# The Application of Evolution Process in Multi-Agent World to the Prediction System

**Krzysztof Cetnarowicz, Marek Kisiel-Dorohinicki, Edward Nawarecki**
Institute of Computer Science, University of Mining and Metallurgy
al. Mickiewicza 30
30-059 Krakow, Poland
cetnar@uci.agh.edu.pl

**Abstract**

Software systems become more and more complex thus the application of self-developing distributed and decentralized processing is indispensable. The complexity of such systems requires new tools for designing, programming and debugging processes which implies the fact that new approaches to decentralization should be undertaken. An idea of autonomous agents arises as an extension to the object and process concepts. The active agent is invented as a basic element of which distributed and decentralized systems can be built. The use of evolution strategies in design of multi-agent systems reveals new posibilities of developing complex software systems. The evolution plays also a key role in creation and organization of social structures. In this paper a new technology of designing and building agent systems based on genetic methods and a draft concept of a model-based approach to such systems are described. Also an application of this technology to a self-developing prediction system is presented and results of simulation experiments carried out with the use of 0-1 random time series are discussed.

## Introduction

Genetic algorithms and evolution programming have been known for more than twenty years and the idea of genetic algorithms has been succesfully used as a base for a number of applications (Goldberg 1989), (Michalewicz 1992). However, results of the applications are not sufficiently satisfactory although a number of modifications and improvements have been introduced.

The genetic algorithms and evolution programming, that may be found in the literature ((Goldberg 1989), (Michalewicz 1992), (Koza 1992)), have the following features which limit the development of more complicated systems:

- General common algorithm of selection and new population generation is used which implies that the whole evolution process is centralized.
- Beings taking part in the evolution process are simplified to a gene (or collection of genes) manipulated by the algorithm.
- Only basic evolution operators such as mutation, reproduction or crossover are used in the evolution process.

Therefore they present inconveniences and restrictions as: beings cannot act independently, their perception of the environment is very poor, possibilities of rivalry and competition are limited, social relations are suppressed and many more. Furthermore the use of only such evolution operators as mutation or crossover limits the development of the created algorithm, which looks like tuning to the given conditions rather then creating a new, more complicated one. Even though there are some approaches which try to solve the problem of insufficient evolution operators in genetic programmig ((Koza 1992), (Koza 1994)), in most cases it seems to lead to distorting the logical structure of the environment and makes the whole evolution process difficult to understand.

Evolution process realized in the multi-agent world may be considered as a new approach to the evolution programming, and as such presents the following new possibilities:

- Beings (elements) that participate in the evolution process (agents), environment and relations agent - environment are well defined. ((Cetnarowicz & Nawarecki 1995), (Cetnarowicz & Nawarecki 1993)).
- Evolution process is decentralized and is performed with no common cadence. Agents can act independently and in consequence the social relations in the agents' population may be developed.
- Perception of the environment by the agents and social relations enable the rivalry and competition among agents that assure the decentralized process of the selection of agents.
- Decentralized process of the evolution in multi-agents world enables new operation of algorithm creation.

## The evolution process in the multi-agent world

### Structure and relations

According to the multi-agent world (MAW) definition (Nawarecki & Cetnarowicz 1993), (Cetnarowicz & Nawarecki 1993) the evolution process takes place in the society of agents that remain in a given environment. A state of the environment changes as a result of internal (e.g. agents' activity) or external events (e.g. a new portion of data supplied to the environment).

The environment and the events may be observed by agents, which can execute actions appropriate to the current state of the environment. Every action performed by an agent changes its relation to the environment and other agents. Each agent has to satisfy certain goals during its life. The main goal is always to survive and that is why agents must adapt to current conditions of the environment or try to change them, sometimes cooperating with other ones.

A given problem that is to be solved takes form of such an environment with characteristic features and events that take place in it. The resolution of the problem is obtained as a result of the changes in the environment carried out by a chosen agent, selected group of agents or the whole society of agents.

### Evolution operators in the agent evolution process

From the point of view of a given agent multi-agent world may be cosidered as a relation agent – environment. If this relation is not satisfactory, it may be changed in two ways:

- agent changes itsef - mutation and crossover operations are derived,
- agent changes environment - reproduction, aggregation and escape operations are derived.

Thus the evolution of the world of autonomous agents may be realized by applying one of five following operators (some of them involve a single agent and some need cooperation of a group of agents):

- mutation with respect to a single agent;
- crossover within the chosen group of agents;
- reproduction of a single agent;
- aggregation of the chosen group of agents;
- escape of an agent out of the environment.

