A Robust Engineering Using Biologically-inspired Models of Cell Differentiation and Morphogenesis.

We propose to use morphogenesis and developmental biology as a source of algorithms and general principles for organizing complex behavior from locally interacting individuals. We aim to design artificial systems that replicate biological robustness, and to use insights from these systems to understand the capabilities of biological systems. The goal is to be able to formalize these general principles as programming languages — with explicit primitives, means of combination, and means of abstraction — thus providing a framework for the design and analysis of self-assembling systems.

The results from this research will have significant impact, not only on our engineering principles for robust design but also on our understanding of biological systems. These new programming models will impact the design of and approach to reconfigurable robotics, self assembly, and smart-matter applications. In the long run we believe these ideas will help achieve coherent behavior from aggregates of biological cells. The biological comparisons will significantly improve our understanding of development and creation of morphology. Not only will this promote the use of computational models for understanding systems level biology, but it will also increase collaboration between computer science and biology in this area, where focus has traditionally been narrow.

This research is motivated by emerging technologies that will make possible novel applications that integrate computation into the environment: smart materials, self-reconfiguring robots, self-assembling nanostructures. We are faced with the challenge of achieving coherent and robust behavior from the interactions of multitudes of elements and their interactions with the environment. These new environments fundamentally stress the limits of our current engineering and programming techniques, which depend on precision parts and strongly regulated environments to achieve fault-tolerance. By contrast, the precision and reliability of embryogenesis, in the face of unreliable cells, variations in cell numbers, and changes in the environment, is enough to make any engineer green with envy. We believe that important organizational lessons can be learned from natural biological systems.

Nagpal has demonstrated the power of the programming-language approach by developing a system that combines local primitives from epithelial cell morphogenesis and *Drosophila* cell differentiation, and combination rules from the mathematics of paper folding. The language specifies a folding process on a flexible sheet composed of many identically-programmed "cells". The specification describes a global process that is subsequently compiled into local programs that are run on the individual cells. The process is extremely reliable in the face of random cell death, random cell distributions, and varying cell numbers. The language is scale-independent in that the resulting shape scales according to the size and proportion of the initial sheet without changes to the program.

The goal of the research described in this proposal is twofold: to extend our programming models to new domains, and to investigate the connection of our models to biological processes. We will design new languages that describe the formation of shape by exploiting replication (growth), motility, and deletion (cell death). Just as paper folding is a natural process for shape formation on a sheet, we will need to discover new models to describe shape formation in these new domains.

Our previous work has exhibited several similar traits to biological systems and provided us with insights into how these systems may work. We will exploit these insights to design biological experiments, then identify one or more colleagues in the Biology department and collaborate with them to perform these experiments. Specifically, we will investigate scale independence and morphological differences in closely-related species. We will also investigate the connection between the kinds of failures that we have observed in the simulated morphogenesis processes and those that occur in naturally-occurring birth defects, concentrating on failures of geometry, topology and

synchronization.

B Robust Engineering Using Biologically-inspired Models of Cell Differentiation and Morphogenesis.

We propose to use morphogenesis and developmental biology as a source of algorithms and general principles for organizing complex behavior from locally interacting agents. We aim to design artificial systems that replicate biological robustness, and to use insights from these systems to understand the capabilities of biological systems. We will formalize these general principles as programming languages — with explicit primitives, means of combination, and means of abstraction — thus providing a framework for the design and analysis of self-assembling systems. We have demonstrated this approach through a programming language that specifies a robust process for shape formation on a sheet of identically-programmed "cells", by combining local primitives from epithelial cell morphogenesis and *Drosophila* cell differentiation with combination rules from geometry. The goal of the research described in this proposal is to develop new programming models for domains such as growth and apoptosis (cell death), and to investigate the connection of our models to biological processes.

This proposal will demonstrate the need for new paradigms in programming (section C.1), outline our approach as well as present preliminary work that demonstrates the potential of this approach (section C.2), present our detailed research plan with specific methodologies and objectives (section C.3), and present our long-term research goals and educational goals.

