Multi-agent Architecture Integrating Heterogeneous Models of Dynamical Processes : the Representation of Time

Edem Fianyo¹, Jean-Pierre Treuil¹, Edith Perrier¹, and Yves Demazeau²

¹ Laboratoire d'Informatique Appliquée ORSTOM, 32 av. Henri Varagnat - 93143 Bondy CEDEX France - {fianyo,treuil,perrier}@bondy.orstom.fr

² Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle - Institut IMAG 46 av. Félix Viallet 38041 Grenoble Cedex France Yves.Demazeau@imag.fr

Abstract. Multi-agent simulations aim at representing the dynamics of complex systems as resulting from multiple interactions between autonomous entities including their own perception of local environment. From our point of view, this approach would also require to take into account individualized perception of time. In this paper, we propose a new way to implement time distribution in multi-agent systems by associating to each agent his own time and rythm to achieve his own actions. Our synthetic examples are extracted from our specific research project, dealing with a simulator of irrigated systems, where the parallelism of physical processes appears to be an important component of reality. Keywords: multi-agent simulation, time-representation

1 Introduction

The ideas we deal with in this paper come from the reflexion we follow within the framework of our research project. The latter consists in the integration of different viewpoints on the future of the same geographical region (namely the Ngalenka region in the North Senegal where new irrigated areas are being developed) with regards to the evolution of both ecological land quality (soil salinity, numerical hydrodynamical models, qualitative thresholds on salt contents, etc) and land assessment (empirical rules of prescribed land use, etc). Some of these models already exist and have been developed with different time steps. Other ones are being formalized and their most convenient time step is not known yet. We have chosen to adopt an outsider point of view of the region we work on. This in order not to promote a single mono-disciplinary and specific point of view against another one. Our simulator has first represented the main natural objects recognized simultaneously by the multidisciplinary research team working on the area, and we are going on associating progressively to these objects the specific models that rule their interactions at the scale of our study, whatever their time step is. That is why we look for the most flexible representation of time which will provide the most adaptable way to introduce, as one goes along, new components and new scenarii of future land and water management. More

> Fonds Documentaire ORSTOM Cote: BX 20213 Ex: currique



generally, this paper focuses on time definition, modelisation and management in multi-agent simulation of natural systems. In the first part, we explore different aspects of the time representation issue in the field of multi-agent simulation. Then, we show how one of these aspect (management of different rhythms) is presently addressed in our application. Finally, the third part resumes a more general discussion on rhythms multiplicity and variability.

2 The question of time in multi-agent simulation systems

The representation of time issue in computer simulation is complex. Different notions of time coexist. Moreover, there are different ways of managing the temporal progression of the simulation. We can add to that the problem of the diversity and the variability of the temporal resolution that endow the simulated processes.

2.1 Notion of real time, virtual time and computation time

In the simulation field, let us distinguish three notions of time :

- there is the real time which is the time in which the real phenomena are observed.
- there is the virtual time which is the representation of the real time by the simulator.
- there is the computation time which is the real time that the simulator has taken to built its result, the execution time.

When a spatial cover for forest growth simulation displays a snapshot representing the forest at the 10th year, it is said that this 10th year belongs to the virtual time, because it has not really happened in the reality, it is a view of what the simulator has calculated to happen according to given assumptions. The virtual time is also called simulation time. It could also be called simulated reality time because it is defined to represent the time going on in the real world. In the following of the paper, we will use the term virtual time as it is often employed in the literature.

2.2 Two ways of implementing virtual time in a multi-agent simulation system

The simulated system is in general associated with a virtual clock that indicates how far the simulation has progressed (the virtual time). This virtual clock can advance in regular time intervals or in irregular time intervals [GD95]. Any dynamical simulation can be considered as a succession of state changes of the simulated world, each of these states associated with a virtual date of the world. This virtual time progresses in a discrete manner, from one step to another, in a continuous linear space of time, the computation time [CH97]. There is one absolute rule that a simulation must follow : it is the causality rule which stipulates that the future of the virtual world cannot have influenced the past. That means when one event causes a second event, the latter must be processed after the first one in the computater time.