Mutation, crossover and reproduction processes in the society of agents have the same form as described in (Fogel et al. 1966), (Goldberg 1989), (Michalewicz 1992). But there is one difference: a given agent individually makes decisions on undertaking any activity (for example, he himself finds partners for the crossover operation) and every mutation, crossover or reproduction process has its own independent cadence of realization.

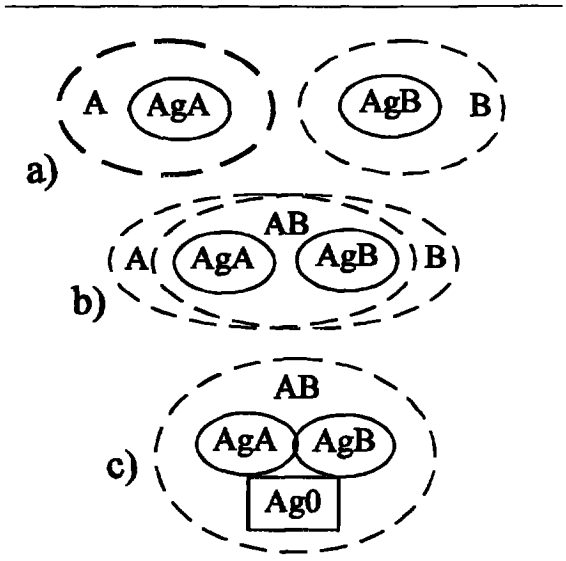


Figure 1: The principle of the aggregation process

The idea of aggregation operation may be considered as a creation of a new environment. The operation of creating a new environment by a group of agents results from the observation that the existing environment is not suitable for these agents. Then a group of agents can make a decision on creating a better environment transforming a part of the one already existing. Cooperating (e.g. by specialization) with one another, the active agents can create a better environment which enables them a more efficient activity. This environment must be created and maintained by the active agents, which in turn, have to act together and behave like a single agent with a new set of characteristic features. The new environment created and the group of agents involved in its creation are considered a new agent originated from the aggregation.

Escape operator makes it possible for an agent to change the environment in which it lives. If the evolution takes place in several environments $E_1, E_2, \ldots E_n$ and each environment has its own characteristic features and different parameters of the evolution process, the agent obtained by mutation, crossover or aggregation that is not well adapted to the environment $E_i$ may go to another environment $E_j$. Then it may start there a new line of descendent agents with valuable characteristic features or return to the former environment.

### The Aggregation Process

The stages of aggregation process (Figure 1) are as follows:

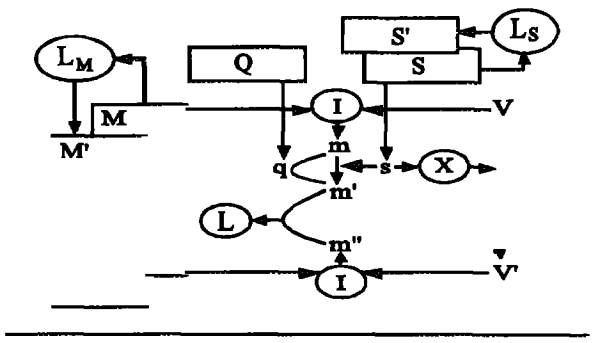


Figure 2: The model of the internal structure of an agent.

1. Let us characterize the environment by two parameters A and B (value of A and B belongs to $\{0, 1\}$). Agent AgA (agent AgB) is able to make the environment parameter A (B) equal to 1 (in its neighborhood). Agents AgA and AgB need the environment with both parameters A and B equal to 1.
2. Owing to the displacement in the environment capability, agent AgA may remain in the close neighborhood of agent AgB and vice versa.
3. Agents AgA and AgB make an arrangement and decide to aggregate together and create a new environment with parameters A and B convenient for them ($A = 1, B = 1$).
4. A new agent Ag0 is created and it stores information of the arrangement of AgA and AgB. The group of agents AgA, AgB, Ag0 and the part of the environment with parameters A and B equal to 1 form a new agent. The agent Ag0 contains (coded in genes) the information necessary for the reproduction of the arrangement (of AgA, AgB, Ag0).

The aggregation process makes it possible for the evolution to realize more complicated algorithms and to create successive new agents, whose population is better adapted to the new environment. It seems that the general principle of intelligent activity of agents in the evolution process consists of decisions properly undertaken by a group of active agents which way of evolution is to be applied. In general it is necessary to decide whether to modify the agent or the environment.

