

Methodological Approaches to Development and Implementation of Multi-agents Investment DSS Using JADE Platform

Andrius Jurgutis and Rimvydas Simutis

Department of Automation, Kaunas University of Technology, Kaunas 51367, Lithuania

Abstract: In these latter days software agents are used for the development and implementation of intellectual decision support systems. In order to implement intelligence in a system some or several dozen of software agents are used and the made system becomes multi-agent. For the development of these systems a set of methodologies, i.e., the sequence of consequent steps of analysis, designing and implementation, is offered. The carried out analysis of the methodologies showed that as a rule they are limited by the spectrum of their pending problem (within the pales of the requirements of specific applied task, within the pales of the possibilities of technical implementation) or within the pales of amount of detail. The variety of methodologies is influenced by the fact that for the development of these systems the requirements and attitudes are offered by the specialists of related spheres such as software, numeral intellect engineers. In the course of the development of hardware and software appeared possibilities to implement mobile multi-agents systems, however, there is no one united mobile multi-agent systems design methodology, whereas existing systems are underdeveloped and their number is small. In this article we introduce the course of the designing of an intellectual real time multi-agent investment management decision support information system adapting and combining some methodologies where the choice to use either communicating or mobile agents is the question of rather technical implementation than methodological. In the article we introduce two ways of system implementation by JADE platform: the first one—using communicating agents, and the second one—using mobile agents.

Key words: Decision support system, intelligent systems, multi-agent systems, knowledge based systems, software agents.

1. Introduction

The development of decision support systems is closely related and keeps pace with the development of hardware, software and numerical methods. Today there is plenty of decision support systems applied in various spheres. Multi-agent investment decisions support system could be attached to knowledge-driven (experts-driven) DSS group or to newly forming Web-driven DSS group [1].

Decision support systems take root in the investment management sphere (business) and both

finance market specialist and their clients repose their trust in them [2-4]. Conducted researches allow drawing the conclusion that the use of these intellectual methods in finance management becomes serious counterbalance as well as the supplement to the classical analytical investment decision-making processes based on investment portfolio management models.

Investment decision support system was implemented with software agents. There are examples of similar systems such as WARREN, MASST [5]. The authors of multi-agent investment management systems notice the majority of the advantages of these systems [3, 6-8].

Investment management is a complex and

Corresponding author: Andrius Jurgutis, Ph.D., MBA, research fields: multi-agents system, intellectual and artificial intelligence systems in finance. E-mail: info@sprendimutechnologijos.lt.

complicated task coming down to the solution of several parallel and consequent real time tasks (tasks execution) and the control and management of separate tasks [3, 4]. In order to solve these tasks and fulfill these actions investment management companies, financial mediators employ various specialists whereas the last-mentioned use different calculation methods, intellectual and adaptive technologies and try to find and make the best investment decisions. You can visually imagine multi-agent investment decision support system as an autonomous virtual investment management and consulting company in which human specialists are replaced with intellectual software agents which execute works more productively. The system is adaptive, i.e., with regard to the situation in the financial market it should offer advices to an investor when and what security paper to purchase/hold/sell, offer to diversify the portfolio and thus to help to manage investments successfully.

Investment decision support system is our unique offered system and implemented with a help of expert systems of investor's risk tolerance assessment, investor's physical assets (property) allocation, financial means assessment [3, 9]. These systems guarantee that investment decision support system can serve the investors of all profiles (types).

It is evident that the system would be useful for an individual investor because it is difficult to him to compete with financial mediators with regard to knowledge because to make an investment decision he executes all necessary actions alone, as if all specialists are collected inside him. Besides, the system would be useful for investment management companies because it would allow them to increase efficiency [1].

Having presented briefly in the introduction the prerequisites of the appearance and functioning abstraction of the multi-agents investment decision support system, the second section of the article briefly reviews the existing methodologies for the

development of multi-agents systems. The third, fourth and fifth sections of the article present in more detail an adapted technique for the development of multi-agents investment decision support systems. The sixth section presents investor's experience and the result of system use.

