Toward a multi-scale approach for spatial modelling and simulation of complex systems

Thi Minh Luan Nguyen , Christophe Lecerf , Ivan Lavallée

Abstract— **Complex systems are composed of many heterogeneous elements organized in a hierarchical way, whose mutual interactions make emergent collective behaviors to appear at the highest levels of observation. In some kind of complex systems, especially in biology as shown by the integrative physiology theory [6], space and geometry have a significant role in the simulation results. In this paper we expose a formalized method for modelling and simulation of complex system, going from structural modelling to dynamic simulation while integrating geometrical information in behavior study. Our solution relies on three kind of concepts and techniques: hierarchical graphs for modelling the system structure and organization, Zeigler's formalisms for the specification of agents [18] and a space aware Multi Agent System for agent-based simulation. It is shown how complex system simulation benefits from the combination of agent-based simulation and DEVS.**

Keywords— **complex system, hierarchical graph, agent-based simulation, DEVS, geometry, multilevel and multi-scale analysis, integration.**

I. Introduction

Although there is not a widely accepted definition of complex systems, it is commonly recognized that they are formed of many heterogeneous elements organized in a hierarchical way whose mutual interactions make emergent collective behaviors to appear at the highest levels of observation.

Moreover in biology, as shown by the integrative physiology theory [6], space and time appear both in the speed of signal propagation (humoral, electrical or chemical), and in the changes of spatial relationships between elements (embryology). This aspect is often neglected but, although oversimplifying is acceptable in the first approximation, space has a significant meaning in biology. We believe that the future belongs to models that can integrate and use geometrical data. We propose here a method and a set of formalisms for studying complex systems that goes from structure to behaviors, taking into account space and geometry.

A complex system is composed of a set of components, each of them being itself a set of sub-components, in which various interactions between different levels of organization take place. Their basic properties are presented in [4], we mention here the most important ones:

- A hierarchy in the structural organization may be described.
- Feedback circuits exist in the functional organization of the system, and in a hierarchy between these circuits as well.
- The system exhibits some emergent properties.
- The dynamics is typically non-linear.

One of the most important tasks in studying complex systems is the representation of their structure. Using the modelling techniques summarized by Jennings in [13] (decomposition, abstraction and organisation), modelling a complex system becomes more tractable and a hierarchical graph (detailed in the next section) can be used to represent the structure and the communication inside these systems better than types of models such as equation-based or cellular automata models. Indeed, equation-based models give a (mathematically) formalized, synthetic comprehension of the studied phenomenon, but they are difficult to improve because of the absence of modularity, and only offer the vision at the macroscopic level behavior because they use aggregate parameters. Cellular automata is a framework to explore the dynamics of complex systems whose components are distributed spatially. Time and space are represented in a discrete way. But neither heterogeneity of the complex system components, nor continuous variations can be easily studied with the cellular automata formalism.

Multi-agent systems are a better candidate for modelling the structure of complex systems. This approach aims to represent a complex system with a set of interacting autonomous entities, the agents. This technique has an important role to play since we wish to study the system behavior at a macroscopic level and we know that is the result of interactions at microscopic level. The hierarchy as well as the geometry are well supported there. On the contrary to equation-based models, multi-agent models focus on the constituent entities of the system rather than aggregate variables representing the average variation of entities. As a consequence, observations are related to population rather than on individual entities but, unfortunately, the modularity of multi-agent method hides a lack of formalism.

L. Nguyen (lnguyent@univ-paris8.fr) and I. Lavallée (ivan.lavallee@univ-paris8.fr), Laboratoire de Recherche en Informatique Avancée, Université Paris 8 et EPHE, 41 rue G. Lussac, 75005 Paris, France

C. Lecerf (christophe.lecerf@ema.fr), Ecole des Mines d'Ales - Site EERIE Parc Scientifique Georges Besse F 30035 Nimes Cedex 1, France

II. A FORMALIZED METHOD

We have developped our study of complex systems on an interdisciplinary approach, using both mathematical, biological and computer sciences concepts. In short, we propose a method for studying system that goes from structure to behaviors, as symbolized in fig 1. The first step is, from the real system, to describe the system structural and functional organization using a hierarchical graph. Then, the behavior of each node (class of components) will be described by Zeigler formalism [18]. Lastly, a space aware multi-agent system will be used as an exploration environment to observe the system dynamics.