## A concept of the multi-agent system using model based approach

Main definition of the multi-agent system (MAS) using the model based approach to the agent is ((Demazeau & Müller 1993), (Cetnarowicz & Nawarecki 1995)):

- $a$ an agent (a given active element of the MAS), $A$ a set of the agents that remain in the system (actual configuration of agents), $a \in A$,
- $E$ - a space in which agents remain (may be defined by its topology $T$, and resources $R$: $E = (T, R)$),
- $V = (A, E)$ - the whole environment,
- intellectual profile of the autonomous agent: $a = (M, Q, S, I, X, L, m, q, s)$, where:

  $M$ is a set of models representing agent's knowledge about the environment (model configuration), $m$ is an actual model of the environment observed by the agent,

  - $Q$ is a queue (a set with a given order) of agent's goals, $q$ is a goal actually realized by the agent, $q : M \times M \longrightarrow \Re$, $q(m, m') \in \Re$,
  - $S$ is a set of strategies that the agent may consider to perform, $s$ is a strategy to be realized, $s : M \longrightarrow M$, $m' = s(m)$,

    $I$ is the observation operator which with the use of the set $M$ builds the model $m$ of the environment, $I : \mathcal{M} \times \mathcal{V} \longrightarrow M$, $m = I(M, V)$,

    $X$ is the strategy $s$ realization operator, when applied it causes changes in the environment, $X : S \times \mathcal{V} \longrightarrow \mathcal{V}$, $V' = X(s, V)$,
  - $L$ is an adaptation operator which adjusts the agent to the particular characteristic features of the environment by changing sets M and S, $L = \{L_M, L_S\}$, where $L_M$ modifies set $M$ and $L_S$ set $S$.
  - The agent $a$ using observation function $I$ builds the model $m$ of the observed environment. Then it selects the best 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 $s$ is then realized using operator $X$. In the end the agent $a$ using the observation function $I$ builds a new model $m''$ of the changed environment and the process of optimal strategy selection and realization 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 the ability of the agent $a$ to act and survive. Changes of the energetic state $P$ correspond to gains and looses of the agent's energy. In common applications the energetic state is represented by a single real value.

## Sample practical realization of the multi-agent system

### Evolution of the agent predicting system

In this sample version of the evolution model the principal goal of autonomous agents is to predict the changes of the environment. In the environment a binary parameter $\alpha \in \{0, 1\}$ is defined. The value of $\alpha$ is changed by a global event which takes place when new value is supplied for the environment. Variations of the $\alpha$ parameter in discrete moments of time may be represented by the binary sequence $x(n)$ ($x(n)$ is

the value of $\alpha$ after n-th successive event, i.e. n-th successive stage of the process).

Value of the $\alpha$ parameter is available for all agents in the environment Each agent tries to predict what value the parameter $\alpha$ will take after the next event. Thus i-th agent taking $x(n)$ as an input generates as an output the binary sequence $y_i(n) \in \{0,1\}$ in such a way that $x(n+1) = y_i(n)$. The environment calculates an average prediction $y(n)$ based on predictions of all agents and gives it as an output of the whole system.

In the considered system an agent performs the prediction applying a finite state automaton which uses input/output language composed of 0 and 1. For the given automaton the transition and output functions are unique and may vary during the evolution process. The automaton plays a role of a gene in which the information about the algorithm of actions of a given agent is saved (cf. (Fogel et al. 1966), (Michalewicz 1992), (Goldberg 1989)).

There are several factors which are responsible for the estimation of predicton quality for an agent. There are two factors which count hits and misses (good and bad predictions):

$\Phi_i^j$ – connected with predictions of j-th state of the automaton of i-th agent,

$\Phi_i$ connected with successive i-th agent predictions.

There is also the most useful one: $\Psi_i$ prediction probability. It estimates the quality of actual prediction of i-th agent in the scale of probability. It uses $\Phi_i$ and $\Phi_i^j$ (j actual state) compared to defined maximum values as a base for calculations.

