**PRIKASPIYSKIY ZHURNAL: Upravlenie i Vysokie Tekhnologii
(CASPIAN JOURNAL: Management and High Technologies), 2014, 4 (28)
THEORETICAL BASES, DEVELOPMENT AND APPLICATION
OF MATHEMATICAL MODELS, ALGORITHMS, COMPUTER PROGRAMS**

УДК 004.4'242

VISUAL PROGRAMMING METHODS IN DEVELOPMENT OF INTELLECTUAL MULTI-AGENT SYSTEMS GENERATION PLATFORM

Статья поступила редакцию 02.12.2014, в окончательном варианте 17.12.2014.

Kravets Andrey D., post-graduate student, Volgograd State Technical University, 28 Lenin Avenue, Volgograd, 400005, Russian Federation, e-mail: kad@gde.ru

During the research authors performed an analysis of a current state in intellectual multi-agent systems field. Also authors analyzed the most known and large platforms and libraries for agent-oriented software engineering. The analysis highlighted the major drawbacks of modern approaches to the process of multi-agent systems development. Following article contains methods and concept usage of which will organize a process of user and platform interaction in more efficient way. Authors propose realization of the conceptual requirements by means of visual programming methods implementation. That approach allows involvement of subject area expert to the project thus improving compliance of a system performance expected by the expert and practical system functionality. During an expanding of visual programming interface authors stated a goal of implementation of rapid prototyping technology by development of previously created agent types database. Rapid prototyping allows obtaining the systems samples at early stages of the project thus increasing a number of iterations of development. By means of sharpening agents adjustments on every iteration expert achieves a goal of the most adequate respond of the system in design during its functioning in subject area.

Keywords: multi-agent systems, system generation, intellectual methods, visual programming, component-based approach

МЕТОДЫ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ В РАЗРАБОТКЕ ПЛАТФОРМЫ ГЕНЕРАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ МУЛЬТИАГЕНТНЫХ СИСТЕМ

Кравец Андрей Дмитриевич, аспирант, Волгоградский государственный технический университет, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. им. В.И. Ленина, 28, e-mail: kad@gde.ru

В ходе проведенного исследования выявлены недостатки современных подходов к процессу проектирования интеллектуальных мультиагентных систем (ИМС). В данной статье рассмотрены методы и концепт, согласно которым должно осуществляться взаимодействие пользователя и проектируемой платформы генерации ИМС. В качестве базовой методики реализации данного концепта предложено использование подхода на основе визуального программирования. Визуальное программирование позволит привлечь к разработке проекта эксперта предметной области, что позитивно отразится на соответствии функционала проектируемой ИМС ожидаемому результату. Такой подход также реализует основы технологии быстрого прототипирования путем применения компонентно-ориентированной архитектуры, что позволит увеличить количество итераций проектирования ИМС и провести более точную настройку агентов системы. Результатом этих изменений является более адекватный отклик от системы в целом в ходе ее работы в целевой среде. В статье представлена практическая реализация интерфейса платформы генерации и ее функционал; приведены результаты экспериментального внедрения платформы.

Ключевые слова: мультиагентные системы, генерация систем, интеллектуальные методы, визуальное программирование, компонентно-ориентированный подход

Introduction. A reason to study and develop a multi-agent system comes from a common necessity to organize cooperative work of a set of software and hardware agents, such as software agents performing information aggregation and analysis functionality throughout the Internet. In this case the Internet acts as a major platform for collaboration between distributed computing units equipped with mechanisms of self-motivation and education. At the moment coverage of theoretical material on multi-agent systems spreads widely on informational technologies mainly in artificial intelligence and distributed calculations. Also a term of multi-agent systems is being used in economics on macro economical level, operations research, analytic philosophy, psychology and linguistics. A sphere of practical usage expands along with the volume of theoretical knowledge. Appliance of multi-agent approach is a wide and rapidly evolving field in software engineering with development rate of about 40 % during the last three years. Expanding of the approach is more apparent in combination with distributed computing technology which covers about 80 % of all software applications on multi-agent basis. Nowadays informational multi-agent systems (MAS) are presented in technics, medicine, agriculture and economics [13].

The goal of the research is providing to expert of ability to develop and change intellectual multi-agent systems by means of MAS generation platform. The goal is achievable by survey of existing methods of user interface design and implementation of these methods in the platform functionality.

