Intl. Conf. RIVF'04
February 2-5, Hanoi, Vietnam

An Agent-Based Simulation Method for Studying Nervous System

Thi-Minh-Luan Nguyen, Christophe Lecerf

Abstract— **This paper describes an agent-based method for modeling and simulation of the cerebellar cortex of nervous system in particular and for complex systems in general. By modular and hierarchical model design, this method allows us to enhance the system model under study without having to rebuild the whole model. We also discuss how well our agent-based method is sound for the study of these systems and how to represent agents' behavioural dynamics of simulation models by DEVS formalism (Discrete Event System Specification) from Zeigler. Conceptualization of such systems is shown in terms of how agents and simulation models may interact with one another.**

Keywords— **Cerebellar cortex, complex system, DEVS, agent-based simulation.**

I. Introduction

For some years now, researchers have been developing models, both in hardware and in software, that mimic a brains' cerebral activity in an effort to produce an ultimate form of artificial intelligence.

The theory of automata with a finite number of states has greatly contributed to the field of neuroscience, particularly in the study of artificial neural networks (ANN). These networks are relatively simple since their organization does not extend beyond two levels: the neuronal level and the level of network of neurons. The neuron is defined as a mathematical entity and neural network consists of an interconnected set of these entities. The neuron functions based on synaptic modification according to specific law of learning [7].

In contrast to artificial neural networks, modeling a real one raises more difficult problems because of the structural and functional complexities involved. The learning rules are not externally imposed as in the case of ANN, they are constructed on the basis of internal molecular mechanisms [7].

In this paper we will study the use of an agent-based method for studying cerebellar cortex's behaviour. It relies on three important notions: S-propagator ([4], [5]), DEVS [22] and multi-agents system. More precisely, we will try to solve the questions related to the representation of system behavioural dynamics by DEVS on one hand and its dynamics simulation by agent-based techniques on the other hand.

For automatic simulators generation purpose, the Zeigler's DEVS formalism seems suitable for representing component dynamics aspect. In addition, modular and hierarchical construction gives us the possibility to easily extend the model under study.

The paper is organized as follows. Section 1 is dedicated to a short presentation of cerebellar cortex and the need of simulation in order to observe its evolution facing to environmental changes. In section 2, cerebellar cortex structure and the related study are presented. We summarize here some important concepts in integrative physiology and the use of our modeling method for nervous systems. DEVS and agent-based method are described in section 3. Finally, conclusions are drawn and future works are outlined in section 4.

We will begin this paper by a brief resume of cerebellar cortex system and its properties.

II. Cerebellar cortex

The cerebellar cortex plays an essential role in movement control and in the coordination of movement which allows reaching a target [8]. The cerebellar cortex is organized into three layers of neurons: the molecular layer, Purkinje cell layer and granule cell layer ([12], [16]). They join to one another through the neurons' synapses. Neurons can respond to stimuli and conduct impulses because membrane potential is established across the cell membrane. We try to model the neurones' behaviours and their relationship in order to study the whole system behaviours.

A. Cerebellar cortex study

A.1 From classical simulation methods ...

Biological processes are modeled mathematically by a set of equations. Since the biological system involves multiples elements that can be modeled in various levels, the global equation depends not only on others global variables but also on local variables. Thus, these equations are sometimes too complicated for an analytical solution.

In general, models can be basically viewed at two levels - one at a micro level and the other at a macro level. Traditional modeling and simulation methods offer just the vision of macro level behaviours; they do not provide insight views of micro level. Macro level involves modeling the general aspects of system like the average of "average glucose concentration in blood", "the amount of CO_2 produced in a respiration cycle ", etc. Modeling of system from this view results in losing some of the detailed aspects of the system. As a result, the diversity of the biological system can not be studied with equation-based method.

In addition, once the system changes, the whole model has to be rebuild completely.

