Autonomous Agent's Behavior Model Based on Automata Theory

Valentin Filatov[1], Valerii Semenets[2]

1. Artificial Intelligence Department, Kharkiv National University of Radio Electronics, UKRAINE, Kharkiv, Nauki Ave. 14., e-mail: valentin.filatov@nure.ua
2. Department of Metrology and Technical Expertise, Kharkiv National University of Radio Electronics, UKRAINE, Kharkiv, Nauki Ave. 14., e-mail: valery.semenets@nure.ua

***Abstract*:** This paper is devoted to a problem of management the corporate system's information space. The management is carried out on the technology of the program agents. The approach to construction the program agent's model on a basis of frame structure is considered. The frame is fixed as universal extension system. For the decision of concrete tasks the functional modules - slots can be added in it. Base structure for model of behavior of the program agents in information environment is the model based on the finite-state machine. The offered models can be used for development the systems for administration of information resources in the allocated computing systems.

***Keywords*:** information system, database, frame, program agent, autonomous agent, automata theory, multi-agent system.

I. INTRODUCTION

The main trends in the development of information systems and distributed databases is, on the one hand, the decentralization and distribution of information management resources, and on the other hand, the heterogeneity of the data storage structures of information systems. Thus, distributed automated systems become the basis of the information structure of data storage, processing and management, and provide the society informatization - the creation of a single information space. Requirements for the performance of distributed automated systems increase with the expansion of regional, national and international integration processes in economy and business.

There is a significant number of works, studies, models, methods and software products, which solve various problems of creating integrated information systems with different efficiency and create opportunities for building intelligent systems based on the technology of software agents.

Research performed by S.J. Russell, P. Norvig, M.N. Huhns, N.R. Jennings, H.S. Nwana, A.R. Simon, P. Wegner, M. Wooldridge etc. is the most recognized in this area. Promising solutions include the development of intelligent systems for the integration and management of information resources based on the technology of software agents [1].

II. PURPOSE OF RESEARCH

The idea of managing information systems and resources with the help of standalone dispatcher programs according to the scenario is not new. Later, real-time dispatcher programs supported distributed computing in mini-computer networks. Now this idea is the basis of the technology of program agents with mobile behavior scenarios implemented in one of the programming languages.

There are a lot of definitions of the concept “agent” depending on the view of distributed processing of knowledge. From the point of view of distributed computing, the agent is a separate process that runs in parallel; it has a certain state and is able to interact with other agents through the transmission of messages. In this sense, it can be considered as a natural development of the paradigm of object-oriented parallel programming. Essentially, program agents perform tasks delegated to them by users. Depending on the specific task, agents can be hosted on personal computers of different purposes up to Web servers. From the point of view of object-oriented programming, a program agent can be viewed as a technology for sharing distributed knowledge and functions. Each agent is a process possessing a certain part of knowledge about the object with the possibility to exchange this knowledge with other agents [2].

III. CURRENT STATE OF AGENT TECHNOLOGIES

In the modern information space, there are intelligent agents that operate in isolation and perform tasks in accordance with specified scenarios.

Currently, there are the following types of intelligent agent architectures:

- Simple Reflex Agents;
- Model-Based Reflex Agents;
- Goal-Based Agents;
- Utility-Based Agents.

Multi-agent systems are increasingly used in information technology. Agents are able to find each other in cybernetic space on the basis of technologies of interaction of program agents that function as coordinators or centers of information exchange, accumulating data about other agents and their functions [3].

Multi-agent systems are characterized by local autonomy, social interaction, adaptability, robustness, and scalability. These are key properties of complex distributed systems: Game playing, Optimization logistics and production processes, Pattern recognition, Computer vision, Speech recognition, Intelligent control, Data mining.

The agent system is a paradigm that takes inspiration from several disciplines, mainly from distributed artificial intelligence. That is related to study and model systems possessing life, i.e. capable of reproducing, surviving and adapting in hostile environments.