Actions performed by an agent depend on the value of its energy. Any action costs some energy and any success gives a corresponding amount of energy. Thus the energy evaluation during the process is defined as:

$$P_i(n) = P_i(0) + \sum_{i=1}^{n} \delta_i(n) \qquad (1)$$

where: $P_i(0)$ – initial energy of i-th agent,
$P_i(n)$ – energy of i-th agent at n-th stage of evolution process,
$\delta_i(n)$ – energy acquired or lost by i-th agent at n-th stage of evolution process:

$$\delta_i(n) = \begin{cases} \delta_i(n) > 0 & \text{when } x(n) = y_i(n) \\ \delta_i(n) < 0 & \text{when } x(n) \neq y_i(n) \text{ or any action is performed} \end{cases}$$

The evolution process involves mutation, reproduction and aggregation operations:

- Mutation operator reverts the output value connected with a given state of agents automaton.
- In the reproduction process a new agent is created by an existing one, with parameters equal or close to its parent.
- The aggregation consists of parallel connection of two or more agents. The prediction of an aggregate is the best prediction of all subagents what makes the aggregate give better results than every subagent itself. The aggregate can be also an object of further evolution.
- The crossover operator can be easily applied by mixing automata of two agents and this way building a new one.

## The model of the predicting agent

The following is the description of the sample prediction system in terms of the model of an agent described above.

**The structure of the environment space**

- The only one resource available for all agents is the value of parameter $\alpha$, $R = \{\alpha\}$.
- The topology $T$ of the environment space is 2D-mesh4 with $5 \times 5$ nodes.

**Intellectual profile of i-th agent** Current model configuration is a set of all subsets of 0-1 sequences in which the beginning $n-1$ elements are the observed values of environment parameter $\alpha$:

$$M(n) = 2^{m(n)}$$

where

$$m(n) = \{\langle x(1), x(2), \ldots, x(n-1), \gamma(n), \gamma(n+1), \ldots\rangle : \forall k \geq n \quad \gamma(k) \in \{0,1\}\}$$

Actual model of the environment is a subset of $m(n)$ consisting of the sequences with all $\gamma(k)$ attainable in $k-n$ successive transitions of the automaton:

$$m_i(n) = \{\langle x(1), x(2), \ldots, x(n-1), \gamma^*(n), \gamma^*(n+1), \ldots\rangle : \forall k \geq n \quad \exists t_i^{k-n}(\gamma^*(k))\}$$

where $t_i^{k-n}(\gamma^*(k))$ stands for $k$–$n$ successive transitions of the i-th agent's automaton leading from the current state to the state with associated prediction of $\gamma^*(k)$.

The goal for an agent is to make a good prediction:

$$q(m_i(n-1), m_i(n)) = \begin{cases} 1 & \text{for } y_i(n) = x(n+1) \\ 0 & \text{for } y_i(n) \neq x(n+1) \end{cases}$$

The set of strategies consists of pairs: $S = \{(s_i, s_e) : s_i$–*intellectual strategy*, $s_e$–*energetic strategy*$\}$. The intellectual strategy is connected with the defined goal $q$. Having applied the operator $X_i$ of $s_i$ strategy realization the model consists of the sequences with n-th element equal to the predicted next value of parameter $\alpha$:

$$m_i'(n) = \{\langle x(1), x(2), \ldots, x(n-1), y_i(n), \gamma^*(n+1), \ldots\rangle : \forall k > n+1 \quad \exists t_i^{k-n}(\gamma^*(k))\}$$

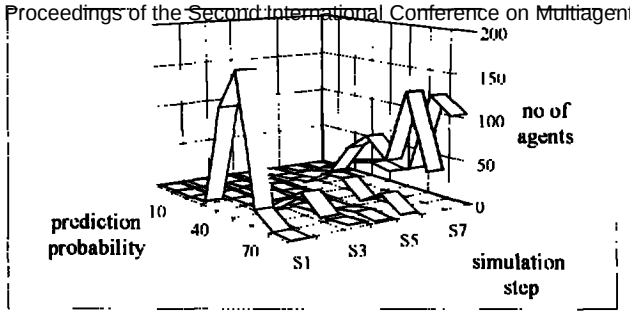


Figure 3: Typical prediction probability distribution changes along simulation

The observation operator $I$ uses aquired from the environment current value of parameter $\alpha$ as input to automaton transition function:

$$m_i''(n) = \{\langle x(1), x(2), \ldots, x(n-1), x(n), \gamma^*(n+1), \ldots\rangle : \forall k > n+1 \quad \exists t_i^{k-n}(\gamma^*(k))\}$$

The obtained model is the initial one for the next stage of the agent's evolution process:

$$m_i(n+1) = m_i''(n)$$

The mutation operator described in the previous section serves as the adaptation operator $L_M$. The set of available strategies $S$ remains unchanged during the simulation therefore there is no $L_S$ operator.

