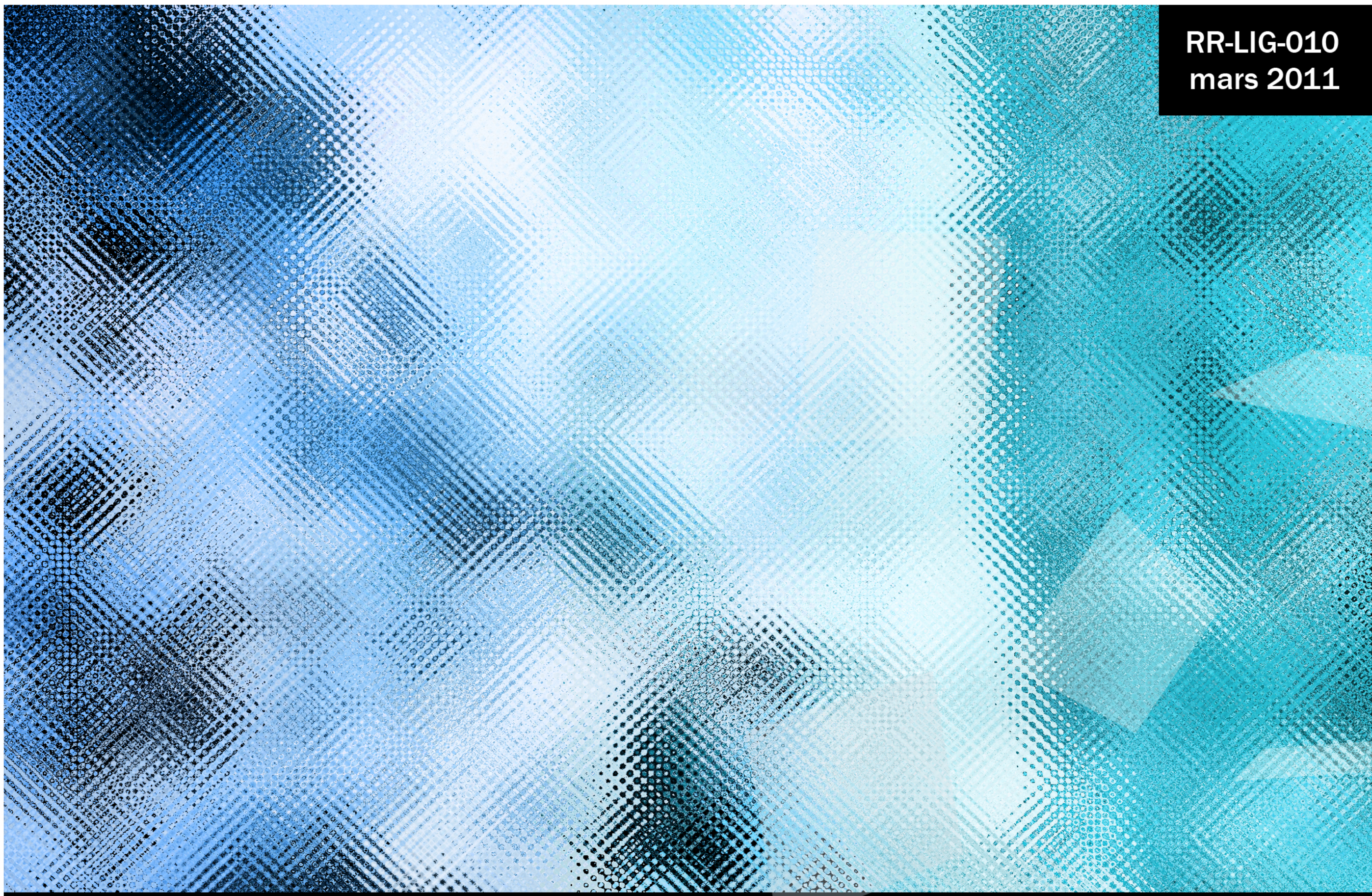
Macroscopic Observation of Multiagent Systems

ROBIN LAMARCHE-PERRIN, Drs, LIG, Grenoble University (UJF), France
YVES DEMAZEAU, Researcher, LIG, CNRS, France
JEAN-MARC VINCENT, Associate Professor, LIG, Grenoble University (UJF), France



RR-LIG-010
mars 2011

http://rr.liglab.fr
ISSN 2105-0422

Macroscopic Observation of Multiagent Systems

February 17, 2011

Robin Lamarche-Perrin
Laboratoire d'Informatique de Grenoble – Université de Grenoble
Robin.Lamarche-Perrin@imag.fr

Yves Demazeau
Laboratoire d'Informatique de Grenoble – CNRS
Yves.Demazeau@imag.fr

