Model reproduction of non-equilibrium thermodynamics principles as a means to provide software self-development

E A Kol'chugina
Department of software and computer application, Penza State University, Krasnaya Street 40, Penza, 400026, Russian Federation
E-mail: kea_sci@list.ru

Abstract Further evolution of technologies in material sciences, hardware and software demands reproduction of self-organization as a manifestation of non-equilibrium thermodynamics principles in new artificial environment. This is especially actual for software engineering, where reproduction of non-equilibrium thermodynamics principles will allow enhancing speed and quality of development process. In this article we reveal, list and analyze principles to be reproduced in artificial systems to provide capabilities for self-development, self-healing and self-improvement. It is shown that self-development process is impossible without reproducing the metaphor of electric charge with its abilities for attraction and repulsion. But this metaphor is usually avoided in computer models. We propose and consider a method to reproduce the metaphor of electric charge in computer models by means of UNIX/Linux operating system tools. The obtained results provide the basis to establish an artificial physics, a new research area dedicated to adequate reproduction of principles of physics in the artificial environment of computer models.

1. Introduction
The reproduction of non-equilibrium thermodynamics principles in computer models became of importance not only for physics, material technologies, natural sciences, but for cybernetics and software engineering also.

In the end, software development has become an industry where it is necessary to engage developers’ teams to implement any more or less competitive project. The epoch of successful individual developers has gone.

Modern software projects involve huge investments of money and human labour. Many of such projects are dedicated for distributed or mobile systems. But the extensive way to achieve high computational performance and powerful functionality faces with two challenges.

The first is a debugging problem. It became impossible to test and debug applications completely during their life cycle.

The second is a problem of adaptation and self-improvement. Typical software is rigid and incapable to interact with its environment for adaptation and improvement. So, changes of environment, e.g. updates of operating system, can lead to destruction of the software due to arising conflicts.

Thus, there is occurring a need of endless revision and release of new software versions.

The solution is to make software more reasonable, if not intelligent, and to endow it with abilities to improve, heal and teach itself. This demands creation of new nonstandard programming paradigms.
The most widely known paradigm to create intellectual software and hardware systems is neural networks. But this paradigm is too complicated to be used for design of simple service software, e.g. utilities. The resulting software should not be too clever and complicated, and, as a result, expensive.

A shorter and simpler way to make software reasonable, but not intelligent, is to reproduce the principles observed in nature and stated in mathematics, physics, chemistry and biology. These principles allow the creation of self-organizing systems, and they were formulated primarily for non-equilibrium thermodynamics of irreversible processes [1-3].

Self-organizing systems exist due to and through the interactions with their environment by energy-matter exchange. Self-organizing systems are capable for self-assembly and self-ordering in response of environmental changes. So, reproduction of non-equilibrium thermodynamics principles in virtual cybernetic medium can endow program systems with abilities for self-development throughout their entire life cycle.

One of scientific branches aimed to reproduce non-equilibrium dynamics of biological systems in cybernetic medium is artificial life [4]. A subdivision of artificial life closely connected with origin of life theory, molecular biology and biochemistry is artificial chemistry [5-21,23,24]. Software artificial chemistry models allow considering programs as artificial molecules. Respectively, the interactions between programs are considered as analogues of chemical reactions which lead to construction and destruction of artificial molecules, i.e. programs. This point of view opens the way to endow programs and groups of interacting programs with abilities for self-ordering and reordering, self-design, adaptation and self-improvement, i.e. continued self-development.

Many software models of artificial chemistry try to imitate the principles of self-organization and non-equilibrium thermodynamics in software systems [5-7]. An overview and analysis of such models was given by the author in [7].

But in [7] it was shown also that attempts to replicate non-equilibrium thermodynamics principles in software systems are imperfect. The reason is that basic principles of physics of the real world are not reproduced in computer models adequately.

In [7] it was revealed the necessity to formulate and study principles of deeper level of reality simulation in the form of artificial physics.