Multi-agent systems are based on a society of distributed autonomous, cooperative entities, each one having a proper role, knowledge and skills, and a local view of the world, being its behavior regulated by simple rules. Agent-based solutions replace the centralized, rigid and monolithic control by a distributed functioning where the interactions among individuals lead to the emergence of intelligent global behavior [4].

There are a number of obstacles to the large-scale deployment of multi-agent systems in the information space. The task of harmonizing standards that ensure the joint work of agents created by different developers should be singled out among the priority problems.

IV. DEVELOPMENT OF THE PROGRAM AGENT BEHAVIOR MODEL

Let us consider one approach to constructing the behavior model of the program agent in accordance with the basic requirements imposed on the properties of program agents - autonomy of functioning and the ability to perform expedient actions. A model based on the theory of an automaton can be considered as a basic structure for constructing the model of behavior of program agents in the information environment, [5,6].

$$\varphi(t+1) = \Phi[\varphi(t), Q(t+1)] \quad (1)$$

$$f(t) = F[\varphi(t)] \quad (2)$$

Equation (1) describes the dependence of the change in the states of the automaton under the influence of the input variable $Q(t)$, and equation (2) describes the action of the automaton $f(t)$ depending on its states. The input variable has only two values and specifies a pair of mappings of the automaton state sets into itself. One of these mappings is specified for $Q=0$, the other is for $Q=1$.

These mappings are stored in the form of a matrix of states $\left\|\alpha_{ij}(Q_l)\right\|, (i, j = 1,2,..., n)$. To determine the problem of the behavior of an automaton, it is necessary to give a characteristic of the medium with which the automaton interacts.

The automaton U is in a random stationary environment $C = C(a_1, a_2, ..., a_x,)$ if the actions of the automaton and the values of its input variable are related as follows: the action $f_\alpha, \alpha = 1,2,..x$ produced by the automaton at time t entails the loss Q = 0 at the time t + 1 with probability $p_\alpha = (1 - a_\alpha)/2$ or the gain Q = 1 with probability $p_\alpha = (1 + a_\alpha)/2$. An automaton with a sufficiently large set of states (memory capacity) possesses an appropriate behavior in the medium C [7,8].

The logical model of a program agent in the form of a frame allows one to apply effectively the theory of automata to construct and study a behavior model. In general, such a model can be written as follows:

$$FR < R_1, C_{11,} >, ... < R_2, C_{21} >, < R_{k1}, C_{km} > \quad (3)$$

The slot in the model (3) is a logical construction for the implementation of specific tasks to the frame-program agent. Slots from the frame can be deleted, added, and the functionality of the job slot can be changed. However, the classical representation of a slot in the form <name>, <value> cannot fully reflect the requirements of the logical model of the program agent in the form (3).

Let us modify the structure of the slot and bring it to the following form:

$$\mathbf{Slot} = \langle \mathbf{Y}, \mathbf{D}, \mathbf{dom}, r_i, \Theta, \Sigma \rangle, \quad (4)$$

where Y – is the set of attribute names, D – is the set of domains, *dom* – is the mapping of $Y \Rightarrow D$, r_i – is the model-tuple of the i-th task of the agent, Σ – is the set of operations on relations. $r_i = \{\{R\}_i\}$, where R_i – is the set of states of the tuple r_i, Θ – is the set that determines the initial conditions and signs of performing actions in the structure of the job. The slot can be designed using a typical set of attributes:

$$Y = \{< OBG >, < ACT >, < CON >, < STA >\}.$$

Description of the possible basic types is given in Table 1.

Table 1

Basic types of attributes

Entity	Entity name	Description
OBJECT	OBG	Database, file, folder, disk, PC
ACTION	ACT	Copy, monitor, protect
CONDITION	CON	IF- THEN, predicate
STATUS	STA	1 – the action on the object was successful, 0 – action failed, * – result of action is uncertainty

In the proposed "frame-program agent" model, the n-slots form a finite number of internal states of the agent $\varphi_i (i = 1,2,...n)$ Each slot solves one task in the information environment and has one specific action $f_i (l = 1,2,...n)$, i.e. one action corresponds to one agent state. For each state, the environment responds with a response signal $Q(t)$. Then the behavior of the program agent can be given by equations of the form (1) and (2).