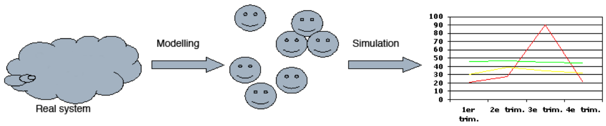

Fig. 1. From sructural modelling to behavior simulation

A. Hierarchical graph

Graph is a structure that is used in the modelling of very diverse situations and, expressed as graphs, many usual problems could be brought back to traditional problems of the graph theory: shortest path, cycle detection, connected components, etc. From a modelling point of view, the graph appears as a coupling intermediate between the physical system and its associated mathematical model. A hierarchical graph offers the ability to represent and view multiple levels in the structure and in the functional organization of the system.

A hierarchical graph (see [9] for a detailed definition) has two types of nodes: atomic and complex.

- The atomic node is the leave of the hierarchy and does not have any internal state.

- On the contrary, the complex nodes have an internal state, which is another hierarchical graph.

A complex node contains a hierarchical graph which may contain other complex nodes in a recurrent scheme. In other words, any complex node in the hierarchical graph may be a component of another complex node of a higher level. Using hierarchical graphs for modelling (i) the structure and (ii) the organization of the system enables multiscale analysis and viewpoints on both the structure and the organization. Classical graph tools such as cycle detection appear to be useful for the analysis of the system. In the end, a unique complex node represents the whole complex system, assumed that each of its components is possibly a complex node: the entire system is just the one in the highest level in the hierarchy.

B. Zeigler formalisms

This set of formalisms was chosen thanks to its capacity to integrate heterogeneous models, its coupling possibilities and its hierarchical decomposition feature.

Ziegler's set of formalisms (DEVS, DESS, DTSS, DEV&DESS, [18]), allows the system dynamics specification in a modular and hierarchical way that is based on the definition of two types of models: atomic models and coupled models. Atomic models are used to specify elementary input/output behaviors. Coupled models are defined by specifying how basic models (atomic models or coupled models at the lower level) interconnect. A coupled model can then be considered as the basic model of a higher level coupled model. In parallel, Zeigler has also developed the concept of abstract simulator. An abstract simulator represents an algorithmic description of implicit instructions for generating DEVS models behaviors. Moreover, the separation between modelling and simulation does not compel us to redefine simulators for newly defined models. In addition, recent work shows that DEVS can "encapsulate" equation-based model ([18], [8]). This formalism is thus well adapted to the specification of the multi-agents models when the constituted models can be expressed in DEVS. Note that Zeigler formalisms deal only with the behavior of the studied systems, the geometrical information cannot be taken into account.

DEVS is a formalism introduced by Zeigler in 1976. This formalism is based on an abstract mathematical object called system, which can be approximated with an automaton. Basically, a system is described by a time base, input, state, output and function for determining the next state and output for a given state. Two founding types, atomic and coupled, are described.

B.1 Atomic model

The atomic model is the basic element of DEVS (see fig 2), it has the following structure:

$A =< X, Y, S, \delta_{int}, \delta_{ext}, \lambda, t_a >$

- X : input set which is the value of input events;
- Y : set of output value;
- S : set of state;
- δ_{int} : internal transition functions. It is used to describe state transition due to internal events;
- δ_{ext} : transition functions due to external events;
- λ : output function which generate external events at the output;
- ta : time advance function;

At any time, the system is in state S. In the absence of external event, system remains on current state during the time given by the time advance function ta. On the

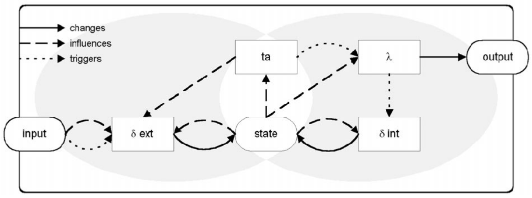

Fig. 2. Internal structure of atomic models ([15])

contrary, it receives external event X by its input port, and the external transition function δ_{ext} will then specify how system changes due to this effect. Then, an event Y which is generated by output function λ is sent to output port. Based on current state, value of external event and the one of time advance function, the next state S is computed. From the outside, this model looks like a black box.

