Plain Talk about Systems Complicatedness

Jozef Kelemen

Institute of Computer Science, Silesian University
Opava, Czech Republic
and VSM College of Management, Bratislava, Slovakia
kel10um@axpsu.fpf.slu.cz

1 Introduction

This extremely informal contribution sketches an extremely informal proposal of classification of systems according their (up to now only intuitively) comprehended *complicatedness* by providing few examples from the fields of (theoretical) computer science, (computer) arts, and economics. The intuitive comprehension provides, however, at least in certain extent, an extension of theoretically well-founded and deeply studied classification of computing systems and their behavior (computation, algorithms) according different complexity measures into complexity classes of systems and behaviors with the same complexity as known in the traditional theoretical computer science. Almost all of our professional considerations are up to now traditional in the sense that almost all of us try to cover all of the appealing objects (languages, molecules, membrane structures), and phenomena (sentence generations, DNA mutations, cells functioning) into the traditional, of course in many situations very well-working, paradigm of the *Turing computability*, and of the spectrum of formal models performing these type of computation.

Our aim is to extend the typology of systems by some intuitive classes of systems with more or less similar level of *complicatedness* in order to include to the potential formalistic debate also systems inspired by some advances in artificial intelligence, artificial life, cognitive science and similar disciplines despite of this notion has - at least up to now - not a very clear status in the hardcore theories of computation.

First of all, it seems to by reasonable to specify at least intuitively what we will in this plain talk understood as a *system*. We will concentrate to systems which produce some symbolic behaviors or structures on the base of transforming another (sensed) symbols or symbol structures into the form of their output structures or their behavioral units. More or less *autonomous agents* as described e.g. in (Kelemen, 2006) are examples of such systems. The next sections will provide a couple of examples.

We note that from our point of view also collections of systems working together in order to generate a behavior (e.g. multi-agent systems, decentralized systems, societies of systems) may be considered as unique systems.

Our second duty is the - at least intuitive - a specification of the meaning of *complicatedness*. By complicatedness we will mean a structural property of a system which provides a base for producing the behavior of this system. If a behavior is given, then "What is the less complicated system which is able to perform this behavior?" seems to be one among the most appealing question related to systems complicatedness.

2 Simple Systems

Simple systems are, according our intuition, systems having a specific property - if we have several such systems, then behavior of the society composed from such systems will be in certain meaning only the simple result of their individual behavior.

More formally and more generally speaking from the point of view of their behavior are simple systems closed under set-algebraic operation like union, intersection, Kleene's * operation, and set complementation operation. From such a perspective, some of the formal grammars - understood as generators of behaviors in the form of sets of strings of symbols (languages) are simple systems: Having two context-free grammars, the behavior which results from the union of their individual behaviors (context-free languages generate by them) is again a behavior (a context-free language) which can be generated by another corresponding context-free grammar. Similar is the situation with other models, too: If we have, for instance, two finite automata, and we form the set-theoretic union (or we use some other suitable operation) of their behaviors (the union of two regular languages accepted by them, or some other operation over these two languages) we receive a regular language again, and we are able to construct a finite automaton which will accept the resulting regular language.

From the position of the theoretical computer science we may state, that simple systems are all theoretical models related to computation (automata, machines, grammars) the class of languages corresponding to which are closed under the traditional (above mentioned) set-theoretic (and also to the other traditional ones - concatenation, and reversal) operations usually studied in theoretical computer science. More details on these models and their closures under set-theoretic (and some other) operations are included into each course-book on theoretical computer science; let us mention e.g. (Hopcroft, Ullman, 1969, Chapter 9). The infinitely large classes of regular, context-free, context-sensitive, and recursively enumerable languages (sets) have this property with respect the operation of union, concatenation, intersection, and reversal; see (Hopcroft, Ullman, 1969, Theorem 9.1).

There are several possibilities of how to define different *complexity measures* which reflects in a very formal, theoretic level some of the characteristics of simple systems, and how to use the differences induced by these measures in order to stratify simple systems according these measures into different complexity classes, to relate these classes, to study the possibility of partial reduction of one to another, etc.; more details provides e.g. (Hromkovic, 1997).

Another possibilities of how to define different classes of languages using some biologically well-inspired models of computing devices can be found e.g. in (Paun, Rozenberg, Salomaa, 1998) for computational interpretations of some of important biochemical processes appearing between nucleic acid macromolecules, and in (Paun, 2002) for the case of computing motivated by biochemical and biophysical processes appearing in (organic) membrane systems like cells, for instance.

