Research Proposal:
Programming technology for molecular-scale computing

MIT Artificial Intelligence Laboratory
Principal Investigator: Prof. Hal Abelson
Other investigators: Prof. Gerald Jay Sussman, Dr. Thomas F. Knight, Dr. Radhika Nagpal

Introduction

Progress in molecular electronics is beginning to yield the technology for creating structures that incorporate myriads of nanoscale computationally active units. These could be fabricated at almost no cost, provided (1) the individual units need not all work correctly; and (2) there is no need to manufacture precise geometrical arrangements of the units or precise interconnections among them. Programming such structures to perform useful computations is a significant challenge.

The objective of the research proposed here is to create foundational programming technology for reliably obtaining coherent, prespecified behavior from vast numbers of unreliable information-processing units, irregularly arranged and interconnected in unknown and even time-varying ways. Our approach combines principles for controlling complexity, drawn from computer science, with techniques for robust design, inspired by biology.

Our work will include: (1) identifying appropriate organizational principles and algorithms; (2) designing new programming languages that can express these algorithms effectively; (3) implementing practical compilation technology for reducing high-level programs into elementary operations that can be carried out by individual information-processing units; (4) demonstrating, in simulation, a few prototype applications.

We will focus on applications that anticipate incorporating molecular-scale computing substrates into intelligent materials. Success in this research would have revolutionary impact by creating the system architectures and algorithms for information-processing agents that can be constructed at prices comparable to the raw material costs. For example, we envision coating bridges or buildings with "smart paint" that senses and reports on traffic and wind loads and monitors structural integrity, or "intelligent blocks" that assemble themselves into prespecified three-dimensional shapes. The resulting system architectures would be physically feasible at any scale. They would be able to coordinate information from vast numbers of distributed sensors and to use this information to control equally vast numbers of distributed effectors, thus enabling the construction of artifacts with unprecedented responsiveness to their environment.

Architecting robustness in the presence of defects

To harness these substrates for useful computation requires developing system architectures that can operate robustly in the presence of defects at the micro-scale. This is a critical challenge. Computer science has been built on a foundation that largely assumes the existence of a perfect infrastructure. Integrated circuits are fabricated in clean-room environments,

tested deterministically, and discarded if even a single defect is uncovered. Entire software systems fail with single-line errors.

One approach to defect-tolerant molecular computing is to have the system begin with a self-discovery and diagnosis phase, where the elements test each other to discover which ones are operational and how they are interconnected. The system then reconfigures itself and its communication paths to avoid the broken parts and compensate for irregular interconnections. This approach is being pursued by a joint team from Hewlett-Packard Laboratories and UCLA, which is developing the molecular-electronics building blocks for such a reconfigurable technology. (See Collier et. al., 1999).

We pursue a different approach to achieving robustness, which we call *amorphous computing*. Amorphous computing exploits software methodologies that do not require precise control over the interaction or arrangement of the individual computing elements. It can tolerate defects, both defects that are present initially, and defects that appear during operation.

Biology provides a rich source of metaphors for inspiring such methodologies. Biological systems rely on local computation, local communication, and local state, yet they exhibit tremendous robustness. No existing engineering framework creates such complex structure from simple, potentially faulty components or maintains such complex behavior of existing structures under dramatic environmental change.

The phenomenon of morphogenesis, for example—the growth of form in organisms—demonstrates that well-defined shapes and forms can evolve from the interaction of calls under the control of a genetic program, even though the precise arrangements and numbers of the individual cells are highly variable. It seems awkward to describe mechanisms such as embryonic development as producing a "right" organism by correcting bad parts and broken communications. The real question is how to abstractly structure systems so we get acceptable answers, with high probability, even in the face of unreliability.

Despite the difference in perspective, we plan to maintain close contact with the Hewlett-Packard team, since we anticipate that the two methods are complementary for achieving effective molecular-electronic computing. It seems plausible that amorphous computing will be most appropriate in coordinating vast numbers of molecular-electronic elements, while reconfigurable logic will be most appropriate in creating the individual elements themselves.

Technical approach

Our work is based on an abstract model of an *amorphous computing medium* (Abelson *et. al.*, 2000). This is a collection of processing elements sprinkled irregularly on a surface or mixed throughout a volume. The elements are possibly faulty, but otherwise identical. They are sensitive to the environment, they may effect actions, and they might be mobile. Each element can communicate with a few nearby neighbors.