Analysis of multi-agent systems development platforms and libraries. During an explanation of the term «agent» it is a common practice to use a concept of object developed in theory of object-oriented engineering. In this case a component-based approach opens new perspectives of programming and design of MAS. Then an artificial agent represents a meta-object vested with some kind of subjectivity, i.e. the object possesses an ability to manipulate other objects, create and destroy them. Also such object must have effective means of recognition and interaction with environment and other objects of its own kind. That means that such «active object», or «artificial actor», functions at a higher level of complexity in comparison to traditional objects in object-oriented engineering and uses these objects in order to achieve its own goals, changing their state and performance by means of management. So a basic set of characteristics in common understanding of «agent» includes such attributes as activity, an ability to organize and realize actions; autonomy, a relative independence from environment or presence of «freedom of will», such behavior has to be supported with sufficient amount of resources; sociability, an attribute grown from necessity to achieve its goals in collaboration with other agents and supplied with efficient communication protocols; purposefulness, which assumes presence of proper origins in motivation, broadly it means development of special intentional characteristics for an agent. Today informational technologies specialists are inventing new approaches to development of intellectual agents' behavior algorithms. A great number of them are based on visual programming which proved to be effective and comfortable for end-user. Yet these technologies are not quite enough developed and researched. Choice of multi-agent technology as a base concept in design of distributed computing systems allows combination of universal protocols with private means of database managing and monitoring in sole system in very easy and effective way. At a stage of design such system obtains flexibility, horizontal and vertical extensibility, and simplifies solution of problems with distribution of load between servers. By means of using intellectual methods in an agent structure it is possible to achieve the most adequate, actual and optimal result of agent performance [14].

Today's market offers a large number of systems and libraries used to work with multi-agent systems. Ideas of software agents in general and intellectual agents in particular draw attention of developers due to an ability to delegate authority in solving complex tasks. Yet development of MAS and actual intellectual agents requires special knowledge and is a complex and resourceful task by itself, since software agents are a new class of software systems which acts in the same way as a user. Thus they become a powerful abstraction in a task of «visualizing» and structuring of more complex systems. But in comparison to procedures, functions, methods, classes and other well-known abstractions which are used by software developers daily, software agents are a principally new paradigm unknown to the most of developers even today [11].

During the research authors analyzed the most known and large platforms and libraries of multi-agent systems development. Amongst them are: multi-agent programming environment NetLogo, multi-agent development library JADE, multi-agent library core MASON. These systems can be used in order to create a multi-agents system of any structure of agents and their collaboration scheme. But organization using multi-agent system has to spend a great amount of resources in a case of including new or changing existing structures and algorithms performed in order to satisfy users' needs and support an adequate level of environmental response restarting a development of MAS almost from scratch [1]. Table 1 shows results of comparative analysis of the systems on different criteria.

Table 1

Comparative analysis of system analogues

System	**Criteria**								
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
AgentTool	3	2	4	1	2	1	4	5	2
Magenta Toolkit	1	4	5	1	3	1	4	1	2
The Multi-Agent Systems Lab	1	2	4	1	2	1	3	1	2
CogniTAO	5	4	1	3	5	3	3	4	5

Criteria of comparison: a) ability to integrate with intellectual methods and algorithms; b) rate of changes implementation in agent structure; c) rate of changing of inter-agent collaboration structure; d) end-user qualification as an information technologies specialist; e) expandability of multi-agent system; F) reusability of previously created structures; g) ability to integrate with standard protocols of data exchange; h) independence of presence of additional software; I) cross-platform ability/.

The systems were estimated on scale from «1» to «5», where «1» stands for the least inherent attribute and «5» – for the most inherent one.

According to the analysis authors composed following list of the major drawbacks MAS development process [3].

1) Requirement of information technologies specialist with high qualification level as a developer. For now a process of multi-agent system development requires an unstoppable interaction of a developer and an expert. Yet some steps vital for the whole project can be performed without acknowledgement of the expert. A great example of such step is a selection of development tools set which critically influences structure of the system in design and the development process in general. Also a vast number of multi-agent systems avoid using of complex agent structure that includes intellectual methods and behavioral algorithms since a question of their inclusion in the project is solved by a developer.

2) Inability of some systems to integrate intellectual methods in agents' structure. Despite the fact that some systems succeeded in development of behavioral algorithms on its platform, every analyzed system is unable to implement intellectual methods that are not used in management – such as data analysis, predictive analysis or pattern recognition.

