A reference model and a theory for multiagent, information systems

Virginia Padilla Sifontes[*]
Universidad Nacional Experimental de Guayana
Ciudad Guayana, Venezuela
`vpadilla@uneg.edu.ve`

Jacinto Dávila Quintero
[*]Centro de Simulación y Modelos
Universidad de los Andes. Mérida, Venezuela
`jacinto@ula.ve`

Abstract

This paper presents a reference model for integrated information systems. The model explains the relations between agents, simulations and geo-spatial databases, with the aim of guiding the implementation of computational platforms that integrate them. For this, the model includes the common conceptual constructs used in several multiagent systems methodologies. We have been building on previous work in simulation[5] which is here connected with the proposal by [2] to produce a multiagent theory for geosimulation to model and simulate complex spatial systems.

1 Introduction

A reference model is the epistemic base of an ontology. It is the description of objects and concepts within some domain of knowledge[1] . In what follows, we propose a mathematical specification of a multiagent system with the purpose of guiding computational implementations of such kind of systems, particulary, those applied to knowledge management and simulation. We have been building upon previous work in AI[6] and Simulation[3, 4] which led to a multiagent theory for simulation[5] which guided the development of the GALATEA simulator (.ibid). Here we adapt that theory, and connect it with the MAGI theory proposal by [2] to produce a multiagent theory for geosimulation, which seeks to explain the relations between agents, databases and geographic information systems as tools to model and simulate complex spatial systems. GALATEA's theory describes the overall dynamics of a multiagent system, depicted as the *Cycle* function, which "moves" the system from one global state to its chronological next. The global state is characterized, at *time t*, by a set of state variables and their values, represented by σ (and the like) in this formalization. However, as proposed by Ferber and Müller[6], the global state also includes a set known as the influences, γ (and the like), which represents all the actions that the agents have executed (their action history) or are currently trying to execute (also known as their *intentions*). An agent is then modeled as a 5-tuple of 1) the set of possible percepts, *Perception*, a function that explains how the agent actually perceives its surroundings, 2) a knowledge base, 3) a set of goals (or intentions), 4) *Update*, a function which is the memorization mechanism of the agent and 5) the *Planning* function that represents the reasoning function that derives new goals and influences, taking into account previous goals and the knowledge base.

The MAGI theory [2] is a meta-model that amounts to a formal theory of a geography with agents and objects in it. This MAGI theory is a perfect complement for our multiagent theory as it provides for 1) the embodiment of agents and 2) a carefully tailored account of the data structures and associated functions required for a geographic information system to efficiently computed answers to queries. A third side-effect of the combination of these theories is the possibility of accounting for the creation of objects and agents. The theory of Galatea did not have those elements. On its side, the MAGI theory acquires an explicit account of time and a DEVS strategy for time management.

---

[1] ontology in its historical sense, rather than the modern, technical meaning.

In the MAGI theory, the environment, Env, is characterized by a 3-tuple from the cross-product of 1) the set of all possible global *parameter=value* pairs to describe a system, 2) the set of all possible global functions operating upon those parameters and 3) the set of all possible layers, $\mathcal{L}$, of objects that may constitute the geography of the system. Each layer, $L \in \mathcal{L}$, is characterized in turn, but another 3-tuple from 1) the set of all possible local parameter, 2) the set of all possible local functions and 3) the set of entities (objects and agents) that populate the system. Agents are, in turn, described by a double record: the agent itself and its type τ. The agent is described by its internal state, its geo-spatial attributes and the set of references to objects and other agents observed by this agent and subjects of its actions. An agent type τ is described by 6-tuple: 1) the set of all possible agent internal states, 2) the set of admissible shapes for this agent type, 3) the set of all possible actions, 4) the set of perception functions, 5) the set of decision functions and, 6) the set of agreement functions by which this agent cooperates with other agents. It should be clear that this corresponds to an embodied accounts of a multiagent system, due to the fact that agents have well defined attributes for their bodies and locations on a physical space.

