

Modelling Distant Learning Activities by Dynamic Agent Task Coalitions

Vadim A. Ermolayev, Vyachyslav A. Tolok
Department of Mathematical Modelling and IT, Zaporozhye State University,
Zaporozhye, 69063, Ukraine

Abstract. Presented is the approach to the design of distant learning facilities for a Virtual/Real University¹ based upon the paradigms of a rational and a benevolent agent, multi-agent system, dynamic task coalition. The particularity of the framework is its capability to perform the tasks without pre-defined task plans. Agents and multi-agent systems inhabit Virtual University Information Space, model real life actors – faculty, technical and administrative staff as well as the users from the outside. Agent coalitions co-operatively provide for the task performance, thus, modelling the processes of university management and distant education. Parametric feedbacks and agents' ability to evolve contribute to the fine-tuning of management routines and to the improvement of teaching and learning. PhD students' recruiting case study provides the illustration of the framework applicability to Virtual University and Distant Education domains.

Distant Education today is the methodology capable to drastically enhance the effectiveness of various kinds of education both in academic and professional branches. Information Technologies (IT) and the Internet are the means providing the capabilities and the infrastructure to organise the process of distant learning in a rather flexible, adaptable and efficient manner. Emerging Virtual Universities (VU) and Virtual Professional Training Centres (VPTC) provide bright examples of how today's intelligent distributed software systems and underlying ITs educate people.

The paper presents the approach to apply formal agent-based framework for the modelling of the processes of information interchange to the design of a Virtual University Information Space (VUIS) [1] inhabited by agents, that form coalitions to facilitate to the execution of the business processes of distant education.

The very high idea of the presented research was inspired by Angehrn's ICDT model [2] of Internet Business Strategies. The concept of VUIS however differs from that of ICDT Virtual Information Space denoted as simply the channel for displaying and accessing information. In the frame of our research VUIS is understood as a Virtual Medium organised on top of the layered mediator IS unifying the hierarchy of the distributed, heterogeneous, interacting and collaborating functional components (departments) and the wrapped distributed heterogeneous information resources. Human users divine VUIS as the model of VU and communicate with it by means of Unified Visual Intranet Interface (UVII) [1]. The concepts of VUIS and UVII are close to the known approaches to Inhabited Information Spaces [3] design. VUIS is inhabited by active functional components (Multi-Agent Systems (MAS) and member-agents) which occupy

¹ The research presented is run in frame of the Project financed by Ukrainian Ministry of Education and Science, Grant № 0199Y1571.

corresponding organisational cells at different levels. From organisational point of view these components are virtual business objects performing business processes in terms of, say, the Enterprise Framework [4].

The particularity of the approach is the attempt to model the processes of Distant Education as business processes within a VU. Business processes are in their turn modelled as the processes of information interchange among various types of human users and different active functional systems/components presented by MAS/agents possessing appropriate roles and distributed over the Internet. The frameworks, architectures and implementations for business process modelling and management in Virtual Enterprise domain are now emerging high and wide (see [4,5] for some examples). However, the diversity of the processes observed in real life is difficult to be modelled by more or less static means provided by, say, CTL based framework [5], ROOM, OOFram role models [6], ICRF [7]. Agents' paradigm provides the way out of this world of predefined workflow and role specifications. The presented approach exploits the metaphor of dynamic agent community² in order to provide better means for the modelling of the intrinsic dynamic character of the domain. This approach is close to that used in RETSINA framework [9] for adaptive collaboration among agents' teams facilitating to solve the tasks of decision making and information management. In the frame of the presented approach the agents are the members of various static MAS representing persistent departments of a VU. The departments communicate with each other via the Proxy Agents acting as the executives who are in charge with some external communications/functions. These Proxies in turn form the university MAS on the higher level. On the lower level each member agent of the department MAS may be expanded into a sub-ordinate MAS having the same generic architecture. As far as these department models represent university functional nodes they are pre-designated to perform tasks. These tasks are merely the tasks of information acquisition, integration, mediation and interchange. Agents' roles [10] are more or less static as far as the agents are capable to perform given sets of atomic works. On the other hand agents' capabilities and beliefs change in time due to changing constraints and experience gained. Moreover, the agents within MAS dynamically form the coalitions, denoted as temporal agent communities, to perform one or another task. The approach presented exploits the Diakoptical MAS framework [10], the model of task execution by agents coalition [11,13]. Human user interface designs are based upon the concept of UVII [1].

