Ubiquity, an ACM publication
October, 2012

Ubiquity Symposium

Evolutionary Computation and the Processes of Life

On The Role Of Evolutionary Models In Computing

by Max H. Garzon

Editor's Introduction

In this article in the ACM Ubiquity symposium on evolutionary computation Max H. Garzon presents reflections on the connections between evolutionary computation, natural computation, and current definitions of computer science. The primary aim and result is a more coherent, comprehensive and modern definition of computer science.

Editor

Ubiquity, an ACM publication
October, 2012

Ubiquity Symposium

Evolutionary Computation and the Processes of Life

On The Role Of Evolutionary Models In Computing

by Max H. Garzon

The definition of computer science requires rethinking so as to be acceptable to scientists in other fields by comparison to theirs. In this article we proceed in three stages to address this overarching problem of what is meant by computer science. First, we characterize the type of phenomena that goes by the name of computation today. This is necessary for our aim because, at least historically, establishing a science has required the determination of an aspect and scope of natural phenomena it is interested in, usually at the exclusion of others. Second, we provide a general characterization of how computer science has addressed these issues. A major issue appears to be that current definitions are so restrictive as to exclude a lot of what one would intuitively deem to be "computing." Finally, we discuss how an integration of important features from evolutionary and natural computing—such as nonsymbolic memory and complexification components—might provide a more distinct, sharper and encompassing definition of computer science. The new focus may reflect more closely the great many flavors of what is considered computation today, stimulate a tighter integration between them, and hopefully drive the science to a deeper definition, more embedded in natural phenomena and so more interactive with other sciences, yet with a more characteristic scope by comparison to them.

What Phenomena Should Computer Science Account for Today?

The opening statement for this symposium and the "What is Computation" symposium [1] provide a summary of the status quo in computer science. We will take a larger view, necessarily more abstract. Unlike the emphasis of previous generations in science and engineering, the fundamental societal concern of our time no longer seems to be matter-energy, machinery or electronic hardware, or even biological function or bioware. The products of scientific and technological advances that brought about the digital age have already penetrated homes and have become an integral part of human lives. The impact was all but unpredictable less than two decades ago. As a consequence, a new world is emerging before our eyes. Distances have been eliminated by new entities such as the Internet, the Web, smart phones, and instant communication around the globe. The computer age marks a significant qualitative change in the old subject matter of “calculating machines” being run at human pace that inspired early computing devices centuries ago and precise definitions of computing nearly a century ago. This new world is sometimes referred to as cyberspace. It makes one wonder, what exactly has our subject matter become and what does it have in store for us in the near future?

A quick run through history may give some insight. At the end of the 19th century attention was focused on electromagnetic forces. The first part of the 20th century saw the emergence of energy, its understanding, and production. The later part of the 20th century saw the emergence of digital devices, their applications in "artificial intelligence" products such as chess player DeepBlue and “Jeopardy” contestant Watson, both capable of defeating human players, and their implications in terms of the human role in the universe. Each of them, energy or electronics, gave impetus to a branch of science that made possible its understanding. Analogously, the single fundamental commodity of our time in cyberspace appears to be *information*. Information is but universally transacted and accepted by society at large, despite the fact that matter and energy still remain major but secondary factors controllable through information processing.

There are various ways to approach a rigorous study of information. One is to attempt to develop some sort of theory of information, akin to what Euclidean geometry does for basic geometric intuitions concerning the phenomenon of areas and volumes, so important to tax assessors in ancient Egypt, using the kind of fictions (lines, areas, and volumes) encountered now in high school. Much later, Claude Shannon developed a theory in the 1940s that has become known as “information theory.” However, it has been enormously useful more as a theory of data communication than a theory of information. Bateson's definition of information as "the difference that makes a difference" assumes again an observer to whom the difference is made, in other words a kind of information processor in itself [2]. The definitions are somewhat circular and not so scientifically sound. More recent attempts at developing an appropriate theory of information expose deeper difficulties. Chapter 1 in Devlin’s *Logic and Information*, provides a sobering assessment of the magnitude of the challenge via the metaphor that we would understand today as much about the answer to “what is information?" as an Iron Age man understood in his time about the answer to "What is iron?" [3]. The best answer that we know requires, at least, knowledge of the atomic structure of matter. Other authors have further explored similar arguments with similar or more radical conclusions, ranging from a general theory of information [4, 5, 6], to sheer impossibility of a unified theory of information [7]. Therefore, the definition of the subject matter of computer science as information and its creation and transformation would seem to lead to a deep dead end for the foreseeable future. And yet, we can sense the phenomenon of "information" lying there in the world, staring at us from every living brain in the guise of data, knowledge and wisdom, and manipulated in an enormous variety of complex forms and ways. So like the ancient Greeks, we will need to continue to use and study "information" in the intuitive sense afforded by our experience of it, not by the theories that we may be able to articulate about it.

