Simulation Methodology of the Evolution Process in Multi Agent World with the SWARM Simulation System.

Rafal Drezewski[*]
drezew@agh.edu.pl

Krzysztof Cetnarowicz[*]
cetnar@agh.edu.pl

Abstract

With the *Evolutionary Multi Agent Systems (EMAS)* it is possible to simulate biological mechanisms of species formation, rivalry and competition between species, and social behavior. But there are many serious difficulties, which one faces in the process of setting up and observing computer simulation of the biological system. Besides, the crucial process of independent verification via replication of results is almost unheard of in the area of computer simulation. Therefore, the appropriate simulation techniques and tools are indispensable. In this paper the simulation methodology of the evolution process realized in the multi-agent world is introduced. In addition we discuss the simulations of the evolutionary multi-agent prediction system carried out with the use of SWARM simulation system.

1 Introduction

It is typical in the research work made with the use of computer simulation that we try to observe and investigate properties of phenomena, which adequate mathematical model cannot be formulated. We can only formulate models of subsystems (i.e. components of a system of interest) and define some basic interactions between them. But the relations that we are really interested in are nowhere explicitly encoded. They rather *emerge* and become accessible for observation as a result of interactions between the subsystems that we simulate. All these things result from the fact that we deal with the complicated structure composed of some elements. The features of such a structure cannot be simply deduced from the features and behavior of its elements when they are taken out from the structure. They can be researched only when these basic elements remain and act within a structure. This fact arises because the structure is not just a simple sum of its components. Structure specific features emerge at a higher level then our model is formulated, as a result of complicated interactions between basic elements of the structure. Therefor, structure specific features can be observed and researched only with the use of computer simulations. This is also the very natural way of developing the artificial intelligence. Artificial intelligence will not be achieved by single, very complicated and sophisticated algorithm. It will rather emerge as a result of interactions between numbers of quite simple entities, which can evaluate and thus adapt to the changing environment's conditions. This is also the way that the natural intelligence was born. The general research program of investigating complex dynamical phenomena using simulation is [6]:

1. We must always be aware that a simulation is generating dynamical phenomena at a level that is *higher* than the level from which the elemental interactions are described. If we are to exploit simulation it is necessary to understand what this capability to produce hierarchies of emergent relations implies.
2. We must have methods with which to identify the elements of the underlying system that create the phenomena of interest.

---

[*] Institute of Computer Science, University of Mining and Metallurgy, Krakow, Poland. This research was sponsored by KBN (Scientific Research Committee of Poland) under Grant No. PB8\,T11C\,008\,12.

3. It is then necessary to formulate models of the important underlying subsystems (those that define the elemental subsystems and the element-element or object-object interactions).
4. Finally, we must create the framework in which the simulation of the subsystems in interaction is composed, and embody the system representation in that framework so that the phenomena of interest can be generated and analyzed.

The technology of *Evolutionary Multi Agent Systems (EMAS)*, that arises as a result of introducing the evolution process into the multi-agent world, can be treated as a new approach to the construction of evolution programs and a new way of developing multi-agent systems.

Evolution process realized in the multi-agent world offers us new possibilities such as:

1. In the process of developing of our system we can make use of existing models of multi-agent systems. This has the effect that agents that participate in the evolution process, environment in which they act, agent-agent and agent-environment relations are well defined. All this features deal with the parts 2-3 of the research program described above.
2. Evolution process is decentralized and is performed with no common cadence.
3. Agents can interact independently with each other and with environment. This cause that emergence of social relations between agents is now possible.
4. Process of selection is decentralized because agents can compete for limited resources of the environment.
5. Last but not least, by introducing evolution process into multi-agent system we provide agents the best mechanism of the adaptation toward changing environment.

With the *Evolutionary Multi Agent Systems* it is possible to simulate biological mechanisms of species formation, rivalry and competition between species, social behavior and so on. It was not possible to introduce all these mechanisms in the case of classical evolution programs [4, 5].