The focus of this publication is the operational aspects, the evolution model as well as the demonstration of the approach applicability to Distant Education domain.

The contribution is structured as follows: Section 2 outlines the modelling framework; Section 3 presents the approach to cope with VUIS inhabitants' evolution; the contribution of Section 4 is the discussion of PhD students' recruiting case study; Section 5 summarises the results. This publication is the revised version of [12]

² Business process is for instance viewed by Jennings et al [8] as the Community of negotiating service providing agents in ADEPT project.

2 Modelling Framework

VIS Functional Face is inhabited by MAS representing functional systems and components at various levels. The member agents dynamically form coalitions for the execution of the tasks of information acquisition, integration, mediation and interchange emerging within the organisation.

The main advantage of the task execution model [11] used in the framework is the absence of statically pre-defined task specifications. The tasks in frame of the presented approach are "summoned" by its Proxy and Facilitator agents and are executed by its middle agents³. Middle agents dynamically form communities to perform emerging tasks. An agent joins the community if and only if it accepts a sensory input containing the (sub)set of atomic works (the part of the task) for the execution. Task execution plan is being developed in more and more details within step-by-step task execution process. The process is conducted by the team of community member agents acting in co-operation with each other. Co-ordination agent serves as teams' co-ordinator and monitors the activities of each team. Coalition member agents act as the models of the functional components of the corresponding real world business object performing a business process.

The framework for modelling of the processes of information interchange comprises the following components: functional system/component model [10], process model [11,13], generic agent model [10], communication model [10], co-ordination model and evolution model.

Framework actors are intelligent (rational - Nwana [14]) software agents capable to communicate with each other by means of the defined set of communicative acts with parametric feedbacks [10, 11]. A task is assumed to be the set of atomic works. Each actor (agent) is capable to perform some atomic works from the set of permissible atomic works of the functional system. These capabilities form the role of the corresponding agent. The notion of role used in the framework [10] is close to that of ICRF [7].

At the agent level the framework provides the key agent's characteristics of situatedness, autonomy, rationality and adaptability. Agent accepts external influences, verifies if the incoming influence complies with the agent's role and finally adjusts its behaviour and performs appropriate macromodel program — i.e. executes or rejects the atomic work requested by the input influence. The function of the macromodel is also to rationally form the feedback containing the results. The results may be presented as functions from the parameters of the incoming influence.

Formally [10, 18], the generic agent is reactive, rational, comprises its sensory interface, the cascade of 3 finite-state machines for incoming influence verification local knowledge base and macromodel execution block. Generic agent is thus the operational shell providing the skeleton for any framework agent. Agents'

³ Middle agents in frame of this work are understood as those ones facilitating to the task execution inside the department MAS and having no direct interfaces to the outside. The notion used is close to that of RETSINA framework [17].

specialisations are merely the sets of their role specific macromodel programs. Macromodel programs are thus considered to be agent's policies and are stored in its local knowledge base.

At the community/coalition level it is assumed that the agents taking part in the process of task execution communicate by means of the following communication acts complying with ACL [15] and KQML [16] capabilities (see [10] for the formal specification and for more details): Directive, Determined Query, Determined Query with Results' Analysis, Undetermined Query with Results' Analyses.

