Dynamic Agent Communities Facilitating to Distant Learning in a Virtual University Information Space

Vadim Ermolayev

***Abstract*--Proposed is the formal approach to the design of distant learning facilities for a Virtual/Real University[1] based upon the paradigms of intellectual software agent, multi-agent system, dynamic task-oriented agent community. The particularity of the framework presented is its capability to perform the tasks of information interchange 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 and assist in their tasks execution. The agents dynamically join the communities to perform the tasks, thus, modeling the processes of university management and distant learning. 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.

***Index Terms* -- Virtual University, Distant Education, Intelligent Agents, Agent Communities, Evolution.**

I. Introduction

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 branch (together with the Electronic Commerce) seems to be one of the most advanced on the way to the Information Society of the XXI-st century.

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 communities to facilitate to the execution of the business processes of distant education.

The very high idea lying in the basement 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 (local legacy IS). 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 corresponding organisational cells at different levels. From organisational point of view these components are virtual business objects performing business processes as workflows in terms of, say, the Enterprise Framework [4].

The particularity of the proposed formal 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 we meet in real life is difficult to be modelled by more or less static means provided by, say, ROOM, OOFRam role models [6], CTL based framework [5], ICRF [7]. Agents' paradigm obviously provides the way out of this world of predefined workflow and role specifications. The approach presented exploits the metaphor of dynamic agent community[2] in

Dept. of Mathematical Modelling and IT, Zaporozhye State University,
66, Zhukovskogo st., 69063, Zaporozhye, Ukraine,
e-mail: eva@zsu.zaporizhzhe.ua

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

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

order to provide better means for the modelling of the intrinsic dynamic character of the Universe of Discourse. 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 (policies). 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 agent communities to perform one or another task. The approach presented exploits the Diakoptical MAS framework [10], the model of task execution by agents community [11]. Human user interface designs are based upon the concept of UVII [12].

It is evident that the problem of modelling a virtual or a real university by means of VUIS inhabited by MAS is twofold. On the one hand, it is the problem of bridging semantic, operational and psychological gaps between human actors and their artificial IT counterparts as well as agent to agent interoperability. On the other side, it is the problem of interoperation among participating heterogeneous distributed components - i.e. operationally - communication, collaboration and co-ordination among these parties. The problem becomes even more complicated if the aspect of the evolution of the artificial actors (as well as of the wrapped information resources described by various data models) is taken into consideration.

In this paper we leave aside in-depth discussion of semantic aspects (ontology representation, interoperability, knowledge sharing etc.) as well as the aspects of co-ordination within dynamic agent communities. We concentrate on operational aspects and on evolution model as well as on the verification of the applicability of the approach to Distant Education domain.

The paper is structured as follows: Section II outlines the framework for the modelling of the processes of information interchange; Section III presents the approach to cope with VUIS inhabitants' evolution; the contribution of Section IV is the discussion of PhD students' recruiting case study; Section V summarises the results.

II. Framework for Modelling of the Processes of Information Interchange

VIS Functional Face is inhabited by MAS representing functional systems and components at various levels. The member agents dynamically form task oriented communities for the execution of the tasks of information acquisition, integration, mediation and interchange emerging within MAS of one or another level.

In real life these tasks are decomposed and executed by groups of functional components. Traditional models of the processes of information interchange are often based on the usage of structured collections of rigid relationships among the participating functional nodes. When modelled within the majority of the frameworks, the tasks (workflows) are represented by plans - pre-defined oriented graphs, Petri Nets [13], etc. These representations are often too much static. If we take the process of discussing the results of a laboratory work (Chemistry course) for an instance, it can not always be adequately modelled by, say, a hierarchy of rigidly positioned actors. Humans often apply much more soft interrelationships: brainstorming, informal discussion, whatever. Another big pitfall traditional representations suffer from is that rigid relationship models are not really scaleable.

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 agents[3]. 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. Community member agents act as the models of the functional components of the corresponding real world business object.

The framework for modelling of the processes of information interchange comprises the following components: functional system/component model [10], process model [11], 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

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].

