Chapter 9
Investigation of Money Turnover in the Computer Agent-Based Model

O. M. Zvereva

9.1 Introduction

There are three main simulation paradigms; they are as follows: discrete event modeling, system dynamics and agent-based modeling (ABM). Discrete-event simulation is used for development of stochastic, dynamic models where simulation state variables change at discrete points in time. System dynamics is an approach aimed to understand the behaviour of complex systems over time using stocks and flows, internal feedback loops and time delays. Agent-based (sometimes called Individual-based) modelling is used to model a system as a collection of communicating entities called the agents.

The third mentioned modelling paradigm has gained increasing attention over the past decade. Macal and North [1] proposed some explanation for this fact. The first reason is that the present-day systems that one needs to analyze have been becoming more and more complex and having various type of interdependences. The second consideration is a consequence of the first—nobody can model these systems adequately. '…We are beginning to be able to relax some of… assumptions and take a more realistic view of these… systems through ABM' [1]. The third and the fourth reasons are related to modern computational knowledge and power, because of which ABM is possible in the real-world scale and can solve the real-world problems. Some modellers contend that ABM 'is a third way of doing science' and could augment traditional deductive and inductive reasoning as discovery methods.

According to [2] every agent-based model consists of the following basic elements:

1. Agents with their attributes, goals and rules of behaviour;
2. Relations between the agents;

O. M. Zvereva (✉)
Ural Federal University, Yekaterinburg, Russia
e-mail: OM-Zvereva2008@yandex.ru

S. I. Kumkov et al. (eds.), *Advances in Information Technologies, Telecommunication, and Radioelectronics*, Innovation and Discovery in Russian Science and Engineering,
https://doi.org/10.1007/978-3-030-37514-0_9

3. Environment: agents communicate through the environment and in some cases can interact with this environment as well.

There are some generic agent characteristics: an agent is autonomous, reactive, pro-active and social.

Agent's autonomy means that the agent encapsulates some state, which is not accessible to other agents. Reactivity is considered as the agent's ability to respond in a timely fashion to changes that occur in the environment in which the agent is positioned. Agent's pro-activeness is realized as its property to exhibit the goal-directed behaviour by taking the initiative. Agent's social ability indicates its competence to communicate with other agents using some kind of a language.

There is no established agent classification because ABM is a rather new paradigm and has been rapidly developing; many new agent types appear every now and then. Some efforts aimed at the development of the agent taxonomy are discussed in [3] in the context of economics.

Agents can greatly differ from each other and their behaviour can be different as well. They can behave in a random way (Zero Intelligence Agents), or they can learn and store their knowledge. But even a simple agent-based model can exhibit complex behavior patterns and provide valuable information about the dynamics of the real-world system which it emulates.

Bonabeau in [4] indicated benefits of ABM over other simulation techniques: ABM captures emergent phenomenon, ABM provides a natural description of a system's behavior, and ABM is a flexible one.

Emergent phenomenon results from the interactions of individual entities. In the engineered model, one determines micro parameters (agent's attributes, rules of interactions) and can measure system macro parameters. Nobody can predict these macro parameters' values, and sometimes they appear to be really amazing. It is rather common to make a project in the 'top-down' manner (especially in the field of software engineering), but, in the case of ABM, another 'bottom-up' approach is used.

In many cases, ABM is the most natural for describing and simulating a system composed of acting and interacting entities. Such systems are common in different scientific fields: e.g. physics [5], chemistry [6], biology [7, 8], history [9], etc. But one can argue that ABM is geared towards social and economic system modelling.

Luhmann and many other sociologists considered society to be a set of communications in its essence [10], and ABM is the most fitting technique for a social communications system simulation. In [11] ABM is even called 'the right mathematics for the social sciences'. In the economic science, one can reveal the mainstream called the agent-based computational economics, which is based on the ABM paradigm (a lot of materials is presented in [12]).

ABM flexibility can be observed along multiple dimensions. In most agent-based models, one can easily (without significant code change) increase the quantity of agents. You can, with minimal effort, change agent's rules of behaviour, include some new elements into their environments, etc.

9.2 Communication Model Specification

A communication process is a process which runs in systems from various scientific domains. Active system entities try to interact with each other exchanging something that has a value for them [10] with the predefined purpose and using a kind of a channel. In a social system, knowledge (or sense) is the value subject for exchange, and, in an economic system, money and products exchanges could be observed.