Therefore, it is not surprising that the electronic age has been born more directly concerned with information processing devices, particularly in their specific electronic implementation, i.e., the conventional computers of our time. A major theme is thus the design and construction of electronic data processing machines. Such devices are called automata, computers, and more recently, smart phones and agents, especially when they are confined to a particular domain of application. The branch of science that has developed around this effort is primarily known as computer science. The ubiquity, importance, and uniqueness of the new tool are reflected in its singular name. We do not have car science, or microscope science, or even mammal science. But we do have computer science. Why could that be?

Upon reflection again, this is not surprising in view of the characterization above. If the subject matter of computer science is information, then computer science must be a second or higher order science because every natural science transacts in information of a particular type—physics about time, space, and matter/energy in general; chemistry about the composition and transformation of matter; and biology about organic matter in the form of what is called life. For example, cyberspace now includes most scientific content today, regardless of the science, and we expect a search engine to return answers about scientific questions as well. Therefore, an appropriate characterization of current computer science cannot focus on specific types of matter (e.g., silicon, or biomolecules, or subatomic particles), although that type of focused research can certainly enlighten the confines and boundaries of the field. Later developments confirmed that the pioneers in the field of classical computing (that largely defines it today, e.g. as in Hopcroft and Ullman [8]) rightly recognized that the abstractions of mathematics and logic were the appropriate setting to develop a science of computing. Since information encompasses concepts even more abstract than taxes, energy, and even quantity and number, it would appear necessary to go even higher in abstraction for a theory of information. On the other hand, as hinted at in the opening statement to this symposium, the models in classical computing are falling shorter and shorter of encompassing most of what is considered computing today. And as Dijkstra pointed out early on, their relation to the science of computing should not be any tighter than that of telescopes to astronomy. Which way to go in moving forward?

What Should be in a Model?

There is a variety of reasons for a science of computers. The most evident is the versatility and universal applicability of computers. Like energy, information is stored, transmitted, transformed, and manifested in many ways, if properly anchored in physical reality. Unlike energy, it seems information can be created, copied, hidden, and destroyed. Information is more abstract than energy, and it certainly can be used to control and manipulate energy and ordinary physical objects. That is the ultimate reason why information processing machines, computers in particular, have a versatility and dexterity unprecedented in history. They allow simulation of virtually everything, from physical phenomena such as wind tunnels and weather, through genetic interactions, through chess players and, according to some, even full human brains. This influence shows no indication of abating and, in fact, appears to have the potential to engulf society. It is thus perhaps inevitable that we will find an increasing demand for a science that provides us with understanding and inspiration to dream up better and more powerful computers today that may perhaps be produced for availability a few months or decades from now. In this work, we will explore what appears to be the most appropriate answer if the overarching eventual goal of computer science is to elucidate the nature of information, including information processing devices (computers), not only as they exist in seller's catalogs, but more importantly, in their fundamental nature as a scientific and technological quest.

How then to undertake a study of this nature? Historically, beginning with early models in physics, understanding any type of phenomenon, be it physical, chemical, or biological, has amounted to the construction of models of the phenomenon that

- Can play an explanatory role (what is really going on?).
- Have relatively simple components (how to build them?).
- Afford a predictive power and a useful role (how to use them to enhance our lives?).

These models have remained appropriate even though they require a human brain to be used, as noted above. Furthermore, increasing use of automation seems to be pointing to the need to build some type of intelligent control or "brain" in these models. Therefore, creating models of information processing is an incredibly complex task in its full scope because even human brains and minds may be construed as information processing devices. If so, trying to create a brain is somewhat analogous to having a personal computer trying to create a better computer than itself by itself. So, for the first time in history, we are no longer trying to understand objects and processes outside the human mind, but something awfully close to the human brain/mind itself. Real human brains/minds might not be appropriately regarded as such models, since they may be too complex to permit the type of manipulation and understanding that models should provide. We thus find ourselves facing another forbidding task, re-inventing brains, a task that is almost as daunting as finding a sound definition of information.

