Software for Simulation of Bio-Like Systems and Processes Using Movable Cellular Automata

Volodymyr Zhikharevich[1] [0000-0003-4882-2954], Kateryna Hazdiuk[2] [0000-0002-7568-4422] and Serhiy Ostapov[3] [0000-0002-4139-4152]

[1-3]Yuriy Fedkovych Chernivtsi National University, Kotsyubynsky str., 2, Chernivtsi, 58012 Ukraine

[1]vzhikhar81@gmail.com
[2]kateryna.gazdyik@gmail.com
[3]sergey.ostapov@gmail.com

Abstract. This paper deals with the development of a universal computer system for simulation biosimilar processes and structures. The main purpose of the system is to provide a wide range of users (scientists, engineers or just interested people) a convenient tool for their research. The simulating method of the proposed computer system is the method of movable cellular automata. An object of simulation, according to the method of movable cellular automata, is divided into small parts (automata), which interact with each other through the interaction's functions. The proposed system contains a library of elementary functions, the parameters of which can be customized according to the requirements of the model. The formation the combinations of elementary functions will allow researchers to visualize the model not only through a set of built-in interaction functions, but also to create their own interactions functions. The basic principles of these interactions are the same, although the examples can be diverse. It should be noted that the models are qualitative, not quantitative, and allow you to demonstrate the fundamental possibility of movable cellular automata for such simulation.

Keywords: Software, Movable cellular automata, Bio-like systems, Computer simulation.

1 Introduction

Simulation of biological processes and systems is one of the most relevant and promising areas of scientific research. The mechanisms of many bioprocesses are currently unknown. Therefore, one of the tasks, that is solved by simulation, is finding answers to questions that are open remain. Those include, in particular, the search for ways of self-organization and the evolution of living matter (Origin of Life).

Nowadays, simulation problems for biosystems are widely developed and require more complex calculations, so traditional approaches become less effective. New ways of improving classical direct imitation are also being explored in this field by

different approaches to the evolution and development of artificial organisms (indirect coding).

Many biological phenomena, such as intracellular localization of molecules, morphogenesis, growth, and dynamics of forest distribution, etc., are related to a spatial component that requires an understanding of spatiotemporal systems. Investigation of the localization and distribution of system components leads to the need to reproduce the shapes and geometry of the studied objects, as well as their possible deformation over time. In image systems, nature, forms, spatial distributions and their temporal dynamics are derived from images. This interprets the image not only as a visualization but also as a quantitative measurement and makes it a major source of data.

However, the use of images as a quantitative dimension raises several key problems. First, we need to investigate and take into account all prior knowledge and hypotheses about the system we are studying. Secondly, we need universal methods that can be applied to more than one specific situation. Finally, we need user-friendly and efficient software that facilitates the transfer of new methods to everyday scientific practice.

2 Previous investigation

A review of existing sources has identified a number of software systems that allow simulating of bio-like processes and structures.

One of these programs is COMPUCELL 3D [1, 2], the creation of which was started in 2004. This development allows us to simulate morphogenetic processes in different organisms, as well as to visualize the phenomena of cell growth, changes in the direction of distribution of chemicals, etc. CompuCell 3D is a three-dimensional modelling environment for complex biological processes that integrates several mathematical models. In CompuCell 3D, the space is discrete and is divided into identical cells that can be square or hexagonal in shape. Thus, the shape of the simulated objects (cells, tissues, or organs in general) is given by a set of such cells, and the processes occurring in these objects are modelled by differential equations. CompuCell3D has modelled such processes as tumour cell growth [3, 4, 5], bacterial colonies growth [6], development of blood vessels [7], and others. However, the COMPUCELL 3D morphogenesis model is limited by the fact that the shape of the modelled organism is a priori a continuous function over time.

Although CompuCell 3D provides the ability to test and experiment with biological models by providing a flexible package with many different levels of control, the user must have a broad range of bioinformatics knowledge to use this system. The disadvantages can be attributed to the poor visualization of the simulated systems. Additions and improvements to the system are still ongoing.