It was decided to create a model with active productive entities which are involved in economic communications. In this system, every agent has a very clear goal: it tries to receive all the resources necessary for manufacturing the specified product volume; in order to succeed in this activity, this agent also tries to sell its own already manufactured product. According to communication process definition [13], every system member is involved in such type of process with the other system members. There is an environment (economics), which delivers money supposed to be means for exchanges. Thus, we have three main components of an agent-based system: a set of agents, an environment, relations between the agents and with the environment as well. In this case, it becomes evident that the most suitable modelling paradigm is the agent-based one.

In order to build a valid model, we have to determine what questions our model will answer, what problems it will help to solve. The main question is—'Under what conditions a productive system will be a sustainable one?' It means that a system can function in a long run: in its every lifecycle it must create a kind of material background for the succeeding lifecycle. This issue is detailed in [14].

Trying to investigate the problem of system sustainability, several additional pieces of knowledge about money turnover were received, namely, we got to know how much money it was necessary to introduce into the system to support its sustainable functioning, and how this money volume has influenced time necessary to carry out all communications, what way of behaviour the agents had to choose to optimize the communication time. Also with the help of the model, some experiments were held to prove (or eject) the hypothesis of virtual, or internal, money usability, their opportunity to play the role of real money.

The main idea for a sustainable productive system model was that some kind of equilibrium should be adopted as its quantitative basis. As this equilibrium, Leontief's static intersectoral equilibrium may be proposed, and his famous 'input-output' model [15] has become an algorithmic basis for the engineered model (Communication Model).

In the Communication Model, there are N agents, where the i*th* agent is characterized by its product volume (x_i), its vector of demands in products of another system agents ($\overrightarrow{W}_i = (w_{ik})_{k=1}^{N}$), and its final demand volume (y_i) that can be spent for meeting its non-product requirements after all communication acts.

As for the system as a whole, it can be described by the following macro parameters: ($\vec{X}(t) = (x_{\mathrm{i}})_{i=1}^{N}$) which is a system product volume vector; ($\vec{Y}(t) = (y_{\mathrm{i}})_{i=1}^{N}$) which is a system final demands vector; and ($W_{NxN} = \left(\overrightarrow{w_i}\right)_{i=1}^{N}$) which is a communication matrix (matrix of mutual payments, as it will be shown further).

For every agent in the model, it is true that:

$$x_i = a_{1i} \cdot x_1 + a_{2i} \cdot x_2 + \cdots + a_{Ni} \cdot x_N + y_i \tag{9.1}$$

In accordance to Leontief's model [15], the technological matrix ($A_{N\times N}$) can be obtained, being calculated on the basis of $W_{N\times N}$, where the matrix element (a_{ki}) determines the k-th agent's product volume which is consumed by the i-th agent for its single product unit manufacturing.

For the system in a whole, the following vector equation can be written:

$$\vec{X} - A\vec{X} = \vec{Y} \tag{9.2}$$

As was mentioned above, economic environment provides money to support exchanges in the system, to determine the money volume Money supplement coefficient (K) was proposed. The money volume circulated in the system is estimated as the multiplication of coefficient K and the total product volume in the system. Every agent receives a sum of money on its account, the initial value of agent's money (m_i) is estimated in the direct proportion to its product volume ($m_i = Kx_i$). Every agent tries to communicate with the other system agents and uses money from its account to provide these communications. Money in this model plays its common role means of exchanges.

An agent can behave in various ways. These different kinds of agent's behaviour were named 'strategies'. In every communication act, it is necessary to determine the partner (or the partners), which product is necessary to receive, and the volume of this exchange. Thus, these strategies differ from each other according to the following three parameters: the number of partners, the rule of the partner choice, and the exchange volume. Thus, strategy and money volume also have become the essential agent's characteristics.

The communication process has to last as long as any communication is possible. Communication stage is divided into exchange cycles (the modelling time is measured in exchange cycles). During every exchange cycle, every agent (chosen in a random way) receives a chance to communicate only once, i.e. to process a single exchange operation in the way assigned by its strategy.

In order to engineer the Model according to the revealed specifications, it is necessary to choose an information toolkit. This problem is discussed in the next section.

9.3 Modeling Toolkit Choice

In the last few years, the ABM community has taken a giant step in developing practical agent-based modeling toolkits that enable individuals to engineer significantly sized and complex applications. There are several surveys which try to evaluate and compare these toolkits. In [16], the authors have evaluated more than 50 frameworks, in [17, 18] only five modelling frameworks are under discussion, but they might be considered as the most popular ones, they are as follows: Swarm, Java Swarm, Repast, MASON and NetLogo.