At the functional system level the agents are considered to be benevolent. The model of a functional system as well as a functional component model is built upon the idea of "absorption" and "generation" of atomic works from the set of the permissible works $W = \{w_1, w_2, \dots\}$ of this functional system. It is considered that the sensory input of the functional component i admits a task $W_i \subseteq W$. A certain part of its works W_i^p may be performed ("absorbed") by the given component and the remaining part of works may be either redirected to another system's components W_i^d in case functional component knows the recipient(s), or rejected W_i^r .

Functional component may as well generate additional set of works W_i^g to complete the execution of works W_i^p . W_i^g as well as W_i^d are redirected to another components:

$$W_i \rightarrow F_O^i(W) \rightarrow \tilde{W}_i, \quad (1a)$$

where: $W_i = \{W_i^p, W_i^d, W_i^r\}$, $\tilde{W}_i = \{W_i^d, W_i^g\}$, $F_O^i(W)$ - macromodel program.

In a special case component i may generate a new set of works W_i^g without been invoked by incoming influence W_i - i.e. may "summon" a new (sub)task:

$$F_O^i(W) \rightarrow \tilde{W}_i, \quad (1b)$$

where: $\tilde{W}_i = \{W_i^g\}$, $F_O^i(W)$ - macromodel program.

More detailed presentation of the system/component model is given in [10, 11]. The model extension as well as the routines for negotiating on sub-task placement and on joining the task coalition within the sub-task arrangement phase are discussed in [13].

A process is denoted as the flow of task execution. Process Π_a starts with generation of the new task $W_a \subseteq W$. Task W_a as well as the additional tasks \tilde{W}_a are considered to be linked to process Π_a and labelled with the unique identifier of this process. The component is considered to be *linked to process* Π_a in case it has absorbed the part of W_a , \tilde{W}_a , or has generated W_a^g . The agent representing this functional component thus *enters the task coalition*.

Process Π_a is considered to be completed in case all the components stopped to absorb the atomic works of the tasks linked to process Π_a . The set of works $W_{\Pi_a}^z$ not absorbed in the process of Π_a is denoted as the set of *inexecutable* works.

Process Π_a modelling (steady-state mode) is performed by applying (1b) and (1a) to all of the components of the system until the process is completed.

For practice the set of system's permissible atomic works is constrained to a finite:

$W = \{w_1, w_2, \dots, w_\sigma\}$. Modelling of a functional system (task coalition) performance is organised as a two-level process in a discrete time space $t_n, t_{n+1} = t_n + \Delta t$. Please refer to [11,12] for the details.

3 Actors' and Resources' Evolution

One of the major characteristics of a VU is its inclination to changes. The framework for VU modelling should therefore possess the means to deal with the changes emerging within the real world. The evolution with respect to the subject under discussion is understood as the process of proactive self-development and self-adaptability of the intelligent active components (the agents) in response to the changes in the environment they inhabit - the VUIS.

The framework distinguishes and handles the movement in:

- Agents' state constraints — the *capabilities* to execute a work
- Agents' conceptualisations (*beliefs*) about their partners — task community members
- Information resources and corresponding metadata.

Capabilities' evolution according to [10] is understood as the process of agent (say, A) transitions from one state s_i to another state s_j . A as an autonomous entity performs these transitions according to its own decisions taken in frame of one or another atomic work execution. Consequently, the "manner" agent A executes policy f , as well as the constraints on policy incoming parameters X_f depend upon the state of agent A . Thus, the evolution of an agent is the evolution of its role.

The set of states of agent A : $S_A = \{s_1, \dots, s_n\}$ - is denoted as the set of 3-nested tuples $s_i, i = 1, \dots, n$:

$$s_i = \{\mathcal{r}(X_A), q(F_A), t(F)\}, \quad (4)$$

where:

$\mathcal{r}(X_A)$ - the set of constraints applied in state s_i over the system parameters X_A of agent A (parameter constraints),

$q(F_A)$ - the set of constraints in state s_i over the set of authorised works of agent A (work constraints),

$t(F)$ - the function denoting transitions from state s_i to another permissible states from S_A resulting from the execution of the works $F = \{f_1, \dots, f_j, \dots, f_m\}$.

