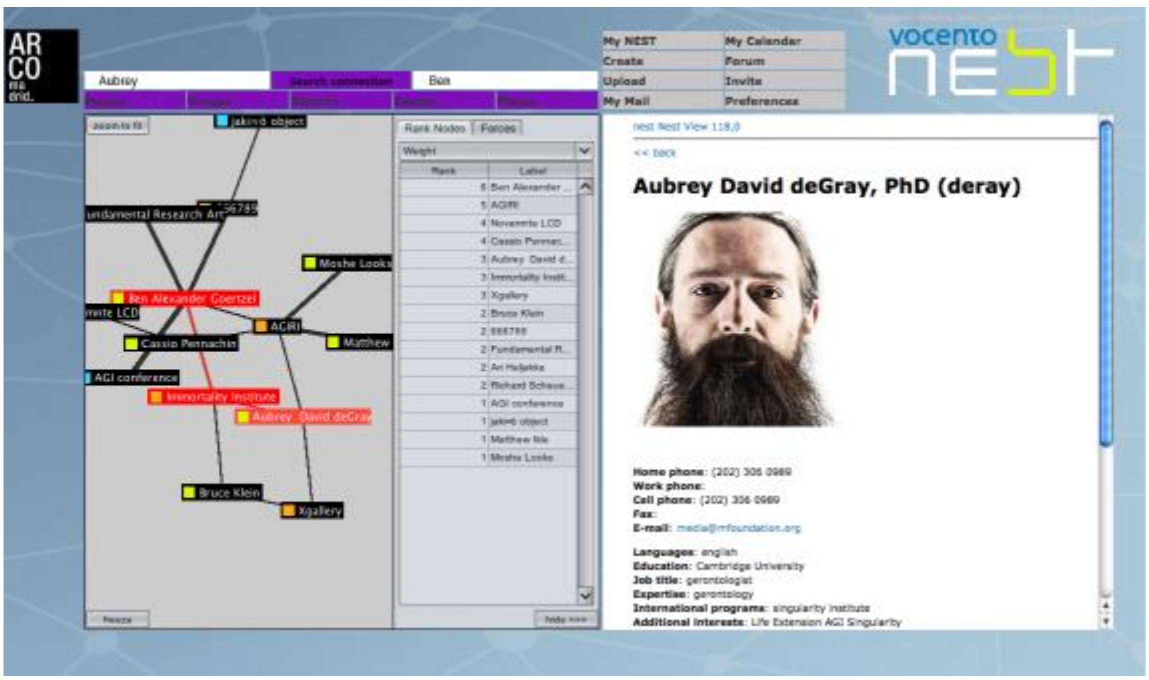
TOWARD COMBINATORIAL COMPUTER SCIENCE

Robert B. Lisek

The primary goal of this article is to focus on the dynamics and processes of creation and transformation of organisms, structures and social organizations. A development of communication networks has caused research on systems that can function as 'neural system' for social organism. However, these earlier models remain on the level of metaphor. Our approach is more formal and uses mathematical tools.



NEST- ARCO Art Fair

1. Acceleration and 'global mind'

Connectivity, bandwidth and the size of resources grow. The increasing flow of information will create a true neural system of society in the future. Anarchy in the Internet proves its potential. The first step is to create a new Internet that learns new associations between data or concepts. Until now, a creation of links is done by hands, by authors of the documents, who determine what other documents are relevant to their text. This process is very inefficient. Search engines are still very archaic and could only partially alleviate this problem. We need 'thinking' Internet, which reinforces connections between documents that are often used together and weakened between rarely used documents. 'Thinking' in means of using a dynamic network of connections for solving problems or responding to questions.

Based on AGI research and Internet extensions, progress will accelerate when a higher than human intelligence emerges. Large fast computer networks connected with human users can "wake up" as superhuman intelligent entities. We can create superhuman artificial intelligences in the next thirty years. Shortly after that, the era of human will end.

2. Portable knowledge development environment

If we want to create a real "neural system" of modern societies and transform the present society into the global mind, we need a good framework - portable knowledge development environment. We are looking for fundamental insight concerning human mind and its relation to the world, not only on a metaphorical level, but also as a strict formal framework. The main features of the framework:

Creativity: rapid construction, comparison and transformation of different structures with local rules and global behaviour

Flexibility: ability to support multiple tasks and representations

Soundness: Rooted in a solid mathematical basis. We use labelled, weighted, partially ordered sets (reflexive, transitive and antisymmetric) - propose them as base for a canonical framework for representation of knowledge and semantic information.