B.1 Motivation: New Computational Environments

Emerging technology, such as MEMs[1] devices, is making it possible to bulk-manufacture millions of tiny computing elements integrated with sensors and actuators and embed these into materials, structures and the environment. Already many novel applications are being envisioned and built: computing elements that can be "painted" onto a surface [6, 46], sensor-covered beams that actively resist buckling [4], an airplane wing that changes shape to resist shear [12], a programmable assembly line of air valves [7], a reconfigurable robot that morphs into different shapes to achieve different tasks [19, 49]. Emerging research in biocomputing may even make it possible to harness the many sensors and actuators in cells and create programmable tissues [25, 44, 21].

These new computational environments pose many challenges to our current programming techniques, that are not met by distributed and parallel computing. These applications will require the coherent coordination of vast numbers of elements, where the individual elements will have limited resources and reliability and the interconnects between elements will be irregular, local and possibly time-varying. In addition the physical embodiment of these elements is strongly intertwined with the individual behavior of the elements and the global goal. Not only are we concerned with topology, but also geometry and physics. Achieving robustness from the local interactions of these vast numbers of elements, and their interactions with the physical world will be one of the major research challenges facing information technology in the coming decade.

These new environments fundamentally stress the limits of our current engineering and programming techniques. Current engineering paradigms rely heavily on precisely functioning parts, strongly regulated environments and strictly managed interactions in order to achieve robustness, and are dominated by a centralized, hierarchical mind-set [5, 39, 27, 28]. Programming strategies

[1] Micro-electronic Mechanical Devices. Integrates mechanical sensors and actuators with silicon based integrated circuits.

within the MEMs community (where much of the underlying device technology is being built) have for the most part been centralized applications of traditional control theory. The few decentralized approaches assume access to global knowledge of the system and tend to focus on hierarchical control [4, 17, 46, 38]. Such centralized hierarchies are not scalable and can be quite brittle, catastrophically failing if a high-level node fails. In the modular reconfigurable robotics community, most of the work has focused on centralized and heuristic searches, which quickly become intractable for large numbers of modules and often require untenable assumptions such as explicitly defining the final configuration and access to global position [37, 50, 15]. The tendency to depend on centralized information, such as global clocks or external beacons for triangulating position, puts severe limitations on the types of applications and environments, and exposes easily attacked points of failure. In general, our engineering methodologies do not easily adapt to errors or unpredictable changes, and adding such fault-tolerance is very costly. These programming strategies put pressure on system designers to build complex, precise (and thus expensive) elements and interconnects rather than cheap, unreliable, mass-produced computing elements that one can conceive of just throwing at a problem. Currently, however, few alternatives exist. Approaches based on cellular automata have been difficult to generalize; local behavior is constructed empirically, without a framework for constructing and composing local rules to obtain any desired goal [39, 10, 3, 29]. Evolutionary and genetic approaches are general, but the correctness and robustness of the evolved system is difficult to verify because of a lack of understanding of the local-to-global relationship [14, 30].

By contrast, biological systems achieve incredible robustness in the face of constantly dying and replacing parts. Cells with identical DNA cooperate to form incredibly complex structures, such as ourselves, starting from a nearly homogeneous egg [41]. The precision and reliability of most developmental processes, in the face of unreliable cells, variations in cell numbers and changes in the environment, is enough to make any engineer green with envy. The plethora of error-tolerant systems in biological systems, in contrast to our non-robust engineering paradigms, suggests that fundamentally different paradigms are used by biological systems to achieve this robustness.

B.2 Biologically-inspired Paradigm for Robustness

In recent decades, there has been significant progress in understanding how cells produce complex pattern and shape during development [41, 26, 48]. Morphogenesis (creation of form) in developmental biology can provide insights for embedding computation in the environment [2, 42]. Even basic developmental processes, such as gastrulation, are based on deliberate cell migration and shape change. Epithelial cells, for example, generate a wide variety of structures: skin, capillaries, and many embryonic structures (gut, neural tube), through the coordinated effect of many cells changing their individual shape [35]. Modeling such behaviors has been attracting increased attention from biologists [36, 21, 24, 23, 35]. We believe that important organizational lessons can be learned from these biological systems.