Here in this article, we make a first step towards forming these principles of artificial physics. We start from analysis of basic categories and principles of the physics of the real world, which are connected with non-equilibrium thermodynamics processes. We are also giving analysis of methods of reproduction of these categories and principles in artificial life and artificial chemistry models.

2. Methods
To analyze the basic categories and principles necessary for reproducing non-equilibrium dynamics in computer models, we need firstly to name and list such categories and principles.

According to the analysis of artificial chemistry models given in [7], these categories, which must be reproduced to ensure the presence of processes with non-equilibrium dynamics, include space, time, energy, entropy, and electric charge. Here we consider different ways of representations of such categories in virtual medium.

2.1. Space and time
In many models of artificial life and artificial chemistry related to the theory of origin of life, such as "Venus" and "Luna" [8], "Tierra" [9-11], "Avida" [12-17], "Amoeba" [18-20], the model space is understood as a memory space of some virtual computer. That space is usually one-dimensional, more rarely it has two or more dimensions. An interesting remark given in [10] is that model space can be considered as non-Euclidian space, since every cell can be reached directly from any other cell by jump performed by central processor unit (CPU).

An analogy between computer memory and model space establishes another important quality of space defined in abovementioned artificial chemistry models: the space in such models is always discrete or cellular.
The model time is organized in more simple and obvious way: it is linear and unidirectional. Really the model time is a CPU time.

2.2. Energy and dissipation

It is known that time and space of the real world possess their own energy. The same is true for the artificial worlds in “Venus” and “Luna” [8], “Tierra” [9-11], “Avida” [12-17], “Amoeba” [18-20], and other similar models of artificial chemistry. In these models, energy is the amount of available CPU time. This establishes a direct relationship between model time, space and energy: the more time the analogue of non-equilibrium process is emulated, the more energy it consumes; the more model energy a computer analogue of non-equilibrium process retains, the more time it is executed; the wider model space the computer analogues of non-equilibrium processes occupy, the more energy they possess [7,9-17].

According to this introduced analogy, the matter in computer worlds is represented by bits and bytes of information stored in memory.

But computer models have an important restriction that seriously limits similarity between physics of artificial and real worlds. Physics of the real world allows possibility of mutual transformations between energy and matter. In artificial computer worlds energy (CPU time) could be transformed into matter (information stored in memory). The converse is not true: information stored in memory can be traded for processor time, but not directly converted to it.

This makes possible to talk about energy dissipation in models of non-equilibrium processes of artificial life and artificial chemistry. An energy, or CPU time, being spent once on changing the state of the memory cannot be restored back.

So software artificial chemistry models trying to reproduce non-equilibrium thermodynamics of real-world self-organizing systems are inevitably dissipative and suffer energy loss.

2.3. Entropy

Definitions of non-equilibrium, non-equilibrium state, and non-equilibrium process are usually given using the term entropy [1-3]. But in computer models, the situation is different. Usually entropy looks like a secondary (compared to energy) principle, which is synonymous with disorder and heterogeneity.

The problem is that it is not clear what entropy itself is. Entropy can be defined in different ways, as shown in [7]. When applied to models of artificial chemistry, entropy can be defined, for example, as a measure of the absence of structures representing data or programs in the memory of a virtual computer [21], or by analyzing the frequency of bit sequences and the speed of their replication by means of statistical mechanics [13].

Thus, to be used in computer models, the definition of entropy should be refined, primarily in physics and natural sciences.

2.4. Self-organization as one of the key principles of constitution of matter

In our opinion, the best definition of self-organization was given by A. P. Rudenko [22]. According to this definition, the main condition for self-organization is non-equilibrium, the main reason is useful work against equilibrium, and the driving force is a part of free energy of exchange process spent on doing internal useful work [22].

In [22] self-organization, along with organization, is understood as one of the key principles of constitution and existence of matter. These principles determine existence of matter in dynamic and static forms. The organization is entropic principle, while self-organization is anti-entropic principle.