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].

A. Generic Agent Model

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 (see [10, 18] for details) 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 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.

B. Communication Model

At the community level it is assumed that the agents taking part in the process of task execution communicate by means of the following communication acts (see [10] for the formal specification):

1. ***Directive*** - the routine to influence the counterpart to unconditionally execute the atomic work.
2. ***Determined request*** - the routine to influence the counterpart to execute the atomic work and to request the results back.
3. ***Determined request with results analysis*** - the routine to influence the counterpart to execute determined request and consequently to reason about the parametrical results received as the reaction.
4. ***Undetermined request with results analyses*** - the routine to broadcast the influence in case the executor agent is unknown. The influence is multicasted to every agent within the community. The results are afterwards compared and appropriate reasoning is performed.

These communicative acts comply with ACL [15] and KQML [16] capabilities.

C. Functional System/Component Model

At the functional system level some basic assumptions are made to simplify the framework and to provide the desired level of agents' reactivity. Functional system members are assumed to be strongly oriented to teamwork and fair play. Successful task execution has higher priority than the local goals of a certain agent. The agents joining the community are bound to deliver truthful results even if it contradicts to their local goals.

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 $W = \{w_1, w_2, ...\}$ 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:

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

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:

$$F_O^i(W) \to \tilde{W}_i, \qquad (1b)$$

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

Thus, the task of a functional component/system model is the appropriate execution of (1a) and (1b). For the moment the model is constrained by the rule that the duration of the execution of each atomic work $w_j \in W$ is the definite time interval Δt. It is however evident that this constraint is not very strong and may be overcome in the future.

D. Process Model

A functional system is tailored to perform processes. 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 community***.

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

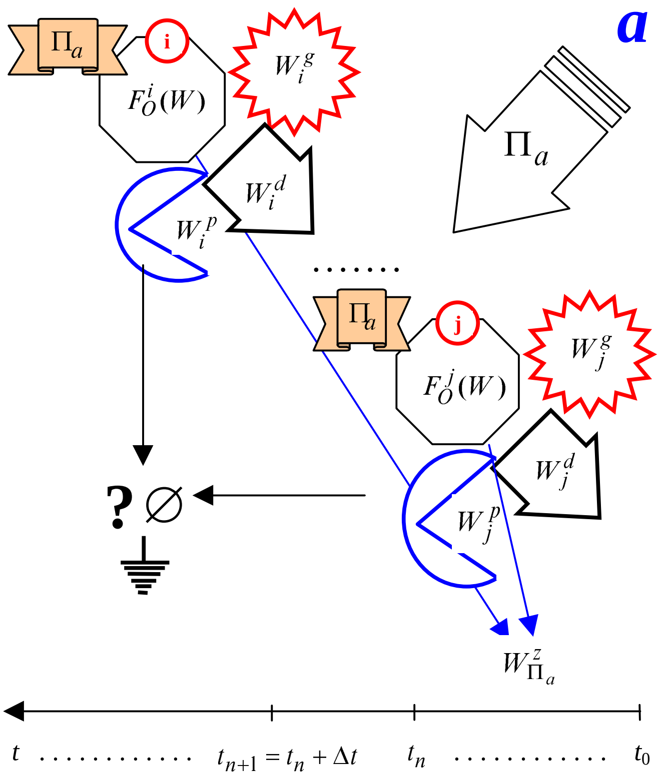

Fig.1. Process model.

in the process of Π_a is denoted as the set of ***inexecutable*** works. See the chart on Fig 1.

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.

Dependency $F_O^i(W)$ is modelled in frame of generic agent model [10], or by any other appropriate method. The requirement for this part of functional component model is the adequate execution of (1b), (1a).

For practice it is reasonable to restrict the set of system's permissible atomic works and to consider it to be finite: $W = \{w_1, w_2, ..., w_\sigma\}$. Modelling of a functional system as a whole while performing a task is herein organised as a two-level process performed sequentially at discrete time points t_n, $t_{n+1} = t_n + \Delta t$.