Simulation directed by clock. In a simulation directed by a virtual clock, the virtual time is discretised in a certain number of identical interval sizes. The interval size is called time-step. The representation of time is symbolized by one virtual clock of reference that constitutes the mover of the simulation and allows the starting of any action from any agent in the simulator. In this case, the time approach is a top-down one. At every time-step, there may be no action to perform, only one action or many actions to perform. In the latter case (many actions), each action is so associated with the same virtual date. The simulator must ensure that the order in which each action is performed will have no influence on the future of the virtual world (synchronous system). There is a lot of multi-agent simulators that cope with time in this way (for example: RIVAGE [PC97], MANTA [DCF93]), because it is the easiest one. But this representation of time is efficient only if each component is known a priori, since one has to choose the step time of the simulator as the smallest time interval among those associated with every agent. This search for the lowest common denominator in time scaling is similar to the search for the finest spatial grid in space representation. The definition of a single relevant time step is not so easy as noticed by LePage[LG97]. This point will be discussed later in the paper.

Simulation directed by events. In a discrete event simulation (DES), the progression of the virtual time depends on the event occurrences that are triggered with a precise virtual date which allows to sort them. With this approach, when there is an inactivity period, the simulation goes directly to the next significant event. It is the synchronization kernel that manages all the events that happen in the simulation. Its role is important especially when several events have been triggered at the same date, to manage the simultaneity (virtual parallelism).

Distributed discrete event simulation. In a distributed DES, the simulated world is modeled as a group of communicating entities, referred to as Logical Processes (or LP). Each LP maintains a local virtual clock that defines the virtual time for that LP (LVT) and the LPs operate as distinct discrete event simulators, exchanging event information if necessary [Rad97]. The simulator itself will be associated with a global virtual time (GVT). Distributed DES attempt to achieve computer parallelism that is to execute events in parallel computation time. The main problem is to ensure that no causality error occurs. To overcome this problem, synchronization between the LPs can be either conservative [Mis86] or optimistic [Jef85].

Conservative synchronization

Under conservative synchronization, events are processed by each LP only when

it can guarantee that no causality (out of order) violation will occur. To make sure that only safe events are processed, all LPs that contain no safe event must be blocked which can potentially lead to deadlock. To prevent deadlock, two kinds of methods have been developed : 1) deadlock avoidance methods and 2) deadlock detection and recovery methods. The latter have the advantage of being easy to implement, but they can not exploit full parallelism of system : LPs behave over pessimistic if the causalities are not frequent [GD95]

Optimistically synchronized simulation

Here the occurrence of causality errors is allowed. To satisfy the causality constraints, some mechanism to recover from a causality violation is defined. The best known is time warp. In time warp, any LP with an event to process is allowed to run without consideration of the progress of the other LPs. Since some LPs will process ahead of others at any given computation time, it is possible for a LP to receive an event from the past, violating the causality constraints of the simulation. In order to recover, the LP receiving these *straggler messages* must rollback to an earlier local virtual time and re-process events in their correct order. Time warp exploits full parallelism of systems : if causalities are frequent, the time warp can be more efficient. Nevertheless, this kind of protocol is hard to implement and to debug because they require complex manipulations and data structures [Rad97], [GD95].

Virtual clock and computer clock. In these two ways of implementing the time in a simulation system, the virtual time is always defined as independent of the computation time. One can imagine to define a virtual time progression founded upon the computation time (e.g. by means of a proportionality coefficient). One can see such a method used in some of the examples illustrating threads use in [Lea97]. It would be interesting to go further with such approaches.

2.3 Multiple variable time-steps or coexisting rhythms

In our project, we are concerned with the coexistence of many different rhythms. The dynamical processes have been modeled according to their rapidity and the frequency of the actions and the events which compose them. The process of evapotranspiration of the culture is a very slow process that we have to link with the irrigation process which is much faster. Moreover, these physical processes may change their rhythm according to external conditions. For example, water will flow faster through the soil according to the height of the water charge put on the ground. Holding on the variability and the diversity of coexisting rhythms is independent of the way virtual time is implemented (by clock or by event). It is a general modelling issue that would need the definition of a computational theory [Mar77]. Here the question is to dynamically define and control the time-step of the simulation or to manage the way the events are generated.

