**Title and Subtitle**

Programming technology for Molecular-Scale Computing

**Author(s)**

Hal Abelson

**Performing Organization Name(s) and Address(es)**

Massachusetts Institute of Technology
Cambridge, MA 02139

**Sponsoring/Monitoring Agency Name(s) and Address(es)**

Navy ONR

**ABSTRACT**

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.

**Security Classification of Report**

U

**Security Classification of Abstract**

U

**Number of Pages**

9

**Distribution Statement**

Approved for Public Release
Distribution Unlimited
Final technical report:
Programming technology for molecular-scale computing

MIT Artificial Intelligence Laboratory
ONR Award number: N00014-02-1-0721
Duration: May 27, 2002 – May 30, 2004
Principal Investigator: Prof. Hal Abelson
Other investigators: Prof. Gerald Jay Sussman, Dr. Thomas F. Knight,
Dr. Radhika Nagpal

1 Abstract

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.

2 Technical objectives

The objective of this work is to create a new engineering discipline for reliably obtaining useful, prespecified behavior from vast numbers of asynchronous, irregularly arranged information-processing units that communicate with only a few neighbors.

Progress in microfabrication and in bioengineering will make it possible to assemble such amorphous systems at almost no cost, provided that 1) the units need not all work correctly; 2) the units are identically programmed; and 3) there is no need to manufacture precise geometrical arrangements of the units or precise interconnections among them. Harnessing these systems for information processing and intelligent materials requires augmenting traditional information processing with new perspectives and techniques from biology.

Computer science is currently 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. In contrast, biological systems rely on local computation, local communication, and local state, yet they exhibit tremendous resilience. 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.
Our new framework combines inspiration about robust design from biology – from morphogenesis and development – with the techniques of organization and control of complexity from computer science.

An effective integration of these perspectives will have a profound impact on computer science, biology, and microelectronics. In biology, we can perhaps address longstanding questions of morphogenesis, and discover a computational understanding of the developmental process. In computer science, we can engineer robust computational infrastructures, out of unreliable components. In microelectronics, we can biochemically pattern nanoscale information rich substrates with atomic precision. We can incorporate robust local computation, sensing, and effectors arrays into structures which exhibit sophisticated global behavior.

Our new framework will give us the power to take control of some biological processes, and to design and construct biological cells and cell assemblies with prespecified behaviors. Such controlled biological mechanisms will give us the ability to construct novel materials with engineered nanoscale structures.

3 Technical approach

In order to exploit programmable materials we must identify engineering principles for organizing and instructing myriad programmable entities to cooperate to robustly achieve pre-established goals, even though the individual entities are unreliable and interconnected in unknown, irregular, and time-varying ways.

We demonstrate the feasibility of these principles in prototype amorphous systems, implemented both in traditional silicon technology and in novel computational substrates that exploit molecular biology. This includes two major thrusts:

1. Invent new programming paradigms and languages for controlling amorphous computing agents.*

- We extend the simulation technology we have already developed to aid in the design and testing of algorithms for amorphous computers.
- We have already demonstrated that amorphous media can be configured by a program, common to all computing elements, to generate highly complex pre-specified patterns. For example, we can specify that an amorphous medium manifest a pattern representing the interconnection structure of an arbitrary electrical circuit.
- We further develop high-level languages for expressing algorithms for amorphous computers. They will provide support for descriptions that do not depend on precision interconnect or perfectly working parts. The concepts involved in the description of an algorithm will be conserved in the face of slow deformation or flow of the amorphous elements.
- We develop languages and programming paradigms for amorphous systems in which the individual elements are sensitive to their physical environment and
can affect it by changing their shapes. If appropriately programmed, collections of such elements can provide an implementation basis for self-configuring physical structures.

- We develop specific applications to demonstrate the utility of the amorphous computer. These applications include methods for the solution of partial differential equations, and coordination and processing of data collected from sensors distributed throughout an amorphous computer.