In artificial chemistry models, the metaphors for non-equilibrium state and free energy are program execution and CPU time, respectively. Useful work against equilibrium is work that provides and maintains self-ordering. In artificial chemistry models, useful and rewarded calculations can be considered as such.
Current modern software is built on the principle of organization: software execution leads to the loss of the model analogue of energy (CPU time) and subsequent termination. After that, complex structures representing code and data are destroyed.

On the contrary, reproduction of self-organization principle in virtual medium implies that:

- software tends to perpetual execution as analogue of non-equilibrium state;
- perpetual execution requires an unlimited source of analogue of energy, i.e. CPU time;
- CPU time should be provided to the software primarily as a reward for performing useful operations.

At the first glance, the means to ensure self-organization in computer medium are obvious. The question is how to represent an analogue of work against equilibrium adequately. In the real physical world, such work is done naturally, spontaneously, and without compulsion. But in [23,7], it was found that the von Neumann architecture is not suitable for reproducing spontaneity, since the concept of this architecture requires that all interactions between programs must be authorized by the central processor unit.

In [7] it was also stated that spontaneous interactions should arise independently, in decentralized manner and inevitably, according to the properties of interacting components and the strength of their mutual attraction and repelling.

And here we face the most important problem: how to ensure decentralized spontaneity. The most typical examples of spontaneous interactions in the real world are those based on an electric charge.

Thus, to solve the problem excavated, it is necessary to imitate in computer models analogue of electric charge.

2.5. Electric charge, attraction and repelling

The basic law concerning to electricity says that like charges inevitably repel and unlike charges inevitably attract. The reproduction of this law in virtual computer medium is of immense necessity for software engineering: this is a key to achieve spontaneous self-assembly and self-development of programs.

Despite the fact that electricity has been actively used by mankind in the last two centuries, it is still not clear what it is. To understand something completely means to be able to reproduce it, but the electric charge and field have never been artificially recreated from scratch until now.

The lack of complete understanding of nature of electric charge in physics makes simulation of electric charge in computer models a great difficulty.

Electric forces of attraction can be seen as a special kind of “glue” that combine and unite atoms and molecules. According to this point of view, there are two ways can be offered to simulate forces of attraction in computer models of artificial chemistry. Any of these methods implies that artificial atoms and molecules can be attracted to each other through something common that can be shared between them.

The first and obvious way to implement such shareable medium is to use common data structures, e.g. variables situated in shared memory. This solution was considered and experimentally studied in [24]. In the conducted experiments it was revealed that common variables can serve as a base for spontaneous formation of dynamical cyclic structures. These structures were formed by simplest functions that run as independent processes and use the same shared variables. The result of simulation in general looked like a spontaneous formation of high-level programs form the variety of simpler ones, where shared variables played the role of “glue” or a means to simulate attraction and repelling forces.

But the relations formed between functions in [24] were weak, and they were easily destroyed by invading functions. The reason is that relations were indirect and mediated. Other drawbacks of approach proposed in [24] were lack of selectivity and asymmetry on the set of possible relations.

The second way to create “glue” between programs is to use common interactions, not objects. These interactions can mimic shared electron clouds that combine atoms into molecules. To implement this, a
process executing program unit destined to be combined with other units must be a system of “sinks” and “sources”.

The “sources” play role of extra electrons on external orbitals, while “sinks” represent unfilled orbitals. The process with excess of “sources” is an analogue of negative ion, while the process with excess of “sinks” is a model analogue of positive ion.

These “sinks” and “sources” can be organized by means of inter-process communication (IPC) system in UNIX/Linux operating systems. Sockets are the most probable candidates, but other IPC tools are also possible to use.

Such approach is similar to those in physics, where interconnections and shared particles are understood as vortexes [25]. We don’t discuss the ideas proposed in [25] from the point of view of physics, but they can serve as useful abstractions from the point of view of cybernetics to imitate particles and their interconnections in impoverished computer medium.