Jean-Marc Vincent
Laboratoire d'Informatique de Grenoble – Université Joseph Fourier
Jean-Marc.Vincent@imag.fr

Abstract

Evaluation of multiagent systems (MAS) requires a level-up in the analysis method: from the microscopic level of the agents to a macroscopic view of the MAS activity. Usually the observer collects and analyzes a huge amount of microscopic data in order to reveal global patterns. By adapting tools from the distributed systems domain to model MAS execution, and by formalizing macroscopic observations as consistent aggregations in space and time of the execution causal structure, we propose a new observation method that directly provides a macroscopic model of the MAS execution. The approach is experimentally validated by implementing probes and algorithms to observe the global behavior of a virtual ant society.

1 Introduction

In most cases, the design of multiagent systems (MAS) focuses on the implementation of agents behavior, whereas MAS evaluation is more interested in the emergent behavior of the whole system. A major issue of MAS understanding therefore resides in the level-up of the analysis method: from a local point of view during the implementation to a global point of view during the evaluation. As regards empirical evaluations, the role of the observation step and its place in relation to the the level-up is therefore significant.

Traditionally, observation is made before the analysis level-up. In that sense, it takes place at a microscopic level. For instance, techniques of postmortem monitoring, originally used for parallel applications management and that have been adapted to MAS observation (see *MAS-Pajé* [Joumaa *et al.*, 2009]), are based on the collection of countless traces locally produced by the agents. As a second step, a processing of local data is achieved in order to reveal emergent behaviors (statistical properties, global patterns, *etc.*). Other similar examples include the use of simulation platforms like in [Louie and Carley, 2008]. MAS are first microscopically modeled, simulated and then observed. Probes and tools provided by those simulation platforms for the executions analysis lean on a processing of local observations.

The first difficulty of MAS observation arises from the decentralization and the asynchronism of common multiagent executions (difficulty 1). Traditional observation methods try to overcome this difficulty by a postmortem synchronization of the microscopic traces [Chassin de Kergommeaux *et al.*, 2001]) or by a simulation core. The second difficulty resides in the data processing step allowing the level-up in the analysis (difficulty 2). Indeed, in order to reveal emergent patterns, the observer has to review all the microscopic interactions taking place between the agents. In the worst case, the processing step has to deal with $(n_a^2 \times n_t^2)$ interactions, where n_a is the number of agents and n_t the number of time steps. Because of this complexity issue, these methods are not suitable for the analysis of large-scale distributed applications developed nowadays, for instance for spatial data handling.

This paper presents a new way of achieving the level-up of MAS analysis. It is based on the concept of *macroscopic observation*, *i.e.* an observation method that directly produces a macroscopic model of the system execution (see figure 1). It consists in a reduction of the model complexity in order to facilitate the analysis and overcome the difficulties 1 and 2. For this purpose, the MAS executions will be represented by aggregates of space and time that keep enough information to allow the evaluation.

In section 2, we design a microscopic model of MAS execution derived from the distributed systems domain. Some domain-related methods for the microscopic observation of distributed executions are also introduced. Section 3 describes the formal notion of macroscopic observation as a transformation of MAS execution models. We focus on a particular kind of model transformations, based on an aggregation principle in space and time. Section 4 experimentally validates this observation method with an example interested in the observation of an ant colony.

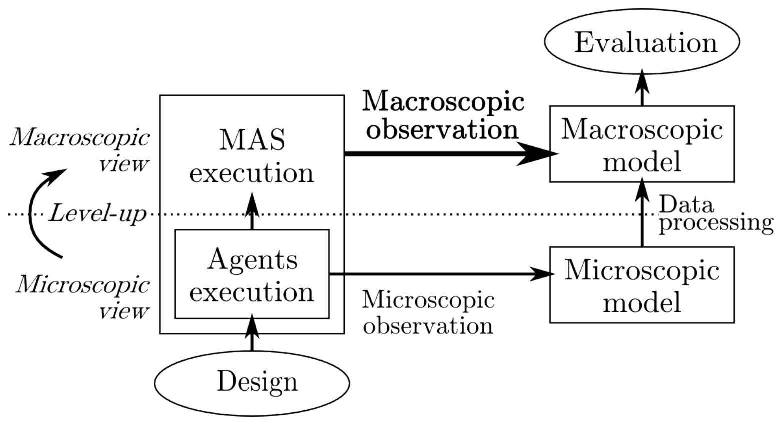


Figure 1: Macroscopic Observation for MAS Evaluation

2 From Distributed Systems to Multi-Agent Systems

Our MAS execution model is grounded on the distributed execution model of [Mattern, 1989] introduced in section 2.1. Section 2.2 presents previous work about distributed systems observation and section 2.3 explains how we exploit Mattern's model for MAS modeling.