Methods of generator interface design. During the preliminary stages of the research project, our goal was identified as aiming to provide software with improved usability to increase experts' involvement in the development of MAS project. Modern systems exploit graphical interfaces of visual language class in order to provide a greater usability for users without IT developers' qualification [5].

Graphical interfaces fall within one of two categories: visual programming (VP) or program visualization (PV), both of which are subclasses of visual languages. VP interfaces allow the creation of programs from graphical elements representing specific static-code procedures, whereas

PV interfaces permit graphical viewing of arbitrary (though finite in range) procedures within a program without any capability to change the program. By such a definition, the multi-agent generation platform would technically be referred to as a VP system, since the «only» qualifier in the PV definition eliminates any other possibility. That is, a VP system can have some PV aspects, such as graphical debugging and simulation, and yet still be considered VP. However, it is unclear that such definitions were meant to apply to a system with extensive use of both PV and VP techniques, as would be needed in a reconfigurable system. The research therefore states a need for a new class of visual language which extensively incorporates both PV and VP techniques. Such a hybrid visual programming language would permit the graphical creation of programs (as in VP) from visualized pre-existing conventional code (as in PV). The pre-existing conventional code could be adjusted external to the system, and subsequent system sessions would then reflect these changes using PV techniques. The code could then be configured interactively using VP techniques. Figure 1 represents concept of VP interface functioning.

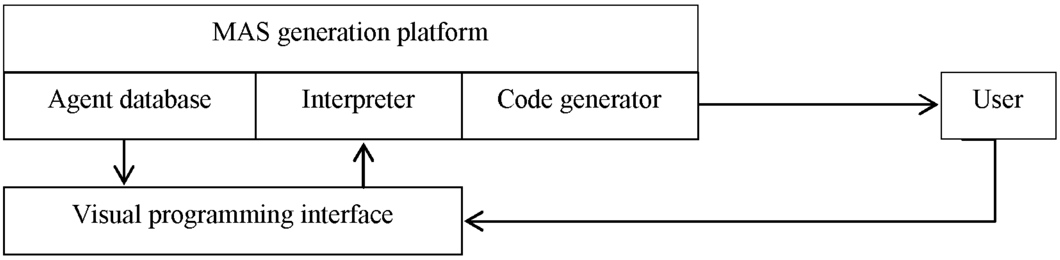

Fig. 1. Concept of VP interface

Respectively to the task the graphical interface must provide a subject area expert with an ability to perform following activities by means of the platform in process of MAS development:

1) design of an agents collaboration structure;
2) setting of necessary adjustments for agents behavior in the system;
3) generation of a system in prototype and productive versions;
4) simulation of MAS functionality and aggregation of its results.

Development of visual programming interface. Authors chose «boxes and arrows» concept as a basis model of visual language of VP interface for MAS generator. In the case «boxes» represent agents of MAS and «arrows» represent connections between agents [5, 10]. Both those elements form inter-agent collaboration structure. In order to supply the interface with types of figures an agent database was created. Agent database is a part of generation platform which consists of agents and connections types. In the database each agent type acquires following attributes: a set of input parameters; an intellectual method realizing agents' functional and behavioral performance; a set of agents' settings which concretize behavior of each generated agent; a set of output parameters. By compiling of intellectual method together with key adjustments and settings which supposed to represent a subject area knowledge of an expert allows platform generates multi-agent system code.

Along with the database methods of VP allow provides an ability to develop MAS to an expert and realizes an idea of rapid system prototyping on a multi-agent basis. Prototyping is the process of developing a trial version of a system (a prototype) or its components in order to clarify the requirements of the system or to reveal critical design considerations. Prototyping can give an expert a chance to «test drive» software to ensure that it is, in fact, what the user needs. Alternatively, engineers may utilize prototyping to improve their understanding of the technical demands upon, and the consequent feasibility of a proposed system. The use of prototyping has been recommended as a way of correcting weaknesses of the traditional «waterfall» software development life cycle, by clarifying important system requirements to the developer, before a full system is implemented.