C. Lecerf, (Christophe.Lecerf@ema.fr), Centre LGI2P Ecole des Mines d'Alès - Site EERIE Parc Scientifique Georges Besse F 30035 Nimes Cedex 1, France

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

In integrative physiology ([4], [5]), while the model is designed with reusability purpose, without distributed computation the size of systems under study is still limited by the computer capacity. Moreover, as in any equation-based method, the micro level relationship can not be explored.

These limitations restrict the efficiency of such models and encouraged us to a new modeling approach based on the interaction analysis of individual entities.

A.2 ... to DEVS and agent-based simulation

A micro level simulation implies modeling each entity involved in the system, i.e., giving each component a set of its own characteristics. The overall behaviour can be viewed as a collective behaviour of individual entities. Agent based modeling is a way of studying the interaction of large numbers of individuals, and the macro scale consequences of their interactions.

Furthermore, one of the principal properties of a biological system is its time scale: certain physiological processes are slow (several hours), on the contrary others are very fast (a few milliseconds). That means we will observe the same variation on a state variable of a slow process at the end of several hours and of the fast one at the end of a few seconds. This property allows us to partially separate slow and fast processes. For example, given two processors X1 and X2 in which X1 varies at each second while X2 has a significant variation only at the end of 100s. Two solutions for simulating these two processes in interaction are given: calculate X1 and X2 at each second, or calculate X1 at each second and calculate X2 all 100s. The second solution saves a lot of calculation cost with a light loss of precision, and gives us a possibility to simulate a large scale system in a reasonable time [10].

If the calculation of a state variable at a given moment is regarded as an event then it is possible to apply discrete events simulation methods. It allows us to manage naturally differences timescale phenomena [9].

In addition, agent-based simulation method does not replace traditional method in biological field. It can be combined with equation-based methods because, within an individual agent, behavioural decisions may be done by equations evaluation ([11]). The system level behaviour is then determined by running the equations describing the interactions among agents.

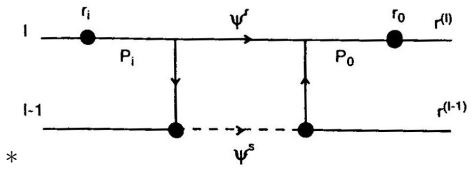
As mentioned by Chauvet in [7], one of the main problem encountered in the neurosciences is that of extending current theory of automata, used in the study of ANN, to real neural networks. The difficulty arises because automata theory fails to take into account multiple levels of biological organization involved in nervous activity [[7]]. Hopefully, the S-propagator framework from Chauvet G. ([3], [8], [7], [9], ...) is born with ambition to take into account the hierarchy of these systems.

Now we will summarize some important points in integrative physiology.

B. Neural field equation

In the integrative physiology conceptual frame, an elementary functional interaction is formally defined by a triplet (source, product, sink) and an equation for a field variable (the product) driven by a time-space field operator that describes the action through time and space of the source on the sink.

Each functional interaction has its own field variable with its own dynamics, formalized by an equation summing three terms and referring to source, sink, time and space in the S-propagators formalism.



$$P_{i0}[\Psi^s] = P_0 \Psi^s P_i$$

Fig. 1. Graph and Equation describing S-propagator (from Chauvet, 1999)

The first term of the equation describes the local diffusion of the product in the physical space around the source. The second one is strictly speaking S-propagator, and represents the non-local interaction due to structural discontinuity. And the third one represents the source, i.e. the internal local mechanisms that lead to the generation of the product emitted by the source.

For instance, S-propagator of the nervous tissue activity can be found at ([8], [7], [9]).

In biological modeling point of view, DEVS (Discrete Event System Specification) of Zeigler [22], is to our knowledge, the best suited attempt to simulate complex hierarchical systems. The next section is dedicated to a brief presentation of Zeigler's modeling and simulation theory.

III. DEVS and Agent-based simulation

A. DEVS

DEVS is a formalism introduced by Zeigler in 1976. This formalism is based on a 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 types, atomic and coupled, were described.

