Adaptation, Bio-Inspiration, Complexity: The Terra Nova of Computing Science

M. Moshe Sipper

"It is fair to say that, in general, no problems have been exhausted; instead, men have been exhausted by the problems."

Santiago Ramón y Cajal,
"Advice for a Young Investigator."

"The essence of science: ask an impertinent question, and you are on the way to a pertinent answer."

Jacob Bronowski,
"The Ascent of Man."

During the past few years a new wind has been sweeping through the computing terrain, slowly changing our fundamental view of computers. We obviously want them to be faster, better, more efficient – and proficient – in their tasks. But that's just part of the story. The other part, and in my mind the more exciting one, is that we've come to demand of computers to stop being so stiff.

Computers are rigid, unbending, unyielding, inflexible, and quite unwieldy. While they have improved our lives in many a way, there is still much to be desired. It all boils down to one thing: at their most fundamental computers lack the ability to *adapt*.

Adaptation refers to a system's ability to undergo modifications according to changing circumstances, thus ensuring its continued functionality. We often speak of an environment, and of the system's adjustment to changing environmental conditions. The issue of adaptation in computing systems has moved into center stage over the past few years.

The archetypal examples of adaptive systems are not among Man's creations – but among nature's. Natural organisms show a striking capacity to adapt to changing circumstances, a fact which has not escaped the eyes of computing scientists and engineers. The influence of the biological sciences in computing is on the rise, slowly but surely inching its way toward the mainstream. There are many examples today of systems inspired by biology, known as *bio-inspired systems*.

These adaptive, bio-inspired systems are complex, which refers to more than their simply being complicated objects, difficult to build and comprehend. As written by Peter Coveney and Roger Highfield in

Frontiers of Complexity: "Within science, complexity is a watchword for a new way of thinking about the *collective* behavior of many basic but interacting units, be they atoms, molecules, neurons, or bits within a computer. To be more precise, our definition is that *complexity is the study of the behavior of macroscopic collections of such units that are endowed with the potential to evolve in time.* Their interactions lead to coherent collective phenomena, so-called emergent properties that can be described only at higher levels than those of the individual units. In this sense, the whole is more than the sum of its components...". Natural organisms are complex adaptive systems, and our artifacts are now beginning to follow in their footsteps.

The research I carry out is within the Terra Nova of computing science. Below, I briefly describe several fields of study in which I'm interested:

Cellular Computing

In recent years we have been witness to a growing number of researchers who are interested in novel computational systems based on principles that are entirely different than those of classical computers. Though emerging from disparate domains, their work shares a common computational philosophy, which I have dubbed *cellular computing.* Basically, cellular computing is a vastly parallel, highly local computational paradigm, with simple cells as the basic units of computation. It aims at providing new means for doing computation in a more efficient manner than other approaches (in terms of speed, cost, power dissipation, information storage, quality of solutions), while potentially addressing much larger problem instances than was possible before – at least for some application domains.

Evolutionary Computation

The idea of applying the biological principle of natural evolution to artificial systems, introduced more than four decades ago, has seen impressive growth in the past few years. Usually grouped under the term *evolutionary algorithms* or *evolutionary computation*, we find the domains of genetic algorithms, evolution strategies, evolutionary programming, and genetic programming. Evolutionary algorithms are common nowadays, having been successfully applied to numerous problems from different domains, including optimization, automatic programming, circuit design, machine learning, economics, immune systems, ecology, and population genetics, to mention but a few.

Fuzzy Systems

Computers are crisp creatures whereas we humans are fuzzy. The question that arises is: can we narrow this gap? Of course, there are two possible ways to do so. The first is by forcing humans to behave more crisply – to be precise and unambiguous; this is exactly the stance that computer programmers must assume: since they converse in the computer's tongue, they must be very "mechanic", avoiding the use of imprecise concepts.

There is another way to narrow the human-computer gap, though, which is much less "painful" to us: having the computer behave in a fuzzier manner. This field goes by the name of *fuzzy logic*. Fuzzy logic allows computing machines to go from "if the room temperature rises above 27 degrees centigrade then increase motor output by 34%" to "if the room is hot then substantially increase motor speed." Fuzziness, though, is more than that: it also refers to a "messier" way of computing, more akin to how things work in nature.

Learning Systems

The ability to learn throughout one's lifetime via interactions with the environment is exhibited by most higher forms of living organisms, and is most pronounced in human beings. This is yet another mode of adaptation, which has served as an inspiration for the development of artificial neural networks. Such networks are able to learn difficult problems from examples, a radically different approach from that of directly programming the desired behavior.

Configurable Circuits

When one sets about to implement a certain computational task then obtaining the highest performance (speed) is unarguably achieved by constructing a specialized machine, that is, hardware; however, the price per application as well as the lead time (from design to actual operation) are both quite prohibitive. Except for a small number of specialized niches, the computing industry has, by and large, converged onto the so-called general-purpose architecture, trading off the best possible performance in favor of a much lower cost per application and shorter delivery time. The gap between these two paradigms has been narrowing over the past few years with the coming of age of configurable computing: configurable circuits can be *programmed* like a general-purpose processor, but at a much lower, *hardware* level.