2. Investigate prototypes, both in traditional silicon technology and in molecular biology.

- We have demonstrated, by simulation of reaction kinetics, the theoretical feasibility of building a family of logic gates where the signals are represented by concentrations of DNA-binding proteins, and where the nonlinear amplification is implemented by in vivo DNA-directed protein synthesis.

- We have installed and equipped a complete recombinant DNA laboratory for constructing and characterizing "cellular logic" gates, based on these theoretical foundations, and we are using this laboratory to construct novel organisms incorporating small digital circuits.

4 Progress

Engineered cell-to-cell communication

One major thrust of this research is to make it possible to use living cells as a substrate for engineering, and to program colonies of simple bacterial cells to be test beds for the organizational principles of amorphous computing.

We have successfully cloned a set of Lux genes from *Vibrio fischeri* and *Photo-rhabdus luminescens*. This naturally occurring genetic circuit combines an intercellular cell density measurement with a complex biochemical light production enzyme cascade. We have cloned this system, isolated the components with three distinct functions, and re-assembled those in several distinct ways.

The three sub-components are the autoinducer sender enzyme, responsible for creating the small signaling molecule N-acyl homoserine-lactone; the autoinducer response protein, responsible for controlled activation of transcription dependent on the concentration of autoinducer; and finally the enzymatic light production cascade. The *P. luminescens* light production cascade was isolated specifically because of its ability to function at normal (37°C) growth temperatures, unlike the corresponding version from *V. fischeri*. This approach of selectively isolating components from a variety of sources, with the explicit intention of creating an easily engineered set of system level components is one important project goal.

To perform this experiment, Weiss and Knight isolated a DNA fragment from *V. fischeri* which, when spliced into a plasmid, caused transformed *E. coli* colonies to glow. They then sequenced the complete region (a gene cluster with 8645 base pairs) and isolated from this structure the gene clusters responsible for light production, autoinducer production, and
autoinducer response. By controlling expression of the autoinducer abd using light production as a sensing mechanism, they created a producer/sensor system that can transmit signals between cells.

Cellular-logic circuit design

We have demonstrated the fundamentals of "genetic process engineering" – taking existing genetic regulatory elements and modifying their DNA encoding so that they can be used in constructing complex in vivo digital-logic circuits. In particular, Weiss was able to mutate ribosome binding sites for the cl repressor and the operator for the bacteriophage lambda P(R) promoter so that the resulting cellular-logic gates had good noise margins and signal-restoration characteristics. This work is important because it shows how to synthesize biological components that can be combined to produce reliable circuits of significant complexity.

Programmable materials

Rather than build precisely engineered mechanical structures, one could program precise complicated structures starting from a single flexible mechanical base. Not only can one design many complex static structures from a single substrate, but one can also produce dynamic structures that can react to environment conditions or affect the environment. Such a programmable material would make possible a host of novel applications that blur the boundary between computation and the environment. Example applications may be a flexible car surface that can change structure exactly at the point of impact, a programmable assembly line that moves objects by producing ripples in specific directions, or manufacturing by programming global shapes on a single, flexible material.

Radhika Nagpal has developed a prototype model for controlling programmable materials. She has shown how to organize a program to direct a sheet consisting of a vast number of autonomous, asynchronous, identically programmed, locally communicating, and irregularly placed agents, that can individually deform ("cells"), to cooperate to construct a large family of globally-specified predetermined shapes. She demonstrated this by presenting a language that allows a programmer to specify a sequence of folds, in a way inspired by Huzita’s axioms for origami, that achieve the desired global arrangement. She showed how this language is compiled into a program that can be distributed to all of the agents. With a few differences of initial state (for example, agents on the edges of the sheet know that they are edge agents) the agents execute their copies of the program, interact with their neighbors, and fold up to make the desired shape.