3 Time representation in our application

Time representation in the present version of our system can be put among the conservative protocol of synchronization. As a matter of facts, only safe actions are processed, all LPs that contain no safe event are blocked until the others LP reach the corresponding virtual time.

The simulator is composed by different types of entities. There are passive entities (e.g. irrigated area) which state evolves according to the action of natural phenomena (rain, climate changes) or human actions (land use or irrigation decision). There are active entities (like natural phenomena, human decisions) that modify the world either by their own will or according to physical determination. These natural phenomena and human actions modify the system entities state by triggering processes like percolation, evapotranspiration or irrigation.

For us, a computer process will be a representation of the dynamics which can describe either a natural physical process or a human action taking. Each process acts via the entities methods that are in the world. It starts or ends because of some events which represent special state changes of the simulated world. Entities and processes are endowed by their own independent life rhythm. The passive entities just update their virtual time. During its cycle, one system component will achieve its set of tasks. When these tasks are completed, it goes forward to the following time lag and increases its own life time, whatever the other ones are doing meanwhile. To ensure the causality coherence of the system, we work on two levels :

- we implement scheduler mechanisms to synchronize components which share the same virtual periodicity time,
- we link different scheduler mechanisms to build a global virtual time.

3.1 Own time definition

Each system entity is endowed with an own-time attribute. This attribute value indicates to the agent its actual virtual date. The entity is associated with a scheduler to whom the entity asks to update its date. The scheduler will respond to its demand after all the entities that are associated with it have done the same demand. Here our scheduler plays the role of a conservative LP. In a Petri net representation, the behaviour of a scheduler can be represented as on figure 1.

7 -

3.2 Relating different own times

In order to define a global virtual time, we use mechanism between schedulers. To simplify the problem, we first assume that relations between different rhythms can be materialized by a proportional number. The global virtual time corresponds to one central entity local time, here the irrigated area time, because almost all the processes act on it.

Theoretically, all the time schedulers are linked with the time scheduler of the irrigated areas following the synchronization mechanism represented on the





figure 2. But as some entities synchronize themselves to others when they process, we choose to link some schedulers to others whose entities interact. This in order to allow more flexibility. We have implemented this time representation on a *clay model* built in Java with the thread mechanisms after a first reflexion on the subject. This first clay model considers the rhythms fixed for good.

4 Taking account of the variability and the diversity of the rhythms when simulating heterogenous processes

In a parallel direction to the time representation that we are implementing, there is another fundamental issue concerning the introduction of variability rhythm concept. Generally, the entities, or processes are equipped with an own rhythm that never changes. We have found that under certain circumstances it is necessary to add another mechanism to change the entities rhythms. As a matter of facts, the process rhythm represents the time-step associated to a significant action of the process on the world. Thus it may change according to



Fig. 2. Synchronization between schedulers

When sheduler EntityTypeB changes state to Sb1, it can choose between two alternatives : to fire Tb2 or to fire Ta2. The link between scheduler EntityTypeA and scheduler EntityTypeB represents the fact that scheduler EntityTypeA associated period unit corresponds to N scheduler EntityTypeB associated period unit, where N is the proportional number materializing the relation between the two rhythms.

the state of the world. For example we can mention the model of water flow in soil ruled by Darcy-Richards differential equation that describes water transfert between the soil surface and the watertable [PLG97]. The numerical solving of this model by finite differences involves both the spatial discretisation of the soil in a set of superimposed layer and a time discretisation. The timestep of this numerical model has to be chosen depending on the input boundaries conditions that the model is going to process. If the water which arrives on the soil has a very high rate, the model is endowed with a rapid timestep in order the water flow not to jump a layer that is not the closest. If this would happen, the model outputs would be biased.

The rate of water that arrives on the soil surface is controlled by the irrigation process which is triggered by the GIE will (the Economical Interest Group composed by cultivators). Since the latter can change it is important to be able to automatically adapt the water flow model rhythm when needed. Conversely, it can also be important to be able to adapt the periodicity or rhythm of decisional processes in order to hold on a priori unknown events that would be triggered by physical processes, for example, to bring moreover water when evapotranspiration has been more important than expected.

4.1 Different types of processes to integrate

The processes that represent the dynamics of the system can be classified in two types.