Equation (1) determines the change in the internal states of the program agent under the influence of the input variable $Q(t)$, and equation (2) is the dependence of the agent's output action on its internal state [9].

Let us assume that the input variable $Q(t)$ in the general case can take three values of “0”, “1” and “uncertainty”. The value of Q = 1 will correspond to the successful completion of the action of the program agent at the time $t = t^*$ to the action of the agent $f(t^*)$. The value Q = 0 – corresponds to the unsuccessful action, and the action in the maintenance or execution mode, will correspond to the state Q = “uncertainty”.

The variable $\varphi(t)$ may take different values $\varphi_1, \varphi_2, ..., \varphi_n$. These values will be called program agent states; n – is its information capacity. Values of a variable $f(t)$ will be called actions of the program agent. Then we can state the following: at the time *t* the program agent A_g is in the *j*-th state $j = 1,2,...n$, if $\varphi(t) = \varphi_j$. The action f_j is called an action corresponding to the state φ_j, $F(\varphi_j) = f_j$. Equation (2) describes the dependence of the actions of the program agent on its states, equation (1) – describes changes in its states under the influence of the input variable. The transfer function of the program agent can be specified by the matrix system. Matrix $\|\alpha_{ij}(Q_l)\|$, $(i, j = 1,2,..., n)$ corresponds to each value Q_l, this matrix determines the change of the states of the program agent under the influence of the signal Q_l.

The advantage of the logical representation of the program agent model is that it can be combined with a state matrix and, thus, reflect the current state and the transfer function of the program agent, depending on the values of the input variable $Q(t)$.

V. Conclusion

The problem of constructing a model of autonomous behavior of a program agent for solving information management tasks in the computing system is considered. The formal approach to constructing the behavior model of program agents based on the theory of automata is substantiated. The proposed approach to constructing the behavior model of a program agent can be effectively used to develop multi-agent systems for solving the problems of managing and monitoring information resources of distributed computing systems.

References

[1] M. Wooldridge, “An Introduction to MultiAgent Systems”, John Wiley & Sons, 2009.

[2] Stuart J. Russell, Peter Norvig, “Artificial Intelligence: A Modern Approach”, 2nd ed., Upper Saddle River, New Jersey, Prentice Hall, 2003.

[3] V. Filatov, “About one approach to the classification of program agents”, *Proceedings of International Conference Modern Problems of Radio Engineering, Telecommunications and Computer Science* (*TCSET*), 2006.

[4] В.А. Филатов, Е.Е. Цыбульник, Л.Э. Чалая, “Модель мультиагентной системы автономного администрирования информационных систем и распределенных баз данных”, Новости искусственного интеллекта, 2002, № 4, с. 620-627.

[5] В.А. Филатов, В.А. Кривоносов, О.Ф. Козырь, “Адаптивные автономные сценарии в задачах управления информационными ресурсами предприятия”, Инженерный вестник Дона, 2013, № 3.

[6] В.А. Филатов, “Модель поведения автономного агента на основе теории автоматов”, Вестник Херсонского государственного технического университета, Херсон: ХГТУ, 2004, № 1 (19), с.108 - 111.

[7] L. Dennis, M. Fisher, N. Lincoln, A. Lisitsa, S.Veres, “Practical verification of decision-making in agent-based autonomous systems”, Autom. Softw. Eng., 2014, pp. 1-55.

[8] G. De Giacomo, and M. Vardi, “Automata-theoretic approach to planning for temporally extended goals”, Lecture notes in computer science, 2000, pp. 226-238.

[9] G. Martius, K. Fiedler, and J. M. Herrmann, “Structure from Behavior in Autonomous Agents”, in *IEEE International Conference on Intelligent Robots and Systems*, 2008, pp. 858-862.