Part 4 of the research program in the area of complex systems refers to the creation of the framework and the system representation. If we think for a while about this step we will quite soon realize that there are many problems connected with the simulation of such a system. There are four main parts of a system creation. First we have to create the space and time in which the entities will act. In the case of *EMAS* systems the space and time are usual discrete. Next we have to create entities (agents) and mechanisms of their low-level interactions. These interactions are explicitly coded and we can formulate their models. The main interests of our research are of course higher level relations, which model cannot be formulated. We must have possibilities to observe our artificial world, thus it is also indispensable to build observing tools. In the end we must schedule events over those entities. In general agent-agent and agent-environment relations can be treated as discrete events of changing internal states of these objects. These changes are then communicated to the other entities that act in the system.

Problem of time is perhaps the most challenging part of the system creation. It is clear that we want to keep our system in the well-defined state of synchronization. For example, we want to update states of one kind of the objects before the other kind of the objects will be updated because they depend on each other in some way. This is connected with the more general problems in the area of computer science of how to use and deal with concurrency. All these problems are quite subtle and sometimes-even people who make simulations are not conscious of their implications and how they affect results of their simulations.

Also, the very important thing is that the crucial in the science process of independent verification via replication of results is almost unheard of in the area of computer simulations. This fact results from the use of different simulation tools, operating systems, and hardware in the research work. Therefor, we must use simulation tools that will allow us to create repeatable and reliable results of our simulations.

2 The formal model of the multi-agent system

As we have said already, there are many models of multi-agent system that we make use of in the steps 2-3 of the research program. In the following sections, we will be using one of these models: the model based on the concept of the M-Agent architecture in order to formulate lower-level models of our sample system. For the convenience of reader we introduce some basic concepts of this model. Details can be found in [1, 2].

The whole multi-agent system AW can be described as $AW = (\boldsymbol{A}, \boldsymbol{E})$, where $\boldsymbol{A}$ is a set of possible configurations of agents, and $\boldsymbol{E}$ is a set of possible configurations of the environment spaces. The whole environment $V = (E, A, C)$, where E is the space in which agents remain, A is a set of agents which act in E, and C is a set of agent-agent and agent-environment relations. The space $E = (R, T)$, where T is a topology of space, and R is a configuration of resources.

An agent can be described from two points of view that correspond with two profiles: intellectual and energetic.

Intellectual profile of the autonomous agent $a = (M, Q, S, I, X, L, m, q, s)$, ([1, 2]) where M is a set of models representing agent's knowledge about the environment; m is an actual model of the environment, $m \in M$; Q is a queue (a set with a given order) of agent's goals; q is a goal actually realized by the agent; S is a set of strategies, which performing agent may take into consideration; s is a strategy to be realized; I is the observation operator, which with the use of the set M, builds the model m of the environment; X is the strategy s realization operator, when applied it causes changes in the environment; $L = \{L_M, L_S\}$ is an adaptation operator that adjusts the agent to the particular characteristic features of the environment by changing sets M and S.

The agent a, using the observation operator I, builds the model m of the observed environment. Then it selects strategy s, which changes the environment according to its model m in the best way from the point of view of the goal q. The selected optimal strategy is then realized using operator X. In the end the agent a, using the operator I, builds a new model m'' of the changed environment and the whole process is repeated.

Energetic profile of an agent a may be defined as an energetic state P, which changes when the agent performs actions. The state P defines ability of the agent a to act and to survive. Energetic profile may be especially useful in evolution process, where it provides a very natural way of eliminating agents with low fitness to the environment conditions from the system.

3 SWARM: the simulation tool

The SWARM simulation system has been created at the Santa Fe Institute [7, 8]. This system can be very useful for the researchers working in the field of computer simulations, especially in the area of multi- agent systems and artificial life.