Other interesting computer simulation system (CSS) is Morpheus [8]. It is a simulation environment for studying high-scaled systems. Morpheus, developed for the study of cellular models under the influence of both internal and extracellular dynamics, which is described in a special declarative language, using the means of numerical

solution of ordinary differential equations, partial differential equations and cellular models of Potts [9], as in the previous CompuCell 3D system.

However, despite the wide range of settings and possible functionality, the construction and study of the dynamics of a multicellular organism is not foreseen.

Ascalaph is the next CSS that was considered [10]. It is a versatile program for molecular dynamics simulating. Under a single graphical environment are presented as authors' own implementations of molecular dynamics, and methods of classical and quantum mechanics of popular programs.

All user manipulations apply to molecular models. In this program, each model has a separate graphic application and file. A model is a set of atoms and their associations, such as bonds, molecules, and residues. Models may also include operating methods, such as simulating parameters, periodic boundary conditions, etc. Other methods may be common to all models, such as geometric measurements.

The disadvantages of this software are that its application to reflect the dynamics of biosimilar processes is quite limited, since the main area of modelling and related relationships are focused on chemical-physical processes.

The concept of a two-dimensional dynamic cellular automaton with a homogeneous structure is proposed in next reviewed study [11]. It extends the capabilities of two-dimensional cellular automata (CA) and allows to apply a CA approach to problems related to the dynamics of the structure of a simulated space in a two-dimensional case. In the process of the research the author has developed a software package that allows the user to program two-dimensional static CA models, two-dimensional dynamic CA models with a homogeneous structure and three-dimensional CA models with a dynamic inhomogeneous cell structure. The visualizer included with the OpenGL library allows you to visualize the simulation process in real time using a graphical accelerator, as well as to build two-dimensional graphs and histograms from the statistics obtained during the simulation process.

The user interface allows you to control the simulating process and set model parameters, most of which can be changed directly during the simulating process. The core of the software package is designed as an open source algorithms and class library that allows any user to add and modify the library code at their own discretion.

All models mentioned above have both advantages and disadvantages. Also, it should be noted that there are other software complexes for simulating of bio-like processes that achieve more precise characteristics, but are narrowly focused. The basis for their creation was a specific task, and therefore the result is aimed at specialists in a particular field (medicine, genetics, etc.). For example, in [12, 13], a 2-dimensional programmatic implementation of a tumour is used. Also, the development of Java 3D imaging of cancer is described in [14]. Similar software complexes also exist to simulate morphogenesis, embryogenesis, etc.

In contrast to the simulation systems mentioned above, this paper proposes the development of a universal computer system that would allow researchers to visualize the model not only through a set of built-in interaction functions, but also to create their own. The main purpose of the system is to provide a wide range of users (scientists, engineers or just interested people) a convenient tool for simulating biosimilar processes and structures. This tool should be simple and flexible when used. In this

regard, the proposed CAD system is based on the method of movable cellular automata (MCA). The core of the system implements an iterative algorithm for selecting automata, searching for the nearest neighbours and their interaction with each other, and the interface part provides visualization of the model and setting of input parameters (set of rules of interactions, properties of automata, etc.). Below we provide a description of the structure and features of the functioning of the CAD-system BLSD (Bio-Like Structure Design, Version 1.0).

3 BLSD CAD system architecture

The system has a modular architecture and is created by means of structured programming. The central part of the system is a procedure that provides the implementation of the algorithm of the method of movable cellular automata (Fig. 1). The method involves the operation of an iterative loop. It means the asynchronous choosing of the automaton from the set of automata, the search for the nearest neighbor and interaction with it. This may be the removal of the automaton or the addition of many new automata. In parallel, there is a graphical representation of a set of movable cellular automata. These two cycles are implemented as separate threads, i.e. the system in this respect has a multi-threaded architecture, which can significantly increase system performance in terms of multiprocessor hardware.