However, a biological system does not contain only such a simple component. In fact, it is composed of many complex components, which are described with sets of subcomponents organized in many levels. Zeigler introduced the coupled model type to fit these requirements.

B.2 Coupled model

A coupled model is composed by a set of components (which are atomic or coupled models, see fig 3) and the coupling of these components. It is defined by a set of input, output ports, a set of constituted components, and coupling among these components. Coupled models allow hierarchical modelling and from a higher level a coupled model can be expressed as an atomic model [18]. The hierarchical aspects of biological system can therefore be naturally modelled with DEVS.

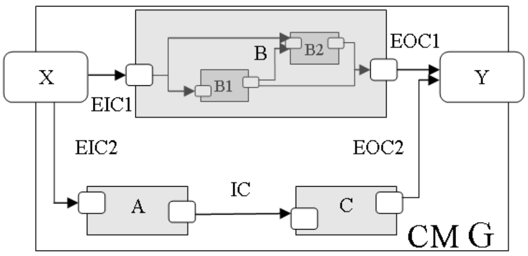

Fig. 3. Coupled models

Coupled model has the following structure:

$C =< X, Y, N, M_d, EIC, EOC, IC, Select >$

- X : set of input ports and values;
- Y : set of output ports and values;
- N : subcomponents list;
- M_d : for each $d \subset N$, M_d is a component described in form of atomic model;
- EIC : external input coupling connect external input to component input;
- EOC : external output coupling connect component output to external output;
- IC : internal coupling connect component output to component input;
- $Select$: the tie breaking function to arbitrate the occurrence of simultaneous events;

Let us consider a coupling component, which consists of a set of atomic components M_d where $d \subset N$. At time t, an atomic component d is in state S_d since e_d (time passed since the last change state of d). The time during which each component d must remain in state S_d if no external event occurred is $ta_d(S_d)$. As a result, a component d will stay at S_d for $\sigma_d = ta_d(S_d) - e_d$. An internal event δ_{int} is scheduled for the component d at $t + \sigma_d$. Suppose that ta is the time scheduled for the first internal event then ta is the smallest value of all $ta_d(S_d)$, that means $ta = Min\{(ta_d(S_d))/d \subset N\}$. The priority list $Select$ allows us to choose among various components having the same σ_d. The atomic component chooses executes its output function and sends the result to all it's influenced neighbours. Then, this component starts the internal transition function δ_{int}, and changes state. We can explore the effects of an arriving external event on an atomic model in the same way. These behavioral components are inter-connected to exchange information through their input/output ports (also called detectors and effectors). Due to the recursion sheme, such a component can be considered in turn like a basic element in a larger model.

Furthermore, Zeigler formalisms do not only model discrete event system, but also deal with continuous and hybrid system thanks to DEV&DESS. This is an extension of DEVS that includes DEVS, DESS and DTSS. Consequently, it is possible to specify some system components by Differential Equations and the others by Discrete Event or Discrete Time systems, the different parts being in interaction to constitute the whole system dynamics.

In biological systems, not only do the dynamic processes vary in time, but also does the topology (see [5]). Clearly, a perturbation of the topology of biological system will affect its evolution both in its (re-)organization and its dynamics. These two aspects are indissociably linked, so that the dynamics may be considered as a consequence of the topological, geometrical and dynamical coupling of the processes involved. Unfortunately, the geometical information cannot be represented in Zeigler's formalisms.

C. Multi-agents systems and simulation

Multi-agent systems (MAS), developed within the framework of distributed artificial intelligence, represent a promising tool to model the dynamics of space aware systems. MAS allow us to represent hierarchy and geometrical informations, but then lack a formal specification, making mathematical demonstrations impossible. Indeed, there are numerous formal specifications for MAS, for example Object-Z [12], Petri nets ([2], [16]), the colored Petri nets [3], etc. These formalisms are used to describe architectures, behaviors, etc of MAS and agents, but the hierarchical concept is not supported by any of them. However, as hierarchy is one of the most important of integrative physiology fundamental concepts [6], we had to combine Zeigler formalisms ans MAS in order to circumvent both the lack of formalism in MAS and the lack of geometrical representation in DEVS.