- There are the continuous processes often described with continuous models. These models are built with mathematical formula like differential equations. They often represent physical processes. When they are implemented in a simulation tool, there are associated with a period of discretisationor rhythm which can be determined by a sensibility analysis, according to the user choice that can be more or less relevant as regards the system dynamics.
- Some processes are intrinsically dicrete so they are described with models that use their intrinsical rhythm. Human actions are often represented with discrete models. For example, the GIE's meeting that start different actions on the world (choice of a new culture, choice of the perimeters that will be irrigated).

4.2 Toward an intelligent control of the periodicity of the processes

Considering the above classification, we claim that there exists at least one type of processes or active entities whose rhythm is a variable that one could find interesting to control. We think that this control could be efficiently achieved by the use of control agents that would encapsulate the active entities. These agents would have the general charge of directing the active entities behaviour. In doing this, they would have to know how the other control agents direct other active entities. They would have to be endowed with an internal representation of these others control agents (especially an internal representation of their control capacity). They would also have to be able to communicate with them. For example, to ask them to change their own rhythm when they detect significative changes in the actions of their own process. So the agent notion takes all its meaning. The agent is nor really the physical entity nor the process that rules it. It is rather the computational agent that realizes these dynamics [TPC97]. This kind of mechanism would allow the genericity of our tool. The reflexion must be deepen further in relation with the work that has been already done in the multi-agent architecture field [GBD97],[GB92].

5 Discussion and Conclusion

Multi-agent systems put emphasis upon interchanging asynchronous messages between agents. The latter are so granted with a temporal autonomy (at the level of computation time) generalizing that of the actors (cf. [Bri89], [Yon92]) When multi-agent systems are used to simulate conversations between indivduals within a group of human beings, this asynchronous feature of message interchanging among 'pure communicating agents' is very convenient, as in [MIT92]. Simultaneity is not the crucial point, and we can easily manage it. Controling interchange is carried out by protocols of interaction, and beyond that, by mechanisms of internal reasoning within the very agents, without considering any explicit time managing.

However the agents encounter the time question as soon as they have to face a partially changing environment, which evolves independently and introduces temporal constraints. This is the case, for example, when we link a multi-agent system with a real time system [OD96]. The agents have thus to account explicitly for time. That is why questions about temporal logic are so abundantly treated in the literature ([Wer96], [Woo96]).

In the field of multi-agent simulation applied to natural physical or biological systems, the explicit representation of time becomes essential. We have to model the true parallelism of nature, the simultaneity and the different rhythms that we observe. We think that the representation of time (as well as the representation of space) deserves special attention, as far as the notion of environment is studied.

In our approach, we consider the world as a set of parallel processes, each one having a specific behaviour, a specific time rhythm, and possibily a specific spatial resolution. In the same way as the spatial resolution means, for spatialised processes, the distance scale from which spatial heterogeneity becomes significant, the time rhythm means, for dynamical processes, the scale of time lag needed by a given process to execute a significant and observable action, as far as the other interacting processes are concerned.

In this paper, we have presented some general principles as regards time implementation that we begun to conceive and test in the framework of our simulator concerning the evolution of a complex set of irrigated areas. For the moment, our implementation of time representation doesn't take account of the variability of the different rhythms. Neverthless, it enables subsequent inputs of new entities associated with yet unknown own times, without modifying the existing scheme. Our current reflexion turns on how to hold on variability and diversity of rhythm in a generic way.

References

[Bri89]

J.P. Briot. Acttalk : a testbed for classifying and designing actor languages in the smalltalk-80 environnement. In Y.Demazeau and J.P.Muller, editors, *Proceedings of ECOOP'89*. Eds, 1989.