In the first survey [16], five widely used characteristics have become the basis for the comparison: complexity of interface and modelling language; operating system required to run the toolkit; type of the license governing the platform (free or proprietary); primary domain for which the toolkit is intended (is it a multipurpose instrument or is intended to model systems of the specific type, i.e. social, biological, economic, etc.); degree of support available to a toolkit user (e.g. documentation quality).

The toolkit language and interface are really important issues. According to this issue, all toolkits can be divided into two groups: in the first group general-purpose programming languages are used such as Java, C++, Python; and the frameworks of the second group have their own modelling languages. Modellers also can be divided into two groups: the first group consists of those who are professionals in the domain area (sociologists, biologists, chemists, etc.), and the second group includes professionals in programming. The first group members are more concerned with the modelling framework ease of use, the degree of programming skills required, and the existence of friendly interfaces to manage simulations. They are not addicted to coding, and it is likely that they will choose something from the second group of modelling toolkits. If the modeller is a professional in programming, he/she will choose a toolkit based on his/her 'favourite' language.

As the exception only proves the rule, we have chosen the toolkit with its special modelling language. Being professionals in IT area, we have decided that we would gain interesting new experience along with a new language implementation. The NetLogo [19] modelling environment was chosen as an instrument for the Communication model development and later simulation. This toolkit has its own modelling language, which is considered to be a Logo dialect extended to support agents.

Considering the main NetLogo characteristics (the latter four from the list above), it runs on different Windows versions (e.g. Windows 7/8/10), and also on Mac OS X 10.4, or newer. It needs Java Virtual Machine (JVM), or Java Runtime Environment (JRE) being preinstalled. NetLogo is a free open source system (it is under the terms of the GNU General Public License). This tool is oriented towards education as its primary specialization, but now it has become a powerful tool widely used in various scientific domains, and it is declared to be geared specifically towards the social science domains.

To discuss the degree of the user support, it is necessary to mention: the well-organized documentation built in the environment (tutorial and vocabulary), the excellent example models library, the well-organized Web site with links to the third party extensions which individuals have developed to fulfill specialized needs.

In [17, 18] NetLogo is one of the toolkits under discussion, this fact can be considered as an additional evidence of its popularity. The survey described in [17, 18] has delivered an interesting approach to modelling toolkits discussion and evaluation. The authors have engineered a set of 16 simple template models named in a whole as the 'Stupid Model'. The template models differ in their complexity: the first model is the basic and the simplest artifact, the subsequent versions incrementally add features that are commonly used in real models. While realizing these models in every framework, it becomes evident how successful and effective modelling process can be, how much effort will be necessary for its completing. According to this survey, one could create all the following models with the help of NetLogo. This proves that our model, which specification was discussed in this paper, with no doubt, could be engineered in NetLogo.

Coding realized in NetLogo has some peculiarities, the first and sounding rather strange is that the modelling language is not an object-oriented but is a procedure-oriented one. It is a common fact that agents are often associated with objects (in the sense of object-oriented programming). In the NetLogo language, we cannot find objects as class instances at all.

All produced code is divided into two parts. One part is collected in the form of procedures under the 'Code' tab of the environmental window, and some code is produced through dialog windows of different GUI controls such as buttons, sliders, monitors, inputs, plots, etc. This approach has the evident disadvantage: there is no single place where one can look to see the whole program code.

There are two obligatory procedures in every NetLogo model. The first one is called 'to setup' and determines all the initial values and primary conditions for further simulation. The second one is called 'to go', and this procedure starts the simulating process itself.

The following lines contain a code example for the beginning of the 'setup' procedure taken from the model code:

```
to setup
clear-all
set-plot-pen-mode 1
set sp_ready []
  ;;set the current directory
  set-current-directory "D:\\models"
;;open the file with input data
file-open user-file
set v_num branches - 1
let temp 360 / branches
;;depict agents in the circle
crt branches [
      set heading temp * who
      set label who
      fd 10]
      …
end
```

One more interesting fact might be mentioned: in the modelling language's basic version, there are no arrays, which are to be the common structures in most of well-known programming languages. Instead of arrays, one can use lists, which have some peculiarities to be compared with arrays. It is more complex to set a single value in the list. For example, the following command set to 0 (zero) the element with number 'who' in the list named 'res_list'.

```
set res_list replace-item who res_list 0
```

On the contrary, it is easier to process the whole list in comparison with an array. For example, following command creates a copy of the list:

```
set res_list_01 res_list
```

In the last NeLogo language versions, it has become possible to use arrays as language extension, but one must point it out explicitly.