Artificial Self-Replication

In the late 1940s the eminent mathematician and physicist John von Neumann had become interested in the question of whether a machine can self-replicate, that is, produce copies of itself. The study of artificial self-replicating structures or machines has been taking place now for almost half a century. Much of this work is motivated by the desire to understand the fundamental information-processing principles and algorithms involved in self-replication, independent of their physical realization. An understanding of these principles could prove useful in a number of ways. It may advance our knowledge of biological mechanisms of replication by clarifying the conditions that any self-replicating system must satisfy and by providing alternative explanations for empirically observed phenomena. The fabrication of artificial self-replicating machines can also have diverse applications, ranging from nanotechnology to space exploration.

Self-Repairing (Healing) Machines

One possible path to attaining more resilient computers is by furnishing them with healing, or self-repair capabilities. There are (at least) two ways in which this can be accomplished. One is based on the immune system of living beings, which is capable of learning, recognizing, and above all eliminating foreign bodies that continuously invade the organism. Moreover, when viewed from the engineering standpoint, it is most interesting that immunity is maintained when faced with a dynamically changing environment. In recent years a number of groups have been investigating the possibility of building artificial immune systems for computers. Another artificial system that heals is the embryonic circuit of the Swiss-based Embryonics project, which is based on the multicellular structure of complex organisms, and on the way such organisms come to be – through the developmental process known as ontogeny.

Adaptive Robotics

Adaptive robots are machines capable of autonomously adjusting their behavior to a changing, partially unpredictable and partially unknown environment, as such representing a promising alternative approach to classic robotics. In this field researchers build actual robots and apply evolutionary- and neural-computation techniques to investigate issues such as mobility, navigation, and collective behavior.

Molecular- and Nano-Systems

Computing devices keep getting smaller and smaller: there are millions of tiny transistors in modern-day chips. Currently we just keep on stretching the existing technology, known as VLSI (Very Large-Scale Integrated Circuits). At some point though (probably within ten to fifteen years) this technology will have to be supplanted by an entirely new one. Molecular biology, giving rise to molecular computing, and atomic physics, giving rise to nanosystems, are two highly interesting recent developments that will probably gain prominence in the future.

Hybrid Systems

This involves the combination of two or more of the above domains, such as evolution of cellular systems, neuro-fuzzy computing, and bio-inspired hardware.

Complexity, Emergence

What makes a system "complex"? How does one construct complex systems? What exactly is emergent behavior? These are but a sample of fundamental questions of interest in these domains.

Selected Publications

[1] M. Sipper, Evolution of Parallel Cellular Machines: The Cellular Programming Approach, Springer-Verlag, Heidelberg, 1997.

[2] M. Sipper, "The emergence of cellular computing," IEEE Computer, vol. 32, no. 7, pp. 18-26, July 1999.

[3] C.-A. Peña-Reyes and M. Sipper, "Evolutionary computation in medicine: An overview," Artificial Intelligence in Medicine, May 2000.

[4] C.-A. Peña-Reyes and M. Sipper, "A fuzzy-genetic approach to breast cancer diagnosis," Artificial Intelligence in Medicine, vol. 17, no. 2, pp. 131-155, October 1999.

[5] M. Sipper and E. Sanchez, "Configurable chips meld software and hardware," IEEE Computer, January 2000.

[6] E. Sanchez, M. Sipper, J.-O. Haenni, J.-L. Beuchat, A. Stauffer, and A. Pérez-Uribe, "Static and dynamic configurable systems," IEEE Transaction on Computers, vol. 48, no. 6, pp. 556-564, June 1999.

[7] M. Sipper, "Fifty years of research on self-replication: An overview," Artificial Life, vol. 4, no. 3, pp. 237-257, Summer 1998.

[8] J.-Y. Perrier, M. Sipper, and J. Zahnd, "Toward a viable, self-reproducing universal computer," Physica D, vol. 97, pp. 335-352, 1996.

[9] D. Mange, M. Sipper, A. Stauffer, and G. Tempesti, "Towards robust integrated circuits: The embryonics approach," Proceedings of the IEEE, April 2000.

[10] D. Mange, M. Sipper, and P. Marchal, "Embryonic electronics," BioSystems, vol. 51, no. 3, pp. 145-152, September 1999.

[11] E. M. A. Ronald, M. Sipper, and M. S. Capcarrère, "Design, observation, surprise! A test of emergence," Artificial Life, vol. 5, no. 3, pp. 225-239, Summer 1999.

[12] M. Sipper, "If the milieu is reasonable: Lessons from nature on creating life," Journal of Transfigural Mathematics, vol. 3, no. 1, pp. 7-22, 1997.