AN INDIVIDUAL-BASED MODEL OF SELF-REPRODUCTION SYSTEM

WOJCIECH JĘDRUCH AND MARCIN GRAMZA

Faculty of Electronics, Telecommunications and Informatics,
Technical University of Gdansk,
Narutowicza 11/12, 80-952 Gdansk, Poland
wjed@sunrise.pg.gda.pl

(Received 7 March 2001; revised manuscript received 15 May 2001)

Abstract: The paper presents results of a few preliminary simulation experiments of a self-reproduction system realized in a programming environment for individual-based modelling of physical systems and discusses the advantages and difficulties of such modelling.

The programming environment named an abstract universe, is aimed at modelling of complex systems, which manifest the self-organization, self-modification, growing and emergent behavior. The central idea of the universe are specific interactions of its entities in a two dimensional space. The entities move and collide according to rules like those of classical mechanics, and interact with one another according to the function encoded in them modifying their structures and functions.

Keywords: individual-based modelling, artificial life, cellular automata, emergent processes, self-modification, self-organization

1. Individual-based modelling and self-reproduction systems

The term individual-based modelling describes a group of simulation methods of physical systems, in which the local rules of interaction between elements of a system are specified, and then an evolution of the system consisting of many such elements is investigated by computer simulation. Using this approach, a complex global behavior of a model is obtained as the result of many simultaneous, distributed in space, local and simple interactions.

Individual-based models are natural for many physical, chemical or biological systems, whose behavior is a result of interactions of many active elements distributed in space. They could be especially beneficial when beside a mere numerical agreement with a physical system the model should map its structure and topology. For such systems, individual-based modelling can be an attractive substitute for models based on differential equations apparatus. Moreover individual-based approach makes it possible to work with models of complex biological systems, which would be impossible even to formulate in traditional mathematical frameworks. Cellular automata, lattice gas automata, molecular dynamics and particle models are well known formal context for individual-based modelling [1–3].

Self-reproduction systems are the ones, which in the course of time produce a copy of themselves. Starting from the work of von Neumann [4] the self-reproduction systems were mainly modelled in cellular automata. The main disadvantage of modelling in the cellular automata environment is the determinism of designed systems, which produce identical copies in the process of self-replication, and in the case of modelling biological cells the difficulties to imitate its geometric structure.

These drawbacks of cellular automata models resulted in design of programming environments, which make it possible to create more realistic models of biological systems. Among them are *tierra*, *avida* [5], and designed independently *universe* [6, 7].

The universe is another universal tool for individual-based modelling. It is a system of entities in a two-dimensional space. The entities move and collide according to rules like those of classical mechanics, and at a higher level of organization they interact between themselves according to functions encoded in them. A computer program can simulate the evolution of the universe starting from any initial configuration of the entities thus allowing to model various systems. The universe has been already used for simulation of such physical processes as diffusion, cluster formation process, chemical reaction kinetic, predator-prey systems, and growth of various complex structures [7–10].

The far-sighted aim of the work is to build a self-reproducing system roughly resembling the structure and functions of a living cell. Having such a system the effectiveness of various self-reproduction strategies can be investigated, and the evolution of the population of artificial cells can be modelled. The aim of this paper is to present briefly the universe and its computer implementation, to describe the results of a few preliminary computer simulation experiments of self-reproduction system, and to discuss the problems arising during modelling.

2. The universe

The universe is defined over a two-dimensional tessellation of identical squares. A square may be empty or contain any number of elements belonging to the set $E = A \cup \{f\}, A = \{0,1\}$. Elements A are called *atoms* and f *photons*. Atoms are permanent elements of the universe; they are not created nor annihilated during its evolution. Photons are temporary elements; they are created by reactions, which dissipate energy.

Atoms occupying a square form a one-dimensional string $a_1, a_2, \ldots, a_N$ called *particle*. Particles in adjacent squares can bond together to form a *complex* of particles. Particles, complexes of particles, and photons can move by jumping to adjacent squares. In the Figure 1 an example segment of the universe space containing particles, complexes and photons has been shown.

The properties of a particle are fully determined by the one-dimensional sequence of its constituent atoms, its velocity and the possible bonds with adjacent particles.

The permanence of a particle depends on its bond energy; this is what suffices to disintegrate the particle into single atoms. The bond energy of a particle is the sum of the bond energies between its neighbour atoms. The values of the bond energy between both types of atoms are specified as the parameters of the system.

All the transformations in the universe observe the momentum and energy conservation laws and are synchronized by a discrete clock.