2.1 Preliminaries

A *distributed execution model* m includes a set of process executions and a set of message exchanges [Mattern, 1989]. A process execution p can be seen as an ordered sequence of events (*i.e.* message emissions, message receptions and transitions between internal states of the process). This sequence represents the *logical time* induced at a local level by the process clock. It defines a total order on E_p, the set of the process execution events. This *local order* is marked $<_p$. The transitive closure of the local orders and the causal relations between emission events and reception events defines a partial order on E_m, the set of all distributed execution events. This *global order* is marked $<_m$. It represents the *global logical time* of the execution.

A *cut* c is a set of virtual events that splits the set of execution events into two subsets: $past_c$ and $future_c$. A cut c is *consistent* when the partial order on E_m is maintained between these two subsets: *none of the future events is a potential cause of any past event.* Formally, c is consistent if and only if for all $e_1 \in past_c$ and $e_2 \in future_c$ we have $e_1 \not>_m e_2$. C_m is the set of all possible consistent cuts of m and $<_{C_m}$ is the partial order induced on C_m by $<_m$.

2.2 Previous Work

Several works in the distributed systems literature discuss the difficulties of distributed execution observation. Some are adapted to very decentralized and asynchronous systems. For instance, the *snapshot algorithm* provides an efficient method to observe a system global state without any centralization process nor synchronization device [Chandy and Lamport, 1985]. This distributed method exploits the communication channels and the message exchanges in order to spread the snapshot from process to process and thus to make consistent cuts in the execution.

Mattern's work on *virtual time* [Mattern, 1989] is in the continuation of [Chandy and Lamport, 1985]. It proposes a distributed representation of time, evolving according to the communications between processes. It is usually used to define a global logical clock that can date and order observed events. Such a clock guarantees that the reconstructed temporal structure of the execution is consistent.

These works provide a strong basis for MAS modeling, mainly because of their efficiency in the construction of consistent execution models in a decentralized and asynchronous context. However, these methods keep the modeling at a microscopic level. The concept of snapshot indeed provides a microscopic view of the global state since it records precisely all the local states in space. In the same way, the virtual time of Mattern gives a microscopic view of event succession in time. In this paper, we exploit these works to define a method of macroscopic observation for MAS execution analysis.

2.3 MAS Execution Model

The *AEIO* design method presented in [Demazeau, 1995] is hereby used to exploit Mattern's distributed system model for MAS modeling. The main idea is that the execution of a MAS can be interpreted as the execution of a particular distributed system.

- *Agent* executions are interpreted as process executions. Hence, the local execution of an agent is described as a sequence of ordered events inducing a *local logical time*.

- *Interactions* between agents are interpreted as message exchanges. Hence, an interaction is characterized by an emission event and a reception event. Particular cases of instantaneous interactions are interpreted as synchronous message exchanges. Agents local times and interactions causality induce, as in [Mattern, 1989], a *global logical time* on the MAS execution.

- *Environments* are fully described by sets of *passive agents*, *i.e.* agents whose internal state cannot autonomously evolve. Such agents are also called *artifacts* in the *A&A model* of [Viroli and Omicini, 2002].

- *Organizations* are interpreted as a particular way of looking at the MAS activity. In this paper, organizations of MAS are not defined as *a priori* specifications that lead the global activity according to predefined design patterns, but as *a posteriori* high-level models based on what is actually observed in the MAS global execution. The next section presents an observation model that can precisely be used in order to interpret the specific activity of MAS executions as organizational patterns.

2.3.1 Formalization

We now give our formal definition of MAS execution models.

A *MAS execution model* m is a pair (A_m, I_m) where A_m is a set of agents executions – the space of the MAS execution – and I_m is a set of interactions between those agents. E_m is the set of all execution events. The application a_m associates each event in E_m with its location in the agents space A_m. $<_m$ is the partial order induced on E_m by local times and interactions causality. $\|_m$ is the relation on E_m^2 such that $e_1 \|_m e_2$ if and only if neither $e_1 <_m e_2$ nor $e_1 >_m e_2$.

In figure 2 the causal structure of execution models are graphically represented as *time diagrams*: agents are *horizontal lines*, events *graduations*, interactions *arrows* and consistent cuts *dotted lines*.

3 Macroscopic Observation

In this section, we formalize the notion of macroscopic observation on the basis of the MAS execution model previously presented. Subsection 3.1 shows how the level-up can be formalized as a pertinent aggregation in space and time of the model logical structure. Subsection 3.2 presents some practical methods to achieve macroscopic observation through the use of probes and a distributed representation of time.