2. Multi-agent Systems Development Methodologies

In the course of the development of software agents theory and their implementation technique several methodologies for agent system development (design, creation) were offered. Such a big amount of methodologies was influenced by some factors: (1) There is no agreement concerning the definitions of single agent and its features; (2) The requirements and attitudes for the development of these systems are provided by the specialists of related spheres but having different viewing angle, such as program, numeral intellect, object-based programming, distributed management systems engineers; (3) There is no one single, deeply-rooted, generally accepted standard for the development of agent systems; (4) The development of software agents as a technology has not stopped. Besides, having remembered what methodology is and what it consists of it is possible to say that agent theory and (or) available technical implementation means do not allow to fill all integral parts of the methodology fully (unambiguously) or there is a possibility for variation what provides the variety of methodologies [10, 11].

The literature offers some grouping methods of multi-agent system development methodologies. One of the methods to divide methodologies is to consider what systems they allow to develop. In this regard methodologies are divided into two groups, i.e., general-purpose methodologies and domain-specific methodologies.

The best known general-purpose methodologies are the following: MaSE (Multi-agent System Engineering methodology) [12, 13], GAIA [14, 15],

Tropos [16], Prometheus [17], MESSAGE [18], methodologies of the authors of JADE platform [19], INGENIAS [18] and others [10-20]. Among the domain-specific methodologies it is worth to mention GAIA2JADE [21, 22], PASSI [23], CASSIOPEIA [24], MOMBAS [25]. Domain-specific methodologies are the narrower versions of general-purpose methodologies. The methodology of the development of the offered multi-agents investment decision support system is obtained naturally in the course of system development in reliance upon financial mediators business model, organizational structure and applying the methodological knowledge of the authors of MaSE, GAIA, GAIA2JADE, JADE. Next, we will shortly present these methodologies.

MaSE [13, 20] and GAIA [15] methodologies are made of two phases, i.e., system analysis phase (at the level of abstraction) and designing phase. Implementation phase is not singled out separately because according to the authors of the methodologies designing phase coincides with implementation phase, i.e., designing phase is an abstract reflection of implementation phase, whereas implementation phase would be technical reflection of designing phase. Methodologies are independent of specific agent platform, programming language and agents communication protocols, so the designed system can later be implemented by different software means. Methodologies are also popular because of the following features: (1) The steps of system designing are executed using graphical display method; (2) The steps present in methodologies can be flexibly adjusted to the needs of the designed system; (3) The system is developed by iteration model. It is possible to mention the following differences of GAIA in comparison to MaSE methodology: (1) more detailed steps; (2) appropriate to copy and apply the features and behavior of the elements of factual social business systems; (3) uses formalization language.

The main steps of the methodologies are the following: (1) to identify and structure system goals,

to make an exact hierarchic tree (the goal is split until it can be accomplished by one simple agent); (2) to provide subsidiary platform agents for system's agents, to identify necessary platform functionalities (services); (3) to identify agents' roles (behaviour and actions to achieve a goal); (4) to identify and design (project) communication relations of agents; (5) to plan agents' position on the platforms; (6) to design (project) agents' relations (links) with external systems.

Other MaSE steps are discussed in more details in Refs. [12, 13], and GAIA in Refs. [15, 17, 21].

Abovementioned methodologies are not methodologically complete, i.e., there is lack of implementation phase. In order to finish the complete cycle of the development of multi-agent investment decision support system, JADE platform was chosen as technical implementation means, and at the same time the methodologies of the authors of GAIA2JADE [21, 22] and JADE [19] platform for the development of agent systems were used. These methodologies differ from above discussed: (1) The methodology of JADE authors is more abstract than formalized; (2) The types and features of agents are identified from the abstract model of the system rather than from the exact hierarchic tree of the system; (3) Agent's role (actions) depends not on the goal provided for it to achieve by hierarchy, but on the agent's role and responsibilities provided by abstract model; (4) During designing phase technological capacities of JADE platform are fully taken into consideration.

Despite such details steps of methodologies they also have disadvantages. The biggest disadvantages are the following: (1) able to create small agent systems; (2) do not include the creation of dynamic multi-agent systems where the number of agents varies; (3) suitable to create closed-type multi-agent systems where communication relations and communication protocols do not vary, one model communicate with one model; (4) do not have automatic code generation system [13, 17]. The

advantages, disadvantages of other methodologies, as well as the comparisons of methodologies and comparative methodologies can be found in Refs. [11, 17, 25].

Having discussed the methodologies the further sections of the article present multi-agent investment decision support system development methodology and pair system implementation ways.