3 Systems with Emerging Behavior

The traditional and most widely used informal definition of *emergence* is formulated in (Holland, 1998, pp. 121-122): Emergence is "... a product of coupled, context-dependent interactions. Technically these interactions, and the resulting system, are nonlinear: The behavior of the overall system cannot be obtained by summing the behaviors of its constituent parts... However, we can reduce the behavior of the whole to the lawful behavior of its parts, if we take nonlinear interactions into account".

In connection with the phenomenon of emergence, another phenomenon appeared very interesting form the computational point of view - the notion of emergent computation. The premise of emergent computation is - according (Forrest, 1991, p. 1) - that interesting and useful computational systems can be constructed by exploiting interactions among primitive components, and further, that for some kinds of problems (e.g. modeling intelligent behavior) it may be the only feasible method. The formal study of such processes and the systems behind them is in the focus of the professional attention up to now, and might be interesting to reflect it not only in experiments in the field of artificial intelligence and artificial life, but also in the context of theories of formal symbolic behavior generators.

Systems with emerging behavior are in fact multi-agent systems because they are set up from a number of individually behaving component systems. Component systems have their own behaviors, and they have also some possibilities to communicate in some indirect ways, sharing the common "environment", for instance, say, e.g., rewriting symbols in a shared string. Good candidates for become to be *systems with emerging behavior* are grammar systems as presented in (Csuhaj-Varju et al., 1994), or *eco-grammar systems* (Csuhaj-Varju et al., 1997).

Consider now component systems to be simple systems from some complexity classes, and consider the behavior of the whole system composed from the component systems now. We may recognize two possibilities:

1. the systems will produce behavior from one of the complexity classes of the component systems, or

2. the system will produce a behavior from another complexity class. In the second case the behavior of the system is emergent, it emerges from the behaviors of the component systems, and the systems will be called system with emerging behavior.

The emergent behavior of such systems satisfies the *emergence test* formulated in (Roland et al., 1999) consisting in the following three basic testing steps:

a) *Design.* The designer designs the systems by describing *local* interactions between components in a language L1.
b) *Observation.* The observer describes *global* behaviors of the running system using a language L2.
c) *Surprise.* The language of design L1 and the language of observation L2 are distinct, and the causal link between the elementary interactions programmed in L1 and the observations observed in L2 are *non-obvious.*

A suitable example of systems with emerging behavior are variants of *grammar systems*, c.f. (Csuhaj-Varju et al., 1994), called *colonies* (Kelemen, Kelemenova, 1992). In the case of colonies finite sets of regular grammars cooperating as members of a grammar system are able to generate the members of all of the family of context-free languages; the relation of this phenomenon to the emergence is discussed in (Kelemen, 2004). Some remarks on a possibility how to attack the formal treatment of the problem of emergence from the positions of the traditional formal language theory and the theory including the theory of abstract families of languages into considerations, is presented in (Freund, Kelemen, Paun, 2003).

4 Hyper-Computing Systems

Hyper-computing systems are, very roughly speaking, systems, which go by their computing potentials in certain senses beyond the limits of traditional Turing-computation. Burgin and Klinger (2004) described the relevant opinions, and in the special issue of TCS Journal in which the just mentioned article is published and which is edited by them, collects a couple of other interesting opinions.

In (Stannett, 2004) the problem of hyper-computation is connected in an elegant way by the Turing machine and the Church-Turing thesis by making explicit the following three points:

1. Computation in a Turing machine is in fact a controlled manipulation of configurations, where each configuration encodes a finite amount of information as a state, a finite amount of information as memory, and a finite amount of information as program.
2. Turing machines control structure is constrained both by the current configuration of it, and by the requirement that only one program instruction is executed at a time.

Then the Church-Turing thesis expresses the conviction that any "cosmetic changes" in the architecture of the Turing machine have no principal influence to its the computational power (may be they have some influence the traditionally understood complexity requirements of performed computation, but what is not computable by a Turing machines remains not computable by other machines, too).

Stannett in the above mentioned article concludes with the statement, that there are only four obvious ways of modifications of the Turing machine: the temporal structure of computation, the information contents of memory, the information content of programs, and the information content of states. Then he provides the list of publications which attacked the problem from the mentioned obvious ways.

