Heterogeneous Behavior Evaluations in Ethically–Social Approach to Security in Multi-agent System*

Gabriel Rojek¹, Renata Cięciwa², and Krzysztof Cetnarowicz³

¹ Department of Computer Science in Industry
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
rojek@agh.edu.pl

² Department of Computer Networks
Nowy Sącz School of Business — National-Louis University
ul. Zielona 27, 33-300 Nowy Sącz, Poland
rcieciwa@wsb-nlu.edu.pl

³ Institute of Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cetnar@agh.edu.pl

Abstract. Ethically–social approach to security problem uses the idea of agents functioning evaluation in a multi-agent system in the analogous way to the evaluation of a person’s behavior in small human societies. This approach involves distributed evaluations made by autonomous agents and processing of the results of this evaluations in order to create a collective decision of a society of agents. Research presented in this paper focuses on the part of domain of ethically–social behavior evaluation that specifies how an agent evaluates the behavior of another agent. The idea of heterogeneous behavior evaluations is formulated and tested in this domain.

1 Introduction

The main idea of security solutions design, that can self-adjust to new and unknown threat, is the use of a model of security mechanisms that function in everyday life and prevent from misusing an individual by others in his environment. Mechanisms inspiring our work are based on the evaluation of observable results of agents’ behavior. In a society every individual observes and evaluates the behavior of all other observable persons. Results of autonomously made behavior evaluations form one decision of the whole society e.g. decision to exclude somebody from a group. Security mechanisms in a society have the decentralized and distributed character — all individuals make their own autonomous evaluations which result in forming one decision of the society.

* This work was partially supported by the Ministry of Education and Science grant no. KBN 3 T08B 042 29.

Having a goal to design security mechanisms in a multi-agent system that are similar to that functioning in human societies, two problems have to be solved. The first problem is the design of evaluations mechanisms which each agent will be equipped with. These mechanisms should enable an agent to evaluate behavior of another agent functioning in a society. Each agent undertakes actions which are perceived in a computer system as objects. That objects registered in a certain period of time create a sequence of actions, that could be processed in order to qualify whether it is a *good* or a *bad* acting agent in the particular system, where evaluation takes place. The second problem is managing, collecting and processing the results of autonomous behavior evaluations in order to state if a particular agent, that is differently evaluated by different agents, is generally *good* or *bad* agent (called also an *intruder*).

The main topic of this paper is the tuning of behavior evaluation mechanisms of an agent to obtain the most infallible decision mechanisms of the whole security system. As it was observed in our earlier works (e.g. [1, 2]) it could incidentally happen that an agent with undesirable behavior was not identified by a society of evaluating agents as *bad* and, as a consequence, this agent was not excluded from that society. On the other hand, it could also happen that a *good* agent could be mistakenly treated as an *intruder*. In order to present stated problems, security mechanisms are presented: in Sect. 2, behavior evaluation mechanisms that are build in every agent in the system and in Sect. 3 mechanisms of management, collecting and processing the results of behavior evaluations. In Sect. 4 the influence of some coefficient's value of behavior evaluation mechanisms is analyzed on the decision of the whole group of agents. In Sect. 5 the idea of heterogeneous behavior evaluation is presented, tested and discussed. The main conclusions of this paper are stated in Sect. 6.

2 The Division Profile

All security mechanisms in which every agent is equipped are named the *division profile*. An agent that posses the division profile can evaluate behavior of an other agent that is visible in an environment of a multi-agent system. Every agent in a secured multi-agent system possesses his division profile, what means that in the system there are made many autonomous behavior evaluation.

