**MEMORANDUM**

**From:** Christoph von der Malsburg

**To:** Michael Arbib, Chair, Computer Science Dept., USC

**Date:** Dec 1, 1999

## The Challenge of Organic Computing

The computing world has several decades of exponential growth behind it. It is natural to expect that some fundamental conceptual adaptations are necessary and imminent. If we want to grasp intellectual leadership, we will have to be among the first to realize developments and stay ahead of them. What I am formulating here is meant more as an aiming point on the horizon than a detailed action plan.

### The Algorithmic Schema

The nucleation point of computing is the algorithmic schema. In its original and ideal form it couples the creative process in a human brain to a logical process in a physical device, linked by the program as the interface. All creative infrastructure (goals, interpretation, methodology, diagnostics and so on) is with the human, all the machine can do is blindly follow commands. In consequence, the logical process has to be deterministic – the slightest deviation of the logical process from the program leads to nonsense, the machine is a blind man on a tightrope. To ensure determinism, machines have to be digital and the creative and the logical process have to be coupled by detailed communication, the human having total knowledge and control of the process as executed.

The classical computer science department was totally dominated by the algorithmic paradigm. Hence the classical canon of courses: algorithms and data structures, theory of complexity and computability, languages and compilers, and computer architecture. There is an obvious trend in favor of fields of application (graphics, computational geometry, multimedia, authoring tools, databases, process control, simulation or various kinds, networking, robotics, artificial intelligence, computer vision and more), and we have to ask the question whether the underlying computing paradigm isn't in need of adaptation and extension or even replacement.

### The Algorithmic Schema is Doomed

The reality of the modern general purpose computer is straining the classical algorithmic ideal to its breaking point. The problem is that the algorithmic schema is appropriate only

if one creative process is coupled to one logical process. In contrast, today's workstations are complex systems of hundreds or thousands of interdigitated programs with a volume of hundreds of MBytes. Many processes, between them often serving many users, are running "independently" of each other, juggled by an operating system administration-style. Process determinism is maintained, but only as a shadow of its original glory, and at a tremendous price. The problem is the lack of coordination in terms of type and timing of interrupts by the operating system, other programs, input signals and users, such that essential organizational aspects of the process can at best be grasped statistically. Furthermore, the exact target hardware substrate of a program is generally not known and may be changing quickly at execution time (when networks of changing load and configuration are at issue).

Large efforts are made to shield the machine code from the programmer and to shield the programmer from the machine code. This shield has the form of a thick layer of intermediate structure. In its 50 years of development, the computing world has developed layers upon layers of infrastructure and architectural elements to keep computing processes reliably on track in spite of programmer errors, and to keep programs simple in spite of the enormous complexity of the developing software systems. Thus, proven structures are offered as templates and to restrict the programmer's freedom of making mistakes, in the form of high-level languages, compilers and debuggers, standardized interfaces, IDEs with structured editors and version control environments, programming styles, and operating systems. Also hardware-specific optimizing compilers or optimizing communication protocols on networks shield the programmer from the details of the executed machine code, and this is also one of the explicit goals of the style of object-oriented proramming. Part of this development can be characterized as a slow migration of the creative infractructure into the machine.

Although the individual programmer has only very indirect and incomplete control of the logical process, since this is co-determined by a large community of programmers and by mere contingencies, tremendous efforts are expended to maintain the facade of the algorithmic paradigm. To this end, data sets and process threads are kept separate by complex security checks and context switching mechanisms (that it may take 50,000 switching operations to move a bit from one location in a complex machine to another) and a semblance of sequential action is strenuously maintained, in spite of a lot of actual concurrency in modern processing units.

All of this works as well as it works, although there are limits to the degree of integration between different programs and the efficiency of the over-all process. However, coming decades will see mounting pressure to improve system integration. In addition, the time is not far that the biggest processor chips in conventional VLSI structure will have a billion transistors instead of today's 10 million, and things may change even more dramatically with the advent of exotic hardware such as molecular computing. One problem with these future structures may be serious difficulty of maintaining digital determinism.