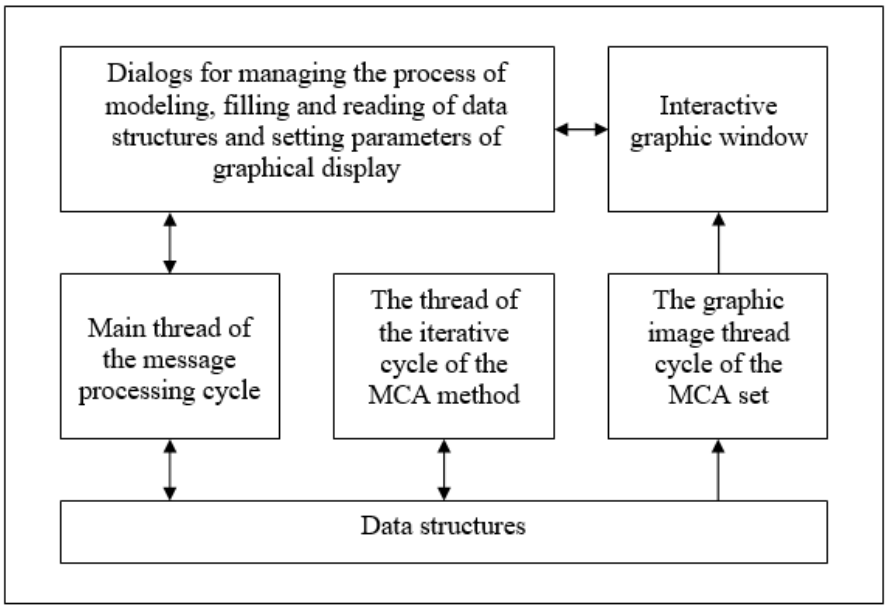

Fig. 1. BLSD CAD-system architecture

The main purpose of the interface part of the CAD system is to control the process of simulation, filling and reading of data structures of models and setting the parame-

ters of graphical display. Graphical rendering is performed using a separate interactive graphical window using the OpenGL graphical pipeline functionality. Interactivity not only displays 3D models on the screen, but also manipulates them with a computer mouse, including: rotate, scale, add or remove individual elements, shift and link them. All system actions, like any standard Windows application, occur when processing the corresponding messages (Windows Messages), which is implemented in the main thread of the message processing cycle.

4 BLSD CAD system data structure

Data structures are implemented in the form of arrays, which store all the necessary information about the models and their parameters in the graphical window. These arrays are read or written to files with the extension "* .blsd".

All information about the set of movable cellular automata is stored in a two-dimensional array E (Elements), the structure of which is shown in Tab.1. The line numbers of the array correspond to the numbers (indices) of movable cellular automata. The number of columns is

$$1+3+Nmax+Pmax$$

The first column corresponds to the type of automaton, then the three columns contain the *x*-, *y*- and *z*-coordinates of the automaton in space. The following Nmax columns contain indexes of adjacent automata. To determine the direct index of the neighbor, it is necessary to ignore (mask) the higher bit of the number obtained from the corresponding cell, and to determine the type of communication with the neighbor one must analyze this bit, since there are two types of neighborhood (unrelated or linked), and this sign is indicated in the higher bit of the neighbor index (0 or 1). Finally, the last columns of Pmax contain the possible values of continuous parameters of cellular automata, for example: the sign of the active or passive state of the automaton, the value of some internal energy, the components of the momentum vector, etc.

Table 1. The structure of E array

Index of an element in an array	**Type of sells**	**Coordinates**			**Neighbours**			**Parameters**		
		x	***y***	***z***	**N1**	**…**	**Nmax**	**P1**	**…**	**Pmax**
1	**S1**	*x*1	*y*1	*z*1	N11	…	N1max	P11	…	P1max
2	**S2**	*x*2	*y*2	*z*2	N21	…	N2max	P21	…	P2max
…	**…**	...	...	...	...	…	...	...	…	...
m	**Sm**	*xm*	*ym*	*zm*	Nm1	…	Nmmax	Pm1	…	Pmmax

In addition to the described data structure, which contains information about the set of movable cellular automata, the system uses structures to store information about the properties of the types of automata, as well as the functions of their interactions. All these data structures are implemented as corresponding arrays. And all arrays of cells of real type, except for the matrix of interactions containing indices of complex functions, for which there are enough cells of integer type. In addition, the names of automata types, their parameters, the names of elementary and complex functions, as well as other text information are contained in the corresponding arrays of cells of the string type.

