computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

AGENT-BASED MODELING OF THE E-LEARNING USERS' BEHAVIOR

Viktor Artemenko

Lviv Academy of Commerce,
str. Tugan-Baranovsky, 10, Lviv, 79005, Ukraine
e-mail: artem@lac.lviv.ua

Abstract: In this paper the methodological and technical approaches are considered to construct the agent-based model (ABM) with built-in artificial neural networks. This model describes the user's behavior of electronic or distance learning in a virtual environment of one of universities. Design of ABM aims to support a development of Web analytics (learning analytics) on the basis of computer experiments evaluating the trends of the knowledge creation and dissemination for e-Learning users. In research AnyLogic software, which is one of the most popular packages for agent-based modeling, is proposed for implementing the created model, and package STATISTICA Neural Networks is proposed for constructing the neural networks. *Copyright © Research Institute for Intelligent Computer Systems, 2014. All rights reserved.*

Keywords: e-Learning, agent-based models, artificial neural networks, software for agent-based modeling.

1. INTRODUCTION

Information and communication technologies (ICT) have a significant impact on almost every aspect of our lives. Under favorable conditions, ICTs become tools of socio-economic development, job creation and continuous improvement of quality of life. Giving each person an opportunity to have access to information and knowledge and to contribute to this field is an important element of inclusive information society, lifelong learning based on ICTs and distance learning technologies.

Distance learning or e-Learning can be seen as an example of a system that consists of agents and has the features of a living structure, as there is a constant exchange of information and knowledge between agents and learning management system.

The term "agent" is widely used in the literature. For example computer models, in which agents are the objects of modeling, called agent-based models or abbreviated ABM [1].

In this article we aim to describe the role and the place of ABM. Among specialized publications dealing with the ABM issues we can refer to the following two [2, 3] and online journals: "Journal of Artificial Societies and Social Simulation" or in short JASSS [4], "Agent-Based Computational Economics" [5] and "Artificial Society" [6]. These sources are devoted to problems of simulation in the social and economic sciences, and development of methodology to obtain new knowledge on the basis of computer experiments with ABM. The ultimate goal of the ABM design is to collect a certain set of rules to conduct specific agents representing an artificial community. Encyclopedia Wikipedia defines this concept as: "Artificial Society is the specific agent-based computational model for computer simulation in social analysis".

Among the ABM software the most popular packages are SWARM and AnyLogic. The first package is a collection of open source software libraries created at the Institute of Santa Fe and available on its website [7]. Some of the libraries are written using scripting language to ensure the use of graphical tools. AnyLogic is a commercial package. Its special AnyLogic agent based library allows specifying the required functionality of the model's agents [8-10].

Analysis of the known examples of agent-based models leads to the following conclusions.

1. Agent is a theoretical abstraction used in the agent-based models in order to represent actors, members of the artificial society, who make decisions with a certain degree of autonomy. The agent can "live" in time, to have a graphical image, respond to actions from the sensors, external programs or specific agents-users.

2. The vast majority of agent-based models are abstract and the main purpose of their creation is a scientific interest, which means the usage of conditional data to test new tools.

3. Among the agent-based models, considering the real phenomena, only a small part is related to the interactions of agents in the field of education.

In our research we aim to study in details approaches, presented earlier in [11], to the agent-based modeling of the e-learning users' behavior with help of artificial neural networks. The creation of such agent-based model is aimed at preparing analytics, which would show trends in the production and dissemination of knowledge of users in distance courses. The results of the ABM testing are based on using the following software: AnyLogic and STATISTICA Neural Networks.

2. APPROACHES TO THE CONSTRUCTION OF AGENT-BASED MODEL OF E-LEARNING USERS

Specifications task of ABM development aimed at creating an artificial society in which interacting agents of such types: A1 – authors of distance courses, A2 – tutors involved in the learning process and A3 – students, users of distance courses. We assume that the agents of artificial society are functioned autonomously. They make decisions, perform actions and interact with other agents of e-learning. Any decision of the agent is sufficient to effect the action. To implement the same interaction is necessary a group decision. Group decisions are subject to certain rules, which must be specified in this case.

Thus, the rules, designed for the agents, who interact in a two-dimensional virtual environment, play the key role in the ABM development.

It is important to note that in our case, during the computer simulation experiments with the means of AnyLogic, two-dimensional virtual environment is presented as a rectangular lattice, which consists of 100 columns and 100 rows (see Fig. 9). The interactions between e-Learning agents take place in each cell (x, y). The virtual environment is limited by the parameters, which are designed on the basis of the analyzed simulation period, namely academic semester (100 days). Any agent that reaches the limits of the simulation period "dies". By that time the authors and tutors of the distance courses fully produce and transmit knowledge, and students receive the full amount of that knowledge.

The move of the analyzed agents in this environment is determined by their behavior during the production and dissemination of knowledge. The main tools in the specification of the behavior of agents are the following variables: timers and statecharts. These variables reflect characteristics of the analyzed agents. Timers can be setup to a certain time, to program when the action will take place. Statecharts allow visually present the behavior of the agents over the time under the influence of events or conditions. They consist of a graphical description of states and transitions between them. Any complex logic behavior of agents in AnyLogic models can be expressed by using a combination of statecharts, algebraic equations, variables, timers and program code in Java.