Algorithms of the division profile are inspired by immunological mechanisms of T cells generation which enable to detect some anomalies. Usage of immunological mechanisms is restricted and the aim of our work is not to present an artificial immune system. In a case of behavior evaluation, immunological intruders detection mechanisms have to operate on observed actions made by evaluated agent. This approach is opposite to the one proposed in e.g. [3, 4] in which immunological mechanisms operate on the structure of resources. An other difference between artificial immunology and ethically-social approach is the autonomy of a process (an agent) in a secured system. In artificial immunology approach one detection system is considered for a particular computer system (or sub-system). In ethically-social approach every agent uses his own instantiation of

detection mechanisms autonomously, what induces the necessity of application of some additional algorithms in order to agree collective decision of all agents.

According to immunological mechanisms of T cells generation the division profile has 3 stages of functioning: creation of collection of *good* (*self*) sequences of actions, generation of detectors and behavior evaluation stage.

2.1 Collection of Good Sequences of Actions

In order to use mechanisms of intruder detection with an immunological approach, the collection of *good* sequences of objects representing undertaken actions has to be specified. The collection of *good* sequences of actions corresponds to the collection of *self* elements in an artificial immunology system. In ethically-social approach to security problem it is possible to assume that an agent evaluates his own behavior as *good*. Taking into consideration this assumption, the collection W of *good* sequences of actions of an agent consists of sequences of actions undertaken by this agent. The length of a sequence is fixed to l . Presuming h last actions undertaken by every agent are stored, his own collection W will contain $h - l + 1$ elements.

In order to generate the collection W an agent should collect information representing actions undertaken by him in the past. But, on the other hand, an agent in order to evaluate behavior of another agent has to collect information representing actions undertaken by the evaluated agent. So an agent should have information about all actions made in the system. This information is stored in the table of actions. In our work every agent stores last h actions of every visible agent in his table of actions.

2.2 Generation of Detectors

Detectors are used to evaluate behavior in the analogous way to the functioning of T-lymphocytes in the immune system. In presented work detectors of an agent are generated only once in the whole time of existence of this agent. Generation of detectors of an agent happens, when this agent "knows" his last h actions for the first time, so when this agent has undertaken h actions (collection W has to be completed before detectors' generation).

The algorithm of detectors' generation uses the negative selection — from set R_0 of generated sequences of length l those matching with any sequence from collection W are rejected. Set R_0 contains every possible sequence (but it is also possible to use a set of sequences generated at random). The length of a detector is equal to the length of a sequence from collection W . Sequence matching means that elements of those sequences are the same. Sequences from set R_0 which will pass such a negative selection create a set of detectors R .

2.3 Behavior Evaluation Stage

Once detectors of an agent have been generated, this agent can evaluate behavior of another agent. The result of behavior evaluation process of an evaluating agent a is a coefficient attributed to an evaluated agent k . This coefficient marked as m_a^k is a number of counted matches between:

- detectors of the agent a which evaluates behavior,
- sequences of actions undertaken by the agent k (this sequences of actions are taken from table of actions of the agent a).

Marking the length of a detector as l and the number of stored actions as h , the coefficient m_a^k is a number from a range $\langle 0, h - l + 1 \rangle$. The maximum of counted matches is equal $h - l + 1$, because every fragment of sequence of actions, which has length equal to the length of a detector, can match only one detector.

3 Mechanisms of Distributed Evaluations Agreement

An algorithm of agents evaluations managing, collecting and processing is used to agree one common decision of all agents which assess behavior evaluations. The difficulty of this agreement is caused by the fact that an agent can be differently evaluated by different agents in the system. This problem is presented in [5]. Only information essential to discuss the main topic of this article are presented in this section. In presented research the power function of behavior evaluation is used.

Each action undertaken by an agent may cause change of the results of behavior evaluations that are done by other agents in the system. This approach lets us formulate *the algorithm of evaluation management* as follows: If agent k undertakes an action, a request of the evaluation of agent k is sent to all agents (except agent k) by the environment.

Agent a in case of receiving a request of evaluation of an agent number k sends back only the coefficient o_a^k in the range $0 \leq o_a^k \leq 1$. The coefficient o_a^k is given by function:

$$o_a^k = \left(\frac{m_a^k}{h - l + 1} \right)^4 \quad (1)$$

where $h - l + 1$ is the maximum of counted matches of agent a . The power function of evaluation behavior increases the weight of high coefficient m_a^k (the exponent was set empirically).