3.1 Transformations of MAS Execution Models

We define an *observation method* as a technical method that associates a MAS execution model to a real MAS execution. An observation method is *microscopic* when the model that it produces contains all the agents and the interactions of the real execution. It is the case for instance for the snapshot concept which gives at an instant t the states of all the processes and the communication channels. An execution model is *microscopic* when it results from a microscopic observation.

3.1.1 Model Transformations

We define a *model transformation* T as a triple (m, m', τ) where m and m' are two MAS execution models – respectively the *source model* and the *transformed model* – and τ is an application from E_m to $E_{m'} \cup null$. Let H be a subset of E_m and H' a subset of $E_{m'}$.

- T is *coherent* on H if and only if:
  $\forall (e_1, e_2) \in H^2, e_1 <_m e_2 \to \tau(e_1) \not>_{m'} \tau(e_2)$.
- T is *complete* on H if and only if none of the events in H are associated by τ with $null$ and if all the interactions within H have an associated interaction in $I_{m'}$.
- T is *closed* on H' if and only if all the events in H' have a preimage under τ and all the interactions within H' have an associated interaction in I_m.

Let T be coherent and complete on E_m and closed on $E_{m'}$.

- T is *spatially invariant* on H if and only if the application a is τ-invariant on H, *i.e.* for all (e_1, e_2) in H^2:

$$a_m(e_1) = a_m(e_2) \to a_{m'}(\tau(e_1)) = a_{m'}(\tau(e_2))$$
$$a_m(e_1) \neq a_m(e_2) \to a_{m'}(\tau(e_1)) \neq a_{m'}(\tau(e_2))$$

- T is *temporally invariant* on H if and only if the relations $<$ and $\|$ are τ-invariant on H:

$$e_1 <_m e_2 \quad \to \quad \tau(e_1) <_{m'} \tau(e_2)$$
$$e_1 \|_m e_2 \quad \to \quad \tau(e_1) \|_{m'} \tau(e_2)$$

- T is *spatially aggregative* on H if and only if there exists a pair (e_1, e_2) in H^2 such that:

  $a_m(e_1) \neq a_m(e_2)$ and $m'(\tau(e_1)) = a_{m'}(\tau(e_2))$

- T is *temporally aggregative* on H if and only if there exists a pair (e_1, e_2) in H^2 such that:

  $e_1 <_m e_2$ and $\tau(e_1) \|_{m'} \tau(e_2)$

3.1.2 Aggregation Transformations

Let us define an *aggregate* E_α as a subset of E_m. In the continuation of this paper, we focus on the transformation functions that are:

- Coherent and complete on E_m, and closed on $E_{m'}$
- Spatially and temporally invariant on $E_m \backslash E_\alpha$
- Spatially and/or temporally aggregative on E_α

We call these particular transformations *aggregations*. They transform the causal structure of MAS execution by replacing the aggregate with a simpler substructure. To that extent, aggregations induce a loss of information. Some interactions can disappear in the transformed model, or else be simplified into a single interaction. In the same way, the logic space and time are simplified by the aggregations. This results in a reduction of the model complexity.

In order to better capture these particular transformations, one can distinguish three kinds of aggregations: spatial (replacing a set of agents with a simpler unique agent), temporal (replacing intervals between consistent cuts with a simpler unique cut) or both (composition of former and latter).

3.1.3 Maximal Spatial Aggregation

A *maximal spatial aggregation* is interpreted as a way of looking at a set of agents as if they were only one single virtual agent. All events in the set are aggregated by the transformation T on a unique agent of m'. Interactions between the agents are not revealed in the transformed model since they are interpreted as an internal activity of the virtual agent. We say that the set of agents is *aggregated* by T. Figure (2.a) presents such a maximal spatial aggregation on the set $\{a_2, a_3\}$.