Expert has an opportunity to interact with the prototype, and give direct feedback to the platform [4, 12]. Sometimes users are not sure that they want certain functions implemented until they actually can try them. Further, the need for certain features may not be apparent until actual use exposes an omission or inconvenience. Users may also find certain features or terminology confusing. Thus it is logical that prototyping tends to help ensure that the first implementation (after the prototype) will meet user's needs, especially when the prototype includes the user interface. That is, the first attempt at developing a system will likely fail to meet user needs, and be discarded. It is better that the first effort be a prototype rather than a final deliverable. That approach provides opportunity to get an actual working prototype of the system rapidly and to test its functionality in subject area. Obtaining a prototype of generated system provides an ability to perform agent adjustment in a very accurate way and to achieve the most adequate agent functioning, thus it positively affects MAS in general.

Figure 2 represents sequences appearing during the platform functioning.

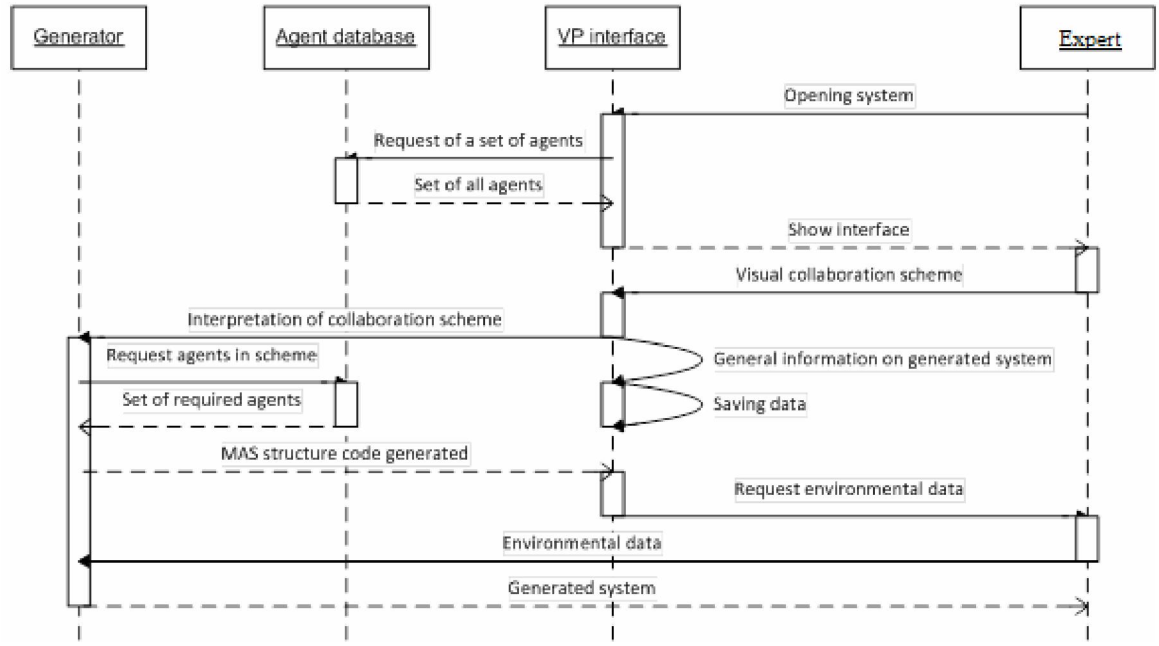
Generator
Agent database
VP interface
Expert
Opening system
Request of a set of agents
Set of all agents
Show interface
Visual collaboration scheme
Interpretation of collaboration scheme
Request agents in scheme
General information on generated system
Set of required agents
Saving data
MAS structure code generated
Request environmental data
Environmental data
Generated system

Fig. 2. Sequence diagram

MAS generation platform interface. Result of the described development is an interface of the platform which provides all the abilities stated earlier to the expert.

The interface combines methods of visual programming with a component-based architecture realized through implementation of agent database [9]. Figure 3 represents the interface.

At figure 3 numerals indicate following objects:

1) list of agents from agent database;

2) work area is used to construct an inter-agent collaboration scheme using agents and connections;

3) functional buttons are used to interact with the work area of the module (that group includes «Add agent», «Save», «Add connection», «Generate» buttons);

4) system mode buttons. These buttons are used to switch between modes of system functioning, such as agent testing, agent settings or visual programming module.