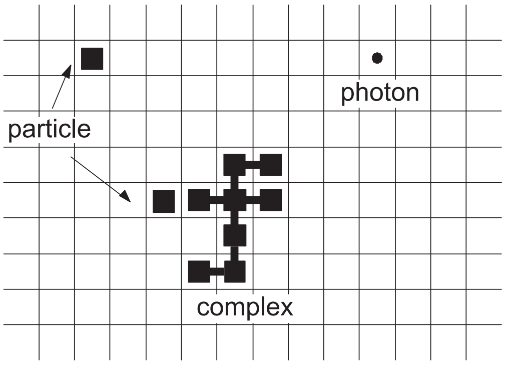

Figure 1. An example segment of the universe containing unbonded particles, a complex of particles and a photon. The particles represented in the figure by black squares consist of strings of zeros and ones (atoms)

2.1. Movements and collisions

During each time step, particles and complexes can move by jumping randomly to adjoining squares in the x and y directions with probabilities:

$$P_x = \begin{cases} |v_x/s| & \text{if } v \le s \\ |v_x/v| & \text{otherwise} \end{cases} \qquad P_y = \begin{cases} |v_y/s| & \text{if } v \le s \\ |v_y/v| & \text{otherwise} \end{cases} \tag{1}$$

where v_x and v_y denote the velocities of the particle in the x and y directions, respectively, $v = \sqrt{v_x^2 + v_y^2}$, and s is a constant. If $v \le s$, then the average Euclidian displacement of a particle is proportional to its velocity v, whereas for $v > s$ the average Euclidian displacement is the same as for $v = s$, which sets up the maximal velocity of one square distance per time step. These rules of movement produce random trajectories distributed around the straight lines, which represents the deterministic continuous movement with velocity (v_x, v_y). To illustrate the rules some example trajectories ate shown in the Figure 2.

Photons can move by jumping to adjoining squares in the x and y directions with probabilities:

$$P_{fx} = |\cos\varphi|, P_{fy} = |\sin\varphi|$$

where φ is a photon direction of movement chosen randomly at the moment of the photon creation.

When particles occupy the adjoining squares or when a photon jumps into the square occupied by a particle, a collision occurs, after which the new states of particles and photons are evaluated.

There are four types of collisions: the elastic and inelastic collision of two particles, and the elastic and inelastic collision of a particle and a photon. The latter collision is further classified into six subtypes: rebounding of the particle hit by the photon from an adjoining particle, setting bond between the hit particle and an adjoining particle, resetting bond, changing the order of atoms in the hit particle, concatenation of the hit particle and adjoining particle, splitting of the hit particle into two particles. The type and particulars of reaction are chosen randomly according to preset probabilities.

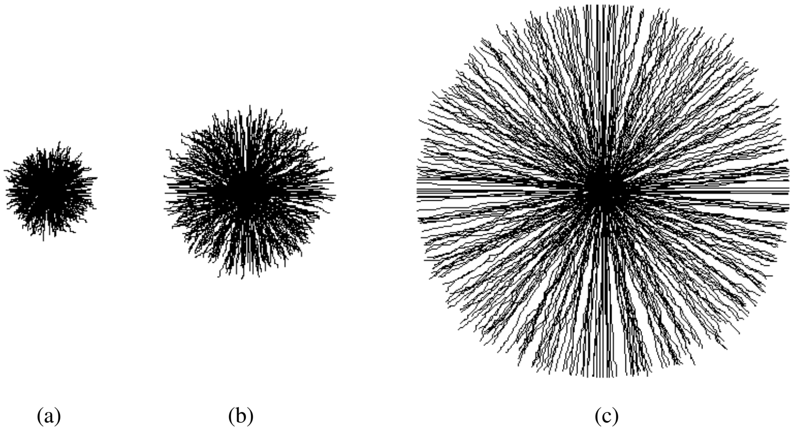

Figure 2. Example of particle trajectories observed for 180 time steps. The velocities of individuals were equal to: (a) 1, (b) 2, and (c) 5, and $s = 5$. For each velocity the set of 360 trajectories was drawn for the directions, which differed by 1°

2.2. Functional interactions

Besides the reactions resulting from the collision of particles with photons, in the universe exists another class of interactions, in which the particles are capable to transform and move other particles in the space around them according to functions encoded in them. The description of the function of a particle is contained in its string of atoms, which is interpreted as a program written in a specially defined language [7, 8]. During interpretation, the string is divided into five-bit portions, which are further mapped into commands.

The function contained in a particle can recognize particular structures of particles and complexes, and transform them by moving them, changing bonds between them, splitting them, concatenating them, or changing the order of atoms in them. Using this transformation the function of one particle can modify the function of other particles.

As an example, the sequence of atoms in a particle: 01100 00010 00001 00000 00001 01111 00000 00111 00000 00000 10100 11100 01010 10000 00010 10001 00000 00001 10010 00000 10101 111 defines function, which moves and attaches to the particle's north square a particle, which first atoms are 1110.

Depending on particles' sequence of atoms (zeros and ones), they can be classified to the following groups:

- particles with not well-formed functions *i.e.* sequences of atoms, which cannot be interpreted as a valid program,
- particles with well-formed functions, but not affecting the universe space (non operating functions),
- particles with well-formed functions, which are able to modify the universe space – called working particles.