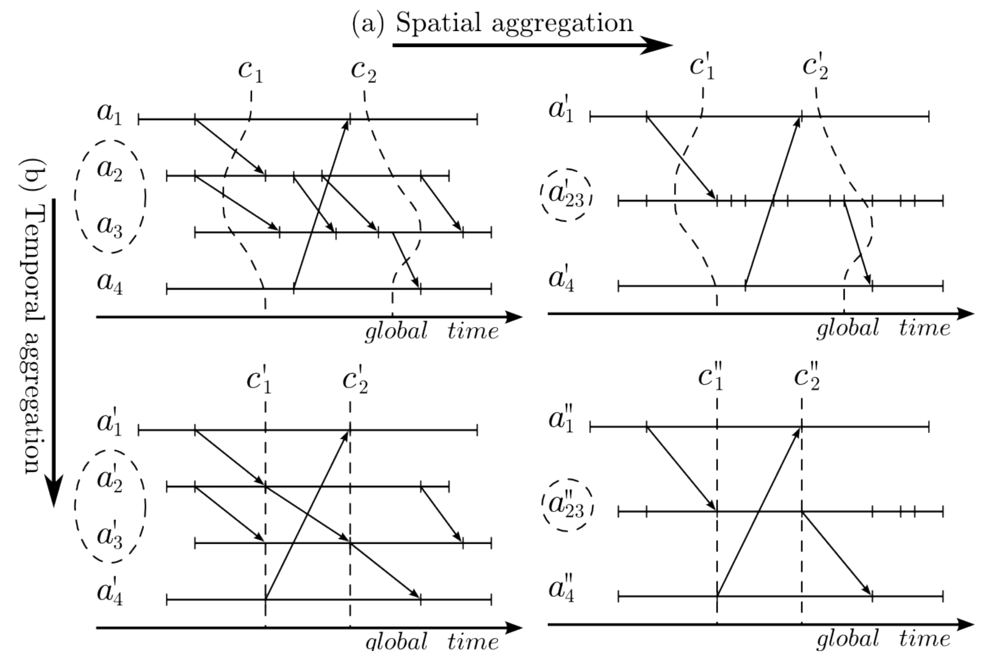


Figure 2: Models Aggregations

3.1.4 Maximal Temporal Aggregation

A *maximal temporal aggregation* should be interpreted as a way of looking at an interval of the MAS execution time as if it was an atomic step of the execution. All the events in the interval are aggregated by the transformation T on two consistent cuts of m'. The multiple interactions that occur inside this interval are not revealed in details by the macroscopic observation. We say that the interval is *aggregated* by T. Figure (2.b) presents such a maximal temporal aggregation on the consistent cuts c_1 and c_2.

3.1.5 Macroscopic Models

A *macroscopic model* is formally defined as the result of an *aggregations composition* working on a microscopic model. Indeed, spatial and temporal aggregations can be composed (see figure 2). Aggregations that are based on disjoint aggregates can also be composed. Ultimately, an aggregation can be characterized by a group of spatial and temporal aggregates partitioning the space and the time of the microscopic model. These partitions are responsible for the reduction of the model complexity.

A *macroscopic observation* is a process that associates a macroscopic model with a real MAS execution *without processing any intermediate microscopic model*.

3.2 Macroscopic Probes and Distributed Representation of Time

An event is located in space and time. From a microscopic point of view, this refers to the real agent of the MAS to which this event is related and to the global logical time at which it happens. Hence, during a microscopic observation, observed events are represented as pairs $(agent, time)$.

In order to implement a macroscopic observation method, a partition of space and time has to be defined. This partition should be dynamic and it should depend on the particular emergent behavior one wants to analyze. This is why the aggregation processes presented in this section are dynamical and partly implemented inside the agents behavior. In opposition to the traditional use of probes and cuts, the observer is not proactive during the observation process. The MAS is for the most part responsible for its own observation.

3.2.1 Macroscopic Space

A spatial aggregation can be implemented with a set of macroscopic probes partitioning the agents space into groups. We call *macroscopic probe* an observation device that interfaces between a group of agents and the observer. Such a device collects data regarding the local activity of the observed agents. Representations of events do not specify the precise agent in which they occur, only the group identifier. The events are thereafter recorded as pairs $(GROUP, time)$ and so they are spatially aggregated according to the macroscopic space of probes. Information losses affect interactions within the groups.

3.2.2 Macroscopic Time

In a temporal aggregation, the global logical time is partitioned into a set of time intervals. A cut can be distributedly represented by internal counters in the memory of agents. The representation of events do not specify the precise logical time at which they occur, but only the macroscopic cut. It is recorded as a pair $(agent, CUT)$. Events are then aggregated according to the macroscopic space of cuts.

In order to guarantee the consistency of the cuts and so the coherence of the observation, one can adapt the *snapshot algorithm* in a multiagent context: during each interaction between agents, the value of the internal cut counter is also emitted. Thus, the agents synchronize their distributed time representations according to their causal relations.

Finally, by using the two methods described in this section, an observer can implement an aggregation that is both spatial and temporal. In this case, the recorded events are represented as pairs $(GROUP, CUT)$.

4 Experimentations

Experimentations have been carried out with the *MASON* environment, libraries and tools [Luke *et al.*, 2005]. They are specially designed for massive MAS modeling, such as swarm systems modeling. We have chosen *MASON* because it constitutes an especially sound environment for large-scale simulations. However, we state that our macroscopic observation method could be easily implemented on a large variety of simulation environments or on real MAS.