3. System Analysis Phase

The offered steps of system development, their content and number of steps are not distinct from other mentioned methodologies and we do not bring in uniqueness within presented methodology in this respect. The uniqueness of the presented methodology appears at the first step of analysis phase which later influences other steps.

Within the analysis phase the above discussed methodologies tend to split goals (MaSE) or actions (GAIA) up to elementary (low) level in order so that the simplest agent could accomplish a goal or an action. The sense of the offered new idea in the analysis phase is not to split up goals and tasks so finely, but to split the system into unique integral components (elements) in relation to functions (function) and responsibilities (responsibility). For example, the recommendation to purchase/hold/sell security paper is given after the fulfillment of fundamental and other analyses, so the methodology discussed by the smallest component would hold the component able to fulfill only fundamental analysis or only other analysis. Based on this attitude independently working analyst having specific knowledge, responsibilities, features, perception of reality and keeping in touch with other system members is replaced with autonomous software agent with corresponding features, knowledge, communication possibilities.

Thus, the first step for the creation of desirable system is to find the closest analog of social system or business model, to conduct analysis, to describe the

system being created abstractedly and to display it graphically. Graphic display of the multi-agents investment decision support system and one version of system abstraction are given in the article [7]. We should not only miss an important fact that in case of multi-agents investment decision support system we can provide two versions (cases) of system abstraction. Substantial difference in abstractions would be planning how the analyst of financial mediator (later replaced with agent) collaborates, communicated with other analysts (agents), information sources (for example, databases, Bloomberg). Having taken in consideration the existing business models of financial mediators, in the first case an analyst can make requests to analysts and (or) information sources and get definite answers, whereas in the second case a financial analyst can go to information source, use its available resources and return with results. Therefore, an agent that fulfills the function of analyst in the first case of abstraction should be able to communicate, and in the second case it should be able to migrate.

The second step of methodology, similar as in MaSE, GAIA is to separate preliminary the unique tasks from an abstract model with regard to functions and responsibilities. They are given in Table 1.

Using tasks (goals) emphasized in Table 1 the third step of the analysis phase is to identify preliminary the software agents, i.e., how many of them, what kind of them, what tasks (goals) they will accomplish. The agents identified during this step are given in Table 2.

The third step presents the details of functions and responsibilities provided for the agent [3, 9].

After the identification of agents, their functions and features the fourth step of the analysis phase identifies their interrelationships, i.e., how they keep in touch with each other as well as with other subsidiary system's means (for example, databases, subsidiary agents of JADE platform, etc.). We would like to draw your attention to the fact that depending on the chosen abstraction version in the first case agents will only communicate whereas in the second

Table 1 Preliminary identified unique tasks with regard to functions and responsibilities.

Task description
1. The providing of recommendation to purchase/hold/sell financial means;
2. The assessment of investor's risk tolerance;
3. The assessment of the investor's investment portfolio allocation;
4. The assessment of financial means recommendation with a help of fundamental analysis method;
5. The assessment of financial means recommendation with a help of technical analysis method;
6. The assessment of financial means recommendation with a help of risk analysis method;
7. The assessment of financial means recommendation with a help of expert analysis method;
8. To coordinate agents' activity.

Table 2 Identified agents

Agent's name	Executing function and responsibility (goal)	Features
Interfaces	The assessment of investor's risk tolerance; The assessment of investor's investment portfolio allocation.	Autonomous, active, pro-active.
Fundamental	The assessment of financial means recommendation with a help of fundamental analysis method.	Autonomous, active, pro-active, communicating or mobile
Technical analysis	The assessment of financial means recommendation with a help of technical analysis method.	
Risk	The assessment of financial means recommendation with a help of risk analysis method.	
Expert	The assessment of financial means recommendation with a help of expert analysis method.	
Intellectual	The providing of recommendation to purchase/hold/sell financial means.	
Coordinating	To coordinate the activity of agents.	

case they will be mobile. Having displayed it graphically we have two separate schemas which are shown in Figs. 1 and 2. On the schemes a thin dashed line with arrows joins intercommunicating agent, and a thick dashed line with arrows shows agents' migration path, a solid line with arrows shows SQL requests.

The fifth step discusses where and how agents, secondary systems will be positioned. Depending on the chosen abstraction version two agents' layouts are possible. They are showed in Figs. 1 and 2.