The parameters of elementary functions that are part of complex functions are contained in the three-dimensional array FP (Function Parameters). Cells with zero addresses indicate the number of elementary functions that are included in the complex, as well as the indices of these elementary functions. A separate one-dimensional FS (Function String) array contains the names of complex functions. The two-dimensional FES (Function Elementary String) array contains the names of elementary functions (in null columns) and the names of their parameters. The interaction matrix is contained in a three-dimensional FM (Function Matrix) array, which has two layers – one (0) for the two-dimensional matrix of rules for interactions between unrelated neighbours and the other (1) for connected neighbours.

Properties of automata types are contained in the corresponding arrays, the cells of which are filled when working with the Properties of Elements dialog box. In particular, the upper part of the dialog handles elements of the two-dimensional array A (Attributes of Elements), and the lower part - elements of the three-dimensional array C (Characteristic of Continuous Parameters). The names of automata types and their parameters also include two separate arrays: one-dimensional AS (Attributes String) and two-dimensional CS (Characteristic String).

5 BLSD interface

The system has a multiple window interface. The main window (Fig. 2) is intended for manipulation of the model and consists of two parts: the field of control of the modes of operation of the program and the field of output of three-dimensional graphic image of the model.

The main menu contains options for invoking standard File Save or Open dialogs, Editing Rules for Interaction Rules and movable automata Properties (Edit), and a System Information (Info) dialog. When you save or open an arbitrary model in a file with the extension "* .blsd", it writes (or reads) all data structures containing information about the state of the set of movable cellular automata, as well as the rules of their interactions. Selecting the Edit menu shows the following dialogs:

- a window for editing properties of automata (Properties of Elements);
- a window for editing the DataBase of interaction rules (Database of Functions);
- window for editing Matrices of Functions.

In general, the system can operate in two modes: construction and simulation, which is determined by the state of the respective controls of the GUI system. The upper left part of the main window contains buttons for controlling the simulation process (start, pause, step, stop). A field of choice of the mode of construction of model elements (Elements), containing three items exist below:

- Adding / Deleting the automaton (Add / Del);
- Moving the automaton (Move);
- Tying up automata (Tie).