Each element has modest computing power and a modest amount of memory. The elements do not have any *a priori* knowledge of their positions or orientations. They are not synchronized, although we assume that they compute at similar speeds, since they are all fabricated by the same process. The particles are all programmed identically, although

each particle has means for storing local state and for generating random numbers. We assume that the number of elements is very large. Thus, the entire amorphous medium can be regarded as a massively parallel computing system.

To get a sense of what it would be like to program an amorphous system, consider a simple process of wave propagation. An initial particle, chosen by a cue from the environment or by generating a random value, broadcasts a message to its neighbors, which propagate the message to their neighbors, and so on, to create a diffusion wave that spreads through the system. The message can contain a hop count, which each particle can store and increment before rebroadcasting, ignoring any subsequent higher values to prevent the wave from propagating backwards.

Wave propagation with hop counts is evocative of the gradients formed by chemical diffusion that are believed to play a role in biological pattern formation. Consequently, we can attempt to organize amorphous processes by mimicking gradient phenomena observed in biology. As an example, we can use diffusion waves to produce regions of controlled size, simply by having the processors relay the message only if the hop count is below a designated bound. Once a region is generated in this way, we can use it to control the growth of other regions.

Diffusion waves are very simple mechanisms, but Coore (1998) has shown that they are powerful enough to support programs that can generate any prespecified planar pattern, up to connection topology, provided that the distribution of particles is sufficiently dense. Nagpal (2001) has extended this work to produce "intelligent surfaces" that can fold themselves into predetermined three-dimensional shapes under program, together with a compilation technology that translates description of macroscopic shape into programs that effect local deformations by individual elements embedded in the surface.

These programs are well-matched to amorphous computing, because the gross phenomena of growth is insensitive to the precise arrangement of the individual particles, as long as the distribution is reasonably dense. In addition, if individual elements do not function, or stop broadcasting, the result will not change very much, so long as there are sufficiently many elements.

This kind of robustness in the face of failing elements is different from traditional approaches to fault tolerance. Traditionally, one uses redundancy to identify faulty components and repair or replace them. In contrast, for these amorphous algorithms, it simply doesn't matter whether isolated components fail—the behavior of the overall system will still be acceptable. We believe that the principle of designing for *acceptable behavior of the aggregate* rather than *correct behavior of individual components* is critical for the effective exploitation of vast aggregates of molecular-electronic components.

Proposed research

We propose to use the amorphous computing model to assemble a catalogue of basic computational constructs and algorithms designed to exploit the potential of molecular computing. Although this work is broadly foundational, we are motivated by possible applications of intelligent materials with embedded molecular-electronic components. Such applications include shape formation, pattern recognition, and environmental sensing. While this work

will be done via simulation, we will remain in close contact with researchers working to fabricate molecular-electronic technology at Hewlett-Packard and elsewhere.

Statement of work (two-year effort)

1. We will create a catalogue of computational primitives, suitable for implementation by vast collections of possibly faulty molecular-scale computing elements. These will include important mechanisms abstracted from biology, such as the formation of local coordinate systems, inhibition, repetitive implementation, local recruitment, growing points, and tropism.

2. We will design programming languages that incorporate the above primitives, and use these to express algorithms that support prospective applications of intelligent materials.

3. We will design development tools to aid an engineer in the design, analysis, and debugging of algorithms and systems to be run on vast systems of molecular-scale elements.

4. We will implement compilation technology to translate programs written in these languages into combinations of elementary operations that can be realized by individual molecular-scale elements.

5. We will test the behavior of the above systems and algorithms, using both simulation and theoretical analysis, paying particular attention to robustness in the face of defective elements and unreliable communications, and we will report on these results.

6. We will demonstrate the utility of these robust mechanisms by constructing sample applications that degrade gracefully, producing acceptable behavior under extreme conditions of failure of individual components.

References

[1] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsey, T.F. Knight, R. Nagpal, E. Rauch, G.J. Sussman, and R. Weiss. Amorphous Computing, *Comm. of the ACM*, May 2000.

[2] C.P. Collier, E.W. Wong, M. Belohradsky, F.M. Raymo, J.F. Stoddart, P.J. Kuekes, R.S. Williams and J.R. Heath. Electronically Configurable Molecular-Based Logic Gates. *Science*, v.285, 16 July 1999.

[3] D. Coore. *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*, Ph.D. thesis, MIT Department of Electrical Engineering and Computer Science, December 1998.

[4] R. Nagpal. *Programmable Self-Assembling: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics*, Ph.D. thesis, MIT Department of Electrical Engineering and Computer Science, May 2001.