Agents are implemented to have internal data representation (memory or state). They possess also means for modifying their internal data representation (perception) and means for modifying their environment (behavior). Different types of agent and their concrete implementation can be found in ([14], [1]). For our study purpose, we used exclusively situated reactive agents. A situated agent lives in an environment where the space is explicitly described.

A multi-agent system is made up of a set of agents evolving in a common environment. Situated MAS are generally made of elementary memoryless agents with a defined position in time and space. Reactive situated agents perform their actions as a consequence of the perception of signals coming either from other agents or from the environment, and are sensitive to the spatial relationships that determine constraints and abilities for actions as well as privileged cooperation relationships. The environment in which agents are situated can reproduce a physical space.

C.1 MAS decomposition

In the outline of Duboz [8], let us make a formal description of a MAS with DEVS. For this purpose, we consider our model according to the four dimensions identified by Yves Demazeau in his methodology "vowels" [7]: "Agents", "Environment", "Interactions" and "Organization"

C.2 Organization

An organization is considered as a configuration that describes how its members act on each other to achieve the goal. In this context, the DEVS coupled models allow to integrate various agents in order to form composed agent (called group of agents). The whole task is then divided in a set of secondary tasks, which are distributed to the group members of the MAS. The MAS hierarchical organization is naturally described by the definition of the atomic and composed agents. The MAS recursion scheme allows us to represent the hierarchical nature of the functional organization of any biological system according to its hierarchical graph model.

C.3 Agent

C.3.a Atomic agent.

First, we use a DEVS model to describe atomic agents. Based on this elementary agent, higher-level agents called composed agents are built.

$DEVS$ = {X, Y, S, δ_{ext}, δ_{int}, λ,}

$DEVS$ model for an atomic agent

X : sensors set

Y : effectors set

S : agent possible states set

C.3.b Composed agent.

We adopt a recursive definition of composed agent based on atomic agent and composed agent of a lower level.

composed agent → composed agent | atomic agent

The DEVS coupled model for composed agent

$N = \langle X, Y, D, \{M_d\}, \{I_d\}, \{Z_d\}\rangle$

where X is the set of input events; Y is the set of output events; D is an index for the components of the coupled agent, $\{M_d\}$: set of constituted agent and $\forall d \in D, M_d$ is a basic agent (that is, an atomic or composed agent), I_d is the set of influences of agent d (that is, the agents that can be influenced by outputs of agent d), and $\forall j \in I_d$, Z_{dj} is the d to j translation function. We can see that composed agents are defined as a set of basic components (atomic or coupled) interconnected through the agent's interfaces. The translation function is in charge of converting the outputs of an agent into inputs for the others where N describe a composed agent.

C.4 Interaction

The basic interactions between agents are realized by exchanging messages with their environment via sensors and effectors.

Perception is represented in DEVS by the arrival of external events, which cause state changes of at least one component. We consider the perception of an agent as the change of its internal state due to an external event (external stimulus).

Action refers to pro-action and reaction. Regarding agent A as a coupled DEVS, the set of all external transition functions that do not receive events from coupled model input ports, plus all internal transition functions, define the autonomous behavior of this agent (pro-action). All transition functions driven by external events define the reactional behaviour of the agent.

C.5 Environment

Within the framework of integrative physiology [6], the operation of biological systems strongly depends on the geometrical distribution of the constituent entities, and the environment of MAS, i.e. the structure in which agents evolve, will take into account this information. Generally, the environment may be [10]:

- an interaction medium
- a space in which agents can move around;
- a place where resources are available.

For our study purpose, "environment" corresponds to a space in which the agents have an explicit position. It is considered as a surface divided into cells (see fig 4). Environment is viewed as a collection of n $(n >= 1)$ cells and these cells have a definite size (in a two dimensions environment, a cell is defined by its height and its width, as consequence the environment size $t = (n \times height) \times (n \times width)$. These cells form a matrix whose size is determined by grid parameters. Each cell can contain one or more elements. The cells not hosting any entity have value 0. The value of the other cells corresponds to the density of the elements that they contain.