Nagpal’s techniques are quite robust. She has investigated and reported on the range of shapes that can be constructed using her method, and on their sensitivity to errors of communication, random cell death, and density of the cells. We believe that Nagpal’s ideas will have an impact on the theoretical biology of differentiation and morphogenesis as well as on the development of technology for building more robust computer systems.
Self-repairing structures

An amorphous computing medium is a collection of irregularly placed, asynchronous, locally interacting "computational agents sprinkled irregularly on a surface or mixed throughout a volume. The agents, which are all programmed identically, are possibly faulty, sensitive to the environment, and may effect actions. For example, the medium might generate patterns as individual agents change color.

In our previous research at MIT, Daniel Coore demonstrated how an amorphous medium can be programmed to generate any prespecified pattern. Radhika Nagpal developed a prototype model for controlling programmable materials: she showed how to organize an amorphous medium of agents that can individually deform, to cooperate to construct a large family of globally-specified predetermined shapes. Both Coore's techniques and Nagpal's techniques are robust in the face of agent failure: the desired pattern or share will be constructed even if many individual agents have when the pattern or shape is to be generated.

Nagpal and Lauren Clement showed how to extend these techniques to create self-repairing structures. These are active patterns that regenerate and rework themselves when constituent agents fail. The key to this work was the invention of active gradients - organizations of communication patterns among the agents that continually check for failure and regenerate as necessary. The invention of active gradients was directly inspired by phenomena of limb regeneration in newts and insects.

We believe that this active gradient mechanism will be a fundamental building block for achieving robustness in molecular-scale and other amorphous computing applications.

Robust shape generation

In a second series of experiments in robust shape generation, Nagpal, Attila Kondacs, and Catherine Chang demonstrated how to accomplish the synthesis of arbitrary 2D shapes through cell growth. They showed how to compile a predetermined global shape to produce a program for a single seed particle that then "grows" the structure through replication. A key feature of this system is that it has the potential for both self-repair and regeneration after the formation process is over.

The basic idea is to represent the shape as union of overlapping network of overlapping circles. Neighboring circles are linked using local reference points relative to each circle; a circle can use its internal reference points to triangulate the location of all of its neighboring circles centers. A rooted spanning tree in this network represents a process for drawing the structure starting from a particular circle.

The growing process uses a technique of role competition, which automatically incorporates the basis of self-repair: In the process of growing a circle, every agent that hears a growth message from the circle center attempts to reproduce and place daughter agents randomly around itself within a ring. This agent only succeeds in reproducing if there is room at the chosen location for another neighboring agent. Agents are aware of their neighbors, and they resume replication when a neighbor disappears. Hence the structure never really stops growing and constantly replaces dying parts. The system is also capable of
regeneration after a large part of the shape is destroyed. This is a consequence of the shape representation chosen. Each circle contains enough information to generate its neighbors, and this process can occur recursively. As a result agents can rely on their original growing procedure to regenerate.

Distributed hierarchies and persistent nodes for reliable storage

Jake Beal has invented an algorithm for organizing an amorphous aggregate of agents into hierarchical clusters. This Leaderless Hierarchy Algorithm builds hierarchical groups quickly and robustly on the basis of local communication, scales to systems of enormous size, and adapts automatically to agent failure and to disruption of communication. Beal’s work indicates that amorphous computers can support robust addressing schemes that will form the basis of robust implementations of distributed processing algorithms.

One application of Beal’s clustering methods is for reliable data storage and retrieval in amorphous systems. An amorphous system forms a large network, where agents communicate if they are geometrically nearby. Beal demonstrated how to reliably store data in the system by distributing a huge number of copies of the data over a geographically local portion of the network, by means of persistent nodes – local collection of agents that collaborate to store a key/value pairs. Beal’s persistent nodes implement atomic storage. They can reorganize themselves in reaction to agent death or communication failures, and they can and move about within the network to avoid damaged regions. As a result, one can “put a piece of data into the amorphous system” and have it stored reliably and consistently.

References


[19] A. Kondacs, Compilation and Biologically Inspired Self-Assembly of Two-Dimensional Shapes, 18th Int’l Joint Conference on AI, March 2003