The area of the functional activity of a particle is a square region centered around the particle. In a given time step, the functions of particles are activated in a randomly chosen succession; a function activated later may affect the area, which was transformed by functions activated earlier.

The transformation induced by the particle function obeys the momentum and energy conservation laws. This implies that after the transformation the sum of bond energy increment of all particles and complexes participating in the reaction is positive.

2.3. Computer implementation

A computer program simulating the universe has been developed (Pascal, C). The particulars of the algorithm are specified in the following pseudocode:

```
{Main program of the Universe}
Program Universe;
var
  ClockCycleNumber : Longint;
  UniverseBinaryFile : String;
  FinishSimulation : Boolean;
begin
  Init_memory_structures;
  Load_data_file(UniverseBinaryFile);
  FinishSimulation := False;
{Main computation loop begins here}
  while not FinishSimulation do
  begin
    PHOTONS;
    MOVEALL;
    PROGRAMS;
    ClockCycleNumber := ClockCycleNumber+1;
  end
{Store the state of simulation on the disk}
  Save_data_file(UniverseBinaryFile);
  Dispose_memory_structures;
end.

Procedure PHOTONS;
{The procedure realizing the movement and collisions of photons. Photons are
specified as the list P of records of type Photon. Photons are drawn from table P in
random order, according to their state new placement is calculated. Then collisions
with particles are solved. Finally, newly generated photons are placed in the table P}
var
  PhoFinalEnergy : Real;
  ph, newph : Photon;
  i : Longint;
  PhotonHit : Boolean;
begin
  P := RandomPermutation(P);
  for i := 1 to size(P) do
  begin
    MoveMemory(P[i], ph);
    PhotonHit := MovePhoton(ph);
    if PhotonHit then begin
      PhoFinalEnergy := SolvePhotonCollision(ph,newph);
      if PhoFinalEnergy > 0 then begin
```

```
{New photon has been generated}
        MoveMemory(newph, P[i])
      end
    end
    else MoveMemory(ph, P[i]);
  end
end

Procedure MOVEALL;
{The main procedure realizing the collision and movement of particles. The particles are specified as the list U of records Particle. Particles are drawn randomly from the list. For each particle the movement direction is chosen (vertical or horizontal). Then a special structure, called semicomplex is generated. Semicomplex is a list of particles adjoining to the given particle in the given direction. For all the particles belonging to the semicomplex a collision is solved. Then the movement of particles is done}
var
  par : Particle;
  i : Longint;
  MovingDirection : Nil, Hor, Ver, All;
  SemiComplex : ListOfParticles;
begin
  U := RandomPermutation(U);
  for i := 1 to size(U) do
  begin
    par := U[i];
    while not par.moveDir = All do
      begin
{Choose direction of move}
        if par.moveDir = Nil then
          MovingDirection := ChooseRandom(Hor, Ver);
            if par.moveDir = Hor then
              MovingDirection := Ver
                else MovingDirection := Hor
{Build semicomplex (particles adjoining par in the direction MovingDirection)}
        BuildSemiComplex(U, par, MovingDirection, SemiComplex);
{For all particles in a semicomplex solve collisions}
        SolveParCollisions(SemiComplex, par, MovingDirection);
{Move particles}
        MoveParticles(SemiComplex, MovingDirection);
        for All particles in a SemiComplex do
          begin
            if particle.MoveDir = Nil then
              particle.MoveDir := MovingDirection
            if particle.MoveDir = Hor or Ver then
              particle.MoveDir = All;
          end
      end
  end
end
```

```
Procedure PROGRAMS;
{Procedure executes programs stored in the particles. Particles are specified as the
list U of records of type Particle. Particles are drawn randomly from the list. Then
the atom string of a particle is interpreted as a program. The program is executed in
an auxiliary space (called Omega). This space is a copy of fragment of the Universe
space, surrounding the particle, which program is executed. After execution of a
program the bond energy of the modified particles is calculated. If it is not less
than the energy before execution of a program, the auxiliary copy (Omega) is stored
in a global space}
var
  par : Particle;
  i : Longint;
  ProgramValid : Boolean;
  ProgramStr : ProgramData;
  Omega : UniverseTable; {Type representing Universe dynamic data}
  Energy : Real;
begin
  U := RandomPermutation(U);
  for i := 1 to size(U) do
  begin
    par := U[i];
    ProgramStr := DecodeProgram(par);
    ProgramValid := CheckIfProgramValid(ProgramStr);
    if ProgramValid then begin
      CopyTmpDataToOmega(Omega, par)
      Energy := GetEnergy(Omega);
      InterpreteProgram(Omega, par, ProgramStr);
      Energy:=GetEnergy(Omega) - Energy;
      if Energy >= 0 then
{Program has been succesfully executed}
        StoreTmpDataToGlobal(Omega);
    end
  end
end
```

During the simulation, a graphic pattern of the universe is displayed on the screen. Besides the main simulation program, several tools have been designed to facilitate preparation of initial states of the universe and observation of some of its parameters.