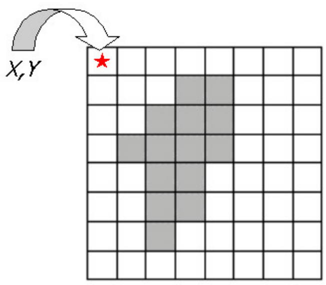

Fig. 4. Environment matrix

Environment constitutes an essential part of situated MAS but only a few works were devoted to their modelling [10]. We believe that it is important to conceive a MAS with a geometrical space representation because an explicit definition of the spatial structure of agent environment allows the definition of distance and adjacency among situated agents. Our solution provides a model that can take into account not only the system hierarchical nature but also the spatial relationship between agents, and even the changes in the system geometry.

III. Application

In order to illustrate the potential of our approach for complex system simulation, we have made an application in the biological neural network field, a case study being the hippocampus.

The hippocampus is part of the cerebrum, and it's one of the area of the brain that deals with memory. The hippocampus plays an essential role in many normal physiological functions, such as information processing, learning and memory formation, as well as in several physiopathological conditions, such as epilepsy and Alzheimer's disease.

We have developed a simple example simulating the hippocampus tissue, using a space aware MAS (a 2D matrix for a hippocampal slice) and the Hodgkin Huxley model [11] for neuron-agents implemented through the following DEV&DESS model:

$HH = \{X, Y, S, ta, \delta_{int}, C_{int}, \lambda, \delta_{ext}, f\}$

The neuron behavior is considered as a hybrid process, the internal evolution is continue, emission and reception of action are discrete. The system has two output: a discrete output "action potential" and a continue output "potential". The continue state variable potential V is used to take into account internal potential evolution. The neuron will produce an "action potential" valued V_{AP} when $V \geq V_\theta$. V is then reset to resting potential V_{rest}.

$$S = S^{discr} \cup S^{cont}$$
$$S^{discr} : \{state|state = \{active, passive\}\}$$
$$S^{cont} : \{V|V \in R\}$$
$$\delta_{ext}(V, x, t) = V + \frac{dV}{dt} + f(w_x)$$

$$\delta_{int}(state, V) \quad \begin{array}{ll} \text{if } state = active & \text{then } state = passive \\ \text{else} & state = active \\ & V = Vrest \end{array}$$

$\lambda(state)$:

if $state = active$	then make an impulsion
otherwise	nothing

$ta(state, V)$:

if $state = passive$	then $ta(state, V) = +\infty$	if $V < V_\theta$
	$ta(state, V) = 0$	if $V \geq V_\theta$
otherwise	$ta(state, V) = t_{ref}$	

$C_{int}(V, x, t)$
=true if $V_0 \geq \theta$
=false otherwise

With Huxley-Hodgkin model, f is described by:

$$f = V' = (\overline{g_{Na}} m^3 h(V_{Na} - V) + \overline{g_K} n^4 (V_K - V) + g_L(V_{rest} - V) + I_{inj}(t))/C_m$$

Results of the simulations are available in the form of videos at http://oss.ephe.sorbonne.fr/∼ntmluan/index.htm.

IV. CONCLUSION

With our method, the integration of well-known approaches (hierarchical graphs, DEVS and agent based simulation) gives a complete process for studying the dynamics of complex systems made up of interacting parts, whatever the field of the considered system. Once defined and build, running such a model relies on instantiating agents population, letting the agents interact in the space aware environment, thus leading to a simulation while monitoring what happens.

In our proposed method, the system modelling process is based on the decomposition of a given real system into various inter-connected elements using hierarchical graphs to represent the system structural organization. Each element is represented by a node, which can be described by a subgraph on a different hierarchical level, and the connection is represented by an edge, forming a multi-scale graph all together. From the behavioral point of view, a hierarchical DEVS formalism is used to describe the behavior of components that are implemented as agents in a situated MAS. At the lowest level, an atomic DEVS component corresponding to an atomic node describes the behavior of an agent in the situated MAS. At the higher level, a coupled DEVS describes a system as a network of coupled components whose connections denote how components influence each other, according to the graph of interactions that represents the organization. Moreover, an explicit definition of the spatial structure of agents environment allows the definition of distance and adjacency among situated agents. Thus, our proposed solution provides a model that can take into account not only the system hierarchical nature but also the spatial and geometrical relationship between agents that we believe has a significant meaning.