The experimentations objective consists in showing that macroscopic observation responds to the difficulties encountered by traditional observation methods, regarding (1) the decentralization and the asynchronism of systems and (2) the cost of the level-up in the MAS analysis.

By ensuring that the implemented functions satisfy our model, we guarantee that the produced execution models will be causally coherent with regards to the real execution (objective 1). The correctness of the implementation has been checked by an empiric comparison of our results with the results obtained via a microscopic approach. However, such an empiric evaluation cannot be generalized as far as the microscopic observation of large-scale systems is impossible in practice.

Objective 2 resides in a reduction of the model complexity that still keeps the emergent behaviors legible. It is discussed in section 4.3.

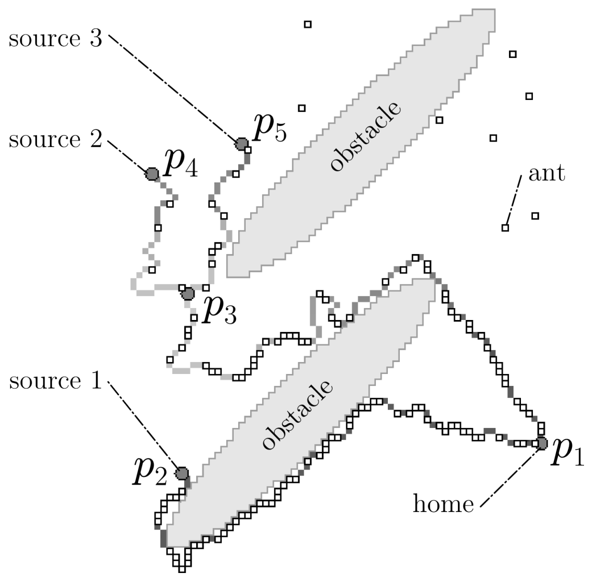


Figure 3: An Ant Colony (200 ants, 2000 steps, 265 100 interactions)

4.1 Ant Colony Model

The MAS model used in the experimentations is the classical model of ant colonies. This model is well-known in the MAS community and thereby constitutes a canonical example of emergent behaviors. This section only presents the model briefly. For a complete description of ant societies simulation, the reader may want to refer to [Drogoul *et al.*, 1995].

MASON provides its own model of ant colonies called *AntsForage*. The decomposition of this model according to the *AEIO* method [Demazeau, 1995] is realized as follows:

- *Environment* is a two dimensional grid including several kinds of areas or artifacts: *ground*, *obstacles*, *home* and *food sources* (see figure 3).

- *Agents* are ants working for the colony (white squares in figure 3) moving on the grid and interacting with the artifacts. They can be in two states: *looking for food* and *bringing back food*. Each state corresponds to a kind of pheromone that the ant drops off on the grid in order to respectively point out its way home or a source of food. When detected, these pheromones are used by other ants to choose their direction.

- *Interactions* between ants are thereby resting on the pheromones. An interaction is thus characterized by the deposit of some pheromones by one ant at one time (emission event) and the detection of these pheromones by another ant at another time (reception event).

- *Organization* of the colony that we try to observe consists in the creation of pheromone paths from home to food sources (grey paths in figure 3) and the back-and-forth of ants on those paths.

4.2 Ant Colony Observation

4.2.1 Microscopic Observation

The microscopic observation of the colony activity consists in implementing a probe for each ant. All local interactions are recorded as pairs $(ant_1, time_1) \rightarrow (ant_2, time_2)$. Leaving aside the centralization and synchronization difficulties, the observer can trace in details the ants movements and produce a microscopic model of the colony activity. An appropriate data processing can then formally aggregate parts of this model and give a global view on the execution.

In the worst case, the number of interactions is equal to $(n_a^2 \times n_t^2)$. For a colony including 200 ants and a simulation of 2000 time steps the data processing and the analysis have to deal with 310 000 interactions in average (see section 4.3).

4.2.2 Macroscopic Space

The purpose of the macroscopic observation is to reduce the space complexity (n_a) and the time complexity (n_t) of the model in order to facilitate the analysis. Thus, it can be applied to large-scale MAS.

The first step that reduces the complexity of the observation method consists in reducing the number of probes. We decide to not observe all the ants behaviors but to attach probes on the grid at locations of particular interest: the starts, the ends and the intersections of pheromone paths indicating food sources. These points are locally detected by the ants. Thus, the whole grid is observed depending on these particular locations. In figure 3, 5 probes are represented (gray circles).

