Algorithm of Behavior Evaluation 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. Behavior based detection of unfavorable activities in multi-agent systems (presented in [3, 4, 5]) is an approach to the problem of detection of intruders. This approach refers to evaluation of behavior of every agent which exists in a multi-agent system. Process of behavior evaluation is distributed – every agent makes autonomous behavior evaluation of other agents. This means that an agent is evaluated separately by all agents in the environment of the secured system. That separate results of behavior evaluations have to be collected and an algorithm should be used in order to elect the *worst* agents which should be eliminated. Collecting and data processing of results of distributed behavior evaluations is the main topic of this article.

1 Introduction

Behavior based detection of unfavorable activities in multi-agent systems is inspired by ethically-social mechanisms that act in human societies. An individual in a society seems trustworthy if its behavior could be observed by others and this behavior is evaluated by majority as good and secure. The decision about trustworthy of an individual takes place in society in the decentralized and distributed way – all individuals in a society make own decisions which form one decision of the society.

Inspiration by ethically-social mechanisms in computer security systems induce decentralization of security mechanisms which should be based on observation and evaluation of behavior of agent functioning in a secured system. Actions undertaken by agents are perceived as objects, which create a sequence registered by all agents in the environment. Registered objects-actions could be processed

in order to qualify whether it is a *good* or a *bad* acting agent in this particular system, in which evaluation takes place. A *bad* agent also could be named *intruder*.

2 Division Profile

Decentralization of security mechanisms is realized in multi-agent systems by means of equipping all agents with some additional goals, tasks and mechanisms. Those goals, tasks and mechanisms are named *division profile*. The name *division profile* is inspired by M-agent architecture which could be used to describe an agent (M-agent architecture was introduced among others in [1, 2]).

Description of division profile was presented in [3, 4, 5]. This article contains only some information that is crucial to the problem of collecting and data processing of the results of distributed behavior evaluations.

Each agent in a multi-agent system has his own autonomous calculated division profile. Division profile of an agent has three stages of functioning: creation of collection of *good* (*self*) sequences of actions, generation of detector set, behavior evaluation. An agent a , which division profile is at his behavior evaluation stage, has division state m_a represented as a vector:

$$m_a = (m_a^1, m_a^2, \dots, m_a^{j-1}, m_a^j) \quad (1)$$

where j is the number of neighboring agents (neighboring agents are agents which are visible for agent a) and m_a^k is the coefficient assigned to neighboring agent number k . Coefficient m_a^k indicates whether the agent number k is evaluated by agent a as *good* or *bad*. Coefficient m_a^k is a number of counted matches between:

- detectors of agent a which evaluates behavior and possesses division state m_a ,
- sequence of actions undertaken by agent number k .

Marking the length of a detector as l and the length of sequence of 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 a length equal to the length of a detector, can match only one detector.

3 Problem of Distributed Behavior Evaluation

In order to choose an agent which should be removed from the system, division states of all agents should be collected and the algorithm of data processing should be used. This problem was solved in our earlier work, but presented solutions seem to be not sufficient in order to obtain a flexible and self-adopting method.

Simulations that in every constant time period Δt each agent in the system executes his full life cycle are presented in [3, 5]. An exemplary agent a calculates

his division state m_a in every life cycle and chooses agent (or agents) number k , that $m_a^k = \max(m_a^1, m_a^2, \dots, m_a^{j-1}, m_a^j)$. Then agent a send a demand of deleting agent (or agents) number k to the environment. To this demand there is coefficient o_a^k equal to the m_a^k attributed.

The environment calculates the sum of coefficients and liquidates an agent (or agents) number n which fulfills two requirements:

- $o_*^n = \max(o_*^1, o_*^2, \dots, o_*^{j-1}, o_*^j)$, where $o_*^p = o_1^p + o_2^p + \dots + o_{j-1}^p + o_j^p$ ($1 \leq p \leq j$),
- $o_*^n > OU$, where OU is constant.

Periodically, after a constant time period Δt , the calculated sums of coefficients are set to 0. Constant coefficient OU is introduced in order to get tolerance for behavior that is evaluated as *bad* in a short time, or is evaluated as *bad* by a small amount of agents.

Presented in [4] simulations are more adequate to a real-world multi-agent system. In that simulations all actions of agents are asynchronous. In each constant time period Δt number of activated agents can be different. Used asynchronous model of agent systems forced some modification in behavior based detections of intruders, particular in algorithm of data processing of behavior evaluations. If an agent number k tries to undertake any action, the environment asks neighboring agents for their "opinion" about him.