Beliefs' evolution is closely tightened to the monitoring of task coalition members' capabilities to perform works. Inter-agent communication and work execution is organised/co-ordinated via parametric feedbacks [10, 11], comprising the information on the current capabilities to execute the certain work. The capability

returned by the executor A to the requestor B is, thus, the function from work parameters $c_A^f = c(X_f)$, $c_A^f \in [0,1]$.

An agent monitors the capabilities of its counter-agents for to intelligently assign works to the executors with probably better capabilities in future tasks. The beliefs on counter-agents' probable capabilities are maintained in the form of matrix \mathbf{C} :

$$\mathbf{C} = \begin{matrix} & \begin{matrix} w_1 & \dots & w_j & \dots & w_m \end{matrix} \\ \begin{matrix} A_1 \\ \dots \\ A_i \\ \dots \\ A_n \end{matrix} & \begin{matrix} c_1^1 \\ \dots \\ c_i^j \\ \dots \\ c_n^1 \end{matrix} \left[\begin{matrix} & & c_1^j & & c_1^m \\ & & \dots & & \dots \\ \dots & c_i^j = c_{A_i}^{w_j}(X_{w_j}) & \dots & & \dots \\ & & \dots & & \dots \\ & & c_n^j & & c_n^m \end{matrix} \right] \end{matrix} \quad (5)$$

The dimensions n and m grow in the process of evolution reflecting the income of new knowledge on counter-agents (n) dimensions and the works they are probably capable to perform (m) to matrix \mathbf{C} . The upper limit for dimension n is the number of member-agents in the MAS comprising the holder of matrix \mathbf{C} . The maximum value for m is the cardinality of the set W of permissible atomic works of the mentioned MAS.

Information resources data and metadata changes are maintained locally by corresponding distributed information systems - resource providers. In frame of the presented research information resource providers are represented by their wrapper agents, which evolve in response to this changes. Wrapper agents are the members (middle agents) of appropriate department MAS.

4 Modelling PhD Recruiting Scenario

PhD recruiting process modelling case has been studied to analyse the applicability of the described approach to distant learning and VU domain. The main reason for to choose this very case was the understanding that a VU needs to be self-regulating to be successful. VU management processes need feedbacks from the processes of distant teaching and learning to adapt to changing students' demands. Otherwise, the routines delivering courses and other knowledge to students should fine-tune themselves grounded on the feedback from improving management facilities. As for the case, PhD students' selection may be considered a management procedure (like hiring personnel). It will be demonstrated below that this process provides new knowledge on the necessity of new courses introduction, thus, feeding back and improving teaching process.

It is assumed that PhD candidates are surfing the VUIS, contacting the departments of their choice via the Proxies and expressing their intents to become students.

It is presumed as well that a Virtual Department is the MAS, comprising at least the following actors:

- Secretary - the Proxy Agent (PA)
- Professors (PRA), Assistants (AA), Course Master (CMA), Librarian (LA) - the Middle Agents

Department MAS also contains utility units providing for scalability, co-ordination and knowledge sharing among its functional actors: Cloning Agent (CA), Co-ordination Agent (COA) with its Shared Data Space (SDS) and Ontology Agent (OA) respectively.

The role of the CA with regard to the case under discussion is to clone a Tutor Agent (TA) each time a new task to process a PhD candidate is "summoned" by the PA in response to the external influence coming from the outer VUIS.

PhD recruiting scenario has been slightly adopted from the real world procedure to highlight the benefits we may obtain from the usage of the modelling approach presented in Section II. The assumptions made here are: participating human actors PhD candidates, Professors, ... are available on-line during the whole scenario; all generated works are accomplished in a reasonably short time.