One of the implicit constraints in the algoritmic scheme is that numerically extensive processes be controlled by simple programs. This imposes constraints on the form of

processes. Thus it is, for example, difficult to deviate from sequentiality. A process may be composed of parallel threads only if those threads are progressing in lock-step, as in vector processing, or if they form highly standardized concurrency patterns, as in RISC architectures. The difficulty lies in the fact that uncertainties in relative timing of threads leads to unforeseeable contingencies. Nonetheless, pressure will mount to make massively parallel computing a reality and to seriously couple thousands of computers over a network. This will in the long run only be possible by bursting the chains of the algorithmic schema.

From all of this the conclusion must be drawn that a new computing paradigm is required.

Organic Computing

The tremendous bottleneck of the algorithmic schema is the required detailed communication between human and machine. The obvious and only way to overcome the problem is to incorporate enough creative infrastructure in the process in the machine to help it keep itself on track. This will make machines more complex, but it will also remove the present obstacles standing in the way of higher complexity and utility. On the other hand, once the need for detailed communication is obsolete the machine will be freed from many of the awkward constraints with which it is bugged today. It is true that the present communication infrastructure (debuggers and so on) are likely to be replaced by diagnostic tools (if only to satisfy our curiosity), but the tremendous hocus-pocus going on to satisfy the algorithmic style is going to disappear.

Organic machines will no longer need to be deterministic in detail and correspondingly the digital style can be given up. Digital bits are realized as high-gain bistable feed-back devices to escape the noise and uncertainty of continuous signals. The digital style is expensive in terms of energy consumption and system complexity. On the other hand, an unbridled analog computer would just be drowned in noise. To keep an analog machine on track it needs mechanisms for pattern restoration (or pattern creation in the first place). It may be supposed that the most important such patterns will be loops of self-consistency, no signal being able to survive long if it is contradicted by its own consequences. Another useful principle will be that each individual signal pathway is embedded in independent alternate pathways which either support or contradict its effect. The system must be given a tendency to favor constellations with positive rather than destructive interference. Another constraint that will fall is that for sequential process structure. Another tremendous relief is going to be the possibility for complete co-location of data and active elements, to the point they can no longer be distinguished.

With many computing processes we are not ultimately interested in the details of the process but rather with their global behavior in terms of comparably simple output patterns. Just think of database search, network communication, virtual reality, robot control, artificial perception, or natural language processing. We therefore should be able to "educate" organic computers by giving them directives on the behavioral level, and by letting them learn, leaving the implementation details to the process itself. The ball is rolling in this direction already, as discussed above, it only needs to be targeted as an end in itself.

## Life as Example

All of this would sound like a pipe dream if it wasn't for the example of life. There can be no doubt that life isn't a computer on any level, if computing is taken in the traditional algorithmic sense: no programmer, no program, no determinism. However, with a somewhat more appropriate concept of computing, life could almost be taken as its definition: it realizes complex and purposeful chains of processes, it self-stabilizes, it adaptively integrates subsystems, and there are probably more such desirable traits of life that we ought to associate with the word computing. This is evident on all levels, from organismic self-maintenance (of cells and multi-cellular organisms, in terms of metabolism, survival and reproduction), to ontogenesis under the control of genes, the evolutionary-genetic apparatus optimized for fast evolution, ecological systems, the brain, and human society with economy and cultural processes including science. It is especially the brain that will have to serve as a model. We are most inclined to see it as a better, intelligent, kind of computer (in distinction to the genetic-evolutionary apparatus or human society, which we don't traditionally see as the intelligent entities that they really are). The brain is way ahead of computers in terms of all desirable achievements, except those of a genuinely algorithmic nature, and it displays all the structural traits discussed above for organic computers (non-digital, massively parallel, held on track by locally acting adaptive principles, co-location of data and active elements, and amenability to education on a global level). The algorithmic scheme has been a tremendous distraction in attempts to understand the brain. What is needed is fundamental re-thinking of the relationship between the two, and more generally of the relationship between our most complex artificial systems and life. Both sides will profit tremendously from exploitation of the analogy with the other.