In order to decide if agent k is in general *good* or *bad*, the environment uses *the algorithm of evaluation collecting and processing*, which consists of following actions:

1. All results of behavior evaluations are stored and, are sent by agents in response to the request of evaluation of the agent k .
2. Gained coefficients are summed and then this sum is divided by $j - 1$ (j is the number of agents):

$$o_*^k = \frac{o_1^k + o_2^k + \dots + o_{k-1}^k + o_{k+1}^k + \dots + o_{j-1}^k + o_j^k}{j - 1} \quad (2)$$

If o_*^k is greater than $\frac{1}{2}$ agent k is eliminated.

4 Tuning of the Division Profile

In this section our aim is to present the influence of the main parameters of behavior evaluation mechanisms on the process of evaluation. Parameters of functioning of division profile are: l — the length of detectors, h — the number of actions stored in table of actions. The quality of ethically-social behavior evaluations has two dimensions: the speed of intruders removing and the rate of self-destruction phenomenon (the number of *good* agents mistakenly evaluated as intruders).

An experimental multi-agent system was implemented in order to discuss the influence of main parameters of division profile on the quality of evaluation process. Two types of resources are in the simulated environment: resources of type A and resources of type B. Resources are used by agents, but refilling of all resources is only possible when each type of resources reaches the established low level. This specific refilling of resources reflects the situation in real computer system with some operations which must be executed in couples, for example: opening and closing a file, connection request and disconnection request. There are a lot of attack techniques that are limited to only one from a couples (or trios...) of obligatory operations (e.g. SYN flood attack [6]). The simulated system has three types of agents:

- 50/50 agents — agents which take one unit of randomly selected (A-50%, B-50%) resource in every full life cycle;
- 90/10 agents — agents which take one unit of randomly selected (A-90%, B-10%) resource in every full life cycle; type 90/10 agents can be treated as intruders, because the increased probability of undertaking only actions of one type can block the system;
- 100/0 agents — agents which take one unit of A resource in every full life cycle; 100/0 agents are also called intruders.

Actions of 90/10 agents are to certain degree similar to actions of 50/50 agents but they are also undesirable in the secured system. The division between 50/50 agents and 90/10 agents is hindered by random character of agents decision which resource select. It could happen that a *good* 50/50 agent takes (much) more units of one resources than units of another resource in certain time period.

4.1 The Self-destruction Phenomenon

A multi-agent system with initial 90 agents was run in order to research the self-destruction phenomenon. All of the agents are 50/50 *good* agents. Measure of the self-destruction phenomenon is the percentage of agents remained in this system after 1000 constant time periods Δt . Because results could differ in every run of the simulation (random character of agent's decision making), the simulation was run 10 times. Average of obtained results are presented in Table 1.

The results for length of detector $l \geq 6$ for number of stored actions $h \leq 20$ indicate that near all agents are deleted because of the self-destruction phenomenon. This dependence is caused by the fact that for so long detectors the

Table 1. Percentage of agents removed after 1000 constant time periods Δt in consequence of self-destruction phenomenon with the coincidence of parameters h — number of stored actions and l — length of detectors