Statechart allows graphically specify the space between states of the agent behavior algorithms, factors that are driving the transitions from one state to another, as well as the actions that occur when states are changing. Statechart are compliant to UML standards. They store graphics, performance attributes and semantics, which are defined by UML (Unified Modeling Language). Statechart in AnyLogic can support the following types of events: 1) signal – the agent can send a signal to another agent to tell him or her something, 2) timeout – over a given period of time in the statechart nothing happens, 3) event of transition – an event in which the Boolean value indicates "true".

The position of the agent in a virtual environment is defined by variables x and y. The agent moves randomly in the process of computer simulation experiments. Each agent initially moves in the direction to the axis X, causing the method RandomMove, and then along to the axis Y, causing this method again. Method RandomMove uses a simple numerical generator of mathematical library Java. It defines the value -1 for the move back; 0 – for the standing still or +1 – for the move forward (AnyLogic allows you to use other random number generators in calibrating ABM). Henceforth, these movements are formalized by using statecharts (see Fig. 6-8).

The movement of agents in a virtual environment is determined by the agents' network or by the location type. There are two the most commonly used among them:

1. *Random placement* – agents are connected by chance, each agent is determined by the specified number of connections.
2. *According to the distance* –agents are connected with each other if the distance between them is less than the specified range of connection (only in the continuous environment).

In our research we use the first type of agents' placement. Therefore, the location of the agents at time "0" is determined by variables x and y. The initial values of variables are determined randomly in a rectangular lattice on the basis of statistical data, which describes the actions of agents (see below). Directions of their movement are determined by the rules given in statecharts (see Fig. 6-8). The movement of students is defined by statecharts as well. The example of their interaction with the authors of e-learning courses is presented in Fig. 8.

Please note that the concept of the *time modeling* is a base in the agent-based modeling. This time is a relatively logical time, which is defined in terms of the behavior of all objects in the model. It consists of a starting time, start date and end date. Two last dates are set by the built-in AnyLogic calendar. Start time is determined by "0" in the settings of computer experiments. This fact explains the expression t+0, which is implied in the following formulas (1)-(12).

Therefore, the key assumptions in the agent-based modeling are as follows:

- in terms of decision-making, agents e-learning move in a two-dimensional space and having a finite horizon of vision;
- agents of the distance courses interact in a virtual environment by the defined rules, have a finite lifetime;
- goal of author of distance course (agent first type) – produce as much knowledge and transmit them to the tutor and students, goal of tutor (agent second type) – to spread knowledge among the greatest possible number of students, and goal of students (third type of agent) – as much as possible to use knowledge.

Behavior of agents e-learning can be described as follows.

Behavior of the author of a distance course. Author of a distance course moves with some predetermined speed in a random direction to search for other agents of e-learning. The search explained by the fact that the demand for training courses for the choice of the student, including and distance learning courses, largely determined by the "Run Games" author of a distance course's, that is, its advertising proposed "by Hartley" information, which can reduce the uncertainty of the agents, the recipients of such information. The author's task is to bring the course to other agents of e-learning. If the line of sight was another agent, the following happens. Author of a distance course rushes to him, until the distance will not be minimal. Then there is their stop, where there are three possible cases.

1. At a meeting of the two authors of distance learning courses for both is proportional to the amount of knowledge gain knowledge of the interlocutor. After the meeting, agents differ and some time available for subsequent contacts.

2. At a meeting with a tutor from the author of a distance course knowledge gain occurs and growth occurs indicator characterizing the transfer of knowledge. At tutor knowledge gain is proportional to the amount of knowledge of the author of a distance course.

3. At a meeting with the student at the author of a distance course is increase (significantly less than at a meeting with the tutor) of the transmitted knowledge, and the student, in turn, increase consumption of knowledge.

The behavior of a tutor. The tutor also moves randomly. Meeting with the author of a distance course described above. At a meeting with another tutor is a mutual exchange of experience, that is, both the amount of knowledge grows proportionally to the volume of knowledge interlocutor. At a meeting with the student is learning, in which the tutor grows in value of the variable, showing the amount of transferred knowledge, the student accordingly their consumption.

The behavior of a student. The interactions of a student with the author and tutor of a distance course described above. At a meeting between the students is an exchange of knowledge is proportional to the accumulated knowledge of the interlocutor.

If an agent sees several other agents, he rushes to the one that gives the maximum gain his objective function.

A formal description of the model.

Legend: a_i – number of the author's of a distance course; a_j – number of the tutor; a_s – number of the student, members of a distance course; $k(t,a_i)$, $k(t,a_j)$, $k(t,a_s)$ – volumes accumulated or available knowledge of the author, tutor and student; $w(t,a_i)$, $w(t,a_j)$, $w(t,a_s)$ – volumes transferred to other agents of knowledge by the author, tutor and student, respectively; x – coordinate along the abscissa (varies depending on the speed); y – coordinate along the ordinate (varies depending on the speed); vx – speed along the horizontal axis (may change when the agent saw interlocutor – the maximum speed); vy – speed on the y-axis (may change if the agent saw the interlocutor – the maximum speed, and after a conversation with the agent); $d1$, $d2$, $d3$, $d4$, $d5$, $d6$ – share knowledge, which is passed from agent to agent; L_a – lifetime agent *a*; K_i, K_j, K_s – the total volumes knowledge produced by the authors, tutors, students of distance learning courses, respectively; K – total amount of knowledge in the analyzed virtual web center; p_i, p_j, p_s – probability of appearance the author, tutor and student of a distance course respectively.