2.4. Properties of the universe

The universe is the programming environment for individual-based modelling. Specific interactions of particles existing in the universe make possible effective modelling of self-modifying systems and among them the processes of emerging and developing of self-organizing structures, which would be very difficult or even impossible to model by other means. The presented universe is a compromise between the need to possess a possibly good imitation of basics properties of modelled environment on possibly low level and the simplicity of specification and effectiveness of computer implementation.

The universe may be seen as an overall model of some reality, on which one can run the simulation experiments. This model fulfills the following postulates given by Langton [11] referred to computer-based *artificial life* models:

(1) they consist of populations of simple programs or specifications,
(2) there is no single program that directs all other programs,
(3) each program details the way, in which a simple entity reacts to local situations in its environment, including encounter with other entities,
(4) there are no rules in the system that dictate global behavior,
(5) any behavior at levels higher than the individual programs is therefore emergent.

Typical way of modelling in the universe consists of preparation of the initial state of the system, and then simulating the evolution of it. Preparation of the initial state of the universe needs specification of the position, bonds, and velocity of all particles and specification of their strings of atoms.

3. Self-reproduction system

The universe makes possible modelling of self-reproduction systems, whose inherent property are aperiodic times of originating and various shapes of growing structures.

In the paper [6] the self-reproduction system modeled in the universe has been proposed. It was to resemble the reproduction of a living cell. However, the need to introduce several subtle mechanisms to control the various stages of self-reproduction process, together with the slowness of previous generation computers hindered the realization of the idea.

Now it seems possible to simulate in the universe a simpler system. It consists of a nucleus, which is responsible for replicating the information and transforming it into working particles, and of a wall surrounding the nucleus as shown in the Figure 3.

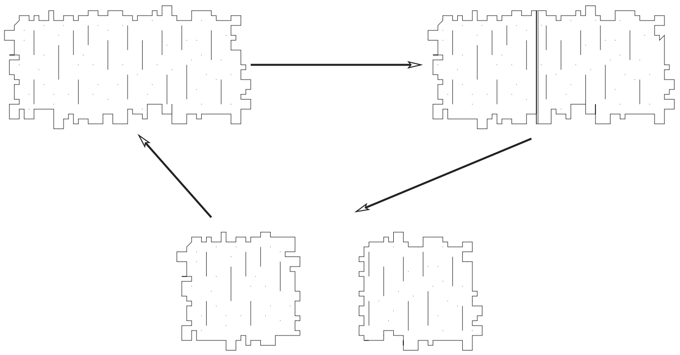

Figure 3. Structure and stages of the self-replication process of the quasi cell. The nucleus (information strings, replicators and decoders) is surrounded by the wall. The vertical lines inside the cell represent the information strings

The nucleus of the system is formed by three types of particle structures:

Information string – the ordered chain of particles. It contains the encoded description of a particle,

D – a particle transforming information contained in the information string into working particle – decoder,

R – a particle replicating an information string – replicator.

There are two parallel processes doing the self-replication. Having only particles of replicator (*R*) system would be able to replicate the information. With particles of decoder (*D*) we are able to obtain new particles, with atom structure depending on information strings. Both particles need a building material to work. This material consist of uniformly distributed set of 5-atom particles with every possible combination of atoms.

The schematic way of work of both particles is shown in Figure 4. One can see that decoder moves along the information string and concatenates appropriate particles into one particle. As a result of the decoding process a new particle originates – its structure is identical to the description contained in the information string. Replicator also moves along the information string but concatenates particles to form complex identical to the information string.

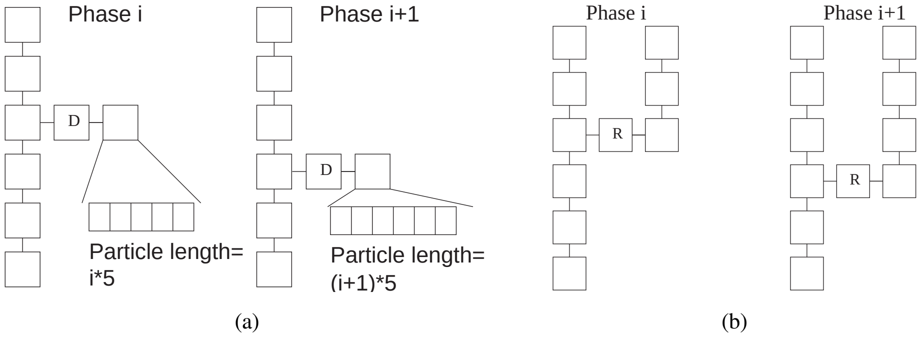

Figure 4. Schema of working decoder (a) and replicator (b)

To make the process of self-replication more effective the nucleus system (information, strings, replicators and decoders) is surrounded by a wall. The wall has the form of an irregular ring consisting of particles called W.