A.1 Atomic model

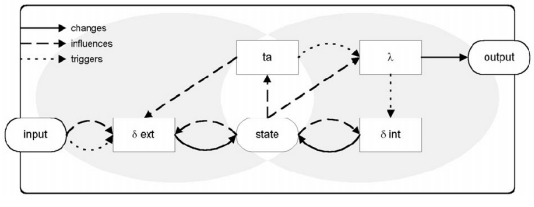


Fig. 2. Internal structure of atomic models (Uhrmacher 1998)

Atomic model is the basic element of DEVS, 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;

Similar to finite state automaton, atomic DEVS models' dynamic behavior is defined by state sets and state transition and output functions. DEVS distinguishes two type of events: internal event are time scheduled and handled by the internal transition function, external event occur upon the arrival of inputs at the input ports and are handled by the external transition 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 contrary, it receives external event X by its input port. 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, next state is computed. That means

On arriving of external event x	
	Execute the event by external transition function
	Change state
	Schedule next internal event
Internal event	
	Execute output function
	Execute internal transition function
	Change state
	Schedule next internal event
	Inform to parent

However, a biological system does not contain only such a simple component. In fact, it is composed of many complex components which in turn are constructed by a set of subcomponents organized in many levels. Atomic model is not suitable to describe such a component. Fortunately, Zeigler introduced also another one: coupled model.

A.2 Coupled model

In DEVS modeling, complex models are built by coupling together atomic building blocks, i.e., connecting the ports of well defined input and output interfaces. Models can be built in a hierarchical manner, i.e., coupled models again can serve as components in more complex coupled models ([21], [17]).

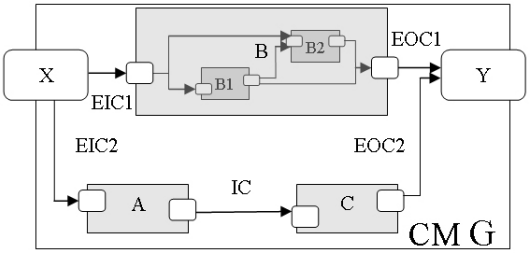


Fig. 3. Coupled models

Coupled model has the following structure:

$$C =< X, Y, N, M_d | d \in N, EIC, EOC, IC, Select >$$

- 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 chosen, executes its output function and sends the result to all its influenced. 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 (or one may use the term detectors and effectors ([13], [15]). For example, the coupled model G of figure 2 comprises two atomic models A,C and a coupled model B which consists of two atomic models B1 and B2.

This type of component can be considered in turn like a basic element in a larger model. The model is created in a recurrent way.

Thus, we have chosen DEVS modeling and simulation approach as the modeling framework. Discrete Event System Specification formalism allows us to express structural and behavioral features of dynamic systems. DEVS model

characterizations in terms of events and states make it suitable for agent technology and thus suitable for studying large scale system.

B. Agent-based simulation

Integration of those concepts allows the automatic generation of a simulator from the model, as defined in the DEVS formalism. Furthermore, we can reference agent internal structure as well as reaction rules to DEVS models. Internal ports are mapped into agent sensors, external ports are mapped into agents' effectors. The agent may order its simulation model to execute primitive actions, which are mapped into external events understandable by the DEVS simulation models [19]. The simulation model, in turn, informs the associated agents about its state and environment. The lowest level of detailed tasks is known as primitive actions are executed by DEVS atomic model. All inferred information, sensory data collected from the simulation will be stored in agent's memory. The integration of well-known approaches (DEVS) together with agents gives us a mean to study dynamics evolution of systems made up of a number of defined interacting parts in a natural way.

C. Neurons dynamics

The neurons can be considered as a "black box" model. We try to describe each process intervening in operation of this organization by identifying input ports, output ports, transformation function and transfer function.

Fig. 4. Neuron dynamics

We turn now on reception and emission process of action potential of neuron u. The neuron u receives a product X via its synapses. Then, neuron u transforms this product into others ones. The transformation is continuous. The last one is then emitted to others neurons via its output ports.