Exemplary agent a in a case of receiving a request (from the environment) of evaluation of an agent number k calculates his division state m_a and sends:

- coefficient o_a^k equal to coefficient m_a^k ,
- additional information:
 - **true** if agent a evaluates agent number k as *the worst* ($m_a^k = \max(m_a^1, m_a^2, \dots, m_a^{j-1}, m_a^j)$),
 - **false** in other cases.

The environment sums these coefficients and eliminates an agent (or agents) number n which fulfills two requirements:

- $o_*^n > OU$, where OU is constant and $o_*^p = o_1^p + o_2^p + \dots + o_{p-1}^p + o_{p+1}^p + \dots + o_{j-1}^p + o_j^p$ ($1 \leq p \leq j$),
- more than 50 per cent of agents evaluate agent number n as *the worst*.

Liquidation of an agent only on the base of the sum of coefficients seemed to be not possible in that research.

It can be noticed that in mentioned solutions there is problem of setting the constant OU . The constant OU was set empirically after series of tests. The constant OU have to be reset in every new type of test. Another inconvenience of presented solution in asynchronous model of simulation is the need to calculate the whole division state m_a (all coefficients assigned to all agents), when evaluated is only one agent number k . In order to reduce computational complexity of behavior based detection of unfavorable activities and make this security solutions more flexible some ideas are presented and tested in the next section.

4 Algorithms of Distributed Behavior Evaluation – Propositions and Experiments

To reduce number of operations undertaken by each agent, which are connected with behavior evaluation, a modification of algorithm of evaluation process was proposed. An agent a in a case of receiving a request of evaluation of an agent number k sends back only the coefficient o_a^k , where $o_a^k = f_{eval}(m_a^k)$. An agent a do not have to calculate the whole division state m_a , but only the coefficient m_a^k . The environment sums gained coefficients. If final sum of all received coefficients o_*^k is larger than $1/2 * j$ agent k is eliminated (j is the number of agents). The function $f_{eval}()$ can be called function of evaluation which should be selected in order to assure removing intruders and leaving good agents.

In order to confirm effectiveness of proposed solutions and to select proper function of evaluation, a multi-agent system with asynchronously acting agents was implemented. In the simulated environment there exist two types of resources: resources of type A and resources of type B. This situation reflect these operations in computer system which should be executed in couples e.g. opening / closing a file. Resources are used by agents, but refilling all resources is only possible when each type of resources reach the established low level. The simulated system has three types of agents:

- *type $g=0$* – agents which take one unit of randomly selected (A-50%, B-50%) resource in every full life cycle;
- *type $g=1$* – agents which take one unit of randomly selected (A-75%, B-25%) resource in every full life cycle; type $g=1$ agents can be treated as intruders, because increased probability of undertaking only actions of one type can cause blocking the system (what is presented in [3, 4]);
- *type $g=2$* – agents which take one unit of A resource in every full life cycle; type $g=2$ agents are also called intruders.

We have simulated the case in which initially there are 64 agents of type $g=0$, 8 agents of type $g=1$ and 8 agents of type $g=2$. All agents in the system are equipped with the division profile mechanisms with parameters $h = 18$ and $l = 5$. The simulations are run to 2000 constant time periods Δt and 10 simulations were performed.

4.1 Linear Function of Evaluation

First a case was simulated in which all agents use linear function for behavior evaluation. An agent a sends back to the environment the coefficient o_a^k in the range $0 \leq o_a^k \leq 1$. This coefficient is given by

$$o_a^k = \frac{m_a^k}{h - l + 1} \quad (2)$$

where $h - l + 1$ is the maximum of counted matches of agent a .

Fig. 1. Number of agents in separate time periods, agents using linear function of evaluation

Diagram in Fig. 1 shows the average numbers of agents in separate time periods. This diagram shows only 100 time periods Δt , during next time periods number of agents remains at the same level.

After series of tests it turned out that intruders were eliminated as well as good agents. Analysis of obtained results indicates that most of agents were deleting successively from 19 constant time period Δt to 28 constant time period Δt . The results of presented simulation enforced a reconfiguration of agents' behavior evaluation algorithm.

Example of three agents: to find a reason for rapidly deleting all agents we have chosen randomly:

- agent number 0 – an agent of type $g = 0$,
- agent number 1 – an agent of type $g = 1$,
- agent number 2 – an agent of type $g = 2$.

The final sum of coefficient gained during their evaluation process was analyzed. Average amount of coefficients returned to the environment is shown on diagram in Fig. 2. The agent number 2 obtains only minimal or maximal values of coefficient. The most of coefficients returned to the environment during evaluation process of agents number 0 and 1 are larger than 0.5. The final sum of coefficients is $o_*^0 = 48.6$, $o_*^1 = 49.7$ and $o_*^2 = 51.0$. The final sum of coefficients for all three exemplary agents is above condition $1/2 * j$ (j is equal to 80 at the time of this evaluation). As a result, agent number 0 is eliminated as well as agent 1 and 2 – intruders.