9.4 Simulation Results

Screenshot of the Communication Model is shown in Fig. 9.1. As was already discussed, this model was intended to solve different problems. The main goal was declared as to find out the conditions of economic system sustainability. It was shown that if the system's characteristics were in accordance with the Leontief's equilibrium (9.2), then all the communications would be done successfully and agents would be ready to begin a new production cycle.

Although money volume and agent exchange strategy do not affect the final successful result, they greatly influence the overall communication time and the character of money circulation in a system.

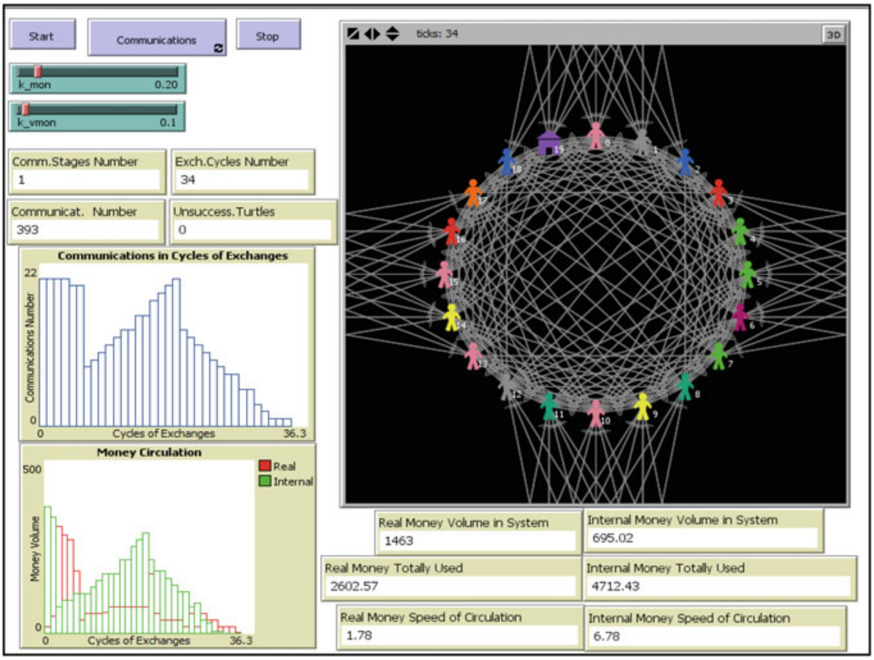

Fig. 9.1 Communication model screenshot

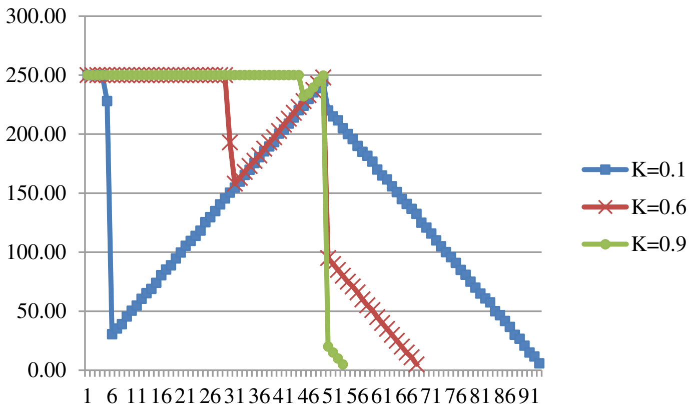

Fig. 9.2 Money turnover in the model with 50 agents with money supplement coefficients $K = 0.1$, $K = 0.6$ and $K = 0.9$

Communication time depends on the money volume in a system. The form of this dependence one can comprehend from Fig. 9.2, it becomes evident that the more money is circulated in a system, the shorter the period of time that is necessary to fulfill all the communications.

Figure 9.2 summarizes the results of experiments in the model with 50 agents with the same strategy used. In case of $K = 0.1$ (the money volume introduced into the system is equal to 10% of overall product value in it), all communications are completed while 94 cycles (in average); in case of $K = 0.6$, it takes 69 cycles, and, in case of $K = 0.9$, it takes only 54 cycles (the minimum value is 49 cycles in case of $K = 1.0$).