The main goal of its authors was to create such a simulation tool that the results obtained with the use of it would be reliable and repeatable. In order to achieve this goal simulation writing is brought up to a higher level of expression. Applications are written with reference to a standard set of simulation tools. The task of using the concurrency is made manageable. SWARM insulates the author of a simulation from all the computer science knowledge that is usually required to implement distributed and concurrent systems reliable. In addition SWARM forces experimenters to make their concurrency assumptions explicit.

SWARM is implemented in the Object-Oriented Programming language Objective C. Computation in a SWARM application is made via objects sending messages to each other.

SWARM applications are structured around the concept of the Swarm. Swarms are the basic building blocks of the SWARM simulation. A Swarm is combination of a collection of objects and a schedule of activity over those objects. The collections are like the matter of the Swarm and the schedule is like the arrow of time.

The core of every application is the *model swarm*. It encapsulates the simulated model, i.e. agents, physical properties and structure of the space etc. In addition to the object collection, the model swarm also contains a schedule of activity over these objects. Model swarm consists of a set of inputs and outputs. The inputs are the model parameters and the outputs are the data, which are the result of the agents' activity.

Second very important part of the system is *observer swarm*. The most important object in an observer swarm is the model swarm that is being studied. In addition observer swarm has a collection of objects (instrumentation), a schedule of activity and a set of inputs and outputs. With the use of this instrumentation we can observe our artificial world, collect data for the future analysis etc. The inputs to the observer swarm are configurations of the observer tools. The outputs are the observations. The observer swarm can run in graphics mode or in batch mode. In batch mode we cannot interact with the simulation. The batch swarm reads the data from configuration file and writes the data to the other files for analysis.

The SWARM system has large number of class libraries, which provides users tools indispensable in the process of creation of the simulation. Detailed description of all these libraries can be found in [8].

4 Sample application

In the following sections we will describe sample application of the *EMAS* technology to the 0-1 time sequence prediction system.

In this system the main goal of the population of agents is to predict the changes of the environment. In the environment a parameter $\alpha \in \{0, 1\}$ is defined. Variations of the parameter α in discrete moments of time may be represented by the binary sequence $x(t)$, where $x(t)$ is the value of α in the time t. Value of the parameter α is available for all the agents acting in the environment. Each agent tries to predict the value that the α will take in the time t+1. So the i-th agent generates the binary sequence y_i, such that $y_i(t) = x(t+1)$.

4.1 Model of the system

Resources available for all the agents are parameter α and energy P_e, so in our case $R = \{\alpha, P_e\}$. The topology T of the environment is graph. Every node of this graph has connections with its eight neighbors.

In the considered system the agent consists of one or more cells. The cell consists of the finite automaton, information about its age and statistics. Each cell makes its predictions with the use of finite automaton, which input/output language consists of symbols 0 and 1. The finite automaton plays the role of the chromosome of particular cell [3, 4, 5].

There are four parameters that show the quality of prediction made by particular agent: $\Psi_{ik}^{j(0)}$ - probability of good prediction connected with the transition through 0 from the j-th state of the k-th cell of i-th agent; $\Psi_{ik}^{j(1)}$ - probability of good prediction connected with the transition through 1 from the j-th state of the k-th cell of i-th agent; $\Psi_i(t)$ - probability of good prediction made by the i-th agent; $\Psi_{\max}^i$ - maximal probability of good prediction made by the i-th agent during its lifetime.

The evolution process involves *reproduction*, *mutation*, *crossover*, *aggregation*, and *escape* operators. In the *reproduction* process an existing agent creates a new one. Chromosomes and other parameters of the new agent are equal to its parent's. *Mutation* operator may alter output symbol and transition between particular states. *Crossover* operator mixes homologous chromosomes of two agents. The *aggregation* operator serves as a mechanism of emergence of social relations between agents. The *escape* operator allows agents to migrate toward different environments.