A good model has to keep a balanced tension between its complexity and understandability, on the one hand, versus the complexity of the real phenomena it should explain and predict, on the other. A computing device must therefore include at least three basic components:

- Some mechanism(s) for reading *and encoding* (one might call it perceiving) *input* information from the external world, which in turn requires things like sensors, memory, internal states, activation levels, and other computational primitives;
- Some *processing* mechanism(s) that dynamically change these internal states, i.e., "crunch" information, or rather, its representations;
- Some mechanism for returning *output* that allows the device to react back to the environment and/or itself, presumably with some effect (achieving some particular goal(s), learning, or survival, for example.)

Note the insistence on the generic word "mechanism" consistent with the desire for a scientific explanation (as opposed to the concept of a black box that just "gets the job done.") Mechanism is used here, for the time being, in a realistic sense and it may, or may not, imply a mechanic or electronic device. Brains excel as information processors despite the fact that many people may have difficulty with a brain being called "a mechanism."

These requirements may appear on the surface to be the standard abstraction of a processor, memory, and I/O that the classical Turing machine and von Neumann architecture give us and thus be nothing new. In view of the previous discussion, however, it is now clear that, as originally conceived in biological systems, these requirements have been essentially ignored in conventional models of computations because there are no specific I/O processes in Turing machines, finite automata, or formal grammars (more below.) As a result, the absence of this component has resulted in "missed opportunities" for models of computation, as pointed out by Burgin [9]. Although in abstract these features are appropriate for a general model of information processing, we submit that an appropriate re-definition of computer science requires a fairly substantial reassessment of their nature, particularly concerning their implementation in the real world. In fact, we are proposing that the definition of computer science should entail giving as complete an explanation as possible of such models of information processing "mechanisms" by providing a less abstract and more grounded characterization of them in physical reality, albeit in a fundamental departure from the classical models, now along three fundamental directions: *memory*, *adaptation*, and *evolution*. We discuss them in turn. The changes are generally motivated by the idea that the devil is in the details, i.e., that the better direction to go is what might be termed "contextualization," "embodiment," and/or "complexification" rather than "abstraction," as discussed next.

Memory and States

There is a phenomenon that implicitly pervades the computing field, but a satisfactory explanation of which has not been made central to the field and thus remains as yet unfledged in computer science. Despite substantial progress to elucidate how it actually works in other areas such as cognitive science, the phenomenon of memory surely holds the key to understanding a number of central issues in computing. In particular, questions such as "What does it mean 'to remember'?" and "How it is possible for humans to remember?" no longer admit a simplistic state-based explanation, especially in view of the fact that current storage-retrieval mechanisms used on computing machines perform so poorly when compared to human memory. Yet, memory seems to be an inherent property of a "brain," human or not. Life in general requires some sort of learning and adaptation, and it appears utterly impossible to argue that these capabilities are possible without some form of memory capability. Therefore any model of a computer must in some way or another involve some sort of memory system. In fact, a key feature of much of classical computer science (Turing machines, data bases) is that they make more sophisticated and appropriate choices for a solution as to the problem of memory. However, memories are not supposed to include means that automatically integrate them into the world, but require external intervention in choosing, rather imposing, the appropriate state set for a Turing machine, or schema for a database. The first departure in the appropriate definition of a computational model is therefore that it must include some "world model" in the form of *a **functional** memory tightly integrated with the input/ouput coming from/to the real world*. Turing machines come close to meeting this condition in abstract, but one must be very lax in accepting the abstraction of a linear tape or finite control as a valid real-world context. Neural nets come much closer in principle, although standard applications have been made in simulation in silicon (which were originally inspired and are still bound by the symbolic linguistic paradigm behind the computer metaphor.) Evolutionary systems likewise use relatively poor representations of the real world (symbolic data structures for genes and genomes), although they appear more tightly integrated to the world by bearing a closer resemblance to biological processes.