The proposed approach will be studied in details in our further works. It is expected that interactions between programs and corresponding processes will be direct, stronger and more stable than it was observed in [24]. But the problem of representing asymmetry on the set of possible interactions remains.

3. Results, discussion and further works

The biggest drawback of any computer model that tries to reproduce self-organization during non-equilibrium processes is the lack of spontaneity and inevitability of interactions. To overcome this, it is necessary to reproduce physical structure of matter in computer models properly.

Here, in this article, we have detected, listed and analyzed physical categories and principles to be simulated, which are necessary to reproduce self-organization artificially. We found out that some terms used in studies of self-organizing systems, e.g. “entropy”, need clearer definition to be represented in computer models (see table 1).

Table 1. Representation of basic physical categories and principles in computer models (a summary).

| Category or principle | Form of representation | Method of realization |
|-----------------------|------------------------|-----------------------|
| Time                  | Explicitly             | Usually represented as linear and unidirectional; identical to CPU time |
| Space                 | Explicitly             | Memory space of the real or virtual computer |
| Matter                | Explicitly or implicitly | Information stored in memory |
| Energy                | Explicitly or implicitly | An amount of available CPU time |
| Entropy               | Explicitly or implicitly | As the definition of entropy is not univocal, this category represented in different ways |
| Self-organization     | Explicitly (as a goal of simulation) or implicitly (as a side effect) | Dynamically through intra system interactions that arising due to modeling rules |
| Electric charge       | Not represented        | Possibly can be represented by means of sockets |

According to the summary given in table 1, most of the considered categories and principles are represented explicitly or implicitly in artificial chemistry models with the exception of electric charge. But here we found that this basic principle is essential for reproducing self-organization in a virtual medium. Electric charge simulation allows running programs to be “glued together” into dynamical structures without central control of the processor unit. We proposed that electric charge, electric attraction, and electron clouds can be represented using a system of “sinks” and “sources”.

We also found candidate tools to simulate electric charge in computer models. These can be sockets in UNIX/Linux operating systems. In our further works we shall study this approach in details.
4. Conclusion
Development of artificial intelligence gave impact to studies in artificial life, a new research area to achieve efficient technologies based on metaphor of self-organization systems in biology. The necessity of deep understanding of the nature of self-organization in biological systems required a new level of research in the form of artificial chemistry models.

In [7] and here it was shown that further research in the field of artificial chemistry and the need for effective reproduction of self-organization in computer models require an adequate representation of physical categories and principles in such models, since the physical level of matter organization is the first where self-organization is observed.

The proposed approaches and the results achieved in this article can become the basis for the development of a new direction of self-organization research in the field of computer science: artificial physics.

The studies of this field are extremely necessary for computer science and technique to endow hardware and software with new abilities for self-organization manifested in forms of self-development, self-healing and self-improvement.