Two agents a_i and a_j make decision of forming an aggregate in the moment t if $d(a_i, a_j) \le d_{\max} \wedge \Psi_i(t) \le \Psi_j(t) \wedge \Psi_{\max}^i(t) \ge \Psi_{\max}^j(t)$, where $d(a_i, a_j)$ is the length of the shortest path between agents a_i and a_j in graph-like environment, and $d_{\max}$ is the maximal length of the shortest path.

The model of the environment of i-th agent in the time t is

$$m_i(t) = \langle (Y_i(t), \Omega_i(t)), (Y_i(t+1), \Omega_i(t+1)), \ldots \rangle, \qquad (1)$$

where $Y_i(u) = \{y_{i1}(u), \ldots, y_{in}(u) : n \text{ is the number of cells}\}$,

$\Omega_i(u) = \left\{\Psi_{i1}^{S_1^u(x_1(u))}, \ldots, \Psi_{in}^{S_n^u(x_n(u))} : n \text{ is the number of cells}\right\}$, for $u = t, t+1, \ldots$.

$y_{ik}(u) = t_{ik}^{S_k^u}(x_k(u))$ is the output connected with the transition of the k-th finite automaton through $x_k(u)$, S_k^u is the state of the k-th finite automaton in the time u,

$$x_k(u) = \begin{cases} x(u) & \text{for } u = t \\ y_{ik}(u-1) & \text{for } u > t \end{cases}, \text{ for } k = 1, \ldots, n.$$

After applying the strategy realization operator X the agent's model of the environment is $m'_i(t) = \langle (y_i(t), \omega_i(t)), (y_i(t+1), \omega_i(t+1)), \ldots \rangle$, where $y_i(u) = y_{ik}(u)$ for $u = t, t+1, \ldots$, k is such that $\omega_i(u) = \Psi_{ik}^{S_k^u(z(u))} = \max\left\{\Psi_{i1}^{S_1^u(z(u))}, \ldots, \Psi_{in}^{S_n^u(z(u))} : n \text{ is the number of cells}\right\}$,

$$z(u) = \begin{cases} x(u) & \text{for } u = t \\ y_i(u-1) & \text{for } u > t \end{cases}.$$

The goal of the intellectual profile is to make correct predictions:

$$q(m_i(t), m'_i(t)) = \begin{cases} 1 & \text{when } y_i(t) = x(t+1) \\ 0 & \text{when } y_i(t) \neq x(t+1) \end{cases}. \qquad (2)$$

Now the time is incremented and the observation operator I generates the model $m_i''(t) = \langle (Y_i(t), \Omega_i(t)), (Y_i(t+1), \Omega_i(t+1)), \ldots \rangle$ (by getting the value of $x(t)$ from the environment). This model is the initial model for the next iteration. The adaptation operator $L = \{L_m\}$ is consisted of all the evolution operators.

Energetic profile of i-th agent is represented by the real value $P_i(t)$. The main goal of the agent in this profile is to maximize the value of $P_i(t)$. In order to realize this goal the energetic strategy s_e is realized by applying one of the following operators: *reproduction*, *aggregation*, *escape*.

The evaluation of the i-th agent's energy is $P_i(t) = P_i(0) + \sum_{k=1}^{t} \delta_i(k)$, where $P_i(t)$ is the energy of i-th agent of age t, $P_i(0)$ is the initial energy of i-th agent, and

$$\delta_i(k) = \begin{cases} \delta > 0 & \text{when } y_i(k) = x(k) \\ \delta < 0 & \text{otherwise} \end{cases}. \qquad (3)$$

4.2 Simulations' results

The main goal of the simulations was to investigate whether new evolution operators: *aggregation* and *escape* can improve quality of the predictions made by agents. The intensity of *mutation* and *crossover* were not taken into consideration in this research. In all the experiments, which results are presented below, the population of agents had to predict the values of α parameter taken from the 0-1 random sequence. This sequence was repeated periodically giving as a result infinite sequence. The first goal of our simulations was to research if the *aggregation* operator improves results of predictions made by the agents. The maximal number of states of the finite automatons was ten. The agents had to predict values taken from the sequence that was three times longer then the maximal number of states of the finite automatons.