E. Task Execution Model

Let $W = \{w_1, w_2, ..., w_\sigma\}$ be the set of permissible atomic works of a functional system.

At the first (upper) level the assembly of all components' states into the conjoint system states model at $t_n + \Delta t$ is performed. The conjoint model is presented in the form of matrix $\Omega(t_n + \Delta t)$ with dimension $m \times \sigma$, where m is the number of system components and σ is the number of atomic works in W. The rows of matrix Ω (Fig. 2.) are the vectors $\Theta_i = \{k_1, k_2, ..., k_j, ..., k_\sigma\}$ reflecting components' states, where k_j is the state of the component *i* with respect to the execution of atomic work w_j. In the simplest case the role of parameter k_j is as follows:

$k_j = 0$ - the component is executing atomic work w_j;
$k_j = l > 0$ - the component is executing the work w_j and *l* similar works are waiting in line;
$k_j = l < 0$ - the component was capable but has not executed *l* atomic works w_j (was idle).

System states matrix $\Omega(t_n + \Delta t)$ is formed by Co-ordination Agent from the matrixes $\mathbf{K}_i$ (dimension $m \times \sigma$) representing component states. Matrixes $\mathbf{K}_i$ are produced by the executable $F_O^i(W)$ of the component model at the second modelling level to provide inputs to the following formula:

$$\Omega = \sum_{i=1}^{n} \mathbf{K}_i \, . \qquad (2)$$

One or more works w_j from W_i may be redirected by a functional component to itself for the execution at the next time point. In this case work delays vector D_a of the process Π_a is updated as follows:

$$D_a[j] = D_a[j] + 1 \qquad (3)$$

At the second (lower) level the production of $\mathbf{K}_i$ is performed by each system component (agent). The components, as mentioned before, are modelled arbitrarily ($F_O^i(W)$), but it is provided that the input information for component *i* is the vector Θ_i, the matrix $\Omega(t_n)$ and the matrix $\mathbf{K}_i$ defined for the previous time point t_n. Matrix $\mathbf{K}_i$ is built according to the rule presented at Fig. 3. and thus reflects the behaviour of the component within the time interval $]t_n, t_n + \Delta t]$,

where:

k_{lj}, $l \neq i$: **1** - component *i* allocates work w_j to component *l*,
0 - otherwise

k_{ij}: **-1** - component absorbs (or is capable to perform) work w_j within interval $]t_n, t_n + \Delta t]$,
1 - component *i* allocates work w_j to itself,
0 - component *i* is not capable to perform work w_j within given time interval $]t_n, t_n + \Delta t]$.

The analysis of $\Omega(t_n)$ values may thus provide to univocally evaluate component load, idle state share within each time interval $]t_n, t_n + \Delta t]$ and therefore to reason on the necessity of changing its behaviour $F_O^i(W)$ in the future.

III. The Evolution of the Actors and the Resources

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 -- i.e. the models applied must cope with various types of evolution. 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*

A. Capabilities' Evolution

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 (policy) 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: $S_A = \{s_1, \ldots, s_n\}$ - is denoted as the set of 3-nested tuples $s_i, i = 1, \ldots, n$:

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

where:

$r(X_A)$ - 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 policies of agent A (policy constraints),

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

B. Beliefs' Evolution

Beliefs' evolution is closely tightened to the monitoring of task community members' capabilities to perform works. According to [10, 11] inter-agent communication and work execution is organised via parametric feedbacks, comprising the information on the current capabilities to