Behaviour the author's of a distance course a_i at the time *t* leads to a state which is described by the following set of parameters: $k(t,\, a_i)$, $w(t,\, a_i)$, x, y, vx, vy.

Tutor and student status is defined similarly.

The meeting of the two authors of a distance learning courses is described as follows:

$$k(t+0,a_{i1}) = k(t,a_{i1}) + d1k(t,a_{i2}), \quad (1)$$

$$k(t+0,a_{i2}) = k(t,a_{i2}) + d1k(t,a_{i1}), \quad (2)$$

$w(t,a_{i1})$, $w(t,a_{i2})$ remain the same.

Meeting the author and tutor of a distance course:

$$w(t+0,a_{i}) = w(t,a_{i}) + d2k(t,a_{i}), \quad (3)$$

$$k(t+0,a_{j}) = k(t,a_{j}) + d3k(t,a_{i}), \quad (4)$$

$k(t,a_{i})$, $w(t,a_{j})$ remain the same.

The meeting between the author and a student of a distance course:

$$w(t+0,a_{i}) = w(t,a_{i}) + d4k(t,a_{i}), \quad (5)$$

$$k(t+0,a_{s}) = k(t,a_{s}) + d4k(t,a_{i}), \quad (6)$$

$k(t,a_{i})$ remain the same.

The meeting between the tutor and the student:

$$w(t+0,a_{j}) = w(t,a_{j}) + d5k(t,a_{j}), \quad (7)$$

$$k(t+0,a_{s}) = k(t,a_{s}) + d5k(t,a_{j}), \quad (8)$$

$k(t,a_{j})$ remain the same.

The meeting of two students:

$$k(t+0,a_{s1}) = k(t,a_{s1}) + d6k(t,a_{s2}), \quad (9)$$

$$k(t+0,a_{s2}) = k(t,a_{s2}) + d6k(t,a_{s1}), \quad (10)$$

$$w(t+0,a_{s1}) = w(t,a_{s1}) + d6k(t,a_{s2}), \quad (11)$$

$$w(t+0,a_{s2}) = w(t,a_{s2}) + d6k(t,a_{s1}). \quad (12)$$

Results of agent-based modeling in a SWARM environment are presented in the paper [12].

We aim to investigate the effectiveness of the AnyLogic package for designing a hybrid agent-based model, which assesses knowledge of the users of distance learning courses on the basis of neural networks, which belong to the new frontier of artificial intelligence. The review of scientific papers in this area [2, 3] suggests that the results of neural networks, which are based on a large number of observations, reflect reality better than expert models, which survey a small number of experts, or fuzzy logic systems, which use the rules laid down by several people.

The conceptual scheme of hybrid ABM of the users of e-learning is presented on the Fig. 1.

From this figure, one may see that in the process of construction of ABM three artificial societies are going to be considered. Each of societies presents the set of agents who are decision-makers in the process of production and dissemination of the knowledge in the virtual environment (Learning Management System – LMS) Moodle. The support of decisions is provided by the means of artificial neural networks. The support of decisions is provided by the means of artificial neural networks.

With the purpose of designing artificial neural networks, we use the data from the activity of web-centre users in one of Ukraine's higher education institutions [13]. Based on the number of clicks, published in the Moodle event log system, it is possible to monitor the performance of the e-learning agents over the past twelve months. It is possible to monitor how the distance course resources were viewed, updated, or what was deleted and added to the tasks. Our research uses the data which characterizes 117000 actions by more than 500 agents. Almost 10% of the agents are authors, approximately 20% are tutors, and the rest are students.

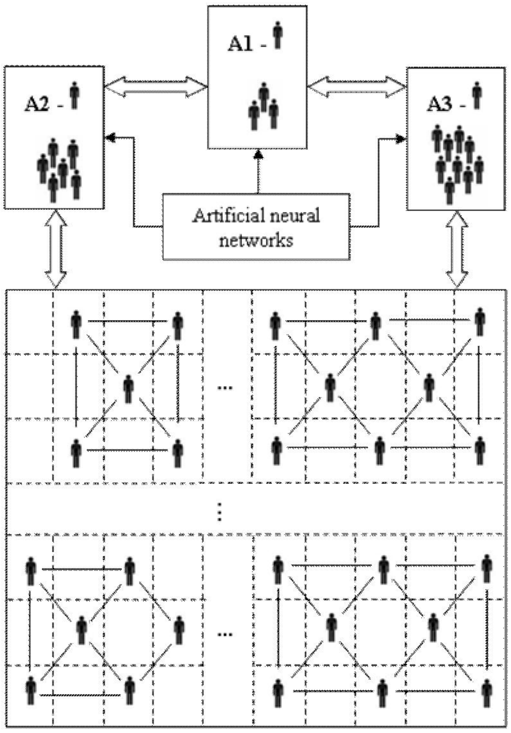


Fig. 1 – The conceptual scheme of hybrid ABM of the users of e-learning

All the data describing the clicks in the event log of the Moodle e-learning management system are divided into the following nonoverlapping click activity data groups: (1) the authors of distance course, (2) tutors and (3) students, participating in the distance course.