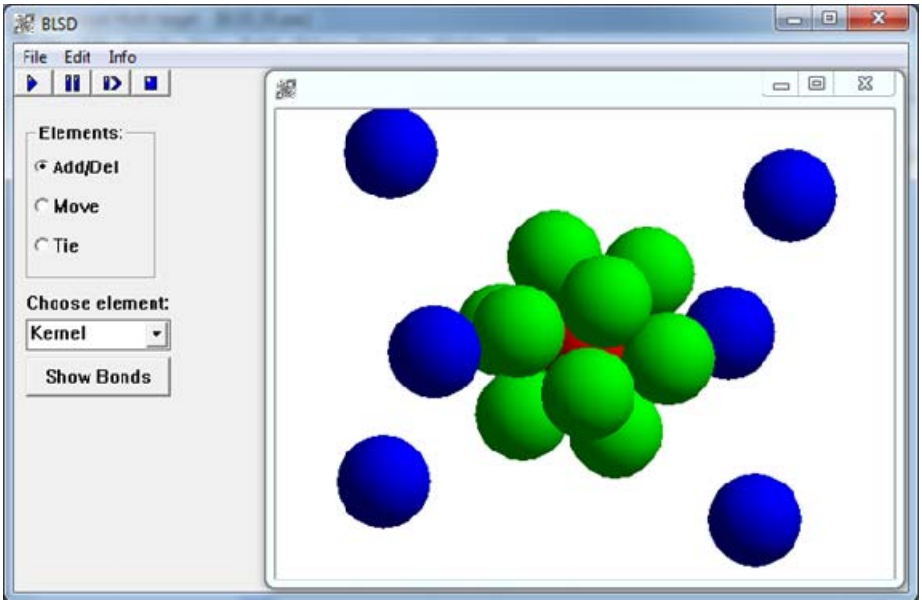

Fig. 2. BLSD main window

Selecting the first item in this field gives you the option to add or delete a movable automaton. Adding happens when you left-click. At the point corresponding to the position of the mouse cursor, a new movable automaton (with a zero-coordinate $z = 0$) will appear in the 3D model display window. The automaton type is being select in the appropriate Choose Element dropdown lists. If you select the second item of the field (Move), you can move an arbitrary automaton by left-clicking on it and moving the cursor without releasing it. Moving occurs only in the plane of the X and Y axes. If there is a need to move along the Z axis, then you should rotate the model 90 degrees relative to the centre, move the required automata and rotate the model back. The model rotates with the mouse so that it is in a transparent sphere that can be "touched" by the surface. In addition, you can scale the model image by rotating the Scroll Wheel and expand the model display window to full screen. Selecting the third item of the field (Tie) gives you the opportunity to connect automata which property assumes the presence of fixed neighbours. First of all, you need to set the connection

mode by clicking the appropriate Show Bonds button. In this mode, the lines will show all the connections between the movable automata that have fixed neighbours.

The properties of movable automata and the rules for their interaction can be edited in the dialogs that appear when you select the appropriate menu item Edit. Consider the window of editing properties of automata (Properties of Elements), shown in Fig.3.

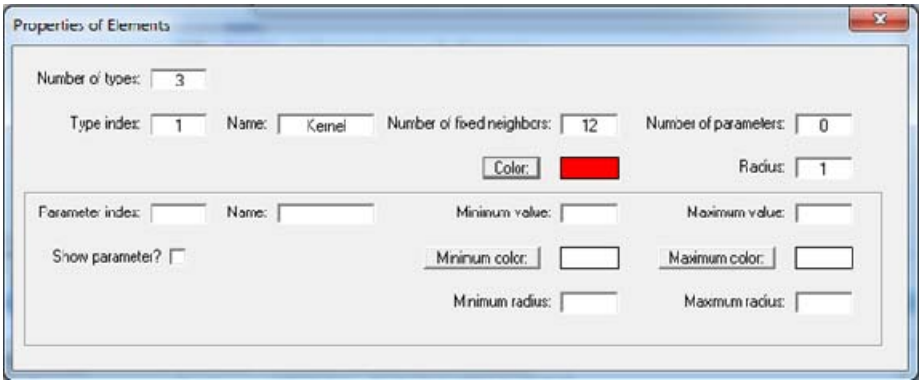

Fig. 3. BLSD Properties of Elements window

This window allows ones to specify the full number of types of cellular automata that can be used when the model is being implement. This number is entered in the "Number of types" field and determines the dimensions of the Matrices of Functions shown in Fig. 5. Selecting a specific type of automaton for editing its properties is made when its number is entered in the field "Type index".

When editing the properties of the automaton, it should be given a name in the "Name" field, specify the number of fixed neighbours (if any) in the field "Number of fixed neighbours", the number of continuous parameters of the automata (if any) in the field "Number of parameters", the radius of the displayed sphere in the "Radius" field in relative units. When you click the Colour: button, a standard colour selection dialog box appears, which allows you to select the colour of the displayed sphere that matches that type of machine.

Consider editing the DataBase of Functions shown in fig.4. This window allows you to form a complete set of complex interaction functions, the indices of which can be entered into the corresponding cells of the interaction matrices (Fig. 5).

This allows you to define the full range of rules for automated interactions. Complex interaction functions are a set (composition) of elementary rules. For a complete list of all elementary interaction rules, see the right in the Set of elementary functions window. These elementary rules implement the corresponding elementary iterative steps to modify the state of the automatons. For example, the Fluctuation rule realizes an elementary probabilistic shift of a movable automaton in an arbitrary direction, and the Attraction rule is an elementary step of displacement toward a selected neighbouring automaton.