-	$h = 16$	$h = 17$	$h = 18$	$h = 19$	$h = 20$	...	$h = 27$	$h = 28$	$h = 29$
$l = 4$	0.22%	0.22%	0.22%	0.00%	0.00%	...	0.00%	0.00%	0.00%
$l = 5$	17.78%	7.11%	2.78%	1.33%	0.44%	...	0.00%	0.00%	0.00%
$l = 6$	97.78%	97.78%	7.78%	97.78%	97.78%	...	0.89%	0.22%	0.22%
$l = 7$	97.78%	97.78%	97.78%	97.78%	97.78%	...	97.78%	97.78%	97.78%
$l = 8$	97.78%	97.78%	97.78%	97.78%	97.78%	...	97.78%	97.78%	97.78%

information stored in the table of actions was not precise enough to generate usable set of detectors. In process of generation of detectors a negative selection is used as it is presented in Sect. 2.2. In the case of not enough information about actions done in the past (to small parameter h), not enough elements are excluded from the set R_0 and detectors become elements of the set R_0 which should be excluded in process of negative selection. Agent a possessing such big set of detectors evaluates all kinds of behavior with great coefficients o_a which means evaluating all kind of behavior as *bad* and as a result self-destruction of agents. However, in the case of $l = 5$ should be stored $h \geq 18$ actions in order to restrict the self-destruction phenomenon, in the case of $l = 6$ the table of actions has to be extended to $h \geq 27$ in order to ensure information to detectors' generation process. Conclusions presented in this paragraph are confirmed by experimental measured number of detectors with the coincidence of parameters h and l (results not presented in this paper).

4.2 The Rate of Intruders Detection

In order to measure the speed of intruder detection a multi-agent system was simulated. At the beginning of each simulation 90 *good* 50/50 agents were in the system. In 50-th constant time periods Δt one intruder was put into the system. The rate of intruders detection is the time needed to remove this agent. As obtained results of experiments show the removing of 90/10 agents is more problematic than the removing of 100/0 agents. 100/0 agents are removed faster and more faultlessly, so only results of the removing of 90/10 agents are presented as more sensitive to parameters tuning. Obtained results are presented in Table 2 (the simulation was run 10 times for each presented case).

Taking into consideration results presented in Table 1 and in Table 2, it could be stated that for length of detectors $l = 5$ the optimal number of actions stored in the table of actions is $h = 18$, however greater number of stored actions involves restriction of the rate of the self-destruction phenomenon. This dependency can be explained by the fact that more information about behavior (more undertaken actions) involves more accurate behavior evaluation. On the other hand more information required to evaluate a behavior (e.g. $h = 19$) needs more time to obtain this information what involves longer time needed to detect the intruder.

Table 2. Results of an 90/10 intruders detection in the form of: (number of simulation which an intruder was removed, average time needed to removing of an intruder in constant time periods Δt) with the coincidence of parameters h — number of stored actions and l — length of detectors

-	$h = 16$	$h = 17$	$h = 18$	$h = 19$	$h = 20$	...	$h = 27$	$h = 28$	$h = 29$
$l = 4$	(10, 31)	(10, 49)	(8, 58)	(10, 69)	(6, 67)	...	(1, 84.0)	(0, -)	(0, -)
$l = 5$	(10, 19)	(10, 20)	(10, 22)	(10, 28)	(10, 32)	...	(10, 105)	(9, 139)	(9, 120)
$l = 6$	(10, 16)	(10, 15)	(10, 17)	(10, 17)	(10, 18)	...	(10, 32)	(10, 35)	(10, 38)
$l = 7$	(10, 16)	(10, 15)	(10, 16)	(10, 18)	(10, 18)	...	(10, 25)	(10, 26)	(10, 31)
$l = 8$	(10, 16)	(10, 15)	(10, 16)	(10, 17)	(10, 18)	...	(10, 22)	(10, 23)	(10, 27)

In order to obtain more precise evaluation of behavior (small rate of the self-destruction phenomenon) longer detectors ($l = 6$) have to be used. Using detectors that have 6 object representing actions ($l = 6$) involves the necessity of increasing the number of stored actions to 28 or greater. Regarding the time needed to detect an intruder for $l = 6$ it could be stated that optimal size of table of actions is $h = 28$.