On the basis of these data, the number of clicks of *j*-type agent $\tilde{x}^{(j)}$ $(j=1,2,3)$ is determined. Each group includes a provisional reference value $\tilde{x}_0^{(j)}$ which is defined within the specified interval of this indicator (between the smallest and the largest possible value of $\tilde{x}_{\min}^{(j)}$ and $\tilde{x}_{\max}^{(j)}$ accordingly) and corresponds to better quality. In this case, expert opinions or certain normative indicators may be considered provisional reference values.

Each of the analyzed parameters of a certain $x^{(j)}$ $(j=1,2,3)$ group is provided by conversion

$$x^{(j)} = 1 - \frac{\left|\tilde{x}^{(j)} - \tilde{x}_0^{(j)}\right|}{\max\left\{\left(\tilde{x}_0^{(j)} - \tilde{x}_{\min}^{(j)}\right), \left(\tilde{x}_{\max}^{(j)} - \tilde{x}_0^{(j)}\right)\right\}}, \quad (13)$$

in which the area of its possible values is defined as a range $[0,1]$. In this case, the zero value of the transformed indicator $x^{(j)}$ will signify the smallest quality of the defined specification while the unit value will indicate the highest quality.

So, if the indicators of a number of clicks are specified under the defined above groups, then it is possible to pass on to the specified $x^{(j)}$ parameters, linked with the input variables $\tilde{x}^{(j)}$ by using transformation (13) above.

Clearly, for each *j*-group the normalized variable $x^{(j)}$ can range from $x^{(j)}=0$ (which corresponds to the worst quality) to $x^{(j)}=1$ (which corresponds to the best quality).

On the basis of transformation results (13) we can proceed to the construction of artificial neural networks. This problem is solved in two stages:

1) by choosing type (architecture) of the network,

2) by selection of network weights (training).

In Fig. 2 are shown these stages of construction and the use of artificial neural networks for subsequent incorporation in ABM.

On the first stage, for each type of e-learning agents (or group), we choose the most effective architecture of neural network. The most popular and well-studied architectures are as follows: multilayer perceptron, linear networks, probabilistic neural networks, generalized regression neural network and others.

On the second stage we have to “teach” the chosen network: that is to find such values of its weights so the network can work satisfactorily. There are specific “teaching” algorithms developed for much architecture, which allow adjusting network weights in a certain way (see Table 1).

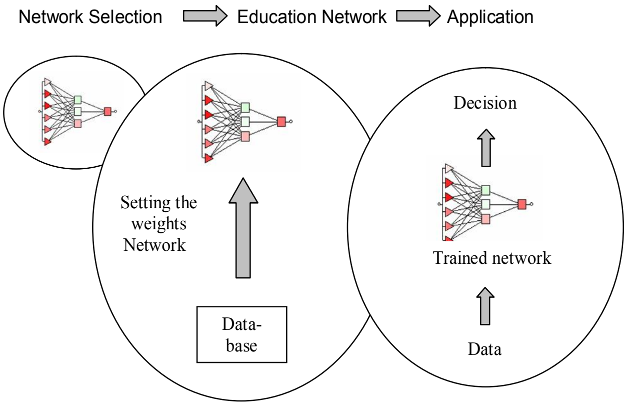


Fig. 2 – Stages of construction and application of neural networks

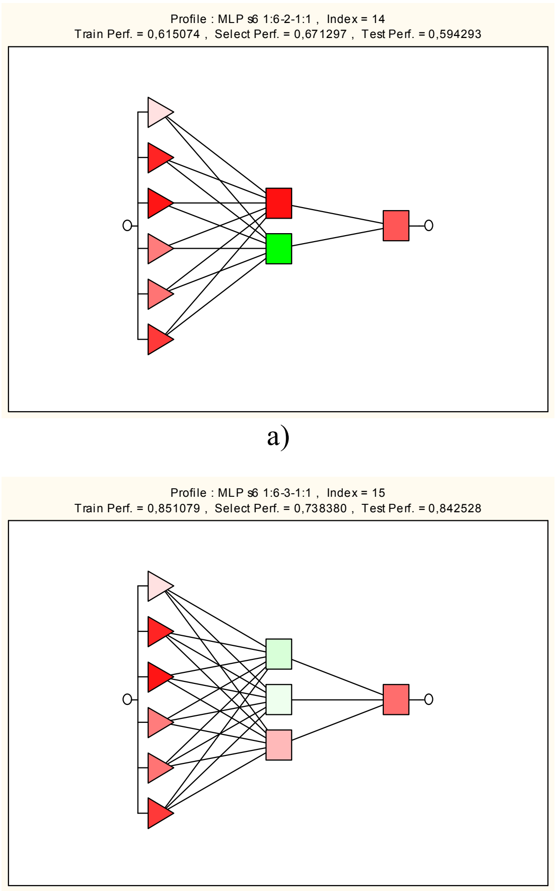
Table 1. The main types of neural networks and algorithms for their teaching.