Self-reference: the structure must be able to relate to itself.

Inference and Data Mining: inferring new information from earlier data. Hyperset transformation and data mining methods (e.g. sequence analysis, network analysis, latent semantic analysis, and other statistical tools) to uncover hidden patterns and structural relations.

Diagram GUI: empirical tests as the way to find the appropriate mechanisms, guide this process by visual insight and graph manipulation.

The Development Framework would enable to build new structures, compare them in detail and then observe them in action. My general intention is to build systems, which have local rules, and then to observe their global behaviour. The environment would enable comparison of how changes in local systems will affect their global behaviour. I'm interested in gaining further insight into how to clarify the notion of transition from the local to the global, and how various elements come together and apart in the evolutionary dance.

As a result, we are interested in creating a software, which will be able to handle a great many different problems quicker and more effectively than man. Here, I’m presenting a conceptual framework for the development of a ‚class‘ of intelligent systems. I emphasise on a ‚class‘, rather than a single system. A system is intelligent not because of what it can do, but because of what it can learn to do. An intelligent system adapts itself to its environment and works with incomplete knowledge and resources. More precisely, the system has a set time and is always ready for new tasks, processing them in real time and learning from its own experience.

3. Retracts

We study systems in relation to their abstract properties, such as structure and organization. In our approach, it is essential to use the theory of ordered sets and fixed points of morphisms. In [1] we present state of knowledge in this field and the results based on new results related to retracts. We draw attention to the methods of representation, manipulation and measurement of partially ordered sets (reflexive, transitive, antisymetric relations). Retracts and fixed points are central for the recursion and computability. Fixed points are important because they accurately characterize solutions of recursive definitions. It is convenient to describe the function using recursion, especially in programming languages. The problem is whether the functions are well defined. The idea is a method of successive approximations. The process of approximation provides fixed points, which are solutions of recursive equations.

Framework should be based on hyperposets - weighted labelled partially ordered sets. They allow the use of n-ary ordered relations represented as hyperedges – such edges that connect n nodes simultaneously. Hyperposets allow the building of hierarchies, ontologies and semantic networks. Posets are a subclass hyperposets. In hyperposets you can connect components in many different ways. Self-reference is possible, you can write an equation f = f (f) with mathematical consistency.

A function f : P -> P is order-preserving if, for all p, q ∈ P, p =< q implies f(p) =< f(q). A subset Q ⊆ P is a retract if there exists an order-preserving map f : P -> Q such that f(q) = q for all q ∈ Q; the map f is a retraction. An element p ∈ P is a fixed point of a function f : P -> P if p = f(p). The poset P has the fixed point property if every order-preserving map on P has a fixed point. A poset has the fixed point property if and only if every retract does .

4. Creation and transformation of organisms, structures and social groups.

4.1 We need formulas that explain how organisms mutate and propagate. Not only as in cellular automata or in genetic programming, but general and basic equations and new formalisms that allow self-reference, which is crucial for describing conscious systems. This means systems, that are able to "think themselves", to understand its limitations, create its own rules and modify their goals and methods. We also need simple equations to measure transformation and spreading. The key thing is to understand the evolution, the way in which new organisms are created and transformed. New structures are created by mutations and functions. For example, point mutations: remove an element (node) or insert a new element (or group of elements).

The most simple way of defining metrics in structure space is the number of point mutations required to transition from one organism to another, from one social organization to another. This is the general idea, but sometimes we need a more subtle way to define the distance of mutations, as - log2 probability that one organism is created from another by single mutation or retraction.