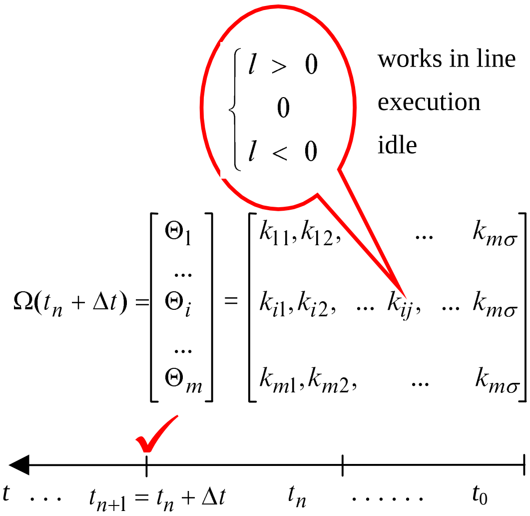

Fig. 2. System state at $t_n + \Delta t$

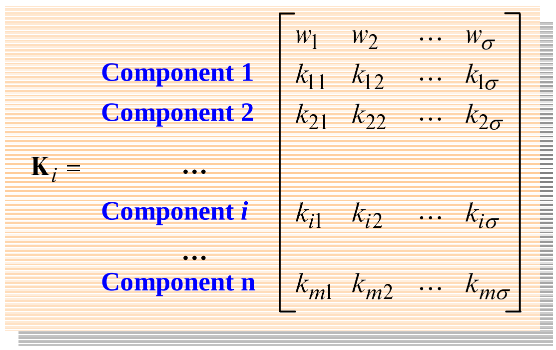

Fig. 3. Component capabilities and intentions within $]t_n, t_n + \Delta t]$.

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

An agent (representing a real world functional component) 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:

$$\mathbf{C} = \begin{array}{c} \\ A_1 \\ \ldots \\ A_i \\ \ldots \\ A_n \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & \ldots & w_j & \ldots & w_m \end{array} \\ \left[\begin{array}{ccccc} c_1^1 & & c_1^j & & c_1^m \\ & & \ldots & & \\ & \ldots & c_i^j = c_{A_i}^{w_j}(X_{w_j}) & \ldots & \\ & & \ldots & & \\ c_n^1 & & c_n^j & & c_n^m \end{array} \right] \end{array}, \qquad (5)$$

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

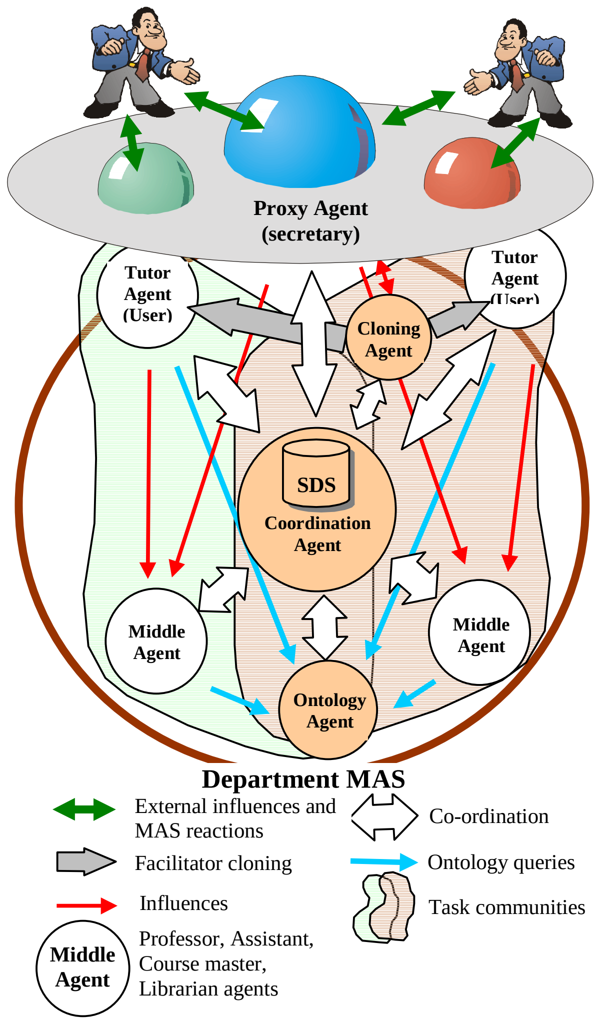

Fig. 4. Virtual Department MAS structure

the set *W* of permissible atomic works of the mentioned MAS.