Multilayer perceptron (MP) The three-layered MP (there is 1 hidden layer) simulates a problem of almost any complexity. MP network can be trained on the basis of such algorithms: - Conjugate gradient method, - Quasi-Newton, - Levenberg-Marquardt, - Converse and Rapid spread, - Delta-delta with a dash.	**Radial basis function (RBF)** These networks have an input layer, hidden layer with radial elements and an output layer of linear elements. RBF relatively quick to learn on-the basis of linear optimization. You can also use the learning algorithms and MP. However, RBF is larger than the multilayer perceptron and therefore are slower.
Linear network (LAN) LAN has only two layers: input and output, the latter has a linear post-synaptic potential function and the activation function. Linear network is better trained by the method of pseudoinverse. This method optimizes the last layer of any network, if it is linear.	**Bayesian networks (PNN, GRNN):** - probabilistic neural network is used only in classification issues, - the generalized regression neural network is used only in regression issues. These networks are trained on the basis of cluster algorithms: k-means, etc.

It's important to note, that the network contains the information estimating production trends and the dissemination of knowledge by agents. This information is recoded from the set of e-learning agents' clicks. Therefore, the quality of network teaching depends on the number of examples in the training set, as well as on how well these examples describe the task.

We have chosen the *STATISTICA Neural Networks* package among other software that is designed to build neural networks [14]. There is a tool called “Problem Solver” in the package, which provides for the construction of neural networks sets with superior characteristics. “Network Constructor”, which includes neural networks selection and teaching with the consideration of advanced users requirements, can also be applied.

Fig. 3 shows examples of artificial neural networks developed by means of *STATISTICA Neural Networks* for a hybrid agent-based model.


Profile : MLP s6 1:6-2-1:1 , Index = 14
Train Perf. = 0,615074 , Select Perf. = 0,671297 , Test Perf. = 0,594293
Profile : MLP s6 1:6-3-1:1 , Index = 15
Train Perf. = 0,851079 , Select Perf. = 0,738380 , Test Perf. = 0,842528

a)

в)

Fig. 3 – Network architectures of master-making STATISTICA Neural Networks

From Fig. 3 one may see that neural networks have identical architecture which is represented by multilayer perceptrons. The differences between these neural networks are in the number of hidden layer neurons and some other parameters.

Numerical values of parameters of neural networks, shown in Fig. 3, are not as important, as the emphasis in this paper is made on the methodological approaches to the development of hybrid ABM. Therefore, we do not need to focus on the technical issues of implementing the neural networks, just simply list below their main purpose for the created agent-based model.

❖ Neural network # 1 (NN#1) determines the activity of agents of the first type based on the forecasting of the possible number of clicks and thus evaluates the level of the knowledge produced and distributed by the e-learning course authors.

❖ Neural network # 2 (NN#2) determines the activity of the second type of agents on the basis of forecasting the possible number of clicks and thus evaluates the level of knowledge distributed by the tutors of a e-learning course.

❖ Neural network # 3 (NN#3) determines the activity of three types of agents on the basis of forecasting the possible number of clicks and thus evaluates the level of knowledge consumed by students participating in e-learning.

So the neural network allegedly accepts a decision of e-learning agents, obtaining an input variable characterizing the growth in usefulness and value of knowledge for the participants of an artificial society of this type. The number of clicks, measured with allowances made for long term threshold values, allows for assessing trends of knowledge production and dissemination in the hybrid agent-based model by agents on the basis of the activity results of each e-learning user.

Note also that *STATISTICA Neural Networks* has a code generator, a separate module that provides this package with an opportunity to create the equivalent of a teaching network as a non-compiled code in C/C++ or Visual Basic. Every calculation and installation of artificial neural network is unlocked and available for viewing user, copy or relevant changes.

Code fragments of a neural network, developed this way, can be embedded as functions for code use in other applications. A huge advantage of this feature is that the *STATISTICA Neural Networks* does not need to be installed on the computer where the generated code is executed.

3. THE IMPLEMENTATION OF A PROTOTYPE HYBRID ABM WITH ANYLOGIC

Agent-based hybrid model was developed in the *AnyLogic* environment. Among utility tools are variables, timers and statecharts (flow charts or diagrams). Variables reflect the change in the characteristics of e-learning course agents. Timers are set for a specific time interval, after which the specified action will be performed. Statecharts make possible to visually present agent's behaviour in time, under the influence of events or conditions, which consists of graphic states and transitions between them. Any complex behavioural logic of the agents in the hybrid model designed in *AnyLogic* can be expressed by a combination of utility tools, as well as embedded code functions of three previously designed neural networks.

To create *AnyLogic* model, we have to create classes of active objects (or use *AnyLogic* library of facilities). With recognition the active objects we create template and particular objects, built in accordance with this template (active object). Those objects can be used as elements of other active objects: avtorDK [..], tutor [..], student [..]. One class is always the root class in *AnyLogic* model. For this root class we generate a copy in the ABM hybrid. The predefined name for the copy is **“Main”**. *AnyLogic* system starts execution with this copy.

Variables which are used to describe the hybrid model have the same structure. The diagram in Fig. 4 explains the variables.