S-propagator ([8], [3], [10], ...) of Chauvet is taken to describe action potential propagation between two neurons.

We now describe briefly neuron dynamics by S-propagators. For details information, refer to [10].

C.1 Presynaptic release and diffusion in the synaptic cleft P_1 and P_2

$$A_0(r,t) = \begin{cases} A_m & \text{if } \psi_p(r,t) > \psi_{th} \\ 0 & \text{if } \psi_p(r,t) \leq \psi_{th} \end{cases}$$

$$\frac{\partial A_{cleft}}{\partial t}(r,s,t) = D_{cleft}\frac{\partial^2 A_{cleft}}{\partial s^2}(r,s,t) - pA_{cleft}(r,s,t)$$

C.2 Neural transmitter propagation: Trans-operator P_3 and P_4 (Postsynaptic binding to the receptor and passive conduction of the postsynaptic currents)

Trans-operator P_3 (Postsynaptic binding to the receptor)

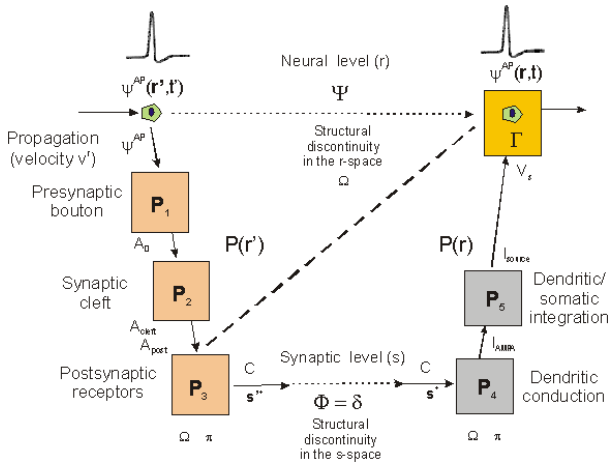


Fig. 5. Functional interaction propagation, Chauvet 2002

$$\frac{dR}{dt}(r,t) = k_{-4}R_d(r,t) + k_{-1}RA(r,t) - (k_4 + k_1 A_{post}(r,t))R(r,t)$$

$$\frac{dR_d}{dt}(r,t) = k_4 R(r,t) + k_{-3}R_d A(r,t) - (k_{-4} + k_3 A_{post}(r,t))R_d(r,t)$$

$$\frac{dRA}{dt}(r,t) = k_1 A_{post}(r,t)R(r,t) + k_r R_d A(r,t) + k_c C(r,t) - (k_{-1} + k_d + k_0)RA(r,t)$$

$$\frac{dR_d A}{dt}(r,t) = k_3 A_{post}(r,t)R_d(r,t) + k_d RA(r,t) - (k_{-3} + k_r)R_d A(r,t)$$

$$\frac{dC}{dt}(r,t) = k_0 RA(r,t) - k_c C(r,t)$$

Trans-operator P_4 (Passive conduction of the postsynaptic currents)

$$R_m(z)C_m(z)\frac{\partial \psi_m}{\partial t}(r,z,t) = \lambda(z)\frac{\partial^2 \psi_m}{\partial x^2}(r,z,t) - (\psi_m(r,z,t) - V_{rest}) + \frac{R_m(z)}{2\pi a(z)}I_{source}(r,z,t)$$

C.3 The source term Γ

$$\Gamma(\psi,t;r) = \frac{1}{r_m c_m}V_s(r,t) + \sum_{ions}\frac{\Delta g_{ion}}{c_m}(\psi_{e,ion} - \psi(r,t))$$

Presynaptic release and diffusion in the synaptic cleft is interpreted as transition function, postsynaptic binding to the receptor and passive conduction of the postsynaptic currents as transfer function and the source term as transformation function.

The S-propagator formalism appears as an efficient mean for representing the hierarchical nature of physiological phenomena. S-propagators give us a mathematical tool for mapping from complex biological system to DEVS and agent-based framework.