The sixth and the last step provides (plans) which subsidiary agents or platform functionalities will be necessary. If communicating agents are chosen, the agent of DF (directory facilitator) platform will be necessary which knows the details (requisites) of other agents. Using mobile agents the agent of AMM (agent mobility manager) platform is necessary which is responsible for the migration of agents.

In summary, the objective of the analysis phase is to obtain from abstract description of multi-agents investment decision support system or from the graphic display based on actually operating social system or by the example of business organization

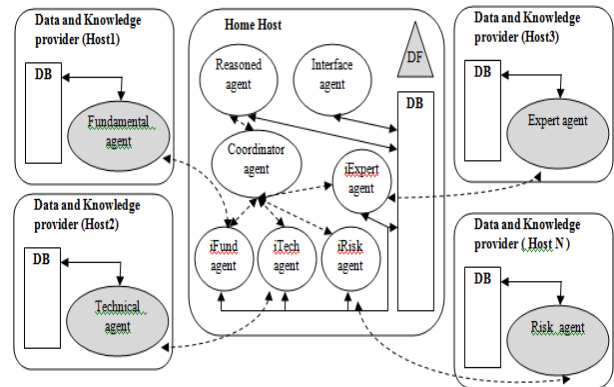


Fig. 1 Agents position scheme when agents communicate.

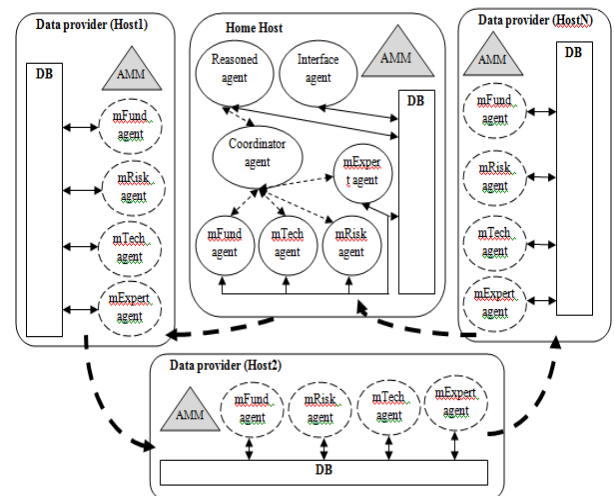


Fig. 2 Agents position scheme when agents are mobile.

activity model system scheme on which the components of abstract model would be replaced with software agents corresponding to their responsibilities and functional qualities. Having such a scheme it is possible to pass on to other phases.

4. System Designing Phase

Designing phase consists of several steps in the course of which agents' inner structure, communication processes and ways are detailed. The consequence of the designing phase is the achievement of definite and sufficient detailing of system components which should be understandable and clear to a programmer who implements the system during the implementation phase.

The first step presents the details of co-called working (task) agents, i.e., the agents which, except for communication, execute certain operations, actions. This step presents the details of the operations performed only inside an agent, actions for goal achievement, whereas communication and mobility are not discussed here. Working agents would be the following: interfaces, intelligent, coordinating, fundamental analysis, technical analysis, expert, risk agents. Both the first and the second versions of abstraction do not influence the number of working agents and their inner structure, while their detalizations are given in Refs. [3, 9]. It is important to mention that all the agents are intellectual, i.e., they include expert fuzzy logic systems which allow transferring the knowledge of the corresponding

experts to agents.

The second step of designing phase details communication relations among agents and relation ways. This step is sensitive to the chosen system implementation abstraction.

First of all, we detail the case when the system is implemented using communicating agents (Fig. 1). Agents in the system communicate by to each other model. It is easy to detail agents' relations, communication protocol in use (according to FIPA (is an IEEE Computer Society standards organization that promotes agent-based technology standards) specifications), and communication moment in Table 3.

In the first column of Table 3 it is possible to see agents' names where the first indicated is communication initiator, and the second is receiver or receiver and responder together. The second column contains communication protocol in use; the third one contains communication moment and finally communication goal.

Further it is necessary to detail the content of the message transferred by communication protocol or in other words a pattern (language in use, ontology and others according to possible FIPA message pattern parameters). Taking into consideration the recommendations of the methodologies message content will include only the short name of the financial means, and used ready-made parameter include ID-conversation (in case of FIPA Request protocol) and language (indicating fipa-sl value).