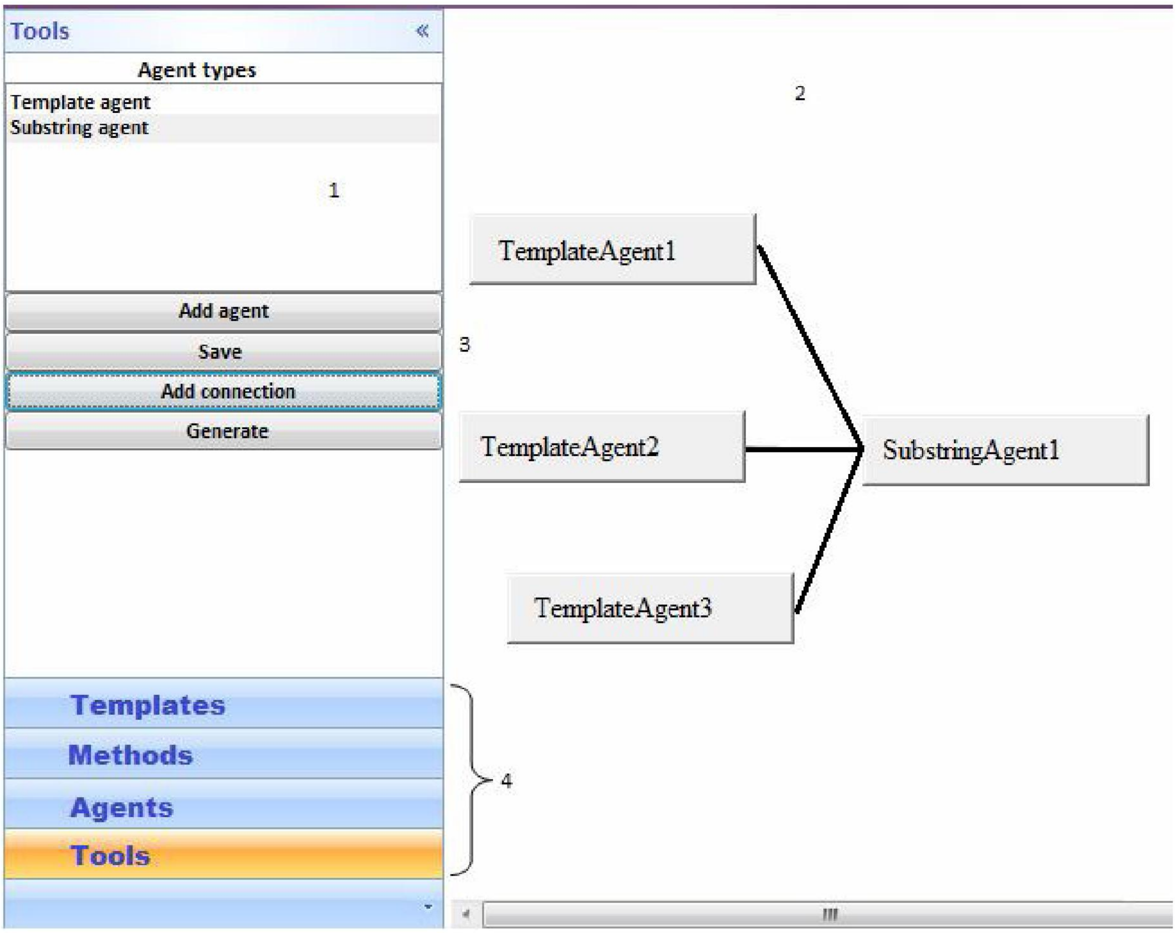

Fig. 3. Platform interface

In order to use the interface user can perform following actions:

1) click type of agents and “Add agent” button to add an agent to the work area;

2) mark two agents on the work area and click “Add connection” button to create a collaboration connection between agents;

3) click “Save” button to save the collaboration structure;

4) click “Generate” button to generate a productive version of MAS;

5) mark an agent on the work area and click “Methods” button to choose a preferable behavioral algorithm for the agent;

6) mark an agent on the work area and click “Agents” button to perform adjustments of the agent;

7) click «Templates» button to switch to the simulation mode of the platform.

The platform have passed the experimental introduction for the development of 3 multi-agent systems: collection and intelligent analysis of data about nanotechnologies MAS for nanotechnology databank [6, 8]; sensor data gathering MAS for real-time maintenance and repair equipment support system [2]; multi-agent module in CRM-system «TOUREAST: CRM AI» for tourist enterprise [7]. The results of experimental implementation are presented in Table 2 and Figure 4.