Let us to provide an example of such systems producing non recursive behavior which is not mentioned in (Stannett, 2004). In connection with another variant of grammar systems - with so called eco-grammar systems (Csuhaj-Varju et al., 1997) - some observations concerning the hyper-computing potentials of this model is provided in (Watjen, 2003). Roughly and informally speaking, Watjen in his just mentioned article proved that if into an eco-grammar system which uses teams of components for generating strings of symbols, a non-recursive function is included which defines the number of components in teams for each step of the derivation process, then such an eco-grammar system is able to generate non-recursive languages. Let us mention marginally, that the pure randomness comprehended as a function (in the above case prescribing teams to derivation steps) seems to be, at least intuitively, non-recursive (in the opposite case it is not random), and that perhaps in the real, non idealized situations the randomness play very crucial role in real physically embodied systems behaviors.

5 Creative Systems

To *create* is usually used for denoting the ability to cause something to come to existence, bring into being, originate something. Creativity is then usually considered as the act of creation, so as a mental and social process involving the generation of new concepts or new associations between the existing concepts, the ability to finding "new ways to look at things"; cf. (Minsky, 1986, p. 134). Two principal attributes are usually required with respect to creative combinations of concepts - the *originality* of the concept, and its *appropriateness.*

In order to decision about the appropriateness, in all of usual cases an anthropocentric test (in certain extent similar to the Turing test known in AI) is used - a test of the appropriateness in a given cultural context of a given human society. The culture of the particular human society dictates what is required and what is (at least marginally) acceptable. This is the important and inevitable outer anthropocentric determination of the inner individual creativity of each human mind or some artificial information processing systems, too.

During the history of the development of computers use, and during the experiments with computational models of mental processes in the field of artificial intelligence and cognitive science numerous systems have been developed which are based on some hypotheses concerning the human mental activities. It have been developed also a challenging area of agents, and multi-agent systems, the fields which are focused to the understanding, construction, and practical use of more and more autonomous computer-based systems - agents as presented e.g. in (d'Inverno, Luck, 2001) -, and societies in which agents may interact - multi-agent systems; see e.g. (Ferber, 1999).

The existence of (societies of) agents opens another, from some perspective - if we consider human societies as a kind of multi-agent systems, and the interactions of agents inside multi-agent systems as some level of the existence of culture in these systems - a more general position for testing the appropriateness of results of creativity.

Some artificially designed systems produce behaviors which, in the case that a human being is considered as the system of this type, are called creative. The author of this contribution does not know any theoretical, formal approach to the study of systems of this type. However, there are some successfully working systems of this type, and now we will mention examples of the two types of creative systems in certain details on the base of (Kelemen, 2009).

In the field of artificial intelligence, numerous systems have been developed which belong to the broad family of the so called *goal-driven systems*. The basic idea common for all of such systems consists in comparing some representation of an actual situation of a given problem with (representation of) a given desired situation of it. The comparison results in an ordered set of formally defined and represented *differences*. The differences are then, step by step, *reduced* using some formally defined *operators* in order to reduce the number of differences between the existing and the desired state of the problem. Operators and differences are connected with respect the ability of the given operator to reduce the related with it difference(s).

The principle was successfully applied e.g. in the famous *General Problem Solver* (GPS) developed by H. A. Simon, A. Newell, and C. J. Shaw during the end of fifties and beginning of sixties of the past century; for more details see e.g. (Ernst, Newell, 1969).

A crucial point in the GPS which makes it relevant for our discussion is that from simple concepts (operators) it constructs autonomously (without any human assistance) a sequence of operators, a more complicated concept, which represents the solution of the problem given at the beginning as an input to the system. This sequence, if GPS is successful in solving the given problem, transforms the starting situation describing the problem (the well-known tower-of-Hanoi or some similar problems), into the situation which represents its solution. In this sense GPS represents a creative system. However, its creativity is fundamentally based on the definitions of operators, differences, situation descriptions, table of operator-differences connections, etc. provided by its human users. So, the success of GPS in solving problems depends on the quality of these human-defined components.