Table 3 Agent's communication relations.

Pair of agents	Protocol	Moment (time)	Goal
Coordinator-iFund	FIPA Inform	At fixed repeating intervals. (for example each 6 hours)	To inform information agents that they should start gathering information about financial means of interest.
Coordinator-iTech			
Coordinator-iRisk			
Coordinator-iExpert			
iFund-Fundamental agent	FIPA Request	At fixed short repeating intervals (for example each 2 minutes)	Inquire of the assessment of financial means by corresponding method. In each case one of two possible answers is obtained: the assessment or the absence of assessment.
iFund-Technocal agent			
iFund-Risk agent			
iFund-Expert agent			
Coordinator agent-reasoned agent	FIPA Inform	At fixed repeating intervals (for example each 6 hours)	To inform the intellectual agent that it is possible to start forming recommendations.

Implementing the system with a help of mobile agents one pair of communicating agents remains (coordinator agent-reasoned agent), whereas other are replaced with mobile agents (Fig. 2) which are created by coordinator-agent at regular intervals, and later when they accomplish their work they disappear. Mobile agents created in agency (agent server) have the list and route of visitable agencies.

When detailing agents' communication relations, the question of system synchronization always arises, i.e., agents should work in certain priority and time sequence. If this priority and time sequence of agents' work is damaged, the system will work inefficiently. System agents (components) can be synchronized in two ways. The first way is to determine experimentally (empirically) how long the work of each agent lasts and on this basis to make agents' work schedule. The second way—to use communication, i.e., when an agent finishes work it informs accordingly and thus initiates the activity of an agent (agents) which should work after it. This way is used when synchronizing agents' activity in the system.

The third step of the designing phase details the communication of agents with subsidiary agents of JADE platform and other systems which are not related to JADE platform (in case of the system under discussion only with databases). If for system implementation communicating agents are used for their registration, search and gathering information about them DF agent of JADE platform is used. Later system agents with a help of corresponding standard JADE platform procedures can apply to DF agent on purpose to find an agent with corresponding name that executes a corresponding function. AMM agent of JADE platform is used when the system is implemented by mobile agents. This agent has procedures necessary to execute the migration of agents. The agents can maintain the (relation) connection with databases in two ways. The first way is to create additional agents to databases, the second

way—agents are able to make SQL requests themselves. In the course of system implementation the second version was chosen because task agents use only two standard SQL requests, i.e., read and record data about financial means into database.

The fourth step of the designing phase details user's link (interface) with the system. Designing the system it was offered to separate user's interface from the agent part of the system, i.e., a system user does not directly interact with agents (does not initiate their appearance, disappearance, behavior, etc.). Both parts work independently from one another. User's Internet zone takes information from agents (agent part) according to updated databases. It is possible to mention several advantages of this separation: (1) Agent part becomes fully autonomous; (2) There is no need in a great number of interface agents; (3) It is possible to modify both parts independently from one another.

The fifth step details agent behaviour. When describing behaviour we understand that it is determined by the fulfillment or nonfulfillment of certain actions. The definition when, how and what actions an agent performs is the detalization of agent behaviour. This step of the designing phase also depends on software chosen for implementation. The chosen JADE platform includes some patterns of agents' behaviour implementation. Unfortunately, there is no accurate technique for the selection of behaviour pattern, while its selection is intuitive taking into account the results of system detalization at the steps of analysis and designing phases. Table 4 presents JADE behaviour patterns adjusted to agents.

These are all steps of the designing phase. The following is an implementation phase, i.e., the record of programming code.

5. System Implementation Phase

Having discussed the system as thoroughly as possible within the steps of two first phases, system implementation and code record become a simple task.

Table 4 JADE behaviour patterns adjusted to agents.