Table 2

The results of experimental implementation

System name	Multi-agent module development time (hours)			Client usage (users number)			Management costs (development costs, thousand rubles RUR)		
	before	*after*	*decrease on, %*	*before*	*after*	*increase on, %*	*before*	*after*	*decrease on, %*
Collection and intelligent analysis of data about nanotechnologies MAS	16	10	38	18	21	17	24.0	15.6	35
Maintenance and equipment repair support MAS	23	10	57	56	62	11	115.0	98.0	15
Multi-agent CRM-system of tourist enterprise	7	4.5	36	34	43	26	76.0	56.2	26

The enterprises participated in the implementation have had another instance of multi-agent system or module implemented already. Thus data on development time, average number of users, and management and development costs per month is adequate to the research and evaluable.

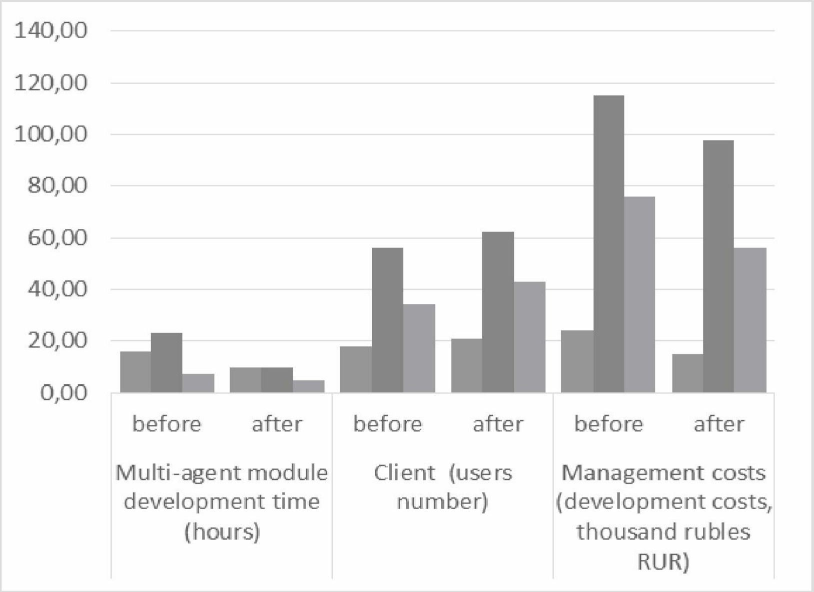

Fig. 4. Experimental implementation results

We achieve improvement criteria namely:

1) through the use of previously developed components and intellectual methods, multi-agents modules development time is ranging from 36 to 57 % of previously spent;

2) the implementation of rapid prototyping technique has allowed to obtain agents, the most adequate and easily accomplishing their goals in a changing environment, which results in the elimination of 11 to 26 % loss of user clients generated multi-agent modules;

3) the expansion of the expert role in the development via visual programming interface has allowed to make changes directly to the generated functional modules from the early stages, reducing management costs for businesses ranged from 15 to 35 %.

The main result of the current research is a development of methods allowing usage of sets of previously developed agent types in multi-agent systems design. It was proven to render a positive effect on a development process increasing its efficiency and decreasing losses in process management.

Conclusion. During the research a visual programming interface of multi-agent generation platform was developed. The interface realizes a concept based on an analysis of multi-agent development systems and study of modern approaches to the visual programming. The system provides an expert with an ability to perform a direct development of multi-agent system via visual programming interface. Rapid prototyping basis of the system allows to get systems prototype at early stages of development. The prototype is proven to be the most useful in question of involvement non-engineer user in the project providing a graphic way of showing drawbacks of system in development. That approach positively affects MAS respond to environment and increases an adequacy of its performance.

Список литературы

1. Будыльский А. В. Управление проектами разработки программного обеспечения с использованием агентных технологий / А. В. Будыльский, И. Ю. Квятковская // Прикаспийский журнал: управление и высокие технологии. – 2013. – № 3(23). – С. 119–128.

2. Кизим А. В. Генерация интеллектуальных агентов для задач поддержки технического обслуживания и ремонта / Кизим А. В., Кравец А. Д., Кравец А. Г. // Известия Томского политехнического университета. – 2012. – С. 131–134.