C. *Changes in Information Resources*

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.

IV. Case Study: PhD Recruiting

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.

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

A. *Virtual Department*

We also presume that a Virtual Department is the MAS, comprising at least the following actors (Fig. 4):

- 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.

B. *PhD Recruiting Scenario*

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 **PRA**s. **PRA**s feed back with their parametric attitudes, having candidate's qualifications as parameters. **TA** determines the best match in case the feedbacks from some **PRA**s 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 than 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].

C. Testing Phase Modelling

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.

At $t = t_n$ **TA** accepts the set of works

$W_{TA} = \{ w_1 = (\text{'}\textit{Require the test}\text{'}, X_1, Y_1)\}$

with the parameters and result descriptions for w_1:

$X_1 = \{$***Edu_Rating*** $=<$ *structure*, **ontology** $=$ *Edu_Rating* $>$,
Q_E_Rating $=<$ *structure*, **ontology** $=$ *Qualif_Exp_Rating* $>$,
Pub_Rating $=<$ *structure*, **ontology** $=$ *Publication_Rating* $>$,
Professor $=<$ *Id*, **ontology** $=$ *Agent_Name* $>\}$,

$Y_1 = \varnothing$.

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 $\tilde{W}_{TA} = \{W^d_{TA}, W^g_{TA}\}$, where:

$W^d_{TA} = \varnothing$ as far as work w_1 is executed and no more works are left for redirection;

$$W^g_{TA} = \{\ w_2 = (\text{'}Provide\_Test\text{'}, X_2, Y_2),$$
$$w_3 = (\text{'}Test\text{'}, X_3, Y_3),$$
$$w_4 = (\text{'}Evaluate\_Results\text{'}, X_4, Y_4),$$
$$w_5 = (\text{'}Analyse\_Marks\text{'}, X_5, Y_5)\}.$$

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

$$W_{PRA} = \{\ w_2 = (\text{'}Provide\_test\text{'}, X_2, Y_2),$$
$$w_4 = (\text{'}Evaluate\_Results\text{'}, X_4, Y_4)\}.$$

Work w_2 is executed and the results

$\tilde{Y}_2 = \{Test\_Form =< FILENAME >\}$

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

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

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 $\tilde{Y}_{PRA} = \{\tilde{y}^1_4 = (m_1, m_2, ..., m_k), \tilde{y}^2_4 = (s_1, s_2, ..., s_k)\}$ 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

$W_{TA} = \{w_5 = (\text{'}Analyse\_Marks\text{'}, X_5, Y_5)\}$

with

$X_5 = \{$***Marks*** $=< \tilde{y}^1_4$, **ontology** $=$ *Mark_per_Project* $>$,

Scale $=< \tilde{y}^2_4$, **ontology** $=$ *Positive_Mark_per_Project* $>\}$

and decides if the candidate may be successful with respect to one of the project vacancies. In case of success W^g_{TA} will contain

$w_6 = (\text{'}Require\_the\_Interview\text{'}, X_6 = \{X_1, X_5\}, Y_6)$, otherwise **TA** generates

$w_7 = ('Inform\_on\_Failure', X_7, Y_7)$.

V. SUMMARY

The paper presents the approach to apply the formal agent-based framework for the modelling of the processes of information interchange to the design of a VUIS inhabited by agents, that dynamically form communities to facilitate to the execution of business processes of distant education. 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 the sub-ordinate MAS having the same generic architecture.

The framework is based upon the paradigms of intellectual software agent, multi-agent system, dynamic task-oriented agent community. The particularity of the framework is its capability to perform the tasks of information interchange 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 middle agents. 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. The workflow is thus being developed in more and more details within step-by-step execution process. The process is collaboratively conducted by the team of community agents acting in co-operation with each other.

The scalability of the framework models is based upon the principle of diakoptics [19] as well as upon the generic character of the agents' architecture.

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 active functional components 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. In-depth evaluation of the proposed approach and the prototype design is planned for future work.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the comments of the anonymous reviewers that helped much to improve the paper.