During the process of self-replication the wall is continuously enlarged. Simultaneously the additional vertical wall, which will later divide the cell is developed. The following particles are taking part in the enlarging and dividing the wall:

W – a particle with no function, used as building material for constructing the wall. It exists in three versions (differing the order of atoms) named W_5, W_7, and W_9, having the bond energies 5, 7, and 9, respectively,

M – a particle with no function consisting of five atoms, used as building material for constructing other particles by the action of decoder D and replicator R. The versions of the particle are named as $M_1, \ldots M_{32}$,

E – a particle with no function, used as a source of energy. It exists in five versions: E_{18}, E_{20}, E_{22}, E_{28}, and E_{30}, having the bond energies 18, 20, 22, 28, and 30, respectively,

A – a particle attaching two particles W_9 to the wall (Figure 5a). The particle A (and also particles S, C, and AW) exists in four variants, which construct lower, left, upper, and right wall. The particle A always exists in a form of complex with particle AW. Prior to the realization of its function the particle A has to have two particles W_7 bonded to itself. During the execution of its function the particle A detaches the particles W_7 from itself, transforms them into W_9, and attaches them to the wall. The transformation of W_7 into W_9 provides energy to make the energy balance of the entire operation positive,

S – a particle shifting the particles W_9 already built in into the wall (Figure 5b). Prior to the realization of its function the particle S has to have a particle E_{20} first attached to itself. During the execution of its function the particle S shifts the particles W_9 within the wall, and then transforms the particle E_{20} into E_{22}, and detaches it from itself. The transformation of E_{20} into E_{22} provides 2 units of energy, which makes the energy balance of the reaction positive,

C – a particle detaching from the wall a pair of particles W_9, which protrude into the interior of the cell, and transforming them into particles W_7 (Figure 5c). The function of the particle is executed when the particle is bonded with a particle E_{20}, the latter being later transformed into E_{28}, and then detached. The transformation of E_{20} into E_{28} provides 8 units of energy, which compensate the loss of energy used for detachments.

AW – a particle bonding two particles W_7 to a particle A. The particle AW is permanently attached to a particle A,

AE – a particle bonding a particle E_{20} to particle S or C,

IW – a particle transporting a particle W_5 into the cell and transforming it into particle W_7. The transformation makes the energy balance of the reaction positive,

IE – a particle transporting a particle E_{18} into the cell and transforming it into particles E_{20}. The transformation makes the energy balance of the reaction positive,

IM – a particle transporting particles $M_1, \dots M_{32}$ into the cell and transforming a particle E_{20} or E_{22} into particle E_{28}. The transformation makes the energy balance of the reaction positive,

RE – a particle removing particles E_{22} and E_{28} from the cell and transforming them into particles E_{30}. The transformation makes the energy balance of the reaction positive,

$BD1$, $BD2$ – a pair of cooperating particles. They construct the vertical wall dividing the cell, and are attached to the end of growing construction. They exist in two variants building the dividing wall in the upper and lower direction.

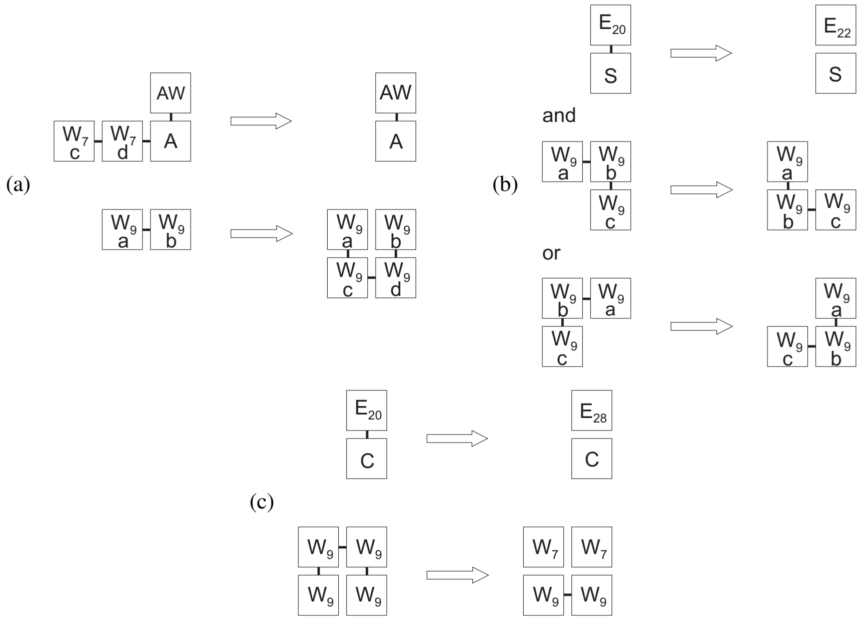

Figure 5. The action of the particles constructing the lower wall: (a) A, (b) S, and (c) C

The particles A, S and C possess functions directly enlarging the walls, while the functions of the particles AW, AE, IW, IE, IM, RE transport through the walls the buildings material (particles W and M) and particles providing energy (E), attach them to particles A, S i C and remove used up particles E.