One can also find some specific features of the money turnover in the model. There are 3 main time periods: the first one—when all exchanges are successful and are fulfilled in the whole possible volume; the second one—the decrease of communications which is resulted in money circulation drop; and the third one—when remained communications are done by degrees.

The second period could be understood as a crisis period, and the third one is the way out of the crisis. In terms of money supplement, one can reveal that the less money in a system circulates the deeper and earlier the crisis occurs, and the longer time is necessary to leave this crisis state. In case of $K = 0.1$, the crisis starts in the 6th cycle and lasts till the 22nd cycle (15 cycles), while in case of $K = 0.9$, the crisis starts in the 46th cycle and lasts not more than 2 cycle (in case of $K = 1.0$ there is no crisis at all).

While simulations, one more phenomenon was revealed—the phenomenon of egoism. It emerges when a single agent changes its strategy and chooses the less optimal one. This agent as a single producer wins, but a system as the whole loses: it takes more time to complete all communications.

To understand possible roles of virtual money [20], a new version of the model was engineered, in which a new agent type was modelled, this type agent was called the External agent.

This new version model has proved to be a model of an opened type system. In the real life, we observe only opened type economic systems, since real economic systems are interrelated with each other, and local economics are involved into the global economic. In the new model one agent starts to play a role of an external environment, and exchanges with this agent simulate exchanges with the external world (i.e. import and export operations). One can find this agent in Fig. 9.1, it is depicted by the house icon.

In this model, two types of money exist: the internal, or virtual, money used only for internal exchanges, and the external, or real, money used for exchanges with the External agent (and also for internal exchanges in the case of the virtual money lack). The virtual money volume is in direct proportion to the preset virtual money supplement coefficient (*K_vmon*), and to the product volume in the system (the same rule is proposed for the real money volume, the corresponding coefficient is denoted as *K_mon*).

After simulations, we can postulate that virtual money can play a role of the real money and improve the communication process. It is clear from Fig. 9.3, where one can find diagrams of virtual, real and total money exchanges in the system with different volumes of real and virtual money.

From the structural point of view, there are internal exchange chains, they are of a closed type, virtual money circulates along these contours, and this accelerates the system communication process. Increasing the virtual money supplement coefficient value, one could shorten the communication time (degree of shortening depends on the agent strategy).

9.5 Conclusions

This paper tries to prove that the agent-based technology could deliver not only interesting, but rather useful results having been implemented for investigating a local economic system. Using the Communication Model, it was revealed that a productive system with industrial agents can become a sustainable one, i.e. work stably for a long time, if this system is based on the Leontief's intersectoral equilibrium. Money circulated in the model influences the communication process time and, moreover, this time depends on the agent behaviour (its exchange strategy).

Some phenomena were revealed during experiments, such as the crisis phenomenon and the phenomenon of egoism. They can hardly be predicted without this kind of simulation.

By exploring circular closed contours, it was found that virtual money introduction is a very useful act. It begins to work in accordance with the real one and significantly accelerates the communication process.

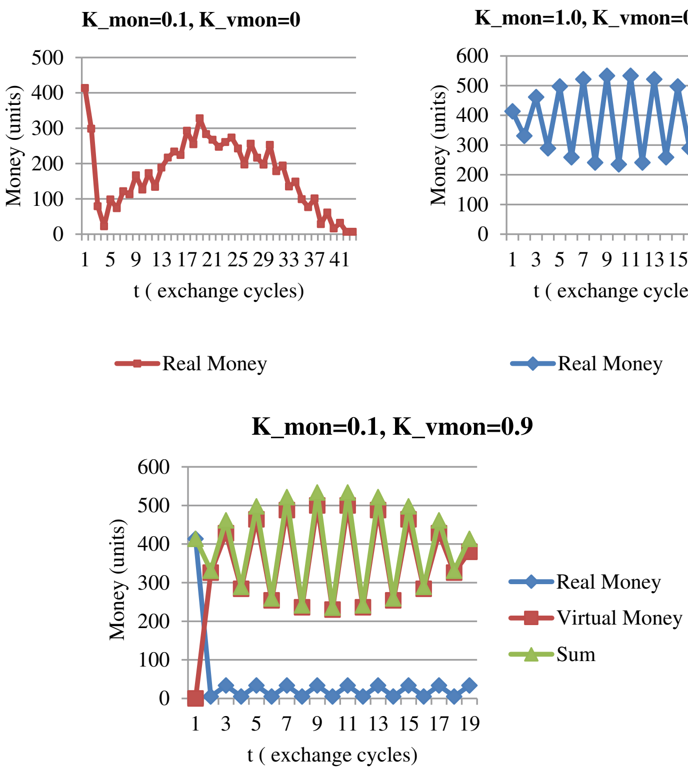

Fig. 9.3 Money turnover in the model with 20 agents in different conditions of money supplement