Each probe defines a spatial aggregate, identified by a group number. This aggregate is made up of all the ants that have met the probe lastly on their trajectory. Each event transmitted to the observer by a probe is localized with its group number.Interactions between probes can therefore be defined as follows: every time an ant meets a probe on the grid, a pair $(GROUP_{last}, time_{last})$ is locally recorded. It indicates the last probe met by the ant and the time when it did. Then, the probe sends a pair of events to the observer: $(GROUP_{last}, time_{last}) \rightarrow (GROUP_{present}, time_{present})$. This is interpreted as an interaction between two probes ($last$ and $present$). The number of interactions between two probes at a given time is interpreted as a flow of aggregated activity between two locations on the grid.

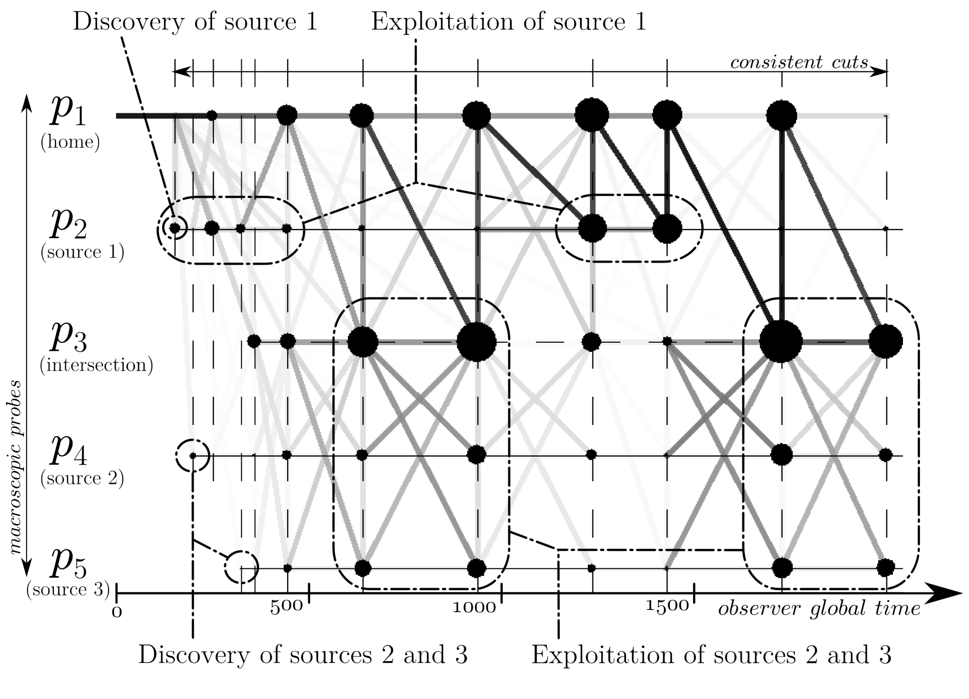


Figure 4: A Macroscopic Model (5 probes, 13 cuts, 139 interactions)

4.2.3 Macroscopic Time

The second step for implementing a macroscopic observation consists in a reduction of the time complexity. Each probe and each ant is marked with a counter that roughly represents the time localization of the agent. This counter indicates the global cut after which the current activity of the probe (or the ant) is executed.

Probes record the ants local events. When a pair $(GROUP, CUT)$ is identical to one that is already recorded, the probe counts the number of identical pairs. When its cut counter is updated (indicating a new consistent cut), the probe sends the recorded interactions $(GROUP_{last}, CUT_{last}) \rightarrow (GROUP_{present}, CUT_{present})$ and their numbers of occurrences. Hence, the interactions between probes are aggregated into flows of activity according to the cuts in which they occurred. In figure 4 the lines brightness represents the intensity of interactions between probes.

Cuts are initiated by particular ants when they interact with particular probes. In our experiments, a given ant initiates the cuts every time it brings back food to the home. The execution is synchronized on the come-and-forth of this particular ant. Probes send data to the observer only when their time counter is incremented. Thus, the MAS execution gets a global clock that synchronizes the data collection. In order to propagate the global cut in the colony activity while preserving its consistency, ants counters are synchronized every time they interact with a probe. Cuts are consistently propagated thanks to the ants activity on the pheromones paths as in the notion of distributed snapshot.

4.3 Complexity Reduction

The example in figure 3 shows the execution of a 200-ants colony on *MASON*. Figure 4 shows the associated macroscopic model we obtain for a 2000-steps execution. Because ants movements are randomized, the simulation has been repeated $R = 100$ times with the same configuration of the environment. The following table presents the average number of probes and cuts used in the case of macroscopic observation, as well as the average number of observed interactions in both cases. The right column also presents the average reduction factor associated with the respective complexity (space, time and interactions).