4. Simulation experiments

A number of preliminary simulation experiments has been carried out. They were run separately for the replication of the nucleus system and for the enlarging and dividing of the wall.

During the functional interactions, in both experiments the photons have been created in the system but by a proper setting of probabilities the effects of collisions of photons with particles have been reduced to exchange the energy of photons for kinetic energy of particles. This way the particles were transformed only by functions of other particles but not by the collisions with photons.

In both experiments the size of the universe was set to 1000×1000 squares.

4.1. The Nucleus

To simulate behavior of the nucleus, the information strings containing the description of particles R and D were used. That way the system is self-replicating.

The parameters of the experiment are listed below:

Total number of particles:	100000
Number of information strings:	6
Number of decoders:	3
Number of replicators:	3

After 100000 cycles there were 2 additional information strings and one more particle of decoder. That meant the system really worked.

The most important problem was slow speed of the self-reproduction processes. To begin the process, either decoder D or replicator R has to bond itself to the top particle of the information string. To do this, it has to move the string to the square adjacent to itself. Successful moving of a complex to a new location requires free space in this location. When the density of particles is high and moved complex is big, the chance of finding such free space is low.

The idea of improving the performance of the system was to divide the information string into smaller parts. That way information strings would be easier to transport across the system. The substrings, representing information, were supplied with tags, describing its proper order, as shown in Figure 6. The program of replicator was unchanged. The decoder has to be equipped with some kind of memory, indicating right order of decoded substrings. The memory is represented by special kind of particle, which is connected to decoder when it reaches the end of the substring. The next substring should have the indicator at its beginning identical to the one connected to the decoder. The last substring in the order has the special indicator - ”end of information”, indicating the end of decoding process.

The results of both variants of experiments are shown in Figure 7, where the number of particles has been shown. Because only decoder can cause decrement of the particles, the figure shows that the system with substring produces working particles more effectively.

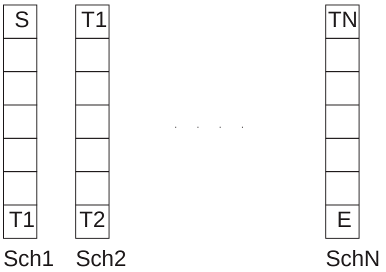

Figure 6. Splitted information chain equipped with tags, where: S – start of the information string tag, T_i, $i = 1,\dots,N$ – ordering tags, E – end of information string tag

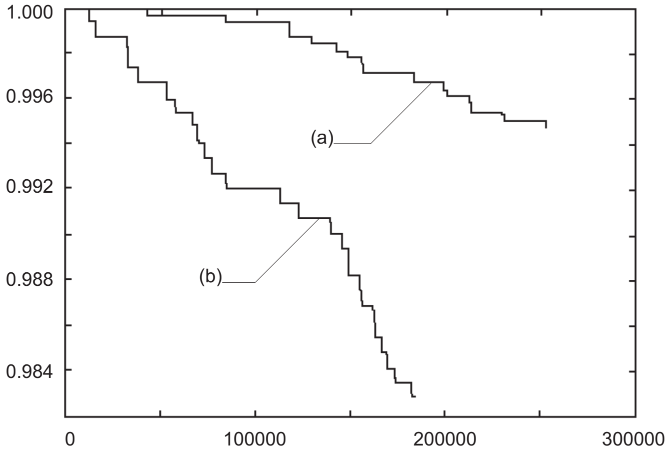

Figure 7. Number of particles in the system modelling the nucleus replication. Decrement of the number of particles is equivalent to the mass of the working particles produced by the decoder D. (a) Results for full-length information strings; (b) Results for information strings divided into substrings

4.2. The Wall

In Figure 8 the initial state of the universe containing the cell, and the states in some characteristic stages of development of the cell have been shown.

In the initial state the universe contained the wall built of 130 particles W_9. Outside the wall were 1500 particles W_5 and 2500 particles E_{18}. Inside the cell were 16 particles A bonded to 16 particles AW, 32 particles S, 16 particles C, 12 particles AE, 4 particles IW, 8 particles IE, and 6 particles RE. Moreover, inside the cell there has been a complex of particles formed of 6 particles $W9$, 2 pair of particles $BD1$ and $BD2$, and 2 particles E. The complex was a seed of vertical wall dividing the cell.

The position and velocities of the particles outside and inside of the cell have been chosen randomly.

During the consecutive simulation steps the system of particles inside the cell moving and colliding gradually enlarges the walls and simultaneously builds up the vertical dividing wall using the particles W as building material and the particles E as a source of energy.

The designing process of the initial state of the cell has been guided by the need of possibly fast build up of the cell wall and its division.