D. Neuron's behavioral description by DEVS

Each neuron is represented by an atomic DEVS model in which state variable is neuron's action potential.

```
class Neuron {
public:
   Input in;
   Output out;
   StateVar state;
StateVar delta int ()
   {
      Reset membrane potential value.
   }
StateVar void delta ext (event x)
   {
      Transformation (the source term in the previous section)
      Compute membrane potential
      Compare with threshold value (P1 and P2)
      If it is equal or greater
      {
         Compute propagation time (P3 and P4)
         Create out-msg() with occurrence time equal to propagation
time.
      }
   }
void lamda(void)
   {
      Send corresponding product to output port.
   }
Time ta()
   {
      Compute next occurrence time.
   }
   ...
}
```

Adopting the abstract simulator concept of DEVS, the model is executed by sending typed messages between simulator agents. Simulators are associated with atomic models and coordinators with coupled models [19].

We now take a closer look on agent structure.

E. Agent-based simulation

Agent is defined as an "active object" that is: autonomous, perceptive, pro-active and communicative [20]. Typical agent objects are composed of two parts: an internal state and behaviour. In brief, agents are implemented to have internal data representation (memory or state). They possess means for modifying their internal data representation (perception) and for modifying their environment (behaviour).

This point is illustrated by the following pseudo-code extracted from [1].

```
Agent object:
        private states:
                preferences;
                wealth_1; /* private wealth */
                ...
        private behavior:
                compare_choices;
                compute_internal_valuations;
                communicate_with(Agent i)
                draw;
                ...
        end.
```

The external transition function encodes the reaction of the agent to incoming events in terms of state changes. The time advance is set to the reaction time, the time an agent needs to produce its output [19].

Part of the agent's activities is communicating actions to other models. The output function takes the first of the intentions and charges its output ports with effects directed to the environment [19].

The internal transition completes the activity by updating the agent's state [19].

The discrete initiation of events can easily be interpreted as activities, and the exchange of information via message passing as the communication between agents ([18]). According to external perturbations (messages), the agent changes its internal state. Since any component of a biological system is modeled by a component in DEVS formalism, when being referred as an agent then this one possesses all mathematical methods to show up a behaviour facing to events received from their environment.

Accordingly, an agent's "output" activities are decoupled from receiving external events. Both, the perception of events and the reaction directed to the environment, interact via state and the time advance function. Agent's first reaction to external perturbations is a change of its internal state. For each state, there is a time advance function associated. It determines the time of the next internal event and output, e.g., the time an agent needs for reacting to external perturbation. External events might shorten or lengthen the time period until the next output, e.g. some external events might require an agent's immediate reaction which is achieved by setting the time-advance close to zero. Besides, modeling the temporal aspect of agent's reaction facing to external events, the time-advance function allows agent proactive behavior to be modeled since outputs depend on agent's current state and are triggered by time. Thus, an agent does not require any external event to become and stay active [19].

Running such a model consists of instantiating an agent population, letting the agents interact, and monitoring what happens [18]. We have a typical agent-oriented program presented by Axtell in [1]:

```
program typical_agent_model;
        initialize agents;
        repeat:
     agents_interact;
     compute_statistics:
        until done;
end.
```

Agent-based simulation method does not replace traditional method in biological field [11]. It can be combined with equation-based methods because, within an individual agent, behavioral decisions may be done by the evaluation of equations. The system level behavior is then determined by running the equations describing the interactions among these agents.

Furthermore, agent-based method is not targeted for a given physiological system. So we can apply this method for other problems that have an emergent behaviour produced by a complex set of connected individual interactions.

IV. Conclusion

The paper presents a generic agent based simulation approach taking into consideration requirements deemed necessary for agent/simulation architecture. An agent based simulation environment accepts the simulation model as its environment. The agent reacts to the events happening in the simulation environment and further may behave in proactive ways.