Art is another field in which the agent paradigm works well, and creativity exhibits its potentials very clearly. Perhaps the most popular among the computer-based art-producing machines is the system called AARON. It began - according its author Harold Cohen (Cohen, 1995) - its existence some time in the mid-seventies of the past century; see e.g. (Cohen, 1973). The earliest versions of the system used some perceptual primitives only for producing (drawing) images. It has the ability to differentiate between figure and the ground, to differentiate between open forms and closed forms, and to differentiate between insideness and outsideness. Moreover, it has the capability to perform various simple manipulations on the structures it produced. Time-to-time, this more or less randomly executed manipulation on primitive line-structures resulted in figures having in certain sense figurative contents, like "human face" represented for instance by the expression as follows: human-face IS (INSIDEclosed-form (UPPER-POSITION(closed-form, closed-form) CENTRAL-POSITION(open form), DOWN(closed-form)))

This symbolic representation may look in corresponding free-hand drawing representation like in Fig. 1 (a). A more complicated picture - an original drawing made by AARON Fig. 1 (b) - of the "human body" may be represented in similar symbolic way. Of course, the human face can be then sophisticated also to more and more complicated pictures, e.g. as the one in Fig. 1 (c), produced with a more sophisticated version of AARON. Realize that what the human-face expresses, is the definition of the human face for AARON.

We decided, associating the human-face with its formal expression that the randomly scrawled lines remind the line-drawing of a human face in our minds. Now we are able to instruct AARON to draw a new human face. But the result will be not the same, as in the previous case, the same will be only the structure expressed by the formal definition of what we consider to be a human face. In such a way we can produce more and more concepts and instruct AARON to draw more and more complicated drawings, each with certain degree of "freedom" of AARON's drawing. For generating Fig. 2 the requirement was a picture of a botanical garden with seven human beings (five distressed women and two men) inside it.

AARON is - from the point of view of knowledge processing technologies - in fact a *rule-based knowledge-system* in the usual meaning used e.g. in (Stefik, 1995). Without going into the technical details of the construction (programming) of it, we may conclude, in concordance with conclusions made in (Cohen, 1995), that AARON constitutes an existence proof of the power of machines to do some of the things traditionally connected with human thought and his creativity. "If what AARON is making is not art, what is it exactly, and in what ways, other than its origin, does it differ from the "real thing"? If it is not thinking - and let us to be more explicit: creative thinking -, what exactly is it doing?" Cohen asks in last strokes of (Cohen, 1995).

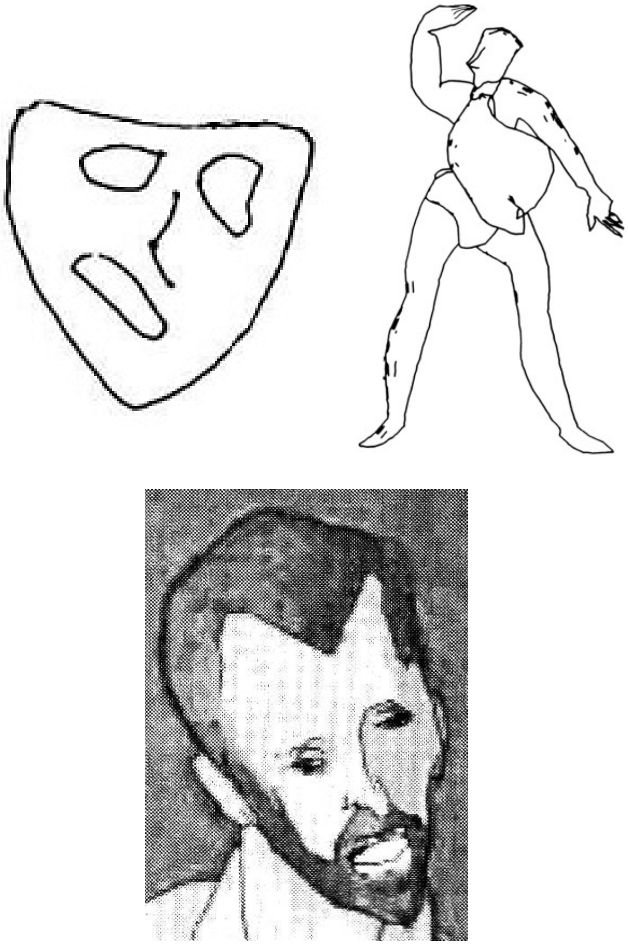

Fig. 1. (Parts of) pictures. The first drown in AARON style; the second produce by AARON, and taken from (McCorduck, 1990, p. 105), the third one produced by AARON, and taken from (Kurzweil, 1999, p. 167).

6 Man-Machine Systems

The emergent nature of some creative phenomena appearing in complicated systems - like the *man-machine societies* are - we may also test using the so called test of emergence proposed in (Ronald et al., 1999). We provide here an example of such a system from the field of the interactive art discussed in more details and with more examples in (Horakova, Kelemen, 2010).