In general, we can distinguish three qualitatively different kinds of memory, corresponding to three stages of evolution, characterized by the key assumptions they make: *symbolic*, *subsymbolic,* and *embedded*. In symbolic computing, it is assumed all kinds of information can be expressed into finite strings of symbols from a basic finite alphabet of building blocks. Cognitively, a symbol is in general an empty box whose meaning is to be contextually interpreted by an observer in a specific interpretation of the symbol. The early bulk of the work in computer science—reflected in current commercial products (conventional digital computers and software)—is based on this type of representation, which has been the prevalent common assumption about data. Symbols are meant to be read sequentially, and so they are associated with sequential information processing, although symbolic representations are also used in processing by parallel machines. Sequential processing reflects the hard constraints of time and its passing, as it appears in ordinary experiences. By contrast, subsymbolic representations in the second kind are more distributed both in space and, hence, time. For example, the basic representation of heredity—the genome in a biological organisms—is a complex chain of basic DNA nucleotides a, c, g, and t; likewise, the physical manifestation of complex living organisms (protein ensembles) are known to be the genetic code of aminoacids, fairly complicated chemical compounds hardly reducible to simple character strings over an alphabet. More complex biological representations of information, such as organs and even brains, have a more complex structure. This type of extra-linguistic representation was achieved through experimentation in early physiology and psychology, and more recently in cognitive science and neuroscience. The representation of information here is by patterns of activation in the form of distributed representations across a large number of cells, each in itself only possessing a nearly negligible amount of memory and processing power. Eyes and brains—animal and human—are primary processors of this kind. Examples of classical abstract models that process this kind of information are cellular automata, neural networks, and even analog and quantum computers. In the past, these processors were called massively parallel because they involved many more processors than sequential computation, although it is becoming clear now that they pale in comparison with natural systems such as eyes, brains, and even DNA molecules. The ontological status of symbolic and subsymbolic types of information is really an open question. Symbolic information is usually thought of as an abstract mathematical string or integer, while a neural activation level is conceptualized as requiring more complex quantities, perhaps requiring values in a continuum.

A third kind of information is expressed in and transformed by entire populations of separate individuals somewhere embedded in the real world. A genome (seen as a collection of chromosomes) is an example, but a zygote (fertilized biological ovum) is a better example. Its "information content" (one might dare say, its memory) originates and accumulates in context- and environmentally-dependent ways, often spread in a large number of cells across the population of cells and exhibiting a *prima facie* summary of the temporal sequence of events that formed it and put it there. It is transformed by processes of interaction of the individuals among themselves and with their environment. The individuals themselves undergo changes through evolutionary processes that adapt and shape them, but which appear unpredictable and entirely dependent on environments, populations, and their histories. The representation, usually called a genome, an organism, or even a society or culture, encodes only a number of key features that serve as a seed to the organism or device, but this representation needs to grow under various environmental conditions through a morphogenetic process into a fully operational information processing entity and, moreover, eventually decay into something else. The mature entity has usually generated a new genome that inherits most of its prior features, although some new ones may be incorporated in the process by genetic operations such as mutation and crossover with other individuals, or even, as recent evidence indicates, as a result of interaction with their environment in the same individual. These operations bear resemblance to biological processes, which were, in fact, the original inspiration of the concepts. Today, they go well beyond the ordinary biological interpretation.

The overall point, however, is that a substantial departure of the classical model means that these types of representation may not be exhaustive. Information may be represented and transformed in nature and the universe in a variety of other ways, some of which we are merely beginning to suspect. They involve such greater numbers of individuals than molecular densities localized in space that counting their number or quantifying their "magnitude" may be meaningless. For example, quantum computing is based on highly distributed representations known in physics as fields. Electromagnetic fields, for example light and electricity, are usually spread across space and change with time. A field at a point of space-time can be quantified by real numbers in some physical or mathematical model. But nothing prevents us from allowing a fragment of physical reality—such as a population of DNA molecules or subatomic particles in an entangled state in some isolated container—from being the model itself, even if we may not be in possession of a analytical physical, chemical, or biological model of that fragment of reality, as simple and desirable as it may be. The point is abstract state-based memories cannot be offered as a universal model of computation.

Input/Output

The second essential ingredient in a computing device is some form of input/output. The obvious way to communicate it is by reading/printing strings over a fixed alphabet, as done in the classical computer models. However, many other ways naturally occur in the animal kingdom, even in verbal communications involving facial expressions and body language, or similar scents spread across a number of sporadic bushes in a jungle. Once again, interaction with the world may be also appropriate through physical signals (the electromagnetic spectrum, even physical interaction), particularly if interacting directly with a nonconventional memory. Sensors by themselves (such as reading heads in a Turing machine), however sophisticated, are not enough generally, because appropriate processing of the data picked up or sensed must take place by the device for it to be of any use. Of course, this argument does not preclude input tapes and symbolic representations as appropriate for some applications. The point is, they cannot be offered as a universal model for every application, particularly in the future.