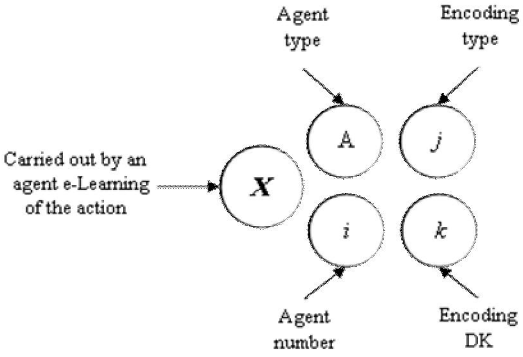


Fig. 4 – Identification of variables in the ABM.

X indicates the action (click), carried by the e-learning agent. Examples of such actions could be as follows: P – browse the resources and tasks in distance course (DK); O – update resources and assignments in DK; D – add resources and tasks in DK; U – remove resources and tasks in DK. As a rule, it is clear from the context which action is involved.

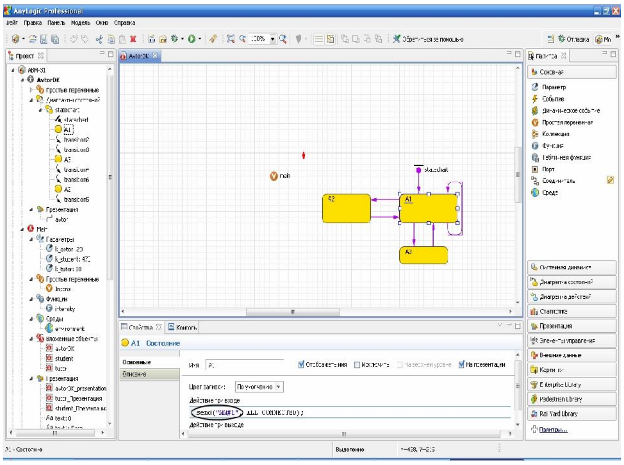
The index *k* determines the electronic distance learning course in which the agent performs its action. The agent’s number is marked with the index *i*, and *j* is the index which indicates the type of agent (A1 – author, A2 – tutor, A3 – student).

Fig. 5 shows an example of agent behaviour specification, which reflects the interaction between authors of distance learning courses, tutors and students participating in e-learning.

As can be seen from Fig. 5, the “Entry action” defines the mutual action of the agents using an artificial neural network #1 (NN#1) above. This network sends the message with a prediction about the amount of clicks by the authors of electronic distance courses to all e-learning agents and thus evaluating the level of produced and distributed knowledge.

Exit action provides for the following three cases.

1. During a contact of two authors distance courses both increase knowledge proportionally with the interlocutor’s knowledge that is presented on the Fig. 6.



Имя A1 ☑ Отображать имя ☐ Исключить ☐ На верхнем уровне ☑ На презентации

Цвет заливки: По умолчанию

Действие при входе

```
send("NN#1", ALL_CONNECTED);
```

Действие при выходе

Fig. 5 – Agent-Based Model Specification Example.

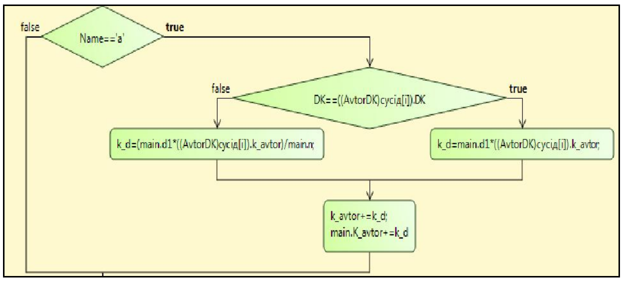

false
Name=='a'
true
false
DK==((AvtorDK)сусід[i]).DK
true
k_d=(main.d1*((AvtorDK)сусід[i]).k_avtor)/main.n;
k_d=main.d1*((AvtorDK)сусід[i]).k_avtor;
k_avtor+=k_d;
main.K_avtor+=k_d

Fig. 6 – Statechart of knowledge transfer algorithm – if AvtorDK meet with AvtorDK.

2. When the distance course author contacts a tutor, he does not gain any knowledge, although the knowledge transfer indicator increases. The tutor gains knowledge proportionally to the author’s knowledge of the distance course (Fig. 7).

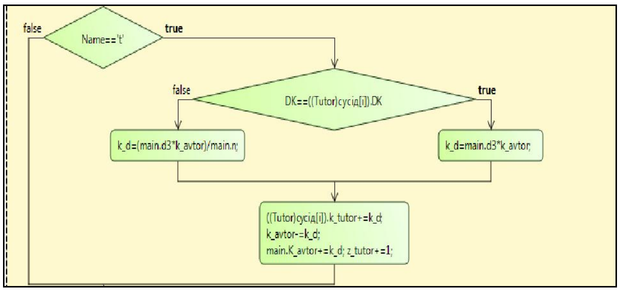

false
Name=='t'
true
false
DK==((Tutor)сусід[i]).DK
true
k_d=(main.d3*k_avtor)/main.n;
k_d=main.d3*k_avtor;
((Tutor)сусід[i]).k_tutor+=k_d;
k_avtor-=k_d;
main.K_avtor+=k_d; z_tutor+=1;

Fig. 7 – Statechart of knowledge transfer algorithm – if AvtorDK meet with Tutor.