Agent name	System when agents communicate	System when agents are mobile
Interface agent	TickerBehaviour	TickerBehaviour
Reasoned agent	CyclicBehaviour	CyclicBehaviour
Coordinant agent	TickerBehaviour	TickerBehaviour
Fundamental agent	CyclicBehaviour	-
Technical agent	CyclicBehaviour	-
Expert agent	CyclicBehaviour	-
Risk agent	CyclicBehaviour	-
iFund agent	CyclicBehaviour	-
iTech agent	CyclicBehaviour	-
iRisk agent	CyclicBehaviour	-
iExpert agent	CyclicBehaviour	
mFund	-	SequentialBehaviour
mTech	-	SequentialBehaviour
mRisk	-	SequentialBehaviour
mExpert	-	SequentialBehaviour

Firstly, it is necessary to remind that JADE platform itself is implemented in JAVA programming language, what means that agents' code will be written in JAVA language. The authors of JADE platform chose JAVA because of its flexibility, i.e., applications written in JAVA language are compatible with different hardware and software.

Secondly, software agents can exist and operate in a certain environment called agency. Agency is a physical place of the existing and operating of a software agent (in a computer, on a server, etc.), as well as present there necessary software environment (rendered services). The agency necessary for the multi-agent investment decision support system is created with a help of JADE platform. Depending on a system abstraction model in the course of system implementation services necessary for communication or services necessary for agent mobility implementation are activated. JADE platform is started and services are actuated with a help of command line some examples of which are given in Table 5.

As you can see from the examples of command lines shown in Table 5 in the course the implementation of the system which uses communicating agents it is not necessary to activate

communication service additionally; it will be activated automatically when JADE is actuated. Using migrating agents mobility service should be activated additionally indicating that in the command line with a mark-services.

In the space of JADE start-up command gui and service activation command services in the command line (-agents) are indicated and in such a way software agents are activated indicating their unique name and JAVA class generated from a software (program) code. After that the command line creates agency which are called containers according to JADE terminology and agents in them start performing their actions, tasks. It is possible to create as many containers as necessary on a computer, and different services can be activated in them.

The third important moment within the implementation phase is to discuss how the knowledge of investment experts is transferred to software agents looking on the part of the record of programming code. As it was mentioned above expert knowledge is transferred to agents using fuzzy logic systems. Necessary fuzzy logic systems were implemented using "MatLab Fuzzy ToolBox". Later using "MatLab Builder JA" MatLab software (program) codes with fuzzy logic systems were converted

Table 5 Examples of JADE command line.

Explanation	Command line
When use communicative agents	
Start home agency and information, intelligent, coordinating agents	-gui informexpert:lt.ajurgutis.webapp.agents.Inform_expert_agent;informfund:lt.ajurgutis.webapp.agents.I nform_fund_agent;informrisk:lt.ajurgutis.webapp.agents.Inform_risk_agent;informtech:lt.ajurgutis.we bapp.agents.Inform_tech_agent; coord:lt.ajurgutis.webapp.agents.Coordination_agent;reasoner:lt.ajurgutis.webapp.agents.Reasoner_age nt -agents
Start other agencies and tasks agents	-gui risk:agents.Risk_agent;tech:agents.Tech_agent;fund:agents.Fundamental_agent;expert:agents.expert_a gent -agents
When use mobile agents	
Start home agency and intelligent (reasoned), coordinating agents. Start mobility service.	-gui coord:lt.ajurgutis.webapp.agents.M_coordination_agent;Reasoner:lt.ajurgutis.webapp.agents.Reasoner _agent jade.core.mobility.AgentMobilityService;jade.core.migration.InterPlatformMobilityService -accept-foreign-agents true -agents -services
Start other agencies and mobility services there.	-gui jade.core.mobility.AgentMobilityService;jade.core.migration.InterPlatformMobilityService -accept-foreign-agents true -services

into JAVA classes which can be used by the agents for the performance of their tasks.

The fourth and the last important thing discussing the record of agent code, i.e., agent code structure and content. Agent code record on JADE platform is not absolutely “free” style in comparison to the record of simple JAVA application code because of some reasons. Firstly, JADE platform is implemented as a set of JAVA classes. Methods implemented in a class correspond to FIPA standards applied for the creation of software agents. For the implementation of each agent several JADE platform classes are used and later the methods of these classes are used in agent code content. Secondly, agent code has clear arrangement structure, i.e., all actions performed by an agent are detailed (recorded) inside agent behaviour.

6. Investor’s Experience Using IDSS

Having implemented multi-agent IDSS (investment decision support system) it was offered to investors for testing purposes. This section presents the experience of one investor and investment results using multi-agent investment decision support system.