Complexification Processes, Self-modification, and Evolution

Even a complete description of a genome will not, in general, give us an idea of how to build the corresponding organism (e.g., a human), let alone understand its information processing abilities. As pointed out by Lewontin, "DNA is a dead molecule, amongst the most nonreactive, chemically inert molecules in the living world. … Only a whole cell may contain all the necessary machinery for self-reproduction. A living organism, at any moment in its life, is the unique consequence of a developmental history that results from the interaction of and determination by internal and external forces …" [10]. Genetic material needs to be expressed, and the expression depends substantially on the environment in which it takes place. A crucial difference in evolutionary approaches to computation lies in accepting the fact that systems should not be entirely predetermined as independent or fixed entities, but rather should be left open to change so that context and environmental conditions can be accounted for. In particular, emphasis on programming should be placed in identifying a set of interaction constraints, as opposed to abstracting details of reality in an artificial environment devoid of a physical "home." The bulk of the information processing lies in the process of exchange with the environment, which requires the ability to sense input and produce some output, perhaps as novel and unprogrammed as the inputs themselves. Therefore, this process should not be produced by following a fixed set of rules insensitive to the situation where the computing device finds itself, but fully embedded in the context of a physical reality. Thus, computational models should always be embedded in an environment and therefore be subject to an evolutionary process that constantly seeks to add fundamental new components through learning, self-modification, automatic "re-programming," and/or evolution. This is one key feature that the new field of interactive computation captures, including evolutionary computation.

For centuries, traditional sciences, and even more recently computer science, have strived to identify key components in their objects of study in a reductionistic program that is supposed to explain complex phenomena with a choice of a few parameters governed by simple rules, usually blended together in a single equation or formula, such as $E=mc^2$. They are attempts to find simple models of complex phenomena, such as chess playing or deductive reasoning. Yet, the focal phenomena of many sciences seem impervious to this type of attack. Good examples are biology and psychology [11]. And as a higher order science, a science of information processing should be even more so. The growth of a zygote into an adult organism is a very complex process that can be affected by multiple factors beyond the initial genome contained in the zygote. In fact, much of the contribution of natural selection and evolution—and a key component in evolutionary strategies—is to build complex models of apparently simple phenomena, such as perceiving, walking, or growing, in an attempt to understand how complexity and information builds up in us and the processes around us [11], not just how it is distilled down into a single bit. Attempts to build such models (see for example, reflexive Turing machines in Burgin [12]) still impose Turing style rules on abstract models, contrary to the open ended nature of interaction in the real world. Thus far, therefore, most of the natural complexification process remains essentially ignored and unexplored in information processing, although it is obviously one of the most important aspects in the mechanisms in biology inherent to life processes. Nature versus nurture is an essential duality in the understanding of life. The point is that it should also be a primary and defining concern for the understanding of information processing machines in general as well (in both definition and efficiency assessment), ranking at least as high as the duality device versus process (machine/program versus algorithm) does in traditional models.

Summary of Current Progress

In summary, we are proposing that the appropriate definition of computer science for our time is the design and construction of devices capable of (a) building by themselves, perhaps from some initial condition and "genetic materials," a model of the world and the environment around them (the so-called "memory module"); (b) sensing their physical environment and possibly affecting changes in it as a result of their combination with their memory (the so-called "interaction module"); and (c) change their internal structure and organization unenthropically so to speak, i.e., by effecting in parallel a process of self-modification so as to increase the fitness with their ever changing environment and internal structure (the so-called "complexification module").

Naturally, the definition arises many questions. How does it change the current conceptions of computing? What does this re-definition imply for the practice of the science? What limitations may it eventually impose on it? How will it reshape the field? Below is rationale for the proposal and just a few initial observations about its potential implications.