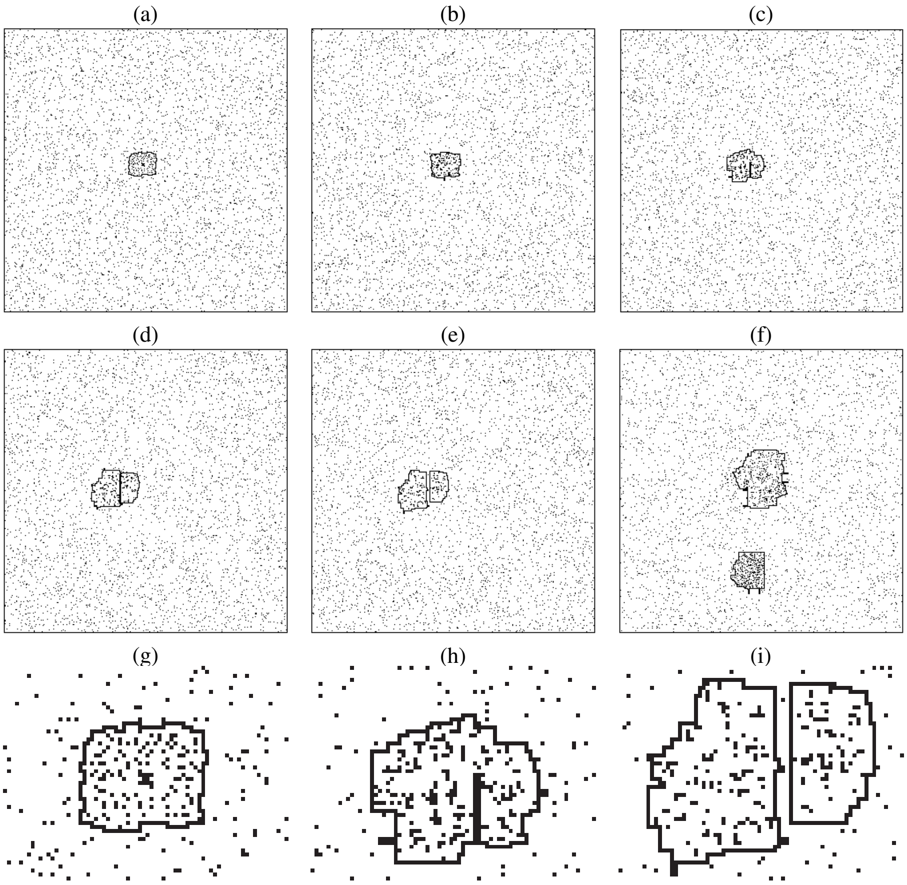

Figure 8. Build up and division of the cell. (a) Initial state of the system, (b)–(f) the states after 200, 600, 1350, 1400, and 6500 steps and (g), (h), and (i) zoomed parts of the figures (a), (c), and (e), respectively

The number of particles possessing functions and placed inside the cell, has been chosen experimentally to obtain relatively steady and possibly fast growing of the wall. Taking in the introductory experiments too many particles IW (importing into the cell particles W) resulted in the course of simulation in growing density of particles W inside the cell, which then leaded to complete blockage of particle movements, and in consequence to complete halt of building up of the wall. To obtain optimal initial numbers of particles directly (without realizing experiments) some regulation mechanism should be introduced to the system, which would stabilize the proper density of various particle types.

Another difficulty was the problem of recognition by functions of particles other particles and their structures. The recognition was based only on characteristic fragments of recognized structures because of the sometimes large length of recognized particles or the large size of recognized structures of particles. This leads sometimes to errors difficult to predict. An example of the consequence of such error is shown in Figure 9, where by some coincidence the particle D instead of detaching the particles W from the wall detached the particle $BD1$ building up the dividing wall, and the detached particle in the new situation has cut the cell's wall dividing it into two parts. The difficulties with recognition grow with

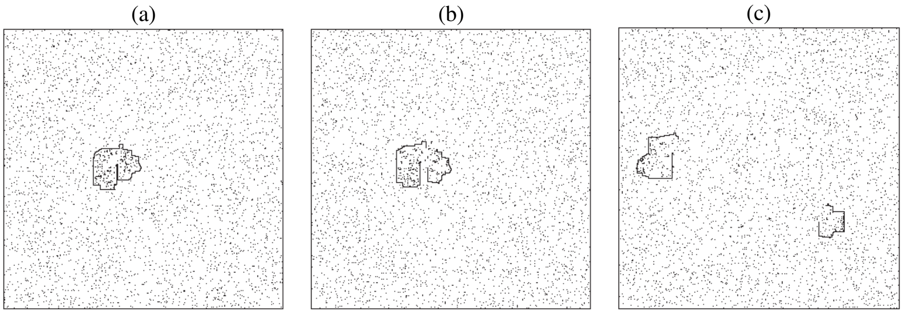

Figure 9. An example of failure. Resulting from erroneous recognition of the wall, the particle D cut off the particle $BD1$ instead of the particle W. In consequence the particle $BD1$, which normally is attached to the built up dividing wall moving freely approached the cell's wall and cut it dividing it into two parts

the number of types of particles and the number of their specific structures, and may be a serious obstacle during construction of the complete self-reproduction system.