Figure 1 shows how the mechanism of natural selection affects the population of agents when only classical evolution operators were active. In the first step of simulation all the agents have 50% of correct predictions (this is the initial value). The number of agents significantly drops in the next steps. The next generations of agents are descendants of the agents with the best fitness to the environment conditions. Therefore, we can observe the new maximum rising in the group of agents with 75-80% of correct predictions.

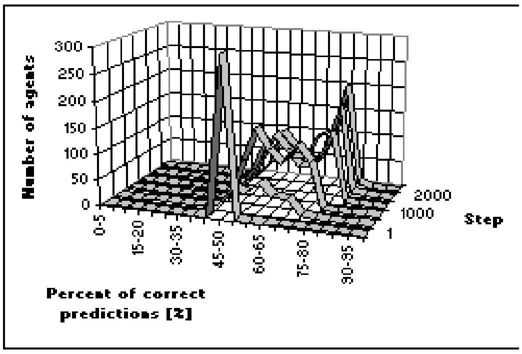

Figure 1 Typical percentage of correct predictions' distribution changes during the simulation. Only *classical* evolution operators.

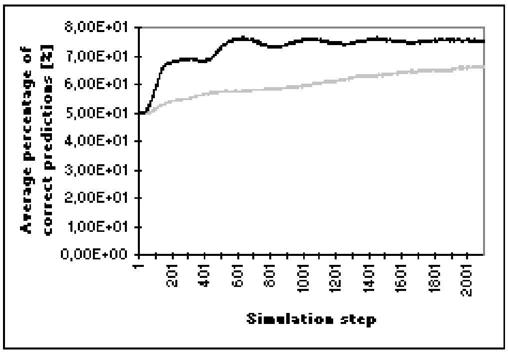

Figure 2 Average percentage of correct predictions. Darker line shows results with *aggregation*.

Figure 2 shows the average percentage of correct predictions during the whole simulation. It can be seen that *aggregation* operator significantly improves results. The average percent of correct predictions is at the level of 75% during almost the whole simulation. When only *classical* evolution operators are activated the average percentage of correct predictions approaches the value of 65%.

The simulations with the *escape* operator were made in order to research the influence of the *escape* evolution operator on the evolution process in prediction system. The maximal number of states of the finite automatons was five. The first type of simulations aimed at comparison between *classical* evolution operators and *classical* evolution operators together with the *escape* operator. There were created three environments. In the main environment the agents' goal was to predict values from the sequence that was eight times longer than the maximal number of states of the finite automatons. In addition to this environment there existed two other environments. In these environments agents had to predict values taken from the sequences, which did splitting the main sequence into two parts create. When the *escape* operator was active the simulation consisted of two parts. The first part of the simulation was the process of species formation. The agents lived in the two environments with the shorter sequences. After some time they were forced to move towards the environment with the main sequence. The results can be seen in the Figure 3. The *escape* operator significantly improved the results of the evolution process. We can observe many more agents with the higher percentage of correct predictions in the population than in the case when there were activated only the *classical* evolution operators.

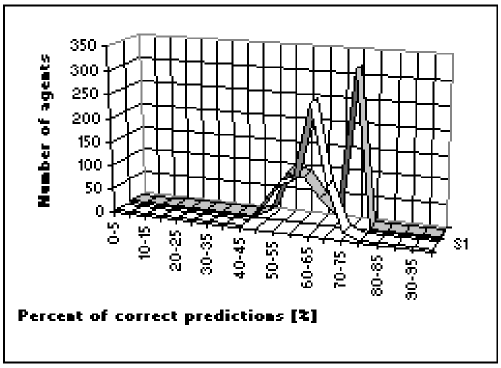

Figure 3 Percent of correct predictions' distribution in the last step of simulation without *aggregation*. Darker line shows results with *escape*.