According to the questionnaire filled in by the investor, risk tolerance assessment system attached the investor to the speculative investor group (tolerates

high risk), and after the additional evaluation of investor’s age and income it was offered to allocate the portfolio in the following way: 13% low-risk assets, 40% moderate-risk assets, 47% high-risk assets. The initial cost of the investor portfolio on October 26, 2012 was 100,000 USD. The system offered the investor financial means of different risk levels from USA market. The investor configured the system in such a way that offered to buy financial means would be assessed in all aspects provided in the system, i.e., applying fundamental, technical analysis, risk and expert assessments. The investor could easily choose to buy financial means which have a recommendation “to buy” and strictly to sell from the portfolio financial means which have a recommendation “to sell”. For the impartial comparison of investment results by the investor we constructed comparative index (Benchmark), 90% of which value was USA S&P 500 index and 10% of which value was USA government 5-7 years bond US15TR index.

The investment results of the investor till February 13, 2013 together with comparative index are showed in Fig. 3. Fig. 3 shows that using investment decision support system the investor obtained better results (+8.46% return within mentioned period) than average results of the similar investor in the market (+6.92%

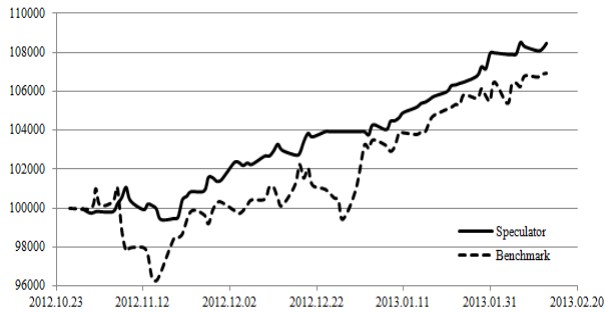


Fig. 3 The diagram of the values of investor's investment portfolio and Benchmark.

Benchmark return within mentioned period).

Risk parameter of portfolios were compared, i.e., standard deviation of portfolio return within one day and one day VaR with 95% reliability level. The values of the mentioned risk parameters of investor's portfolio were respectively 0.29% and -564.87 USD, and Benchmark portfolio respectively 0.69% and -1342.97 USD.

The analysis of other investment results obtained by investors showed that using multi-agent investment decision support system the results were better than average in the market. However, taking into consideration short testing period it is before time to make summarizing conclusions.

7. Conclusions

Guided by the methodology presented in the article the multi-agent investment decision support system was successfully and fully implemented and it can be tested at www.sprendimutechnologijos.lt/webapp. Methodology analysis and the number of designing steps and detalization depth is sufficient performing later system programming with JADE.

Having implemented both abstraction versions of multi-agent investment decision support system they were compared. After the comparison it is possible to make a conclusion that the system implemented by mobile agents is superior and more effective than the system implemented by communicating agents. Its superiority and efficiency is manifested in the following aspects: (1) The system implemented by mobile agents works more quickly; (2) The system

implemented by mobile agents "does not block" and does not use Internet network intensively; (3) The system implemented by mobile agents uses smaller number of agents; (4) In the system implemented by mobile agents it is easier to synchronize agents' work; (5) This multi-agent system is less sensitive to the breakdown and troubles of system components.

The investment results achieved by investors using multi-agent investment decision support system show that the system is able to help to make more profitable investment decisions than market average.