We presume that the procedure of PhD recruiting comprises the following steps:

- A PhD candidate submits the CV and indicates his/her intention to become a PhD student
- The CV is analysed and the best Professor Match is searched
- Qualified candidate passes the test from the chosen professor
- Successful candidate is interviewed and assigned to a research project
- The professor and his assistant prepare the individual curriculum for the accepted candidate as well as the list of recommended reading

Agents' activities within these phases are as follows.

Phase 1. *Establish connection and submit the CV.* PA accepts the external influence, generates the new task. First atomic works within the task are: CA - to Clone the Tutor Agent; PA - to Pipeline the candidate to TA's human to agent interface, TA - Require CV and Extract Qualification Data)

Phase 2. *CV analysis and search for the best match.* TA submits Candidate's qualifications to Department PRAs. PRAs feed back with their parametric attitudes, having candidate's qualifications as parameters. TA determines the best match in case the feedbacks from some PRAs fit the qualification cluster region. In case the candidate appears to be not up to the level TA generates a work for the proxy to notify the requestor and to recommend him to contact other Departments. In case the best match is found the candidate is recognised to be qualified and TA summons the following Testing Phase.

Phase 3. *Testing:* TA requests the test from PRA. PRA provides the test. TA requests the candidate to fill in the test form and passes it to PRA. PRA evaluates the exercise and replies with the parametric marks (depending from the research project). TA executes the marks' analysis (similarly to Phase 2) and either qualifies the candidate as successful and launches the Interview Phase or entrusts PA to notify the candidate on his failure.

Phase 4. *The interview:* TA generates the task for the PRA to interview the candidate. TA pipelines successful candidate to PRA. PRA arranges on-line communication between his master (human professor) and the candidate. PRA requires his human master to fill in the PhD recruiting form. PRA influences TA to process the PhD recruiting form. TA analyses the PhD recruiting form and either passes it to the Personnel Department's PA to hire the accepted candidate to the project or entrusts PA to notify the candidate on his failure. In case the candidate has successfully passed the interview and thus became PhD student TA launches the Curriculum Phase.

Phase 5. *Curriculum preparation:* TA generates the task for the PRA to prepare the curriculum and the working program for the PhD student for the 1-st semester.

PRA redirects the task to his AA adding his course recommendations to the parameters' list. AA prepares the working plan and the curriculum and then requests the necessary electronic courses from CMA. CMA analyses the request and, if necessary, issues the Call for the unavailable courses – see details in [18].

Let's assume that at $t = t_n$ TA initiates the Testing phase and examine the activities of TA, PRA and PA agents within the Phase 3.

At $t = t_n$ TA accepts the set of works $W_{TA} = \{w_1 = ('Require\ the\ test', X_1, Y_1)\}$ with the parameters and result descriptions for w_1 :

$X_1 = \{Edu_Rating = \langle structure, ontology = Edu_Rating \rangle,$
 $Q_E_Rating = \langle structure, ontology = Qualif_Exp_Rating \rangle,$
 $Pub_Rating = \langle structure, ontology = Publication_Rating \rangle,$
 $Professor = \langle Id, ontology = Agent_Name \rangle\}, Y_1 = \emptyset.$

Atomic work w_1 is accepted and executed as far as all of the parameters X_1 (obtained as the results of the previous works at previous phases) are available from COA's SDS. While executing w_1 TA, as "subscribed" by its appropriate macromodel, generates the tasks $\tilde{W}_{TA} = \{W_{TA}^d, W_{TA}^g\}$, where:

$W_{TA}^d = \emptyset$ as far as work w_1 is executed and no more works are left for redirection;
 $W_{TA}^g = \{w_2 = ('Provide_Test', X_2, Y_2), w_3 = ('Test', X_3, Y_3),$
 $w_4 = ('Evaluate_Results', X_4, Y_4), w_5 = ('Analyse_Marks', X_5, Y_5)\}.$

Works w_2, w_4 form W_{PRA} and w_3, w_5 form W_{TA} for the next step $t = t_{n+1}$.