With Ziegler's formalisms, agent-based models can be combined with equation-based models because, within an individual agent, behavioral decisions may be done by evaluating theses equations [17].

We have presented in this paper a general modelling and simulation method based on (i) hierarchical graphs enabling multiscale analysis, (ii) DEV&DESS bringing a hierarchical formalism for agent specification, and (iii) a situated MAS reflecting the system geometry. The attractiveness of this method lies on its ability to be used in various domains, and thus to reduce the model building cost.

REFERENCES

[1] F. Brazier, B.D. Keplicz, N.R. Jennings and J. Treur. (1995). *Formal Specification of Multi-Agent Systems: a Real-World Case*. First International Conference on Multi-Agent Systems (ICMAS'95). San Francisco, 25-32.

[2] I. Bakam, F. Kordon, C. Le Page and F. Bousquet (2001). *Formalization of a spatialized multi-agent system using coloured Petri nets for the study of a hunting management system*. J.L. Rash et al. (Eds): FAABS 2000, Lecture Notes in Artificial Intelligence, 1871: 123-132.

[3] K. Jensen (1997). *Coloured Petri nets: basic concepts, analysis methods and practical use*. Berlin: Springer.

[4] Center for the study of complex systems, *The Study of Complex Systems* http://www.pscs.umich.edu/CSCS/complexity.html.

[5] G. A. Chauvet (1993). *Hierarchical functional organization of formal biological systems: a dynamical approach. I. An increase of complexity by self-association increases the domain of stability of a biological system* Phil Trans Roy Soc London B 339: 425-444.

[6] G. A. Chauvet (1996). *Theoretical systems in Biology: Hierarchical and Functional Integration* volume I, II, III. Oxford : Pergamon.

[7] Y. Demazeau (1995). *From interactions to collective behaviors in agent-based systems*. In Proceeding of First European Conference on Cognitive Science.

[8] R. Duboz (2004). *Intégration de modèles hétérogènes pour la modélisation et la simulation de Systèmes complexes: Application la modélisation multi-échelles en Ecologie marine*. Thèse. Ecole doctorale de l'Université du Littoral - Côte d'Opale.

[9] G. Engels and A. Schürr (1995). *Encapsulated Hierarchical Graphs, Graph Types, and Meta Types*. Departement of computer science, Leiden university.

[10] J. Ferber (1995). *Les systèmes multi-agents: vers une intelligence collective*. InterEditions, Paris.

[11] A. L. Hodgkin and A. F. Huxley (1952). *A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve*. Journal of Physiology. 500-544.

[12] V. Hilaire, A. Koukam, P. Gruer, J-P Muller (2000). *Formal Specification and Prototyping of Multi-agent Systems*. Lecture Notes in Computer Science (vol 1972: 114-127).

[13] N. R. Jennings (2001). *An Agent-based Approach for Building Complex Software Systems*. ACM 44(4): 35-41.

[14] P. Paruchuri, A. R. Pullalarevu and K. Karlapalem (2000). *Multi agent simulation of unorganized traffic*. Proceedings of the first international joint conference on Autonomous agents and multiagent systems. Bologna, Italy. 176 - 183.

[15] A. M. Uhrmacher and B. Schattenberg (1998). *Agent in discrete event simulation*. European Simulation Symposium "Simulation in Industry - Simulation Technology: Science and Art" (ESS'98), Nottingham, SCS Publications, 129-136.

[16] F. Vernadat, A. Lanusse and P. Azéma (1994). *Modélisation par réseaux de Petri d'un langage acteur : Application à la vérification de systèmes multi-agents*. Actes des 2èmes Journées Francophones IAD-SMA.

[17] H. V. D. Parunak and R. S., Rick L. Riolo (1998). *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide*. Proceedings of Multi-agent systems and Agent-based Simulation, Springer.

[18] B.P. Zeigler, H. Praehofer and T.G. Kim (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press.