References

- [1] Power, D. J. 2011. *A Brief History of Decision Support Systems*. DSSResources.COM, version 4.1., World Wide Web, <http://DSSResources.COM/history/dsshistory.html>.
- [2] Bojadziev, G., and Bojadziev, M. 2007. *Fuzzy Logic for Business, Finance, and Management*. 2nd ed.. NJ, World Scientific Publishing Co. Pte. Ltd., 232.
- [3] Jurgutis, A., and Simutis, R. 2009. "Building of Securities Valuation IT System Using Multi-agents Approach." In *Proceedings of the 4th International Conference on Electrical and Control Technologies*, Kaunas, 156-61.
- [4] Wooldridge, M. 2006. *An Introduction to Multi-agent Systems*. John Wiley & Sons, UK, 348.
- [5] Wooldridge, M., Sycara, K., and Jennings, N. R. 1998. "A Roadmap of Agent Research and Development." *Autonomous Agents and Multi-Agent Systems* 1 (1): 7-38.
- [6] Jurgutis, A., and Simutis, R. 2011. "An Investor Risk Profiling Using Fuzzy Logic-Based Approach in Multi-agents Decision Support System." In *Proceedings of the 17th International Conference on Information and Software Technologies*, Kaunas, 23-30.
- [7] Negnevitsky, M. 2005. *Artificial Intelligence: A Guide to Intelligent Systems*. Addison-Wesley, UK, 415.
- [8] Nwana, H. S. 1996. "Software Agents: An overview." In *Knowledge Engineering Review*, UK, Cambridge University Press, 1-40.
- [9] Jurgutis, A., and Simutis, R. 2011. "Investor's Portfolio Allocation by Financial Asset Classes Using Fuzzy Logic-Based Approach in Decision Support System." *Computer Technology and Application* 2 (10): 757-64.
- [10] Zambonelli, F., Jennings, N. R., and Wooldridge, M. 2001. "Organizational Rules as an Abstraction for the Analysis and Design of Multiagent Systems." *International Journal of Software Engineering and Knowledge Engineering* 11 (4): 303-28.
- [11] Sturm, A., and Shehory, O. 2001. "A Framework for

- Evaluating Agent-Oriented Methodologies.” *Agent-Oriented Information Systems* 3030: 94-109.
- [12] DeLoach, S. A., Wood, M. F., and Sparkman, C. H. 2001. “Multiagent Systems Engineering.” *International Journal of Software Engineering and Knowledge Engineering* 11 (3): 231-58.
- [13] Wood, M. F., and DeLoach, S. A. 2000. “An Overview of the Multiagent Systems Engineering Methodology.” In *Proceedings of First International Workshop on Agent-Oriented Software Engineering*, Limerick, Ireland.
- [14] Wooldridge, M., Jennings, N. R., and Kinny, D. 2000. “The Gaia Methodology for Agent-Oriented Analysis and Design.” *Journal of Autonomous Agents and Multi Agent Systems* 3 (3): 285-312.
- [15] Zambonelli, F., Jennings, N. R., and Wooldridge, M. 2003. “Developing Multiagent Systems: The Gaia Methodology.” *ACM Transactions on Software Engineering and Methodology* 12 (3): 317-70.
- [16] Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., and Perini, A. 2003. “TROPOS: An Agent-Oriented Software Development Methodology.” *Journal of Autonomous Agents and Multi-agent Systems* 8 (3): 203-36.
- [17] Melomey, D. A. 2012. “A Methodology for Modelling Mobile Agent-based Systems (MobileAgent Mobility Methodology – MaMM).” A Master Thesis, University of East London, 222.
- [18] Maalal, S., and Addou, M. 2011. “A New Approach of Designing Multi-agent Systems (with a Practical Sample).” *International Journal of Advanced Computer Science and Applications* 2 (11): 148-57.
- [19] Caire, M. G., and Bahri, P. A. 2006. “A Methodology for the Analysis and Design of Multi-agent Systems Using JADE.” *International Journal of Computer Systems Science and Engineering* 21 (2): 1-40.
- [20] Gomez-Sanz, J., and Pavon, J. 2004. “Methodologies for Developing Multi-agent Systems.” *Journal of Universal Computer Science* 10 (4): 359-74.
- [21] Moraitis, P., and Spanoudakis, N. I. 2006. “The GAIA2JADE Process for Multi-agent Systems Development.” *Journal of Applied Artificial Intelligence* 20 (2-4): 251-73.
- [22] Moraitis, P., and Spanoudakis, N. I. 2004. “Combining Gaia and JADE for Multi-agent Systems Development.” In *Proceedings of the 17th European Meeting on Cybernetics and Systems Research* (EMCSR 2004).
- [23] Cossentino, M., and Potts, C. 2002. “A CASE Tool Supported Methodology for the Design of Multi-agent Systems.” Presented at The 2002 International Conference on Software Engineering Research and Practice (SERP’02).
- [24] Collinot, A., and Drogoul, A. 1998. “Using the Cassiopeia Method to Design a Soccer Robot Team.” *Applied Artificial Intelligence (AAI) Journal* 12 (2-3): 127-47.
- [25] Tran, Q. N. N., and Low, G. 2008. “MOBMAS: A Methodology for Ontology-Based Multi-agent Systems Development.” *Information and Software Technology* 50 (7-8): 697-722.