All these elements are formalized in the new combined theory, which constitutes a formal explanation of the relationship between an agent and its environment. Notice that the environment can have its own dynamics, sometimes elaborated to the point that its could be seen as an agent. Therefore, one could think of that simple two-agent system as a minimal layout for a system specification: an individual agent faces the agent environment and interacts with it. This could be easily generalized to many agents facing the same environment.

In what follows (section 2), we produce the formalization and then, in section 3, we explain the relations between the formal model and a databases model before presenting the conclusions.

2 A new approximation to a formal and embodied model of a multiagent system

What follows is the top-most mathematical description of a system populated by agents, depicted as the $Step_{out}$ function, which "moves" the system from one global state to its chronological next. The functions $Evolution$, states that the system evolves as described by: $Evolution : \Im \otimes Env^* \to \epsilon$

$$Evolution(t, env) = Evolution(Step_{out}(t, env)) \tag{1}$$

where $env \in Env^*$ is an embedding structure containing all the global state, as the elements of Env^*, but also the influences in the system:

$$Env^* =< P_G, F_G, \mathcal{L}, \Gamma > \tag{2}$$

where P_G represents the set of all possible global *parameter=value* pairs to describe a system and F_G the set of all possible global functions operating upon those parameters, while $\mathcal{L}$ is the set of all possible layers of objects that may constitute the geography of the system, and Γ, is the set known as the influences, which represents all the actions that the agents have executed (their history) or are currently trying to execute (also known as their intentions).

$Step_{out}$ formalizes the system stepping from an overall situation to its next by the agents computing their actions, and, therefore, $Step_{out} : \Im \otimes Env^* \to \Im \otimes Env^*$:

$$< t'', env'' >= Step_{out}(t, env) \tag{3}$$

where $env'' =< p'', f, l'', \gamma'' >$ and $env =< p, f, l, \gamma >$, and $p'' \in P_G$, $f \in F_G$, $l'' \in \mathcal{L}$ and $\gamma'' \in \Gamma$. Each layer l is described by $l =< p_l, f_l, A >$ where A is the set of agents in this layer.

The environment reacting to the agent's actions by changing its state:

$$< p'', f, l'' >= React_{realtime}(t', p', f, l', \gamma') \tag{4}$$

and the actions being executed by the environment:

$$< \gamma'' >= Exec(p', f, l', \gamma' \cup_i \gamma'_i) \tag{5}$$

where $\gamma' \cup_i \gamma'_i$ is the set of perceptions of the environment and its past history in t'.

We use γ_i to represent the influences produced by agent i and s_i is its mental state. If r_i is the time that an agent uses while reasoning, then the behavior of an agent is modeled by the equation:

$$< s''_i, \gamma'_i >= Behaviour_i(t', r_i, s'_i, \gamma) \tag{6}$$

These equations account for the overall behavior of the whole system. However, just before that overall behavior is triggered, the system progresses through a subtler activity by means of the $Step_{in}$ function in: $Step_{in} : S \otimes \Im \otimes Env^* \otimes \Gamma \rightarrow S \otimes \Im \otimes Env^* \otimes \Gamma$:

$$Step_{in}(< s_1, s_2, .., s_n >, t, < p, f, l >, \gamma) =<< s'_1, s'_2, .., s'_n >, t', < p', f, l' >, \gamma' > \tag{7}$$

This activity, formalized by equations 8 through 12 , accounts for a simulation process in which there may be agents included, as modeled by equation 9. In turn, with Λ, description of the system through the laws of change and, β, structure and support information about the system, $React_{simulationtime} : \Lambda \otimes \beta \otimes \Im \otimes Env^* \otimes \Gamma$:

$$<< p', f, l' >, \gamma' >= React_{simulationtime}(<\lambda_1||...||\lambda_m>, t, < p, f, l >, \gamma \cup_j \gamma_j) \tag{8}$$

$$< s'_j, \gamma_j >= Behaviour_j(t, r_j, s'_j, \gamma) \tag{9}$$

As part of the simulation activity, equations 10, 11 and 12 describe the usual elements of a DEVS simulations [10][9], where env is a model of a simulated system which includes its set of components and the laws of behavior of that system (which are then selected as $< \lambda_1||...||\lambda_m >$), and whose associated events (i.e. ξ) drive the whole dynamics inside a simulation. It is worth noticing that within the real time period t to t', both points on the real scale of time, time progresses as dictated by the simulation. So, at least from a timing perspective, reality subsumes the realm of simulations.