## Organic Computing as Research

The Human Genome Project and many related efforts are at the present time producing terabytes of genetic information. There is much talk about the importance of building the new field of Bioinformatics to support this effort in terms of better algorithms and software packages for reconstruction and database search. It will become clear soon that this perspective on the role of computer science falls way short of what will be required to digest the flood of genetic information and, above all, to make sense of it. The activation of each gene is controlled by a number of others, which in turn are controlled by still others, the whole forming a hierarchical edifice of enormous complexity. The maintenance of cells and whole organisms, and the control of ontogenesis is the result of complex regulatory processes certainly as complex as those active in a modern computer.

Now, this analogy of the genetic apparatus to the working computer makes it all too evident that here is a very important unrealized research subject in the field of genetics, and that biology needs all the help it can get to cope with this problem. The first programs ever written where strings of machine commands. It then became quickly clear that a program beyond a hundred lines could no longer be handled directly, due to the spaghetti code problem. As a result, computer science built up the complex tools and control hierarchies already mentioned. Life has run into the same problem 3 billion years ago,

and has responded in the same way: inventing structured programming, so to speak, with the equivalent of operating systems and programming styles and subroutines and control parameters and so on. It is a striking fact that this realization hasn't dawned yet on biology, with very few exceptions, and that biology sees the genome still as nothing but spaghetti code, to be deciphered in terms of individual gene control interactions, the equivalent of individual machine instructions.

From this situation arises a great opportunity for an exciting new research field. It is high time to start modeling genetic control hierarchies, throwing at the problem all the insight, knowledge and methodology that computer science and related fields have amassed in recent decades. There is virtually no activity in this field, and yet it is not a day too early to start this effort. Biological progress can be accelerated enormously if clearcut ideas and models are formulated for global genetic control structures and if methods are made available to extract from the quickly emerging pool of information evidence on the actual structures realized. The genomes for a number of whole organisms is already available. In the case of the roundworm C. elegans we even know the output of the genetic program down to the history of individual cells. What is missing is the connection between the two.

Organic Computing as a Technology

This and similar efforts, directed at research subjects such as the brain, ecology, economy, or the information society, are not only a hunting ground for theoretical efforts, but there will be tremendous pay-backs to computer science in its effort to develop the theme of organic computing. It is not clear what the implementation technology of computers is going to look like in 20 year's time. Although it likely will be a version of VLSI, exotic alternatives may also come to the fore – molecular or optic or quantum computing. It is likely that that technology will lie at the intersection point of nano-, info- and bio-technology, one of the buzz-phrases of today, with the bio-aspect possibly concerning more the style than the substrate. In any case, however, the drive for a new style of computing will mount, under the dual pressures of complexity and microscopic indeterminism. Although it is likely that the first organic computers will be "simulated" on the basis of a fundamental algorithm run on a deterministic machine, algorithmic control and digital determinism will eventually be given up altogether (except of course in that important subset of problems which are algorithmic in their very nature, such as monetary accounting, solving differential equations, encryption, or signal multiplexing).

Conclusion

What I am writing sounds like science fiction, and indeed it is. But there is no doubt that I am describing the future of computer science, the main uncertainty being when that future will start, and who is going to be the leader. Let our department be it!! I do think we have many components of the necessary effort already in place in the department and at ISI, and that USC is an ideal environment for the venture, with several schools on the move in terms of fund-raising and the willingness to identify new goals, and several centers and initiatives already under way, ready to be gelled into a grand challenge that deserves its name.

Let's face it, the current reality of computing, software or hardware, is dominated by companies and markets rather than by science. Genius is measured in terms of clever market ideas rather than fundamental scientific insights. Winning structures, such as operating systems, interfaces, communication standards, net browsers, spread-sheet programs or editors, are the result of billion dollar market battles and not of intellectual achievements. In that particular climate, academic departments are not in the best of positions to excel and are in danger of degenerating into training centers for IT personnel. A CS department can attract the best minds among the young only by projecting intellectual leadership. Organic computing is going to be the barycenter of computer science in the coming century. Let's be the first to realize it. Let's put it up as our strategic plan and line up behind it!!