- [CH97] Patrick Coquillard and David R. C. Hill. Modélisation et Simulation d'écosystèmes - Des modèles déterministes aux simulations à événements discrets. MASSON, 1997.
- [DCF93] Alexis Drogoul, B. Corbara, and D. Fresneau. Manta: New experimental results on the emergence of (artificial) ant societes. In C. Castelfranchi, editor, *Simulating Societies Symposium*,, 1993.
- [GB92] Les Gasser and Jean-Pierre Briot. Distributed Artificial Intelligence : Theory and Praxis, chapter Object-Based Concurrent Programming and Distributed Artificial Intelligence, pages 81–107. Kluvwer, 1992.
- [GBD97] Zahia Guessoum, Jean-Pierre Briot, and Michel Dojat. Des objets concurrents aux agents autonomes. In Joel Quinqueton, Marie-Claude Thomas, and Brigitte Trousse, editors, actes des Journées francophones d'IAD et SMA, pages 93-106. Hermes, April 1997.
- [GD95] Zahia Guessoum and Pontien Deguenon. A multi-agent approach for distributed discrete-event simulation. In DISMAS'95 proceedings, 1995.
- [Jef85] D.R. Jefferson. Virtual time. ACM Transactions on Programming Languages and Systems, 7(3):405-425, July 1985.
- [Lea97] Doug Lea. Concurrent Programming in Java Design Principles and Patterns. Addison-Wesley, 1997.
- [LG97] Christophe LePage and Vincent Ginot. Vers un simulateur générique des peuplements piscicoles. In Hermes, editor, Actes des 5èmes Journées Francophones d'Intelligence Artificielle et Systèmes Multi-Agents - JFIADSMA'97, 1997.
- [Mar77] David Marr. Artificial intelligence a perosonal view. Artificial Intelligence, 9:37-48, 1977.
- [Mis86] J. Misra. Distributed discret-event simulation. Computing surveys, 18(1):39-65, March 1986.
- [MIT92] T. Maruichi, M. Ichikawa, and M. Tokoro. Modelling autonomous agents and their group. In Y.Demazeau and J.P.Muller, editors, *De*centralized Artificial Intelligence I. UCL Press, 1992.
- [OD96] Michel Ocello and Yves Demazeau. Une approche du temps reel dans la conception d'agents. In J.P. Muller J. Quinqueton, editor, *IA distribuee et Systemes multi-agents.JFIADSMA'96*. Hermes, 1996.
- [PC97] Edith Perrier and Christophe Cambier. Une approche multiagents pour simuler les interactions entre des acteurs hétérogènes de l'infiltration et du ruissellement d'eau sur une surface de sol. In Tendances nouvelles en modélisation pour l'environnement. Elsevier, 1997.
- [PLG97] Edith Perrier, Christian Leclerc, and Patricia Garnier. Profile-scale simulation of water flow: a computer program to visualize and to estimate soil hydraulic properties effects. In proceedings of the International Workshop on Characterization and measurement of the Hydraulic Properties of Unsaturated Porous Media. Riverside, USA, 1997.
- [Rad97]
- Radharamanan Radhakrishnan. Formal specification and discrete event simulation algorithm. Technical report, dept. of ECECS - university of Cincinnati, 1997.

- [TPC97] Jean Pierre Treuil, Edith Perrier, and Christophe Cambier. Directions pour une approche multi-agents de la simulation de processus physiques spatalisés. In Intelligence Artificielle et Systèmes Multi-Agents - 5ème journées JFIADSMA'97, 1997.
- [Wer96] Eric Werner. Logical foundations of distributed artificial intelligence. In John Wiley Sons; editors G.M.P. Hoare and N.R. Jennings, editors, Fundations of distributed artificial Intelligence, 1996.
- [Woo96] Michael Wooldridge. Fundations of distributed artificial Intelligence, chapter Temporal Belief Logics for Modeling Distributed Artificial Intelligence Systems. John Wiley and Sons; editors G.M.P. Hoare and N.R. Jennings, 1996.

[Yon92]

92] A. Yonezawa. ABCL : An object-Oriented Concurrent System. MIT Press, 1992.



- Workshop Pre-Proceedings

MABS'98

Multi-agent systems and Agent-Based Simulation

in association with SimSoc

July 4 – July 6, 1998 Cité des Sciences - La Villette, Paris, France

> Technical report of the LIP6 Computer Science Laboratory University Pierre et Marie Curie / CNRS



These pre-proceedings have been prepared for the MABS workshop participants. They should not be quoted without the authors' permission.

The Proceedings of the workshop will be published by Springer Verlag in the *Lecture Notes on Artificial Intelligence* series at the end of 1998.

© The authors