$$\xi = NextEvent(\gamma \cup_j \gamma_j) \tag{10}$$

$$< p, f, l >= Scan^*(env, \xi) \tag{11}$$

$$t' = TimeOf(\xi, t) \tag{12}$$

Then the top-most structure env is scanned to identify the actual component where the next event will happen and to recover its associated instructions. Thus this is a specific kind of scanning, guided by the geographical attributes of the system.

Agents are modeled as follows. An agent is a 4-tuple:

$$a_\tau =< k, goals, georefs, Context > \tag{13}$$

where $k \in K_\tau$, $goals \in G_\tau$, $georefs \in Shapes_\tau$ and $Context$ is the spatial context explained above. Notice that, as indicated by the subscript, this agent is associated to a agent type τ which, in turn, is formalized by the 8-tuple:

$$\tau =< K_\tau, G_\tau, Shapes_\tau, \sum_\tau, P_\tau, Perception_\tau, Update_\tau, Planning_\tau > \tag{14}$$

where K_τ is the set of possible knowledge bases for this type of agent, G_τ is the set of possible goals for this agent type, $Shapes_\tau$ is the set of admissible shapes the body of the agent (of this type) can adopt,$\sum_\tau$ is the set of possible actions this agent can execute, P_τ is the set of possible observations this type of agent can perform, and *Perception*, *Update* and *Planning*, model the connections between perceptions and actions for this type of agent. All these specific structures can be connected with the rest of the system by a new of agent behavior function:

$$Behaviour_a : \Im \otimes \Re \otimes K_\tau \otimes G_\tau \otimes \Gamma \rightarrow K_\tau \otimes G_\tau \otimes \Gamma$$

$$< k', goals', \gamma_a >= Behaviour_a(t, r_a, k, goals, \gamma) \tag{15}$$

where

$$k' = Update_a(t, Perception_a(\gamma), k)$$

$$< \gamma_a, goals' >= Planning_a(t, r_a, k', goals)$$

and where $< k', goals', \gamma_a >$, depicts the knowledge base, the goals of agent a and the influences,γ_a, this agent is posting to its environment as actions it intends to execute.

3 An agent-oriented model for databases and information systems

The complexity of the formalization presented in the previous sections is well justified. Its main purpose is to support the development of computational system that combine databases, simulators and information systems.

Databases research is a field with a long and productive tradition [8] which, somehow, could discourage new developments. Fortunately, some researchers insist [1] there still are many open challenges, such as big and complex data management, integration of structured and unstructured data and applications on mobile and social networks. This has encouraged us to include a database's analysis in our work. We believe the theory presented in section 2 could serve as a model for the full range of functionalities from a simple database system up to complex multiagent systems.

To support such a statement, let us consider the formal model of a deductive database proposed by Gallaire[7] which was intended to account for active databases as well. A deductive database's static components can be described as a 3-tuple $< T, IC, Q >$, where T is a theory that includes domain closure, uniqueness of names, equality and completion axioms, elementary facts and the deductive rules; IC is the set of integrity constraints and event-condition-action, ECA, rules and Q represents the set of possible queries.

The agent model subsumes these by regarding $T \subseteq K_a$ and $IC \subseteq G_a$, being K_a and G_a the agent's knowledge base and goals, respectively, as described in previous sections. Also notices that we make $Q \subseteq P_a$, i.e. Q corresponds to the set of possible inputs a databases can get (as queries). We could also associate the database with a particular geophysical agent (of type τ) by using K_τ, G_τ and P_τ as suggested by the theory.