5 Heterogeneous Behavior Evaluations

Conclusions gathered in Sect. 4.2 indicate that in order to get precise detection of intruders the number of stored actions has to be extended, which results in longer time of detection of an intruder. In all the tests done in order to show this dependency, the agents have the same parameters (l , h) of behavior evaluation, but in a multi-agent system it is possible to set up different agents with different parameters. This idea is named *heterogeneous behavior evaluations*. Our aim of stating this idea is to use "slow" agents ($l = 6$, $h = 28$) to restrict the rate of the self-destruction phenomenon and simultaneously to use "rapid" agents ($l = 5$, $h = 18$) to speed up detection of an intruder. In order to test the idea of heterogeneous behavior evaluations, experiments are done that are analogous to presented in Sect. 4.1 and Sect. 4.2. Obtained results are gathered in Table 3.

Table 3. Results of experiments with heterogeneous acting agents

-	Self-destruction phenomenon	An intruder detection
30% "slow", 70% "rapid"	1.67%	(10, 24)
50% "slow", 50% "rapid"	0.60%	(10, 26)
70% "slow", 30% "rapid"	0.56%	(10, 28)

Results gathered in Table 3 in comparison with the results presented in Table 1 and Table 2 indicate that heterogeneous evaluations are desirable in ethically-social approach to security. Implementation of the proposed idea enables to obtain evaluation possibilities which unite merits of "slow" and "rapid" agents

without manifestation of disadvantages of this two agents' groups. Analyzing results for 50%/50% case the time needed to detect of an intruder is approximate to time of detection for agents with $l = 5$ detectors ("rapid" agents which perform more rapid evaluation of behavior), but the rate of the self-destruction phenomenon is rather specific for agents with $l = 6$ detectors ("slow" agents which perform more precise evaluation of behavior).

6 Conclusion

Dependencies between parameters of behavior evaluation process and the quality of behavior evaluation (rates of intruders detection and the self-destruction phenomenon) have been presented in this paper. The main conclusion is that in order to obtain precise detection great amount of information about agents' behavior have to be used, which involves using long detectors ($l = 6$) and storing great number of actions performed in the past (the size of table of actions $h = 28$). On the other hand, using shorter detectors ($l = 5$) and smaller tables of actions ($h = 18$) causes acceleration of the process of intruder detection, however, this process is not so precise as using longer detectors ($h = 6$, $l = 28$).

Distributed character of behavior evaluation performed by agents in multi-agent system led us to the idea of heterogeneous behavior evaluations. In this approach different groups of agents in the secured system are set up with different parameters. Performed tests indicate the usefulness of this idea. Obtained possibilities of evaluation due to heterogeneity demonstrate mutual advantages of using short ($l = 5$) and long ($l = 6$) detectors.

References

1. Cetnarowicz, K., Cięciwa, R., Rojek, G. (2005) Behavior Evaluation with Actions' Sampling in Multi-agent System, In *Lecture Notes in Artificial Intelligence*, Vol. 3690, Springer-Verlag Berlin Heidelberg, 490–499
2. Cetnarowicz, K., Cięciwa, R., Rojek, G. (2005) Behavior Evaluation with Earlier Results Collection in Multi-agent System, In *Preprint Proceedings of The Agent Days 2005*, Malaga, 7-8 July 2005, 77–84
3. Forrest, S., Perelson, A. S., Allen L., Cherukuri R. (1994) Self-nonsel Self Discrimination in a Computer. In *Proc. of the 1994 IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society Press, Los Alamitos, 202–212
4. Hofmeyr, S. A., Forrest, S. (2002) Architecture for an Artificial Immune System. *Evolutionary Computation*, vol. 7, no. 1, 45–68
5. Rojek, G., Cięciwa, R., Cetnarowicz, K. (2005) Algorithm of Behavior Evaluation in Multi-agent System, In *Lecture Notes in Computer Science*, Vol. 3516, Springer-Verlag Berlin Heidelberg, 711–718
6. Schetina E., Green K., Carlson J. (2002) *Bezpieczeństwo w sieci*. Wydawnictwo HELION, Gliwice