At $t = t_{n+1}$ PRA accepts $W_{PRA} = \{w_2 = ('Provide_test', X_2, Y_2),$
 $w_4 = ('Evaluate_Results', X_4, Y_4)\}.$

Work w_2 is executed and the results $\tilde{Y}_2 = \{Test_Form = \langle FILENAME \rangle\}$ are passed to COA for further use. At meantime w_4 is redirected to PRA for the next step – the results of w_3 , which form the parameters of w_4 , are not yet available from COA. At the same time TA accepts

$W_{TA} = \{w_3 = ('Test', X_3, Y_3), w_5 = ('Analyse_Marks', X_5, Y_5)\}$

and redirects both works to himself for next steps waiting for the results of respectively w_2, w_4 .

At $t = t_{n+2}$ TA executes w_3 . At $t = t_{n+3}$ PRA executes w_4 and passes the result vector $\tilde{Y}_{PRA} = \{\tilde{y}_4^1 = (m_1, m_2, \dots, m_k), \tilde{y}_4^2 = (s_1, s_2, \dots, s_k)\}$ to COA. Here, k is the quantity of PRA master's projects with PhD vacancies, m_i is the candidate's mark in case he/she pretends to work on project i and s_i indicates what the professor thinks about the level, starting from which the mark may be considered to be positive.

At $t = t_{n+4}$ TA accepts $W_{TA} = \{w_5 = ('Analyse_Marks', X_5, Y_5)\}$

with $X_5 = \{Marks = \langle \tilde{y}_4^1, ontology = Mark_per_Project \rangle,$

$Scale = \langle \tilde{y}_4^2, ontology = Positive_Mark_per_Project \rangle\}$

and decides if the candidate may be successful with respect to one of the project vacancies. In case of success W_{TA}^g will contain $w_6 = ('Require_the_Interview', X_6 = \{X_1, X_5\}, Y_6)$, otherwise TA generates $w_7 = ('Inform_on_Failure', X_7, Y_7)$.

5 Summary

The contribution presents the approach to apply the formal agent-based framework to the design of a Virtual University Information Space and model Distant Learning activities. In the frame of the presented approach the agents are the members of various static MAS representing persistent departments of a VU at different levels of organisation model. Agents dynamically form coalitions to perform the tasks related to the business processes of distant education. The departments communicate with each other via the Proxy Agents acting as the executives who are in charge with some external communications/functions. These Proxies in turn form the University MAS on the higher level. On the lower level each member agent of the department MAS may be expanded into the subordinate MAS having the same generic architecture.

The underlying modelling framework is based upon the paradigms of intelligent software agent, multi-agent system, dynamic agent task coalition. The particularity of the framework is its capability to perform the tasks without pre-defined task plans. The tasks in the frame of the presented approach are "summoned" by Proxy and Facilitator agents and are executed by the coalitions of benevolent middle agents. An agent joins the coalition if and only if it accepts a sensory input containing the (sub-)set of atomic works (the sub-task) for the execution and the mutual agreement on delegating the work to this very executor is gained within the arrangement phase [13]. The task workflow is thus being developed in more and more details within step-by-step execution process and is collaboratively conducted by the coalition agents team comprising co-ordination agent.

The framework provides the model of the agents' evolution to better cope with the diverse changes emerging in real life. Evolution is understood as the process of proactive self-development and self-adaptability of the intelligent functional actors in course of their task execution activities and in response to the changes in the environment they inhabit - the VUIS. Parametric feedbacks and agents' ability to evolve promote to the fine-tuning of management routines and to the improvement of teaching and learning.

PhD students' recruiting case studied provides the illustration of the framework applicability to VU and Distant Learning domains.