First, it is clear that the classical definition fits this re-definition because it focuses on building models that share some of these basic characteristics, albeit in a very abstract sense. Classical Turing machines use a finite memory/control and symbolic module of the world (the read/worktapes). However, their interaction module is reduced to barely reading and rewriting symbols on a tape in isolation in order to produce a response to the algorithmic problem that it may be trying to solve. And their complexification module is nonexistent, or rather being implicitly outsourced to the creator of the machine and the default forced on it by the environment, or simply ruled utterly undesirable (who wants to change the machine that correctly solves an algorithmic problem frozen in time by the software requirements?). More recent models have attempted to address these issues by adding self-modifying components [5, 9, 12], but they are not powerful enough to escape Turing's logical gravitational field in either definition or consequence because they are usually primarily evaluated against the standard model, achieving, at best, hyper-Turing capabilities, so far removed from the capabilities that ordinary life forms exhibit so readily. Neural networks fit the definition better. They have made significant contributions in this definition, not only in the memory module (long-term memories in synapses among neuronal units), but more importantly in the interaction module (via associative memories, such as at the Hopfield model and Kohonen's self-organizing maps) and the complexification model (via learning algorithms, both supervised and unsupervised, if they were to be made an integral part of the models, and not just a preprocessing step to reduce them into the classical models.) Evolutionary models still use a symbolic memory module, but focus largely on the interaction and complexification modules (via genetic algorithms, genetic programming, and evolutionary strategies). On the other hand, quantum computers use subatomic particles in a highly nonsymbolic memory module, as well quantum physical processes as a critical part of their interactive module, although completely lacking a complexification module (dis/entangling particles would be fine examples, if they were an integral part of the quantum computer.) Finally, molecular computers use native DNA molecules as a memory module, which gives them an advantage from the beginning since they already "know" how to interact with the world (at very low cost in "programming" effort), while offering a good potential as a memory module via DNA indexing [13]. Their greatest advantage, however, resides in the complexification module because they readily offer the advantage of self-assembly, critical in natural biological systems [14, 15] . What is hard to find is a model that fully takes advantage of the three modules all together. Even more vexing is the fact that not one model thus suggested has been capable of creating its own model of such fundamental notions as space or time, something offspring of most living organisms seem to achieve so effortlessly and instantly. So geometry appears to be, after all, still an unresolved problem in computing.

Second, this re-definition would make other scientific developments, computational and non, relevant to computer science. For just one example, the area of robotics aims at building agents that can perform tasks automatically or with guidance, and exhibit a sense of intent (agency). A robot, however primitive, must possess at least some form of a memory and interaction modules, acting in tight synergy within and with its environment to be considered successful. The point here is, the development of these three basic modules and their synergy cannot be done appropriately in isolation and/or as an afterthought, but they must be done as in integral part of research and development in computing. Design of computing devices should be more of an evolutionary process of trial and error within the model itself, rather than an external design process, as it has been primarily considered and practiced so far. Such a device will at some point have to know when it has found what it is looking for (see Franklin and Garzon [16] for some elaboration), halt on that quest, and move on with the business of just becoming.

Finally, what are the major consequences of adopting such a re-definition? Perhaps the most fundamental to the current definition today is the renunciation of the primary role of the symbolic and abstract paradigm for information encoding and processing, in exchange for a return to a grounding of computer science in natural sciences. The many benefits and recent developments make this change both desirable and inevitable. One critical consequence is that the asymptotic behavior of algorithms and program would no longer be the primary criterion for assessment, simply because every computing device would have to be developed and evaluated as embedded in some physical environment (natural or simulated) to make sense, as opposed to being made into an abstraction in the neighborhood of some infinity, devoid of an anchoring reality. The synergy of interaction among the various modules and of the model with the environment would become far more important, regardless of whether the latter is discrete or continuous. Interactive computing, as is known today, would play a more important role to assess the performance and feasibility of a model. A second consequence is that the early themes of morphogenesis (in Turing's models) and self-reproduction (in von Neumann's models) would experience a revival. In fact, we suspect that topics such as morphogenesis, homorrhesis, and phylogenesis would take central stage in order to develop the complexification strategies to be embedded in superior computing devices. Higher-order learning (learning to learn [2]), would also come into the picture. A third consequence is computer science would develop a much more interdisciplinary nature, in more direct interaction with other natural sciences than symbolic and abstract computation has allowed so far; it would also be forced to develop more complex and interactive models than the oversimplified and insufficient models of traditional sciences would allow. This interaction is likely to provide the science with both, new powerful sources of inspiration and a far deeper and extensive range of applicability, as surprising as it may seem upon renouncing abstraction in implementation. And so, perhaps in a not so distant future, we may have a chance at staking an insightful definition of information that bears fundamental principles informing us of the scope and limitation of computing devices, extant or yet to be invented, in answering important questions to human kind in other areas like biology (such as phylogenetic questions [17]), cognitive science, or even chemistry and physics. Relativity will indeed be an advance to a Newtonian-like cyberspace.