	Microscopic observation	**Macroscopic observation**	**Factor**
Space (n_a)	200 ants	6.7 probes	33
Time (n_t)	2000 steps	18 cuts	110
Interactions	3.1×10^5 interactions	2.4×10^2 interactions	1700

The complexity of the execution space and time is reduced thanks to the macroscopic observation. The observer needs to implement 33 times less probes in average than with a microscopic observation method. The number of synchronizations between the probes and the observer (in order to transmit data) is 110 times less. For these reasons, macroscopic observation is much easier to achieve than microscopic one.

More importantly, the microscopic observation has to deal with about 310 000 interactions to reveal the MAS emergent behaviors. With 1700 times less recorded interactions, the macroscopic observation can still render an account of the MAS dynamics. Indeed, figure 4 represents the evolution of activity flows between grid locations during time intervals. This information allows us to reveal the global behaviors that we wanted to observe: (1) the creation of pheromone paths between the home and the sources of food and (2) the come-and-forth frequency of the ants on those paths. To conclude, with much less information retrieval, and without any postmortem data processing, the macroscopic observation can still participate in the analysis of MAS executions.

Other experimentations show that the gain in interactions complexity increases with the size of the MAS execution. For a 1000-ants colony and a 10 000-steps execution (on $R = 100$ simulations) the number of observed interactions is divided by 3100 in average. These experimentations finally show that the complexity gain is not only related to the system size, but mostly to the complexity (*e.g.* number and configuration of the food sources and the obstacles) of the problem the system has to solve.

5 Conclusion and Perspectives

In this paper, MAS execution models have been formalized as causal structures between events localized in space and time. A macroscopic observation method generates a model such that substructures of space and time are aggregated and simplified with regard to the microscopic model. Consistency of generated models is maintained during the aggregation in order to produce a coherent image of the execution.

The presented method has been validated by experimentations interested in the observation of an ant colony. Probes and distributed representations of time are used to significantly reduce the spatial and temporal complexity of the model. In that way, we overcome problems related to MAS decentralization and asynchronism that make the microscopic observation difficult (overcoming of the difficulty 1). Ultimately, the global behavior of the system is properly presented as flows of activity between particular points of interest in the MAS environment. These flow are stretched over consistent time intervals. No data processing is needed to get this macroscopic model, hence the cost for analyzing emergent behaviors is radically reduced (overcoming of the difficulty 2). To go further, the case of open MAS, for which space size is variable, will be handled in the future. We believe that the work of [Fidge, 1991] on variable-size distributed systems can be exploited to use our method in a highly dynamic context.

To conclude, the observation method we designed contributes to the evaluation of large-scale multiagent systems. Where traditional observation methods fail to reveal emergent behaviors, because of the cost of data processing, this new approach will allow, in the future, to see for example what has never been seen before in the case of large-scale executions: the observation and the analysis of MAS including a million agents and operating on a long-life term. We believe that this important issue for the future of the multiagent domain can only be handled by a macroscopic approach like ours.

References

[Chandy and Lamport, 1985] K. M. Chandy and L. Lamport. Distributed Snapshots: Determining Global States of Distributed Systems. *ACM Transactions on Computer Systems*, 3:63–75, 1985.

[Chassin de Kergommeaux *et al.*, 2001] J. Chassin de Kergommeaux, É. Maillet, and J.-M. Vincent. Monitoring Parallel Programs for Performance Tuning in Cluster Environments. *Parallel Program Development for Cluster Computing*, pages 131–150, 2001.

[Demazeau, 1995] Y. Demazeau. From Interactions to Collective Behavior in Agent-Based Systems. In *1st European Conference on Cognitive Science*, 1995.

[Drogoul *et al.*, 1995] A. Drogoul, B. Corbara, and S. Lal. MANTA: New Experimental Results on the Emergence of (Artificial) Ant Societies. *Artificial Societies*, pages 190–211, 1995.

[Fidge, 1991] C. Fidge. Logical Time in Distributed Computing Systems. *Computer*, 4:28–33, 1991.

[Joumaa *et al.*, 2009] H. Joumaa, Y. Demazeau, and J.-M. Vincent. Performance Visualization of a Transport Multi-agent Application. In *PAAMS*, 2009.

[Louie and Carley, 2008] M. A. Louie and K. M. Carley. Balancing the Criticisms: Validating Multi-Agent Models of Social Systems. *Simulation Modelling Practice and Theory*, 16(2):242–256, 2008.

[Luke *et al.*, 2005] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. MASON: A Multi-agent Simulation Environment. *Simulation*, 81(7), 2005.

[Mattern, 1989] F. Mattern. Virtual Time and Global States of Distributed Systems. *Parallel and Distributed Algorithms*, pages 215–226, 1989.

[Viroli and Omicini, 2002] M. Viroli and A. Omicini. Modelling Agents as Observable Sources. *Journal of Universal Computer Science*, 8:423–452, 2002.