References

1. Ermolayev, V. A., Pletsy, S. U., Tolok, V. A. (1998): The Architecture of the Unified Information Space of a Virtual University. Lecture Notes of Zaporozhye State University 1(2), 44-53 (in Russian)
2. Anghern, A. (1997): Designing Mature Internet Business Strategies: the ICDT Model. European Management Journal, 15(4), 361-369
3. I3Net European research initiative. <http://www.i3net.org> – URL was accessed May 8, 2001
4. Papazoglou, M. P., Van der Heuvel, W.-J. (1999): From Business Processes to Cooperative Information Systems: an Information Agents Perspective. In: M. Klusch (ed.): Intelligent information agents: agent based information discovery and management on the Internet. Springer-Verlag, Berlin Heidelberg New York, pp. 10-36
5. Davulcu, H., Kifer, M., Pokorny, L. R., Ramakrishnan, C. R., Ramakrishnan I. V. (1999): Modeling and Analysis of Interactions in Virtual Enterprises. Proc. of the 9-th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE'99), Sidney, Australia, March 1999
6. OMG Unified Modeling Language. Specification. Version 1.3. June 1999. <http://www.rational.com/uml/resources/documentation/> – URL was accessed May 8, 2001
7. Lupu, E., Milosevic, Z., Sloman, M.: Use of Roles and Policies for Specifying, Building and Managing Virtual Enterprise. Proc. of the 9-th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE'99), Sidney,, Australia, March 1999
8. Jennings, N. R., Faratin, P., Johnson, M. J., Norman, T. J., O'Brien, P., Wiegand, M. E. (1996): Agent-based business process management. International Journal of Cooperative Information Systems. 5(2, 3), 105-130
9. Sycara, K., Decker, K., Pannu, A. Williamson, M. and Zeng, D. (1996): Distributed Intelligent Agents. IEEE Expert. Dec. 1996, 36-45
10. Ermolayev, V. A., Borue, S. U., Tolok, V. A., Keberle, N. G. (2000): Use of Diakoptics and Finite Automata for Modelling Virtual Information Space Agent Societies. Lecture Notes of Zaporozhye State University. 3(1), 34-44
11. Borue, S. U., Ermolayev, V. A., Tolok, V. A. (1999): On Diakoptical Approach to Process Modelling in Multi-Functional Information Systems. Proc. of International Conference Knowledge-Dialog-Solution (KDS'99), Kaciveli, Ukraine, September 1999, 211-219 (Russian)
12. Ermolayev, V. (2000): Dynamic Agent Communities Facilitating to Distant Learning in a Virtual University Information Space. Proc. of International Conference Emerging Technologies and New Challenges in Information Society (IS2000), Aizu-Wakamatsu, Japan, November 2000, 488-495
13. Ermolayev, V. A., Borue, S. U., Tolok, V. A. (2001): Co-operative Tasks Execution by the Coalitions of Rational Software Agents. Submitted as a regular paper to the 5-th Intl. Workshop on Cooperative Information Agents (CIA'01), Modena, Italy, September 2001
14. Nwana, H. S. (1996): Software Agents: an Overview. Knowledge Engineering Review. 11(3), 205-244
15. Foundation for Intelligent Physical Agents (FIPA) Spec.: DRAFT, Version 0.2, Agent Communication Language, 1999. <http://www.fipa.org> – URL was accessed May 8, 2001
16. Finin, T., Fritszon, R. (1994): KQML - A language for protocol and information exchange. Proc. of the 13-th DAI Workshop, Seattle, WA, USA, 1994
17. Sycara, K. (1999): In-Context Information Management through Adaptive Collaboration of Intelligent Agents. In: M. Klusch (ed.): Intelligent information agents: agent based information discovery and management on the Internet. Springer-Verlag, Berlin Heidelberg New York, pp. 78-99
18. Borue, S. U., Ermolayev, V. A., Tolok, V. A. (2000): Application of Diakoptical MAS Framework to Planning Process Modelling. Proc. of the 2-nd International Scientific - Practical Conference on Programming (UkrPROG'2000), Kiev, Ukraine, May 2000, 488-500.