3. Кравец А. Г. Генератор агентов мультиагентной системы сбора данных о перспективных технологиях / А. Г. Кравец, С. В. Шевченко, А. Д. Кравец // Віснік Національного технічного університету «ХПІ». Збірник наукових праць.Тематичний випуск «Системний аналіз, управління та інформаційні технології». – 29'2012. – С. 92–97.

4. Кравец А. Д. Генерация интеллектуальных агентов / А. Д. Кравец, Д. А. Абрамичев // Конференция-конкурс научных, конструкторских и технологических работ студентов ВолгГТУ : тез. докл. / под ред. В. И. Лысак. – Волгоград : ВолгГТУ, 2012. – С. 45–48.

5. Кравец А. Д. Проектирование генератора интеллектуальных мультиагентных систем / А. Д. Кравец, А. Г. Кравец, С. А. Фоменков // Современные проблемы и пути их решения в науке, транспорте, производстве и образовании'2012 : сборник научных трудов Sworld по материалам междунар. науч.-практ. конф. – 2012. – Т. 13, № 4. – С. 42–46.

6. Кравец А. Д. Разработка модели генерации интеллектуальных агентов / А. Д. Кравец, С. А. Фоменков, А. Г. Кравец // Сборник научных трудов SWORLD. – 2012. – Т. 5, № 3. – С. 59–61.

7. Романенко Е. В. Некоторые вопросы проектирования и реализации распределенной информационной системы "TOUREAST: CRM AI" / Е. В. Романенко, А. Г. Кравец // Прикаспийский журнал: управление и высокие технологии. – 2013 – № 4 (24). – С. 165–175.

8. Умаров А. С. Некоторые аспекты создания информационных систем для сбора и хранения научной и наукометрической информации / А. С. Умаров, Н. В. Попова, В. А. Зелепухина // Прикаспийский журнал: управление и высокие технологии. – 2013. – № 3 (23). – С. 111–118.

9. Korotkov A. A Intelligent Multi-Agent Systems Generation / A. A. Korotkov, A. G. Kravets, A. D. Kravets // World Applied Sciences Journal (WASJ). – 2013. – Vol. 24, spec. issue 24 : Information Technologies in Modern Industry, Education & Society. – Pp. 98–104.

10. Kravets A. G., Component-Based Approach to Multi-Agent System Generation / A. G. Kravets, A. D. Kravets, S. A. Fomenkov // Knowledge-Based Software Engineering : Proceedings of 11th Joint Conference, JCKBSE 2014 (Volgograd, Russia, September 17–20, 2014) / ed. by A. Kravets, M. Shcherbakov, M. Kultsova, Tadashi Iijima. – 2014. – Vol. 46. Series: Communications in Computer and Information Science. – Pp. 483–490.

11. Kravets A. G. Corporate intellectual capital management: learning environment method / A. G. Kravets, A. S. Gurtyakov, A. D. Kravets // Multi Conference on Computer Science and Information Systems 2013 (Prague, Czech Republic, July 23–26, 2013) : Proceedings of the IADIS International Conference ICT, Society and Human Beings 2013 and IADIS International Conference e-Commerce 2013. – Prague : International Association for Development of the Information Society, 2013. – Pp. 3–10.

12. Kravets A. G. Multi-agent systems component-based generator: development approach / A. G. Kravets, A. D. Kravets, S. A. Fomenkov, V. A. Kamaev // IADIS Applied Computing 2013 conference.– Forth Worth, Texas, USA, 2013 – Pp. 178–182.

13. Shoham Y. Multi-agent systems: Algorithmic, Game-Theoretic, and Logical Foundations / Y. Shoham, K. Leyton-Brown. – London : Cambridge University Press, 2009.

14. Wooldridge M. An Introduction to MultiAgent Systems / M. Wooldridge. – USA : John Wiley and Sons, 2009.

References

1. Budylskiy A. V. Upravlenie proektami razrabotki programmnogo obespecheniya s ispolzovaniem agentnykh tekhnologiy [Project Management of Software Development using Agent Technologies]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Management and High Technologies], 2013, no. 3 (23), pp. 119–128.

2. Kizim A. V., Kravets A. D., Kravets A. G. Generatsiya intellektualnykh agentov dlya zadach podderzhki tekhnicheskogo obsluzhivaniya i remonta [Generation of intellectual agents in technical service and repair support problem]. *Izvestiya Tomskogo politekhnicheskogo universiteta* [Bulletin of the Tomsk Polytechnic University], 2012, pp. 131–134.