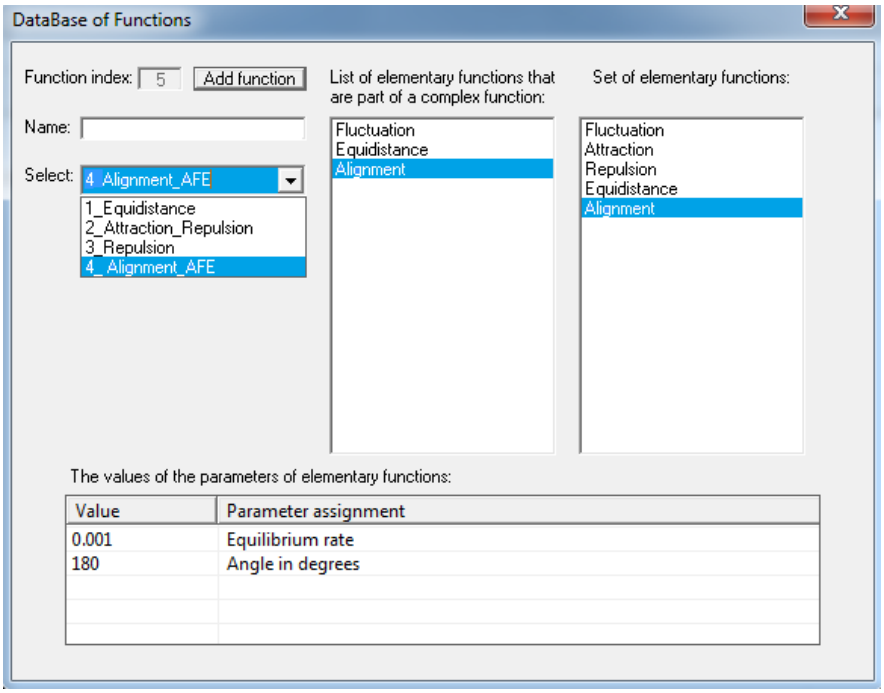

Fig. 4. Editing window of interaction functions DataBase

When you right-click on an elementary rule from the list of “Set of elementary functions”, an information box will appear with a description of this rule and its parameters (Fig. 5). The window contains a simple illustrative image, which may include an animated image obtained from the corresponding gif file.

The edit rule base editing window allows you to create complex rules that consist of a set of elementary rules. To do this, ones should enter the name of the complex rule in the "Name:" field and create a list of elementary functions that will be included in it. An elementary function is added to the List of elementary functions that are part of a complex function by double-clicking the left mouse button or pressing Enter on the selected function in the Set of elementary functions field. Deleting a function by right-clicking or deleting a selected function in the "List of elementary functions that are part of a complex function" field.

When you select an elemental function that is part of a complex function, a table is displayed at the bottom of the dialog box in the field "The values of the parameters of elementary functions:" and a description of the parameters included in this function. It is thus possible to change the values of these parameters by editing the corresponding cells in the table.

Then, when you click the "Add function" button, a newly created function is added to the DataBase of complex functions, which is automatically assigned a new index (Function index) and is assigned at the beginning of the function name. Further, if necessary, it is possible to edit existing rules in the DataBase. To do this, select the

appropriate function from the "Select" drop-down list, and then perform the steps described above. In particular, add or remove elementary functions and change the values of their parameters.

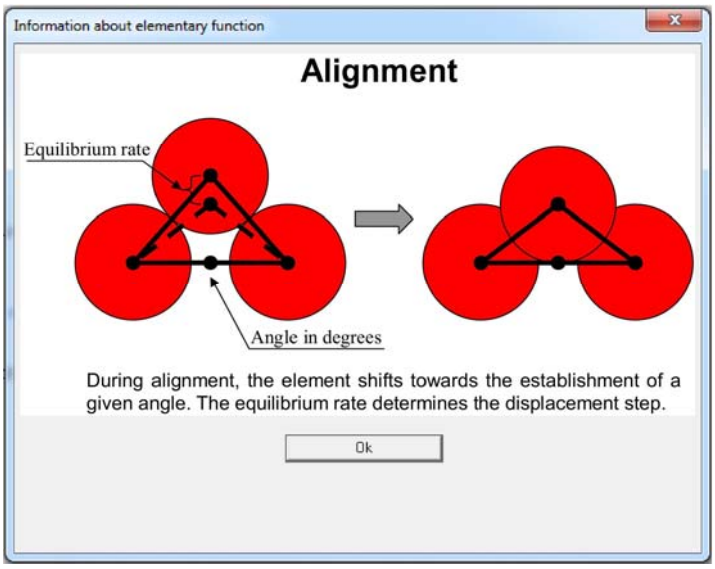

Fig. 5. Tooltip with information about elementary interaction function

Let us consider editing the Matrices of Functions shown in Fig. 6.