3. When the author contacts a student, he gains substantially less knowledge than from contacting with the e-learning course tutor. At the same time the student’s knowledge consumption increases (Fig. 8).

Behaviour of Tutor’s (A2) and student’s (A3) is described in the first Model using neural networks #2 and #3 above.

The designed agent-based model for e-learning allows participants to observe visually their actions.

To view the values of characteristic variables for any agent, you can use the control panel system to pause the simulation.

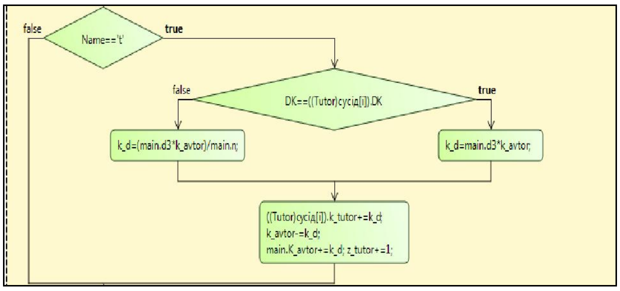

false
Name=='t'
true
false
DK==((Tutor)сусід[i]).DK
true
k_d=(main.d3*k_avtor)/main.n;
k_d=main.d3*k_avtor;
((Tutor)сусід[i]).k_tutor+=k_d;
k_avtor-=k_d;
main.K_avtor+=k_d; z_tutor+=1;

Fig. 8 – Statechart of knowledge transfer algorithm – if AuthorDK meet with Students.

With the help of created model, we may see how much knowledge per type of agent was produced in general and on average (Fig. 9).

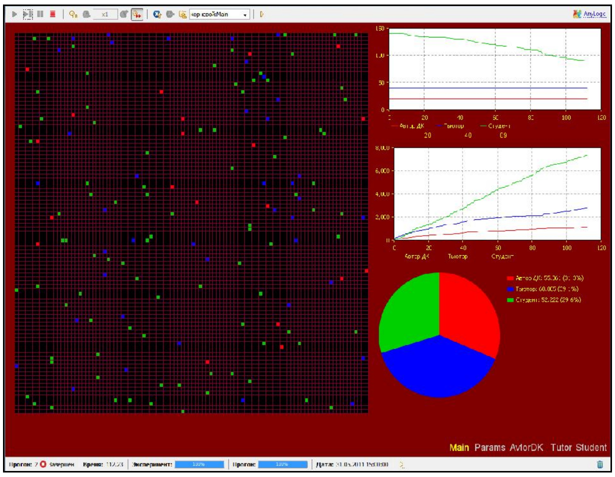


Fig. 9 – The main window of computer experiments.

Here, the upper graph shows the dynamics of change in the number of agents of each type. The X axis shows time of courses, and the axis Y – number of agents. The middle graph represents the amount of knowledge produced by each type of agents. Such a large number of consumed knowledge among the students, represented in the graph, is due to the fact that students rate in comparison with others is remarkably higher: number of students – 140, number of tutors – 40, DK authors – 20. The pie chart represents the average number of produced and consumed knowledge for each type of agents.

We can also observe the statistics of the residual amount of knowledge and the number of meetings each agent after of computer simulation experiments (Fig. 10).

Substantive interpretation of diagrams is shown (see Fig. 10). The top chart shows the amount of knowledge (variable k_avtor) for each AvtorDK. Other charts below describe the number of meetings of each agent with other types of agents. For example agent A(3), for which the value variable k_avtor = 30.304, met in computer simulations 0 times with other DK authors, 3 times with other tutors, and 14 times with the students.

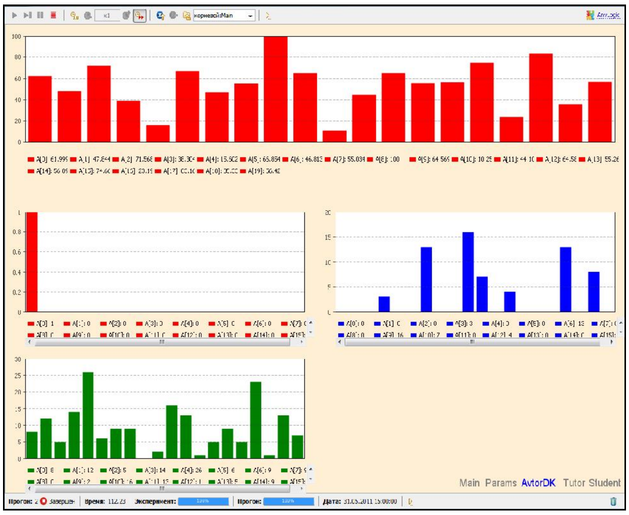
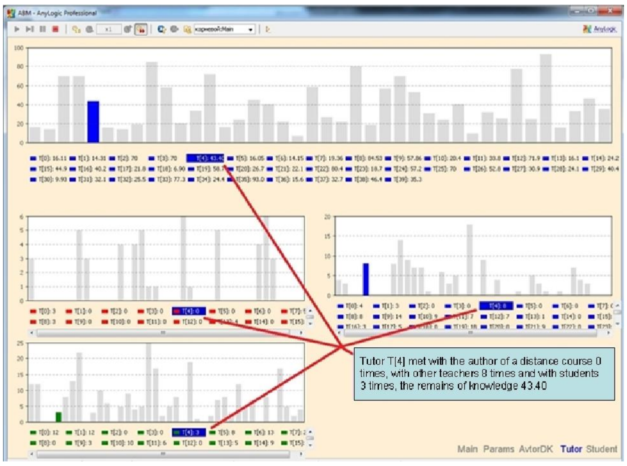


Fig. 10 – Diagrams balance of knowledge and meeting agents on the example of Authors DK.