**Energetic profile of the agent** Energetic profile is represented by an integer value P as described in the previous section. Energetic strategy $s_e$ is realized by applying one of the evolution operators (death, reproduction and aggregation) described in details also in the previous section

## The simulation of the evolution of predicting agents

The results of the simulation of the predicting automata are attached as an illustration of the considerations presented above.

The variation of $\alpha$ parameter is generated as a binary sequence of defined length by applying a random generator with uniform distribution. This sequence is periodically repeated giving an infinite input sequence. Autonomous agents are to predict the next following value of the sequence using finite automata. The agents are exposed to the action of the evolution including mutation, reproduction and aggregation procedures. Decisions on which of the evolution processes to undertake are made by every concerned agent upon the state of the energetic and intellectual profile.

The evolution process has been studied with the change of three parameters responsible for the intensity of evolution operations:

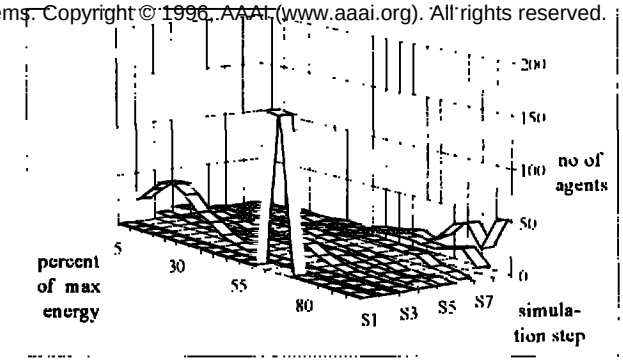


Figure 4: Typical energy distribution changes along simulation

$\Gamma_d$ death intensity,

$\Gamma_r$ reproduction intensity,

$\Gamma_a$ aggregation intensity.

The intensity of mutation did not pertain to this research. Apart from these parameters intensity of all evolution operations depend on the number of agents in the environment in the following way: the more agents are in the environment the easier they die and aggregate but the harder they undergo the reproduction process.

The most important feature of the behaviour of the environment is illustrated by two last figures (Figure 6 and 7). In most cases the number of agents involved in the simulation drops at the very beginning. There is a longer decline. Then it starts to rise and after several slight falls reaches upper limit of number of agents in the environment. This limit is defined to stop the simulation when the agents' predictions are good enough for the environment to give satisfying results. Having reached that state the evolution parameters in the environment should change in order to check the upgrowth of the system. That is why the evoltion intensity parameters depend on the number of agents in the environment. For the same reason there is an upper limit of energy, that can be possessed by an agent, defined as well.

The stages of the system in the evolution may be presented as a histogram. The first two (Figure 4 and 3) show how energy and prediction probability change during a typical simulation. Every agent starts with energy of above half a maximum value and prediction probability of 50%. Agents' predictions cannot be good at the beginning of the simulation because finite automata are initialized at random. It causes a fall in energy and predition probability. Many agents die because of lack of energy and sometimes it leads to a simulation breakdown, especially when the death intensity is too high in comparision with the reproduction one. After this fall a new generation of descendents of the

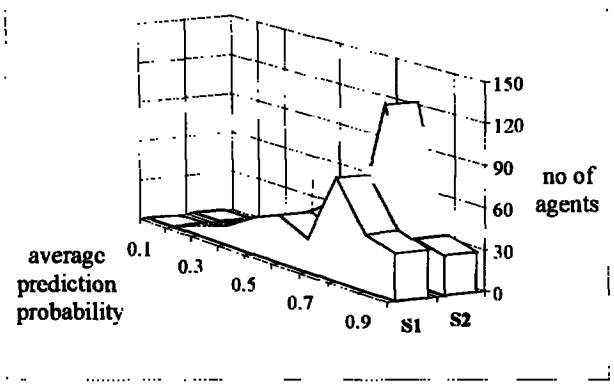


Figure 5: Average prediction probability from 20 simulations with (the higher one) and without (the lower one) the use of aggregation

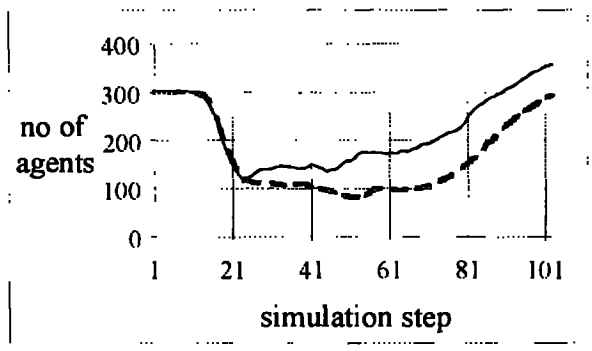


Figure 6: Average number of agents from 20 simulations with (the upper one) and without (the lower one) the use of aggregation

best agents is developed. A more or less linear rise in agents' energy up to the maximum value appears and the number of agents with prediction probability of 80-90% grows. It leads to a balance state in which there are regular slight parameter fluctuations and the predctions of the whole environment are 100% or almost 100% correct.

