Agent Oriented Simulation with OOCSMP. An example in evolutionay ant colonies.

Juan de Lara[1,2]

[1] School of Computer Science
McGill University
3480 University Street
Montrèal, Quebèc, H3A 2A7, Canada
Juan.Lara@ii.uam.es

Manuel Alfonseca[2]

[2] Dept. Ingeniería Informática,
Universidad Autónoma de Madrid
Ctra. De Colmenar, km. 15, 28049 Madrid,
Spain
Manuel.Alfonseca @ii.uam.es

1.Introduction

Agent-based simulation [1] is a powerful and natural way to carry out complex simulation experiments, in which many autonomous and interacting entities take part. The key abstraction in this methodology is the autonomous agent, which interacts via discrete events.

OOCSMP [2] is an object oriented continuous simulation language. A compiler (C-OOL) was built for this language to produce C++ code or Java applets from the simulation models. This approach would simplify the generation of simulation based web courses. A number of them have been generated using this language which can be accessed from: `http://www.ii.uam.es/~jlara/investigacion`.

Although it was conceived as continuous, OOCSMP has features that allow certain degree of discretization, such as: possibility of handling discrete events, to iterate over matrix and vector indexes and blocks to convert continuous to discrete values.

2.Extending OOCSMP for Agent-Oriented Simulation.

Several extensions have been added to OOCSMP in order to perform agent-oriented simulation, such as: Multiple object constructor invocation; Objects can be eliminated from the simulation; A new output form represents the position and the state of the agents and Multicast and Broadcast message-passing mechanisms. This allows methods to be invoked on objects (point-to-point), classes (broadcast) or collections of objects (multicast). The way the elements of the class/collection are accessed can be sequential (first to last or last to first), specified by the user in a vector or random. This can be necessary to avoid phenomena that arise due to accidentally imposed inter-agent correlation.

3. An example: Simulation of an evolutionary virtual ant colony

Our *vants* communicate directly with other vants when they are near, not by dropping pheromones. They live in a two-dimensional grid and their objective is to find food. When this happens, they eat a portion (extending their life span), and take another portion to the nest. This may be repeated until the food is depleted. Several locations with food may exist at the same time. When a vant arrives at the nest, it rests there for some time. When two agents meet outside the nest, they can exchange their knowledge about the food position. If a vant does not find food during a certain period of time, it returns to the nest.

Our vants have several parameters that control their behaviour: *Activity*, for its speed; *Communicative* to control if it will communicate with another agent when they meet; *Scepticism* for its credulity; *Lie* to control the degree to which it will lie when informing the others of the food position and *Memory*, that controls the probability of forgetting the food position. The five parameters are encoded in binary and concatenated, making a genotype. When two agents meet at the vanthill, they can reproduce if there's enough food in the nest. In each reproduction, two new agents are created, with '*genetic*' information resulting from their parents genomes after the operations of mutation and crossing-over have been applied to them, a typical procedure in genetic algorithms.

4.Conclusions and future work

The *Activity* and *Memory* parameters in the population grow quickly to its maximum. The other parameters may oscillate, but when food is scarce, agents compete between themselves and liars begin to proliferate. As this parameter goes up, scepticism also grows. When there's plenty of food, it's better for vants to co-operate, and liars and sceptics may disappear quickly. This is an emergent behaviour of the system.

Acknowledgment

This paper has been sponsored by the Spanish Interdepartmental Commission of Science and Technology (CICYT), project number TEL1999-0181

References

1. Wooldridge, M. 1999. "*Intelligent Agents*". In "*Multiagent Systems. A modern approach to Distributed Artificial Intelligence*" (Weiss ed.). pp. 27-77, The MIT Press.
2. Alfonseca, M., de Lara, J., Pulido, E. *"Semiautomatic Generation of Web Courses by Means of an Object-Oriented Simulation Language"*, special issue of "SIMULATION", Web-Based Simulation, Vol 73, num.1, July 1999, pp. 5-12.