References
[1] Prigogine I 1955 Introduction to Thermodynamics of Irreversible Processes (Springfield: Thomas)
[2] Gyarmati I 1970 Non-Equilibrium Thermodynamics Field Theory and Variational Principles (Berlin-N.Y: Springer)
[3] Ziegler H 1963 Some extremum principles in irreversible thermodynamics with application to continuum mechanics Progress in Solid Mechanics 4 ed. I. N. Sneddon and R. Hill (Amsterdam, The Netherlands: North-Holland Publishing Company) 2 p 91-193
[4] Langton C G 1989 Artificial life Artificial Life: Proc. of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (September 1987, Los Alamos, New Mexico) (SFI Studies in the Sciences of Complexity VI) ed. C. G. Langton (Redwood City, CA: Addison-Wesley) p 1-47
[5] Dittrich P, Ziegler J and Banzhaf W 2001 Artificial chemistries – a review Artificial Life 7(3) 225-75
[6] Banzhaf W and Yamamoto L 2015 Artificial Chemistries (Cambridge, Massachusetts; London, England: The MIT Press)
[7] Kol’chugina E A 2019 Self-organization and self-development of programs in artificial chemistry models (in Russian) (Kazan: OOO “Buk”)
[8] Rasmussen S, Knudsen C and Feldberg R 1991 Dynamics of programmable matter Artificial Life II: Proc. of Workshop on Artificial Life (February 1990, Santa Fe, New Mexico) (SFI Studies in the Sciences of Complexity X) ed. C. G. Langton, C. Taylor et al (Redwood City, CA: Addison-Wesley) p 211-54
[9] Ray T S 1991 An approach to the synthesis of life Artificial Life II: Proc. of Workshop on Artificial Life (February 1990, Santa Fe, New Mexico) (SFI Studies in the Sciences of Complexity X) ed. C. G. Langton, C. Taylor et al (Redwood City, CA: Addison-Wesley) p 371-408
[10] Ray T S 1997 An evolutionary approach to synthetic biology: zen and the art of creating life Artificial Life: an overview ed. C. G. Langton (Cambridge, Massachusetts: The MIT Press) p 179-209
[11] Thearling K and Ray T S 1994 Evolving multi-cellular artificial life Artificial Life IV, Proc. of the 4th International Workshop on Synthesis and Simulation of Living Systems (6-8 July 1994, the Massachusetts Institute of Technology) ed. R. A. Brooks and P. Maes (Cambridge, MA: MIT Press) p 283-8
[12] Adami C and Brown C T 1994 Evolutionary learning in the 2D artificial life system «Avida» Artificial Life IV, Proc. of the 4th International Workshop on Synthesis and Simulation of
[13] Adami C 1994 On modelling life *Artificial Life IV, Proc. of the 4th International Workshop on Synthesis and Simulation of Living Systems (6-8 July 1994, the Massachusetts Institute of Technology)* ed. R. A. Brooks and P. Maes (Cambridge, MA: MIT Press) p 377-81

[14] Adami C 2002 Ab initio modeling of ecosystems with artificial life *Natural Resource Modeling* **15** 133-45

[15] Adami C 2003 Sequence complexity in darwinian evolution *Complexity* **8**(2) 49-56

[16] Orfia C and Adami C 2002 Evolution of genetic organization in digital organisms *Evolution as Computation; DIMACS Workshop (January 1999, Princeton)* ed. L. F. Landweber and E. Winfree (Berlin Heidelberg : Springer-Verlag Berlin Heidelberg) p 296-313

[17] Orfia C, Bryson D M, Wilke C 2009 A software platform for research in computational evolutionary biology *Artificial Life Models in Software, 2nd edition* ed. M. Komosinski and M. Adamatzky (London : Springer) p 3-35

[18] Pargellis A N 1996 The spontaneous generation of digital “Life” *Physica D* **91**(1-2) 3-35

[19] Pargellis A N 1996 The evolution of self-replicating computer organisms *Physica D* **98**(1) 111-27

[20] Pargellis A N 2003 Self-organizing genetic codes and the emergence of digital life *Complexity* **8** 69-78

[21] Kol’chugina E A 2009 Non-equilibrium programming *Izv. Vyssh. Uchebn. Zav. Povolzh. Region. Techn. Nauki* **3**(11) 25-31

[22] Rudenko A P 2013 Self-organization and synergetics *Complex Systems* **2**(7) 4-39

[23] Kol’chugina E A 2016 Self-organization of program systems based on artificial chemistry’s models *Neurocomputers* **2** 77-83

[24] Kol’chugina E A 2018 Spontaneous emergence of programs from “primordial soup” of functions in distributed computer systems *Automatic Control and Computer Sciences* **52**(1) 40-8

[25] Kasterin N P 1937 *Generalization of the basic equations of aerodynamics and electrodynamics: Report at a Special meeting at Acad. Sciences of the USSR 9 December 1936* (Moscow: Academy of Science of USSR Publishing Office)