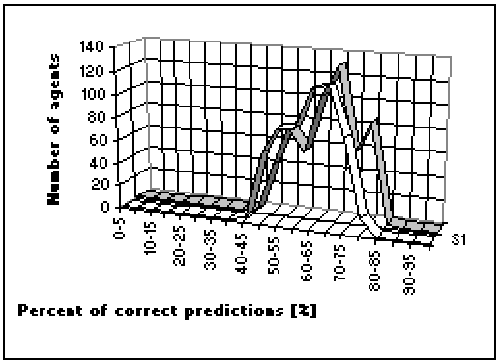

Figure 4 Percent of correct predictions' distribution in the last step of simulation with *aggregation*. Darker line shows results with *escape*.

In the second type of simulations all conditions of the simulation were the same except of that the *aggregation* operator was activated. The *aggregation* was allowed only in the environment with the main sequence. The agents could form the aggregates consisted of two cells. Figure 4 shows the results of these simulations. Also in this case the application of the *escape* evolution operator improved the results of predictions made by the population of agents.

5 Conclusions

In this paper the simulation methodology of the evolution process in multi-agent world has been presented. It is typical to the research in the area of *Evolutionary Multi Agent Systems* (and more general to every research made with the use of computer simulations) that we try to observe and investigate the *emergence* of phenomena, which mathematical model cannot be formulated explicitly. Therefore the appropriate methodology of this kind of research is indispensable. The program of research can be expressed in two main steps:

- formulate models of the system's components (subsystems) and its interactions at the lower level, and
- create the framework in which the subsystems and their interactions can be simulated in order to generate and observe the phenomena of interest.

In our paper we have proposed the model based on the concept of the M-Agent architecture. This model allows us to formalize the properties of the multi-agent system (environment, agents, agent-agent and agent-environment relations). Thus this model can serve as a tool in the first part of research. The second part of research made with the use of computer simulation requires appropriate programming tools. These tools should allow us to obtain reliable and repeatable results of our simulations. In this paper we have proposed the SWARM simulation system for this part of research. This system has all the features, which are indispensable in the *EMAS* and artificial life simulations. We have applied this methodology to the research concerning the emergence of social relations between agents (the *aggregation* operator) and the process of species formation (the *escape* operator). These phenomena were researched in the random 0-1 time sequence prediction system. It has been shown that these two new operators significantly improved the evolution process in the multi-agent prediction system. Future research will be focused on introducing new evolution mechanisms based on the biological evolution, and application of the neural network instead of the finite state automaton. Therefor, it will be possible to research different phenomena that exist in the process of biological evolution.

References

[1] Cetnarowicz E., Cetnarowicz K., and Nawarecki E. Agent oriented technology of decentralized systems based on the M-agent architecture. Submitted to Sao Paulo, 1997.

[2] Cetnarowicz K. M-agent architecture based method of development of multi-agent systems. In *Proc. Of the 8th Joint EPS-APS International Conference on Physics Computing.* ACC Cyfronet, Krakow, Poland, 1996.
[3] Fogel L.J., Owens A.J., and Walsh M.J. *Artificial Intelligence Through Simulated Evolution.* John Wiley, 1966.
[4] Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.
[5] Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer -Verlag, 1996.
[6] Rasmussen S., Barrett C.L. Elements of a theory of simulation. *ECAL'95, Lecture Notes in Computer Science.* Springer-Verlag, 1995.
[7] *Brief Overview of Swarm.* Swarm Development Group, http://www.santafe.edu/projects/swarm.
[8] *Reference Guide for Swarm 1.4.1.* Swarm Development Group, http://www.santafe.edu/projects/swarm.