4.2 Discrete Function of Evaluation

On diagram in Fig. 2 we observe that amount of returned coefficient for both agents number 1 and number 2 reaches maximal value for $o_a^k = 1$. Thus we have

Fig. 2. Average amount of coefficient returned during evaluation process of exemplary three agents

simulated a case in which all coefficients o_a^k , sent to the environment, have value from the set $\{0,1\}$. This coefficient is given by

$$o_a^k = \begin{cases} 1, & \text{if } m_a^k = h - l + 1 \\ 0, & \text{if } m_a^k \neq h - l + 1 \end{cases} \quad (3)$$

Diagram in Fig. 3 shows the average number of agents in separate time periods. All good agents have remained in the system, bad agents were deleting successively from 19 constant time period Δt to 28 constant time period Δt , agents of type $g=1$ were eliminated only in 71%. Obtained results indicate that discrete function of evaluation can be inefficient for agents of type $g=1$, because, as we can see in Fig.2, they are evaluated also in a way that does not indicate the maximal value of coefficients in division states.

4.3 Power Function of Evaluation

In order to increase a weight of high coefficients we have simulated also a case in which all agents use the power function for behavior evaluation. An agent in a case of receiving a request (from the environment) of evaluation of an agent k sends back to the environment the coefficient o_a^k in the range $0 \leq o_a^k \leq 1$. The coefficient o_a^k is given by

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

Diagram in Fig. 4 shows the average number of agents in separate time periods. All bad agents were deleting successively from 19 constant time period Δt

Fig. 3. Number of agents in separate time periods, agents using discrete function of evaluation

Fig. 4. Number of agents in separate time periods, agents using power function of evaluation

to 28 constant time period Δt , the agents of type $g=1$ were eliminated in 99% and good agents were also eliminated in 8%.

Randomly selection of resources cause that in a particular situation actions undertaken by a good agent could be considered as undesirable for a system. In the environment, with agents using power function of evaluation, such agent is immediately eliminated.

5 Conclusion and Further Research

In this paper the problem of collecting and data processing of distributed behavior evaluations in the multi-agent system was discussed. Three functions of evaluation: linear, discrete and power were proposed and researched. Proposed mechanisms in algorithm of behavior evaluation allow to reduce computational complexity of earlier algorithms presented in [3, 4, 5].

In order to confirm the effectiveness of proposed conceptions a multiagent system with agents acting asynchronously was implemented. The results obtained in Sect. 4.1 demonstrate that using the linear function for behavior evaluation causes elimination of intruders as well as good agents. However results obtained in Sect. 4.2 and Sect. 4.3 indicate that the proposed methods detect an abnormal behavior of agents in the system. Discrete function for behavior evaluation can be useful in case of application in which it should be sure that any good agent is not removed, but elimination of all intruders is not crucial. Power function of evaluation should be used in application in which elimination (or only indication) of all agents that can be considered as undesirable is crucial.

Further research in behavior based detection of unfavorable activities in multiagent systems should include methods for precise recognizing and eliminating all intruders from the system. One of the goals is to extend algorithms of distributed behavior evaluation with mechanisms, which makes it possible to take into account also the final sum of coefficients obtained during earlier live cycles of an agent. Such algorithms could let us avoid situations, in which good agents are eliminated because of short-term deterioration of their actions.

References

1. Cetnarowicz, K.: M-agent architecture based method of development of multiagent systems. Proc. of the 8th Joint EPS-APS International Conference on Physics Computing, ACC Cyfronet, Kraków (1996)
2. Cetnarowicz, K., Nawarecki, E., Żabińska, M.: M-agent Architecture and its Application to the Agent Oriented Technology. Proc. of the DAIMAS'97, St. Petersburg (1997)
3. Cetnarowicz, K., Rojek, G.: Unfavourable Behaviour Detection with the Immunological Approach. Proceedings of the XXVth International Autumn Colloquium ASIS 2003, MARQ, Ostrava (2003) 41–46
4. Cetnarowicz, K., Cieciva, R., Rojek, G.: Behavior Based Detection of Unfavorable Activities in Multi-Agent Systems. In MCPL, Conference on Management and Control of Production and Logistics, Santiago - Chile, (2004) 325–330
5. Cetnarowicz, K., Rojek, G.: Behavior Based Detection of Unfavorable Resources, in: Lecture Notes in Computer Science, Proceedings of Computational Science - ICCS 2004: 4th International Conference, Springer-Verlag, Heidelberg (2004) 607–614