4.2 Self-production.

We refer to older tradition - cybernetics and systems theory. Although some old cybernetic models are based on central, hierarchical structure, new methods emphasize self-organization, autonomy, decentralization and interaction of many factors. Several models have been developed that can be used both for organisms and social systems: Miller (1978) - living systems theory, Matura and Varela (1980, 1992) - theory of autopoiesis, Powers (1973, 1989) - perceptual control theory and Turchin (1977) - metasystem transition theory. In this approach, both social and biological organisms can be seen as special cases of more general category - "life" or "autopoietic system". Autopoietic (greek word for "self-production") system consists of a network of processes that recursively produce their own components and separate itself from environment. For example: living cells can be characterized as a complex network of chemical processes, which constantly mutate and produce molecules necessary for the functioning of cells. Reproduction is often seen as a function for defining "life". However copying without autopoiesis, which can be described more accurately as the replication, does not mean life: some crystals and viruses can replicate without life. We can talk about autopoesis of societies where physical components of society can be described as its human members and their achievements (buildings, cars, roads, computers, books, etc.). Each of these components is produced by a combination of other elements from system.

4.3 The problem of the border. In the original definition of autopoiesis a fact is added, that the autopoietic system should produce its own borders or spatial or topological separation of system from the environment. In contrast to biological organisms, there is no clear boundary in social systems. For example, the state can produce most of its basic elements internally but it also needs to import some components (people, artefacts) and expertise from outside. This means that borders in social system are fuzzy.

4.4 Dynamics and spreading. Morphisms Processor

The most important is the transformation of structures. Morphisms are transformations of ordered sets. A topological structure of the network (weight of links and priority distribution) is changed through the use of morphism. We define a new concept: the energy of morphism. The energy of the morphism of an order set is a scale-invariant of morphism: function from morphism to rational numbers. The energy of set O is the infinitum of energy of the set of all the morphisms of the order set of type O. The type and canonical form of an order set is based on a number of irreducible elements. Intuitively, the connection between the complexity of the morphism and its energy is simple: the more complicated morphism, the higher the energy.

These kinds of studies have many applications in creation of Artificial General Intelligence and finding solutions for problems associated with the emergence of complex, combinatorial objects and large data sets.

5. Automatic Programming

5.1 Automatic programming is a strong field from the very beginnings of the AI. PSI project at Stanford [2], which constructed a LISP, seems to be an important point of classical automatic programming. The automatic programming is "AI complete" in the sense that it requires general knowledge to understand what program/user wants to achieve (without determining details in a programming language). Artificial General Intelligence (AGI) is connected with 'unlimited' ability to learn. Framework must be experimental, cognitive architecture, which learns by modifying and extending itself, including developing an ability to learn (better learning). Let's assume that learning is treated as programming: learning is designing a new algorithm. Learning as an ability to write programs. Learning as a general knowledge of the world is to build a model that is able to predict the consequences of events or activities. In this perspective, general program learner is a program that writes programs. It is a program that has to invent both the algorithms and representations. Such programs can easily operate on each other and produce other programs, much in the way that molecules can react and produce other molecules. For human programmers, the programs become more complex, it makes sense to develop new languages in order. This regularity also holds for AI systems. Unlimited learning system will be a system that invents new programming languages.

5.3 Architecture and machines based on partially ordered sets

LISP introduced abstract syntax tree (AST) as a base type of data and automatic memory management method. It was also probably the first functional programming language. PLANER, PROLOG and a variety of systems have introduced automated theorem proving, now widely used in most modern languages. The objects oriented semantics of many modern languages, derives from the architecture of systems from the 70s'. Currently, we need to create a new class of languages in which a base type of data will be partially ordered set. Ordered sets are richer structures than chains and trees used in all languages and computing machines that we know. At the same time posets are structures that we can strictly control. Therefore, building a new architecture and computers based on posets would be an important step towards the implementation of AGI and global mind idea.

References and Notes:

[1] Robert B. Lisek, Retracts And Fixed Points In Theory Of Ordered Sets. Towards Combinatorial Computer Science, CCS USA, 2007

[2] Cordell C. Green, Richard J. Waldinger, David R. Barstow, Robert A. Elschlager, Douglas B. Lenat, Brian P. McCune, David E. Shaw, and Louis I. Steinberg, Progress report on program-understanding systems., Stanford University, Stanford, CA, 1974

[3] Humberto R. Maturana, Francisko J. Varela, Autopoiesis and cognition: the realization of the living, Kluwer, 1980

[4] David Vernon, Giorgio Metta, and Giulio Sandini, A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents, IEEE Transactions on Evolutionary Computation, Special Issue on Autonomous Mental Development, 11(2), 2007