3. Kravets A. G., Shevchenko P. V., Kravets A. D. Generator agentov multiagentnoy sistemy sbora dannykh o perspektivnykh tekhnologiyakh [Generator of nanotechnology data gathering multi-agent system agents]. *Visnik Natsionalnogo tekhnichnogo universitetu «KhPI». Zbirnik naukovikh prats. Tematichniy vipusk «Sistemniy analiz, upravlinnya ta informatsiyni tekhnologiї»* [Bulletin of National Technical University "KPI". Special Issue "System Analysis Management and Information Technologies"], 2012, no. 29, pp. 92–97.

4. Kravets A. D., Abramichev D. A. Generatsiya intellektualnykh agentov [Generation of intellectual agents]. *Konferentsiya-konkurs nauchnykh, konstruktorskikh i tekhnologicheskikh rabot studentov VolgGTU : tez. dokl.* [Conference-contest of Scientific, Construction and Technical Student Projects of Volgograd State Technical University], Volgograd, Volgograd State Technical University Publ. House, 2012, pp. 45–48.

5. Kravets A. D., Kravets A. G., Fomenkov P. A. Proektirovanie generatora intellektualnykh multiagentnykh sistem [Development of multi-agent system generator]. *Sovremennye problemy i puti ikh resheniya v nauke, transporte, proizvodstve i obrazovanii'2012 : Sbornik nauchnykh trudov SWORLD on materialam Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Actual problems and their solutions in science, transport, manufacture and education'2012. Digest of scientific works SWORLD on materials of the Scientific and Practical Conference], 2012, vol 13, no. 4, pp. 42–46.

6. Kravets A. D., Fomenkov P. A., Kravets A. G. Razrabotka modeli generatsii intellektualnykh agentov [Development of intellectual agents generation model]. *Sbornik nauchnykh trudov SWORLD* [Digest of scientific works SWORLD], 2012, vol. 5, no. 3, pp. 59–61.

7. Romanenko Ye. V., Kravets A. G. Nekotorye voprosy proektirovaniya i realizatsii raspredelennoy informatsionnoy sistemy "TOUREAST: CRM AI" [Design and Implementation Issues of Distributed Information System "TOUREAST: CRM AI"]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Management and High Technologies], 2013, no. 4 (24), pp. 165–175.

8. Umarov A. P., Popova N. V., Zelepukhina V. A. Nekotorye aspekty sozdaniya informatsionnykh sistem dlya sbora i khraneniya nauchnoy i naukometricheskoy informatsii [Some Aspects of the Development of Information Systems for the Collection and Storage of Scientific and Scientometric Information]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii [*Caspian Journal: Management and High Technologies], 2013, no. 3 (23), pp. 111–118.

9. Korotkov A. A, Kravets A. G., Kravets A. D. Intelligent Multi-Agent Systems Generation. *World Applied Sciences Journal (WASJ,* 2013, vol. 24, Spec. Issue 24 : Information Technologies in Modern Industry, Education & Society, pp. 98–104.

10. Kravets A. G., Kravets A. D., Fomenkov P. A. Component-Based Approach to Multi-Agent System Generation. *Knowledge-Based Software Engineering : Proceedings of 11th Joint Conference, JCKBSE 2014 (Volgograd, Russia, September 17–20, 2014)*, 2014, vol. 46. Series: Communications in Computer and Information Science, pp. 483–490.

11. Kravets A. G., Gurtyakov A. P., Kravets A. D. Corporate intellectual capital management: learning environment method. *Multi Conference on Computer Science and Information Systems 2013 (Prague, Czech Republic, July 23–26, 2013) : Proceedings of the IADIS International Conference ICT, Society and Human Beings 2013 and IADIS International Conference e-Commerce 2013*, Prague, International Association for Development of the Information Society Publ. House, 2013, pp. 3–10.

12. Kravets A. G., Kravets A. D., Fomenkov P. A., Kamaev V. A. Multi-agent systems component-based generator: development approach. *IADIS Applied Computing 2013 conference*, Forth Worth, Texas, USA, 2013, pp. 178–182.

13. Shoham Y., Leyton-Brown K. *Multi-agent systems: Algorithmic, Game-Theoretic, and Logical Foundations*, London, Cambridge University Press Publ., 2009.

14. Wooldridge M. *An Introduction to MultiAgent Systems*, USA, John Wiley and Sons Publ., 2009.