With this proposal, we hope to extend our simulation model aimed to incorporate Purkinje cell model, as well as other neurons in the cerebellar cortex to study hippocampus. These models will be closely based upon known structural and physical properties of this region of the cerebellum and will produce neuron-like outputs that can be compared to data from actual physiological experiments.

References

[1] R. Axtell, *"Why agents? On the varied motivations for agent computing in the social sciences"*, working paper n17 at Center on Social and Economic Dynamics, 2000.

[2] J. Banks (Editor), J. S. C., Barry L. Nelson, David M. Nicol (2000). *Discrete-Event System Simulation*, Prentice Hall.

[3] G. Chauvet (1996). *On the hierarchical organisation of physiological systems: the example of bone physiology.* Cells and materials, vol 7(4), p 259-269.

[4] A.G. Chauvet, *Theoretical systems in Biology: Hierarchical and Functional Integration.* Volume II: *Tissues and Organs*, Oxford: Pergamon, II , 1996

[5] A.G. Chauvet, *Theoretical systems in Biology: Hierarchical and Functional Integration.* Volume III: *Organization and regulation*, Oxford: Pergamon, III , 1996

[6] A.G. Chauvet, *S-propagators: a formalism for the hierarchical organization of physiological systems. Application to the nervous and the respiratory system*, Inter Journal of General Systems, 28 (1)53-96, 1999

[7] G. Chauvet (2002). *On the Mathematical Integration of the Nervous Tissue Based on the S-propagator Formalism: I. Theory* . Journal of Integrative Neuroscience, vol 1(1), p 31-68.

[8] B. Daya, G. Chauvet (1999). *On the role of anatomy in learning by the cerebellar cortex.* Mathematical biosciences, vol 155, p 111-138.

[9] G. Chauvet, P. Chauvet (2002). *On the Mathematical Integration of the Nervous Tissue Based on the S-Propagator Formalism: II.Numerical Simulations for Molecular-Dependent Activity.* Journal of Integrative Neuroscience, vol 1(2), p 157-194.

[10] P. Chauvet (2003). *LifeLike: un outil de simulation de systme biologiques.*

[11] H. V. D. Parunak, R. Savit, R. 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, p 10-25.

[12] F.W. Howell, J. Dyhrfjeld-Johnsen, R. Maex, N. Goddard, E. De Schutter (1999). *A large scale model of the cerebellar cortex using PGENESIS.*

[13] J.H. Holland, *"Hidden order: how adaptation builds complexity"*, Addition Wesley, 1996.

[14] C. Lecerf, M. Bui. (2001). *Modeling physiological function: parallels with distributed processing.* Eurosim.

[15] R. Norvig, *"Artificial Intelligence: A Modern Approach"*, Prentice Hall, 1977.

[16] E. De Schutter, and R. Maex, (1996). *The cerebellum: cortical processing and theory.* Curr. Opin. Neurobiol. 6 759-764.

[17] H. Praehofer and G. Reisinger (1994). *Distributed Simulation of DEVS-Based Multiformalism Models* Proc. 5th Conf. on AI, Simulation and Planning in High Autonomy Systems, ed. PA Fishwick, JW Rozenblit, IEEE Computer Society Press, Los Alamitos CA, p. 150-156.

[18] A.M. Uhrmacher (1996). *Concepts of Object- and Agent-Oriented Simulation.* In Workshop on Multiagent Systems and Simulation MASSIM'96, Ulm, ASIM Mitteilungen, Number 53, p. 1-8

[19] A. M. Uhrmacher, B. Schattenberg (1998). *Agent in discrete event simulation.* 10TH European Simulation Symposium 'Simulation in Industry - Simulation Technology: Science and Art' (ESS'98), Nottingham, SCS Publications, 129-136.

[20] M. J. Wooldridge N. R. Jennings. (1995). *Intelligent Agents: Theory and Practice.* The Knowledge Engineering Review. 10(2). 115-152.

[21] B.P. Zeigler, *"Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems"*, Academic press, 1995.

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