The discussed difficulties indicate the practical impossibility of direct, faultless design of initial state of the system ensuring expected complex behavior. This is the result of ambiguity of the aims and difficulty of predicting the evolution of the system. Obtaining of the proper predictions is additionally difficult because of the possibility of occurrence of self-modifications. All this is a specific manifest of unpredictability in universal computing systems, which is also the property of universe [12]. Similar situation arises in the real economical or political systems, where the aims are ambiguous, there are no exact mathematical models or they have limited range of applicability, and the effects of some decisions can be seen only during evolution of the system in time.

Another observation is that improving the self-reproduction system by the deduction or by trial and error method one can come across problems and then overcome them by finding the solutions, which are similar to those found in the real systems. The need to move into the cell the particles rich in energy and after use removing them is an example of solution of such problem. Another example is (not realized yet) a need to control the density of various types of particles inside the cell, or very important problem of recognizing the types of particles inside the cell. In the case of competing various self-reproduction systems the problem of differentiating the particles belonging to the system from the ones belonging to the other system will arise.

The existing similarity of solutions may be explained by the fact that in the real systems taking into account the existing laws of physics, there exist solutions in some sense optimal, which are found by biological systems and their dispersion is limited. If we had an individual-based programming environment, which would be in basic assumptions similar to the real environment, than the structures realizing some functions built in this environment would have to be similar to their counterparts in the real world. The existence of such similarities of solutions shows that the presented universe in spite of many simplifications, is modelling environment, which is a good model of the real environment, in which biological life takes place.

5. Concluding remarks

The use of the abstract universe – a universal individual-based modelling environment to modelling of self-reproduction system has been presented. The preliminary simulation experiments illustrate the effects of cooperative behavior of simple entities interacting with one another and leading to arising of complex processes. The presented experiments show another feature offered by the individual models – they can mimic nature by the "first principles" and do not use any averaging; just operate on the finite number of elements.

The proposed nucleus is a perfect copying system. One always obtains an ideal copy of the information string as well as the ideal particle built from the description of the chain. So it is impossible to obtain any new quality in that system. The term "new quality" means the information or new kind of particle, which was not present at the beginning of experiment. In natural living system such new artifacts are the source of diversity. One of the sources of diversity is mutation, *i.e.* some destruction of information string or imperfect decoding process.

Modelling of mutation can be realized by introducing special kind of a working particle, which would change the order of atoms of other particles. Another way of realizing mutation is to enable the transformations of particles by interactions with photons (by setting the proper parameters of the simulation). As was observed in one simulation experiment, mutation of information string leads to fast degradation of the system [12].

It seems that after overcoming some above mentioned difficulties it is possible to simulate the proposed complete self-reproduction system in the near future. The observation of its evolution would be very informative especially when the random modification were allowed, or in the system would exist cooperating or competing different self-reproduction systems.

So we have discussed the system where conditions were set at the start, and they quite precisely define the behavior of the system. One can ask about the system where there is hardly any information, and one can expect that such information is to emerge during the simulation. That information is represented by the particles having useful functions, originated during the simulation. The experiment with self-organization is quite simple to conduct in the universe environment. One needs to put some particles and allow them to concatenate. After some cycles more lengthy particles should appear in the system. Some of them may contain useful functions. Those are not known at the start of experiment.

References

[1] Hockney R W and Eastwood J W 1988 *Computer Simulation Using Particles* Bristol, Institute of Physics Publishing

[2] Hoover W G 1986 *Molecular dynamics* Lectures Notes in Physics t. 258, Berlin, Springer-Verlag

[3] Toffoli T and Margolus N 1987 *Cellular Automata Machines* Cambridge, Mass., MIT Press

[4] Neumann von J 1966 *Theory of Self-reproducing Automata* Burks A (ed.), Champaign-Urbana: University of Illinois Press

[5] Adami Ch 1998 *Introduction to artifial life* Springer, New York

[6] Jędruch W and Sampson J R 1987 *Bio-Systems* **20** 329

[7] Jędruch W and Barski M 1990 *Bio-Systems* **24** 99

[8] Jędruch W 1997 *Programming environment for molecular modelling of complex systems* WPG Gdansk (in Polish)

[9] Waniewski J and Jędruch W 1999 *Mathematical Biosciences* **157** 23

[10] Waniewski J and Jędruch W 2000 *Int. J. Appl. Math. and Comp. Sci.* **10** (1) 175

[11] Langton C G 1989 *Artificial life, Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, Langton C G (ed.), Los Alamos, New Mexico, September 1987, Redwood City, Addison-Wesley

[12] Gramza M and Kujawa J 1999 *Individual-based modelling of the biological systems on the supercomputers*, MSc Thesis, Technical University of Gdansk (in Polish)