The first interactive piece we mention is based on some ideas from the experiments executed in the field of artificial life, and is created by Christa Sommerer and Laurent Mignonneau. The project named A-Volve has been presented first in

Fig. 2. Five distressed women and two men (one hiding, upper right). Taken from (McCorduck, 1991, p. 135)

1994, and developed it during few next years. The audience of the A-Volve session has possibilities to design artificially living creatures and to provide for them different modes of their behavior (aggression, snuggles, curiosity, caution, friendship, etc.). In the virtual world behind of the screen a society of such creatures mutually interact, and behaves according the habit of the members, and have been influented also if the visitors touched the screen; see Fig. 2. More details on the A-Volve can be found in (Whilelaw, 2004).

Let us now to analyze the project from the position of the *test of emergence*: *Design*. The language L1 is the language in which the system is implemented, so, a purely technical computer programming device used with a specific intention to provide a usable, user-friendly software product for well-specified purposes. *Observer*. The previous language substantially differs, of course, form the language L2 in which the audience - the observers - interacts with the systems A-Volve. This language contains tools for defining new creatures, and contains also kinds of gestures for interacting with the creatures through touching the screen. *Surprise*. The surprise follows then from the observation of the new created creatures as members of he existing community behind the screen, and from the direct interaction with the creatures through touching the screen. This is the reason why we propose that the real artistic creativity emerges in the case of the systems A-Volve from interaction of human being with the machine.FFF

Fig. 3. Interaction with the A-Volve. Photo Ch. Sommerer and L. Mignonneau, 1994.

7 Conclusion - What About to Study Reflexive Systems?

The emergence of creativity mentioned in the previous two chapters can be discussed also in the broader context of a special kind of systems derived from study of some specificity of economic systems and their behavior, and called by George Soros as *reflexive systems* (Soros, 1994). For explaining what kind of systems we have in mind we borrow an example from (Soros, 1994, p. 42).

Let us suppose that active agents belonging to a given system work according two functions. The first function, say f, is defined on the situations appearing on the system. We will call it the cognitive function, because the participants - the agents - effort to understand the system depends on perception of the systems. More formally (but not in a pure formalistic manner) we have $y = f(x)$. The second function defines the participants' participation on the changes of the system. This participation is supposed to be rationalistic, so is based on the understanding of the system and changes the situation inside of the system. We express this dependence of the behavior on the understanding by the function $x = g(y)$ and call this function, according Soros proposal as the participating function. So, as the result we have: $y = f(x)$, and $x = g(y)$, what gives $y = f(g(y))$, and $x = g(f(x))$. This is, roughly speaking, and not in a very well formulated way, the fundamental property of the reflexive systems.

An example: Suppose that a globally influential group of economic experts start to speak and write on a good functioning bank in a highly critical tone and start to hesitate with respect of its economic future. What will happen? This bank will quickly go to problems and in the more wrong case to bankruptcy. More generally speaking, in certain situations, in certain types of systems the observer's observations change the observed object. So, "objective" observations are not possible in this type of systems. This type of systems are reflexive.

The development of a methodology for systematic study of such systems might perhaps start form experimentations with artificially created societies of simple economic agents as presented in a very impressive way in (Epstein, Axtell, 1996). The experiments prove the way how some simple economic laws, e.g. the famous *Pareto law*, emerges - laws originally formulated on the base of observation of the behavior oh human economic societies - in very simple societies with some basics of economic behavior of their members, and how many other interesting economic and social situations and processes can be observed and experimentally tested in the specific test-bed of the multi-agent system the author used in experiments collected and analyzed in the above mentioned book .

What was presented concerning the creativity is another illustration of the behavior of reflexive systems. Thanks to the reflexivity of the human society the new creations are first surprising, but then become to be accepted, so, they become appropriate for the society in which they have first the attribute of innovations (technical innovations, artistic innovations, fashion innovations, etc.).

However, the study of reflexive systems are up to now and according the author best knowledge, out of the scope of interest of theoretical computer scientists, despite of the fact, that reflexivity is perhaps the property of majority of the complicated information processing systems (like human brains, computer networks, man-machine systems and societies, etc.). What about to look for the much more suitable formal frameworks in order to make first steps towards formal understanding of this type of systems from computationalist positions similarly as we do that with the simplified models of real computing engines in the field of traditional computer science?