Acknowledgements The work was supported by Act 211 Government of the Russian Federation, contract N 02.A03.21.0006.

References

1. C.M. Macal, M.J. North, Agent-based modeling and simulation, in *Proceedings of the 2009 Winter Simulation Conference*, pp. 86–98(2009)
2. C.M. Macal, M.J. North, Tutorial on agent-based modelling and simulation. J. Simul. **4**, 151–162 (2010)
3. S.-H. Chen, Varieties of agent-based computational economics: a historical and interdisciplinary perspective. J. Econ. Dyn. Control (2011)
4. E. Bonabeau, Agent-based modeling: methods and techniques for simulating human systems. PNAS **99**(suppl. 3), 7280–7287 (2002)
5. A. Troisi, V. Wong, M.A. Ratner, An agent-based approach for modeling molecular self-organization. Proc. Natl. Acad. Sci. USA **102**(2), 255–260 (2005). www.pnas.org/cgi/doi/10.1073/pnas.0408308102. Accessed 12 Jan 2018
6. E. Tatara, I. Birol, F. Teymour, A. Cinar, Agent-based control of autocatalytic replicators in networks of reactors. J. Comput. Chem. Eng. **29**, 807–815 (2005)

7. K.J. Mock, J.W. Testa, An Agent-Based Model of Predator-Prey Relationships Between Transient Killer Whales and Other Marine Mammals (University of Alaska Anchorage, Anchorage, AK). www.math.uaa.alaska.edu/orca/. Accessed 11 Jan 2018
8. V.A. Folcik, G.C. An, C.G. Orosz, The basic immune simulator: an agent-based model to study the interactions between innate and adaptive immunity. Theor. Bio. Med. Model. **4**(1), 39–56 (2007)
9. M. Gavin, Agent-based modeling and historical simulation. Digit. Hum. Q. **8**(1) (2014)
10. N. Luhmann, *Social Systems. Sketch of the General Theory* (Science, St. Petersburg, 2007), 668 p
11. P.L. Borrill, L. Tesfatsion, Agent-based modeling: the right mathematics for the social sciences, in *Working Paper No 10023*, July (2010)
12. L. Tesfatsion, Agent-based computational economics. www.econ.iastate.edu/tesfatsi/ace.htm. Accessed 19 Jan 2018
13. V.B. Kashkin, *Intoduction To Communication Theory: A Tutorial* (Flinta, Moscow, 2013), 224 p
14. D.B. Berg, O.M. Zvereva, S. Akenov, «Economic Microscope»: the agent-based model set as an instrument in an economic system research, in *ICNAAM 2016 Conference Proceeding,* Vol. 1863, Paper No 050006, 21 July 2017
15. V.V. Leontief, *Essays in economics. Theories, theorizing, facts and policies* (Politizdat, Moscow, 1990), 415 p
16. C. Nikolai, G. Madey, Tools of the trade: a survey of various agent based modeling platforms. J. Artif. Soc. Soc. Simul. **12**(2) (2009)
17. S.L. Railsback, S.F. Lytinen, S. Jackson, Agent-based simulation platforms: review and development recommendations. Simulation **82**, 609–623 (2006)
18. S.L. Lytinen, S.F. Railsback, The evolution of agent-based simulation platforms: a review of NetLogo 5.0 and ReLogo, in *Proceedings of the Fourth International Symposium on Agent-Based Modeling and Simulation* (2011). http://www2.econ.iastate.edu/tesfatsi/NetLogoReLogoReview.LytinenRailsback2012.pdf. Accessed 21 Jan 2018
19. U. Wilensky, NetLogo (Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999). http://ccl.northwestern.edu/netlogo/. Accessed 20 Jan 2018
20. B.A. Lietar, *The Future of Money. Creating New Wealth, Work and Wiser World* (KRPA Olymp: AST: Astrel, Moscow, 2007), 493 p