With respect to the dynamic models of the database, both the execution model and the inference engine are included in the agent planning and execution mechanism of the multiagent reference model. Moreover, the multiagent reference model provides an activating mechanism that is not accounted for by the standard database model. The agent is immersed in a dynamic process called the *observe-think-execute* cycle, which continuously monitors the surrounding world and whenever it detects that some of the observations match the conditions of its rules, then the corresponding actions are produced for execution. The agent's cycle *observe-think-execute* is reduced to a process of *observe-think* in a database, where *observe* corresponds to the query being sent to the database and *think* corresponds to the triggering mechanism. In short, an active database is an agent with a peculiar set of beliefs and goals.

4 Conclusion and Future Work

We have developed a general-purpose, mathematical model for knowledge management systems which formalizes the relations between agents in a multiagent system, simulations and information systems. We believe this model establishes the bases for a common language (with various syntactic and graphical expressions) with an extended semantics to integrate rules, events, time management, conditions of operation, and other DBMS constructs into the agent oriented paradigm. Also, we had established a basis for integrating the simulator Galatea and a geographic information systems, to produce a more flexible, adaptable, user-friendly, distributed and computationally efficient tool to simulate complex spatial systems.

We are working to implement a combined GIS-simulator system that embody the model described in this paper. We are confident that the model will support the specification and swift implementations of platforms for agent-oriented, data-intensive, knowledge management services.

References

[1] R. Agrawal, A. Ailamaki, P. Bernstein, E. Brewer, M. Carey, S. Chaudhuri, A. Doan, D. Florescu, M. Franklin, H. Garcia-Molina, J. Gehrke, L. Gruenwald, L. Haas, A. Halevy, J. Hellerstein, Y. Ioannidis, H. Korth, D. Kossmann, S. Madden, R. Magoulas, B. Ooi, T. O'Reilly, R. Ramakrishnan, S. Sarawagi, M. Stonebrakerl, A. Szalay, and G. Weikum. The claremont report on database research. *SIGMOD Rec.*, 37(3):9–19, 2008.

[2] I. Blecic, A. Cecchini, and G. Trunfio. *Studies in Computational Intelligence 176*, volume 176, chapter A Multi-Agent Geosimulation Infrastructure for Planning, pages 237–253. Springer, Berlin, 2009.

[3] J. Dávila, E. Gómez, K. Laffaille, K. Tucci, and M. Uzcátegui. Multi-agent distributed simulations with galatea. In *The 9-th IEEE International Symposium on Distributed Simulation and Real Time Applications*, IEEE Computer Society, pages 165–170, Montreal, Canada, 2005.

[4] J. Dávila, M. Uzcátegui, and K. Tucci. A multi-agent theory for simulation. In *Fifth IASTED International Conference on Modelling, Simulation and Optimization*, pages 285–290, Oranjestad, Aruba, 2005.

[5] J. Dávila, M. Uzcátegui, and K. Tucci. From a multi-agent simulation theory to galatea. In *Summer Computer Simulation Conference (SCSC0'7)*, San Diego, CA, USA, 2007.

[6] J. Ferber and J-P. Müller. Influences and reaction: a model of situated multiagent systems. In *ICMAS-96. Second International Conference on Multiagent Systems*, pages 72–79, 1996.

[7] H. Gallaire, J. Minker, and J-M. Nicolas. Logic and databases: A deductive approach. *ACM Comput. Surv.*, 16(2):153–185, 1984.

[8] A. Geppert and K.R. Dittrich. *Advanced Database Technology and Design*, chapter Component Database Systems, pages 403–435. Artech House, Boston, 2000.

[9] G. Wainer. *Discrete-Event Modeling and Simulation.* CRC Press, Boca Raton, FL, 2009.

[10] B.P. Zeigler, H. Praehofer, and T. Gon Kim. *Theory of Modelling and Simulation.* Academic Press, San Diego, second edition, 2000.