References

1. Burgin, M., Klinger, A.: Three aspects of super-recursive algorithms and hypercomputation or finding black swans. *Theoretical Computer Science* 317 (2004) 1-11
2. Cohen, H.: Parallel to perception - some notes on the problem of machine-generated art. *Computer Studies* 4, No. 3/4 (1973)
3. Cohen, H.: The further exploits of AARON, painter. *SEHR, Stanford Electronic Humanities Review* 4. No. 2 (1995)
4. Csuhaj-Varju, E., Dassow, J., Kelemen, J., Paun, Gh.: *Grammar Systems - A Grammatical Approach to Distribution and Cooperation.* Gordon and Breach, Yverdon, 1994

5. Csuhaj-Varju, E., Kelemen, J., Kelemenova, A., Paun, Gh.: Eco-grammar systems - a grammatical framework for studying life-like interactions. *Artificial Life* 3 (1997) 1-28
6. D'Inverno, M., Luck, M.: *Understanding Agent Systems*. Springer, Berlin, 2001
7. Epstein, J. M., Axtell, R.: *Growing Artificial Societies*. The MIT Press, Cambridge, Mass., 1996
8. Ernst, G.; Newell, A.: *GPS - A Case Study in Generality and Problem Solving*. Academic Press, New York, 1969
9. Ferber, J.: *Multi-Agent Systems - An Introduction to Distributed Artificial Intelligence*. Addison Wesley, New York, 1999
10. Freund, R., Kelemen, J., Paun, G.: A note on emergence in multi-agent string processing systems. *Computing and Informatics* 22 (2003) 623-637
11. Forrest, S. (Ed.): *Emergent Computation*. The MIT Press, Cambridge, Mass., 1991
12. Holland, J.: Emergence. Addison-Wesley, Reading, Mass., 1998
13. Hopcroft, J. E., Ullman, J. D.: *Formal Languages and their Relation to Automata*. Addison-Wesley, Reading, Mass., 1969
14. Horakova, J., Kelemen, J.: On the emergence of creativity in man-machine systems. In: *Proc. 14th IEEE International Conference on Intelligent Engineering Systems, INES 2010*, Las Palmas, Gran Canaria, May 2010 (accepted)
15. Hromkovic, J.: *Communication Complexity and Parallel Computing*. Springer, Berlin, 1997
16. Kelemen, J.: Miracles, colonies, and emergence. In: *Formal Languages and Applications* (C. Martin-Vide et al., eds.). Springer, Berlin, 2004, pp. 323-334
17. Kelemen, J.: Agents from functional-computational perspective. *Acta Polytechnica Hungarica* 3 (2006) 37-54
18. Kelemen, J.: A note on agents, societies, and creativity. In: *Proc. 7th IEEE International Conference on Computational Cybernetics, ICCC 2009* (A. Szakal, Ed.). IEEE CD Publication (IEEE Catalog Number CFP09575-CDR) 2009, pp. 81-84.
19. Kelemen, J., Kelemenova , A.: A grammar-theoretic treatment of multiagent systems. *Cybernetics and Systems* 23 (1992) 621-633
20. Kurzweil, R.: *The Age of Spiritual Machines*. Penguin Books, New York, 1999
21. McCorduck, P.: *Aaron's Code - Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. Freeman and Co., New York, 1991
22. Minsky. M.: *The Society of Mind*. Simon & Schuster, New York, 1986
23. Paun, G.: *Membrane Computing - An Introduction*. Springer, Berlin, 2002
24. Paun, G., Rozenberg, G., Salomaa, A.: *DNA Computing*. Springer, Berlin, 1998
25. Roland, E. M. A., Sipper, M., Capcarrere, M. S.: Design, observation, surprise! A test of emergence. *Artificial Life* 5 (1999) 225-239
26. Soros, G.: *The Alchemy of Finance*. John Wiley and Sons, New York, 1994
27. Stannett, M.: Hypercomputational models. In: *Alan Turing - Life and Legacy of a Great Thinker* (Teuscher, Ch., Ed.). Springer, 2004, pp. 135-157
28. Stefik, M.: *Introduction to Knowledge Systems*. Morgan Kaufmann, San Francisco, Cal., 1995
29. Watjen, D.: Function-dependent teams in eco-grammar systems. *Theoretical Computer Science* 306 (2003) 39-53
30. Whitelaw, M.: *Metacreation - Art and Artificial Life*. The MIT Press, Cambridge, Mass., 2004