Figures 5 and 6 account for the thesis of positive influence of aggregation operator on the evolution process. The first graph shows a comparision of the average prediction probability distribution with and without the use of aggregation operator at similar stage of simulation with the same other parameters. The histogram shows that the number of agents with prediction probability of 80% with the use of aggregation is twice as much as without one, whereas the rest of the histograms look quite similar. Furthermore, the average number of agents (Figure 6) during simulations with and without aggregation is the same till the stage

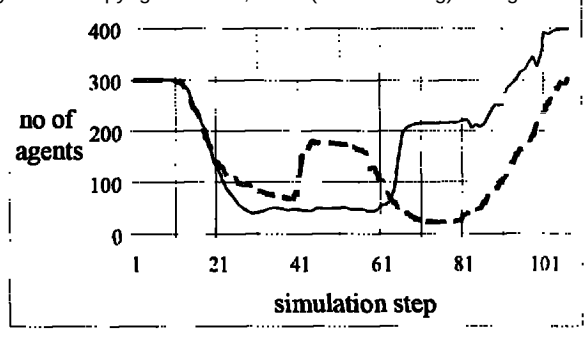


Figure 7: Number of agents during two sample simulations with different parameters

of dropdown and then it rises much quicker in the case of the use of aggregation.

And the last chart (Figure 7) shows how the escape operator could improve the evolution of the multi-agent environment even more. Two plots show the number of agents in two environments simulated with different parameters. Let the escape operator cause agents to move from an environment with more agents to another one. This could balance the actual number of agents in each environment because falls often appear in separate periods of time.

## Conclusions

The following remarks may conclude the presented considerations:

- The application of genetic algorithms to the design of multi-agent systems (MAS) reveals new possibilities of the development of decentralized and distributed systems. It may be also regarded as an extension to the genetic algorithms which enables them to give better results in various applications.
- In such systems it is possible to introduce new operators based on the evolution of the natural world. The use of aggregation operator significantly improves the evolution process of the multi-agent world. It seems that the escape operator may improve it even more.
- The introduction of such group operators as the aggregation one enables creation and development of social relations among agents (the choice between competition and cooperation).
- The application of a general theoretical model of the multi-agent world (MAW) aids the analysis and design of multi-agent systems, also based on genetic methods. Furthermore, it enables the comparison of various agent systems and makes such systems easier to understand and develop.

## References

Cetnarowicz, K., and Nawarecki, E. 1993. Decentralized decision support systems: The evolution of active agent in environment approach. In *Foundations of Computing and Decision Sciences. Institute of Computing Science, University of Technology Poznan, Poland.* volume 19, No 1-2 94, 127–135.

Cetnarowicz, K., and Nawarecki, E. 1995. Système d'exploitation decentralisé realisé à l'aide de systèmes multi-agents. In *Troisième Journées Francophone sur l'Intelligence Artificielle Distribuée et les Systèmes Multiagents*, proceedings. 311 322.

Demazeau, Y., and Müller, J.-P. 1991. From reactive to intentional agents. In Demazeau, Y., and Müller, J.-P., eds., *Decentralized A.I. 2.* North-Holland. 3–10.

Fogel, L., Owens, A., Walsh, M. 1966. *Artificial Intelligence through Simulated Evolution.* John Wiley & Sons Inc., N. Y.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Company, N. Y.

Koza, J. R. 1992. *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* A Bradford Book. The MIT Press, Cambridge, Massachusetts.

Koza, J. R. 1994. *Genetic Programming II. Automatic Discovery of reusable Programs.* A Bradford Book. The MIT Press, Cambridge, Massachusetts.

Michalewicz, Z. 1992. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, Berlin.

Nawarecki, E., and Cetnarowicz, K. 1993. Intelligent decision system with distributed reasoning. In *Proc. of the Eleventh IASTED International Conference Applied Informatics Annecy, France 93* 152 155.