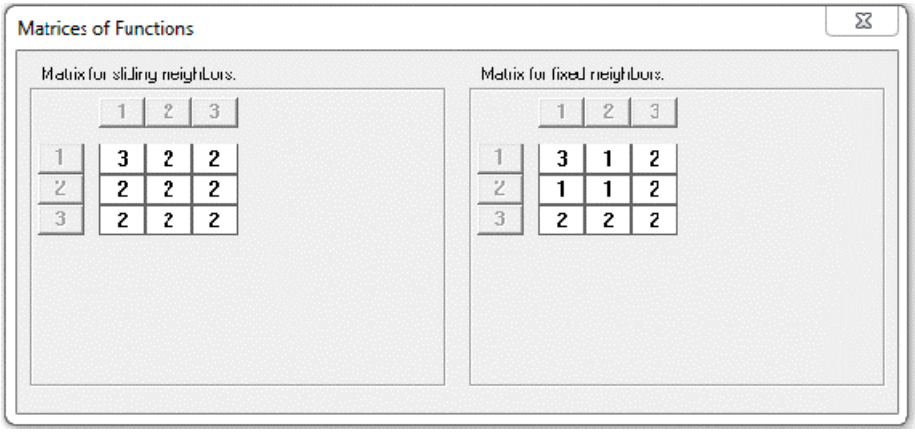

Fig. 6. Editing window of interaction rules matrices

This window allows you to define the rules of interaction between all possible combinations of types of two interacting automata. As mentioned above, the dimen-

sions of the matrices correspond to the number of automaton types in the simulated system. If the dimensions of the matrix are larger than 7 × 7, which exceeds the size of the fixed field of the dialog window, the Scroll Bar buttons appear, which allow to move the visible fragment of the matrix. In the cells of the matrices, a Function index should be specified, which will be used for automata interactions of the corresponding types. Moreover, since there may be different rules of behaviour between unbound and connected automata of the same type, two matrices are also required: Matrix for sliding neighbours and Matrix for fixed neighbours.

It should be noted, that examples which demonstrate the possibilities of the system and proposed method was described in our previous investigations [15-17].

6 Conclusion and future research

In this paper the problem of the construction of a CAD-system for a simulation bio-like structures and processes using movable cellular automata was considered. The paper gave an overview of available computer systems deals with such problem, outlined the advantages and disadvantages each of them. The detailed description of main components of the BLSD system is provided.

The evolution of the proposed system is expected in two main areas:

- Development of a base of elementary rules and means of their editing;
- Enhancing interactive graphics and improving quality.

The current version of the system includes a primary rule base, which does not support flexible modification, like adding new features without having to recompile the system and other related inconveniences. In the future, it is planned to create the ability to download a library of elementary rules from the appropriate DLL (Dynamic Link Libraries), which will allow you to develop a DataBase of rules in any programming language.

The graphics part of the system will also be improved. In particular, it will be possible to choose the type of to display models, in addition to Orthographic projection also Perspective projection. You will also be able to choose from a variety of graphic effects, such as translucency, fog, blur (photographic depth effect) and more.

The interactive part, which allows you to manipulate the elements of the model, will be complemented by the possibility of group selection of elements and their simultaneous copying. This will greatly speed up the process of building structures that contain the same fragments.