Such diagrams of the meetings with respect to transfers of knowledge in a learning management system Moodle can be observed for other e-learning agents. In Fig. 10 tabs for tutor and student are presented as well.

Fig. 11 illustrates some results of a simple experiment with a designed agent-based model hybrid, which assesses the knowledge of e-learning users. The results of the knowledge and the e-learning contact between agents are illustrated through the example of a specific (fourth) tutor.

Ideally, a hybrid agent-based model with integrated neural networks allows describing data of layers, and providing analytics.



Tutor T[4] met with the author of a distance course 0 times, with other teachers 8 times and with students 3 times, the remains of knowledge 43.40

Fig. 11 – Diagrams of the resulting knowledge on the example tutors of the distance course.

We use the multiple data sets and analytical tools in a single interface for the visualization and data presentation. This helps in analysis of patterns of production and knowledge dissemination of e-learning agents.

4. CONCLUSIONS

Thus, analysis of the results of the completed research allows making the following conclusions:

1. It is extremely important to promote a dialogue between researchers and practitioners to create new tools and methodologies supporting the development of Web analytics (or learning analytics) in the field of e-learning.
2. Both agent-based models and artificial neural networks are effective tools for developing Web analytics (learning analytics) in e-learning.
3. Using those tools above can lead to improving distance e-learning courses aimed at creating adaptive mechanisms of interaction between agents of e-learning: webinars, brainstorming and thematic discussions on the forums.

In the future we plan to modernize the ABM model that would take into account:

– all formalized description of the model code for conduct analyzed agents (1) – (12);

– interactions with agents of the university administration and upper-level management – the Ministry of Education and Science of Ukraine.

5. REFERENCES

[1] V.L. Makarov, A.R. Bakhtizin, The new tools in the social sciences – an agent-based model: general description and specific examples, *Economics and Management*, (12) 50 (2009), pp. 13-25 (in Russian).

[2] A.R. Bakhtizin, *Agent-based models of the economy*, ZAO "Publisher "Economy", Moscow, 2008, 279 p. (in Russian).

[3] V.L. Makarov, A.R. Bakhtizin, *Social simulation – a breakthrough new computer (agent-based models)*, Economics, Moscow, 2013, 295 p. (in Russian).

[4] *Journal of Artificial Societies and Social Simulation*. http://jasss.soc.surrey.ac.uk/JASSS.html.

[5] *Agent-based Computational Economics*: http://www.econ.iastate.edu/tesfatsi/ace.htm.

[6] *Journal of Artificial Societies*. http://www.artsoc.ru/ (in Russian).

[7] Website Santa Fe Institute. http://www.santafe.edu.

[8] Anylogic Simulation Software. http://www.anylogic.com/.

[9] Y. Karpov, *Simulation modeling systems. Introduction to modeling with AnyLogic 5*, BHV-Petersburg, St. Petersburg, 2005, 400 p. (in Russian).

[10] A. Borschev, From system dynamics and traditional IT – to practical agent-based models: causes, technology and tools. http://www.gpss.ru/paper/borshevarc.pdf. (in Russian).

[11] V. Artemenko, Web Analytics in e-Learning: agent-based and neural network approaches, *Proceedings of the 2013 IEEE 7th International Computing on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. Vol. 2, Berlin, Germany (September 12-14, 2013), pp. 774-780.

[12] V. Artemenko. Computer modeling of communicative interactions of e-Learning agents, *Educational Technology & Society*, (13) 2 (2010), pp. 345-354. http://ifets.ieee.org/russian/depository/v13_i2/pdf/12r.pdf. (in Russian).

[13] Website of web-center of Lviv Academy of Commerce (LAC). http://virt.lac.lviv.ua. (in Ukrainian).

[14] *STATISTICA* Automated Neural Networks – Site StatSoft is now a part of Dell. http://www.statsoft.com/Products/STATISTICA/Automated-Neural-Networks.

***Viktor Artemenko**, In 1974 graduated from Ivan Franko National University of Lviv, Faculty of Economics. In 1979-82 studied at the graduate shcool of the Central Economics and Mathematics Institute (CEMI, Moscow) at the Academy of Sciences of the USSR.*

In 1982 defended his PhD thesis in economics. Since 1982 he is working at Lviv Academy of Commerce (LAC) as an assistant professor. In 1994-2009 was the head of the Management Information Systems department at LAC. Since 2006 is the head of the UNESCO Chair branch "New information technologies in education for all". Research interests: management of efficiency of regional development of Ukraine, construction and monitoring of life quality indicators, ICT, e-Learning.