REFERENCES

[1] Ermolayev, V. A., Pletsky, S. U., Tolok, V. A. The Architecture of the Unified Information Space of a Virtual University // "Lecture Notes of Zaporozhye State University", ISBN 966-599-007-1, Vol. 1, No 2, 1998, pp. 44-53 (in Russian).

[2] Anghern, A Designing Mature Internet Business Strategies: The ICDT Model.// European Management Journal, Vol. 15, No 4, 1997, pp. 361-369.

[3] I3Net European Research Initiative http://www.i3net.org

[4] Papazoglou, M. P., Van der Heuvel, W.-J. 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, 1999, pp. 10-36.

[5] Davulcu, H., Kifer, M., Pokorny, L. R., Ramakrishnan, C. R., Ramakrishnan I. V. Modeling and Analysis of Interactions in Virtual Enterprises. //In: Proc. of the 9-th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE'99), March 23-24, 1999, Sidney, Australia.

[6] OMG Unified Modeling Language. Specification. Version 1.3. June 1999. http://www.rational.com/uml/resources/documentation/

[7] Lupu, E., Milosevic, Z., Sloman, M. Use of Roles and Policies for Specifying, Building and Managing Virtual Enterprise. //In: Proc. of the 9-th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE'99) , March 23-24, 1999, Sidney, Australia.

[8] Jennings, N. R., Faratin, P., Johnson, M. J., Norman, T. J., O'Brien, P., Wiegand, M. E. Agent-based business process management.// Int. Journal of Cooperative Information Systems, 5(2, 3), pp. 105-130.

[9] Sycara, K., Decker, K., Pannu, A. Williamson, M. and Zeng, D. Distributed Intelligent Agents. // IEEE Expert, Dec. 1996, pp. 36-45.

[10] V. A. Ermolayev, S. U. Borue, V. A. Tolok, N. G. Keberle, Use of Diakoptics and Finite Automata for Modelling Virtual Information Space Agent Societies // "Lecture Notes of Zaporozhye State University", ISBN 966-599-058-4, Vol. 3, No 1, 2000, pp. 34-44.

[11] Borue, S. U., Ermolayev, V. A., Tolok, V. A. On Diacoptical Approach to Modelling of the Processes in Multi-Functional Information Systems // "Artificial Intelligence" Scientific Theoretical Journal №2, 1999, ISSN 1561-5359, Special Issue: Proc. of Intl Conference KDS'99, Katciveli, 13-18.09.1999, pp. 211-219 (in Russian).

[12] V. A. Ermolayev, V. A. Tolok, Interfaces and Human - Agent Interaction in Virtual Information Spaces, Panel talk at ESPIRIT AgentLink SIG on Intelligent Information Agents meeting, London, GB, Apr. 21-23, 1999, 10 p.

[13] Adam, N., Atluri, V. and Huang, W. Modeling and analysis of workflows using petri nets. // Journal of Intelligent Information Systems, 10(2), pp. 131-158, March 1998.

[14] Nwana, H. S. Software Agents: an Overview. // Knowledge Engineering Review, Vol. 11, No 3, pp. 205-244, Oct./Nov. 1996.

[15] Foundation for Intelligent Physical Agents (FIPA) Spec: DRAFT, Version 0.2, Agent Communication Language, 1999, http://www.fipa/org

[16] Finin, T. and Fritzson, R. KQML - A language for protocol and information exchange. // Proc 13th DAI workshop, pp. 127-136, Seattle, WA, USA.

[17] Sycara, K. 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, 1999, pp. 78-99.

[18] S. U. Borue, V. A. Ermolayev, V. A. Tolok: Application of Diakoptical MAS Framework to Planning Process Modelling // in: "Problems of Programming" Scientific Journal №1-2, 2000, ISBN 966-02-1244-5, Special Issue: Proc. of the 2-nd Intl. Scientific - Practical Conference on Programming (UkrPROG'2000), Kiev, 23-26 May 2000, pp. 488-500.

[19] Kron, G. Diakoptics. Macdonald, London, 1963.