References

1. Kumar, S., Das, A., & Sen, S. Multicompartment cell-based modeling of confined migration: regulation by cell intrinsic and extrinsic factors. Molecular biology of the cell, 29(13), pp. 1599–1610., (2018). DOI:10.1091/mbc. E17-05-0313
2. Swat, M. H., Belmonte, J. B., Heiland, R. W., Zaitlen, B. L., Glazier, J. A., Shirinifard A., CompuCell3D Reference Manual. Source page: www.compucell3d.org/Manual/

3. Scianna, M., Preziosi, L., Preziosi L.; Ed. by M. Chaplain, J. Batzel, M. Bachar: New Challenges for Cancer Systems Biomedicine. Hybrid cellular Potts model for solid tumor growth. Springer, pp. 205–224, (2012).
4. Andasari, V., Roper, R. T., Swat, M. H., Chaplain, M. A.: Integrating Intracellular Dynamics Using CompuCell3D and Bionetsolver: Applications to Multiscale Modelling of Cancer Cell Growth and Invasion. PLoS ONE, Vol. 7, no. 3 (2012).
5. Gao, X., Tangney, M., Tabircam, S. A. Multiscale Model for Hypoxia-induced Avascular Tumor Growth. Proceedings of 2011 International Conference on Bioscience, Biochemistry and Bioinformatics Vol. 5, pp. 53–58 (2011).
6. Poplawski, N. J., Shirinifard, A., Swat M., Glazier, J. A.: Simulation of single-species bacterial-biofilm growth using the Glazier-Graner- Hogeweg model and the CompuCell3D modeling environment. Math Biosci Eng. Vol. 5, no. 2, pp. 355–388 (2008).
7. Kleinstreuer, N. A, Dix, D., Rountree, M., Baker, N., Sipes, N., Reif, D., Spencer, R., Knudsen, T.: Computational Model Predicting Disruption of Blood Vessel Development PLoS Comput Biol. Vol. 9, no. 4. 20 pp. (2013).
8. Source page: http://morpheus.gitlab.io
9. Wu, F.Y.: The Potts model. Reviews of Modern Physics. Vol. 54, Issue 1, p. 235. (1982).
10. Source page: http://www.biomolecular-modeling.com/Ascalaph/Documents/TOC.html
11. Vitvitskiy A.A.: Razrabotka i issledovanie metodov kletochno-avtomatnogo modelirovaniya rosta i deleniya kletok jivyih organizmov [Development and research of cellular-automata modelling methods of growth and division of living organisms' cells]: diss. for the phd degrees of Phys.-Math. sciences: Speciality. 05.13.18, "Mathematical modelling, numerical methods and program complexes". Novosibirsk, (2015).
12. Sottoriva, A., Verhoeff, J.J., Borovski, T., McWeeney, S.K., Naumov, L., et al.: Cancer stem cell tumor model reveals invasive morphology and increased phenotypical heterogeneity. Cancer Res 70: pp. 46–56 (2010).
13. Anderson, A.R., Weaver, A.M., Cummings, P.T., Quaranta, V. Tumor morphology and phenotypic evolution driven by selective pressure from the microenvironment. Cell 127: pp. 905–915 (2006).
14. Chen, Y.J., Tabirca, S., Tangney M. Interactive 3D Graphics for Cancer Experiment Data Visualisation. In: IEEE eds. The 2010 International Conference on Bioinformatics and Biomedical Technology (ICBBT 2010): pp. 42–46 (2010).
15. Hazdyuk, K.P., Zhikharevich, V.V., Nikitina, A.M., Ostapov, S.E.: Application of movable cellular automata to modeling the locomotion of worm-like organisms: Mathematical and computer simulation. Series: Technical Sciences: Coll. of sciences. Kamianets-Podilskyi: Kamianets-Podilsk. nats. un-t: pp. 16–25 (2018) (In Ukrainian).
16. Hazdyuk, E.P., Zhikharevich, V.V., Nikitina, A.M., Ostapov, S.E.: The unicellular microorganisms "Amoeba Proteus" locomotion simulation with the use of movable cellular automata method: Applied Discrete Mathematics, № 42: pp. 104–119 (2018). DOI: 10.17223/20710410/42/8
17. Zhikharevich, V., Hazdiuk, K. and Ostapov, S.: Simulation of bio-like systems and processes using movable cellular automata: [Electronic resuorce]. CEUR Workshop Proceedings, Vol. 2353: pp. 664-673 (2019). Access mode: http://ceur-ws.org/Vol-2353/paper63.pdf