References

[1] Denning, P. What is Computation: Opening statement. *Ubiquity* November (2010); http://ubiquity.acm.org/article.cfm?id=1880067.

[2] Bateson, G., and Donaldson, R.E. *A Sacred Unity: Further Steps to an Ecology of Mind*. HarperCollins, New York, 1972; http://www.rawpaint.com/library/bateson/formsubstancedifference.html

[3] Devlin, K. *Logic and Information*. Cambridge University Press, Cambridge, 1991.

[4] Burgin, M. *Theory of Information: Fundamentality, Diversity and Unification*. World Scientific, Hackensack, N.J., 2010.

[5] Burgin, M., and Dodig-Crnkovic, G. Preface: Information and Computation – Omnipresent and Pervasive. In *Information and Computation: Essays on Scientific and Philosophical Understanding of Foundations of Information and Computation.* World Scientific, Hackensack, N.J., 2011.

[6] Burgin, M. Information Dynamics in a Categorical Setting. In *Information and Computation: Essays on Scientific and Philosophical Understanding of Foundations of Information and Computation.* World Scientific, Hackensack, N.J., 2011, 35-78.

[7] Capurro, R. , Fleissner, P. , and Hofkirchner, W. Is a Unified Theory of Information Feasible? In *The Quest for a unified theory of information*. Gordon and Breach Publishers, New York, 1999, 9-30.

[8] Hopcroft, J., and Ullman, J. *Introduction to Automata Theory, Languages And Computation.* Addison-Wesley, Reading, MA, 1979.

[9] Burgin, M. *Super-recursive Algorithms*, Springer, New York, 2005.

[10] Lewontin, R. C. The Dream of the Human genome. *The New York Review of Books* 39, 10 (1992), 32-40.

[11] Bar-Yam, Y. (1992). *Dynamics of Complex Systems*. Addison-Wesley, New York, 1992.

[12] Burgin, M. Reflexive Calculi and Logic of Expert Systems. In *Creative Processes Modeling by Means of Knowledge Bases*. Sofia, 1992, 139-160.

[13] Garzon, M.H., Bobba, K.C., Neel, A., and Phan, V. DNA-Based Indexing. *International Journal of Nanotechnology and Molecular Computing* 2, 3 (2010), 25-45.

[14] Garzon, M.H., and Neel, A. J. Molecule-inspired Methods for Corse-grained Multi-system Optimization. *Computational Neuroscience: Springer Optimization and Its Applications* 38, Part 2 (2008), 255-267.

[15] Qian, L., and Winfree, E. Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades. *Science* 332, 6034 (2001), 1196-1201.

[16] Franklin, S., and Garzon, M.H. On stability and solvability (Or, When does a neural network solve a problem?). *Minds and Machines* 2, 1 (1992), 71-83.

[17] Garzon, M.H., and Wong, T.Y. DNA chips for species identification and biological phylogenies. *Journal of Natural Computing* 10, 1 (2011), 375-389.

About the Author

Max Garzon received a B.S. in mathematics and physics from the National U of Colombia and a Ph.D. at the University of Illinois, joined the University of Memphis in 1984 and has enjoyed sabbatical leaves at various universities in Europe, Asia and Latin America. Early in his career, he did research in the complexity of symmetric computational memory structures for sequential computers and separation of sequential complexity classes. His current research focuses on interactive computing, broadly including parallel, distributed and evolutionary computing and human-computer interaction, both in the traditional areas and the emerging areas of biomolecular programming and bioinformatics. A primary research thread is the development of morphogenetic and self-modifying models of a geometry, space and time inspired by biological systems. In addition to over 150 articles and books in these areas, he has developed or been instrumental in developing software products for complex systems, simulation, and control, such as *Edna* (a virtual test tube), early versions of *Autotutor* (an intelligent computer-based tutoring system for instruction in computer literacy and conceptual physics), online election systems and a variety of software solutions for local business and industries.

DOI: 10.1145/2390009.2390010