We propose to use morphogenesis and developmental biology as a source of algorithms and general principles for organizing complex behavior from locally interacting agents. Currently, developmental biology remains untapped as a source for biologically-inspired computing, in spite of its incredible robustness and complexity. Our preliminary results suggest that general principles for robust design can be extracted from these systems. Our approach is very different from other biologically-inspired computation that has focused on the genetic system and neural systems as the two main sources of inspiration. Instead our work is more along the lines of research on generalizing immune system paradigms to extract programming techniques for achieving robustness in the face of unknown external attacks [40, 13]. Our emphasis is on the design of programming models that capture the general principles for biological robustness and provide a framework for replicat-

ing biological robustness. We expect that these frameworks will give us insights into how complex morphology is created in biology.

We believe that the results of this research will have significant impact, especially as we design applications that that cross the boundary between computation and the physical world. These new programming models will impact the design of and approach to reconfigurable robotics, self-assembly, and smart-matter applications. In the long run, we believe that these frameworks will be useful for achieving coherent behavior from aggregates of genetically-modified biological cells [25, 44].

At the same time, our approach leverages our skills as computer scientists to uncover primitives and abstractions, and may give us a glimpse at how development functions at a systems level. Already, our preliminary work has given us insights into how scale-independent shapes can be achieved through local processes and in this document we propose using these insights to direct biological experiments. We expect the biological comparisons will significantly improve our understanding of development and creation of morphology. Not only will this promote the use of computational models for understanding systems level biology, it will also increase collaboration between computer science and biology in this area, where focus has traditionally been narrow.

B.3 Preliminary Work

In her PhD thesis, Nagpal provides an example of how this approach can be applied to achieving programmable self-assembly [32]. She presents a language for instructing a sheet of identically-programmed, flexible, autonomous agents ("cells") to assemble themselves into a predetermined global shape, using local interactions. With this language, a wide variety of global shapes and patterns can be described at an abstract level, compiled into cell programs, and then synthesized using only local interactions between identically-programmed cells. Examples include flat layered shapes, all plane Euclidean constructions, and a variety of tessellation patterns.

The programmable cell sheet model is inspired by epithelial cell tissues, which form a variety of structures through the coordination of local shape changes in individual epithelial cells. In particular the sensing and actuation model is based on mechanical models by Odell et al. of epithelial cell behavior during gastrulation and neurulation [35]. The programmable cell sheet consists of a single layer of thousands of randomly and densely distributed cells that can coordinate to fold the sheet along straight lines. The cells are identically-programmed and autonomously execute their programs based on purely local interactions with nearby cells (on the order of 15 neighbors).

The self-assembly works as follows: The desired global shape is specified as a folding construction on a continuous sheet, using a language called the Origami Shape Language. The language is based on a set of paper-folding axioms by Humiaki Huzita [20] and is inspired by origami, where a large variety of complex shapes can be constructed from a sheet using a small set of axioms, simple initial conditions (edges and corners of the sheet) and only two types of folds. The program for an individual cell is automatically compiled from the global shape description. All cells execute the same program and differ only in a small amount of local dynamic state. The cell program itself is inspired by biology and is composed from a small set of primitives: gradients, neighborhood query, cell-to-cell contact, polarity inversion and flexible folding. When the cell program is executed by all the cells in the sheet, the sheet is configured into the desired shape.

Figures 1 and 2 show two examples of classes of shapes and patterns that can be self-assembled by this system. In figure 1 a two dimensional surface of cells coordinate to form an inverter-chain pattern. Initially the surface is mostly homogeneous, with only the elements on the border having special local state: a value that indicates which of the four borders it belongs to. This is analogous to the presence of a small number of maternal determinants in an embryo, which determine the

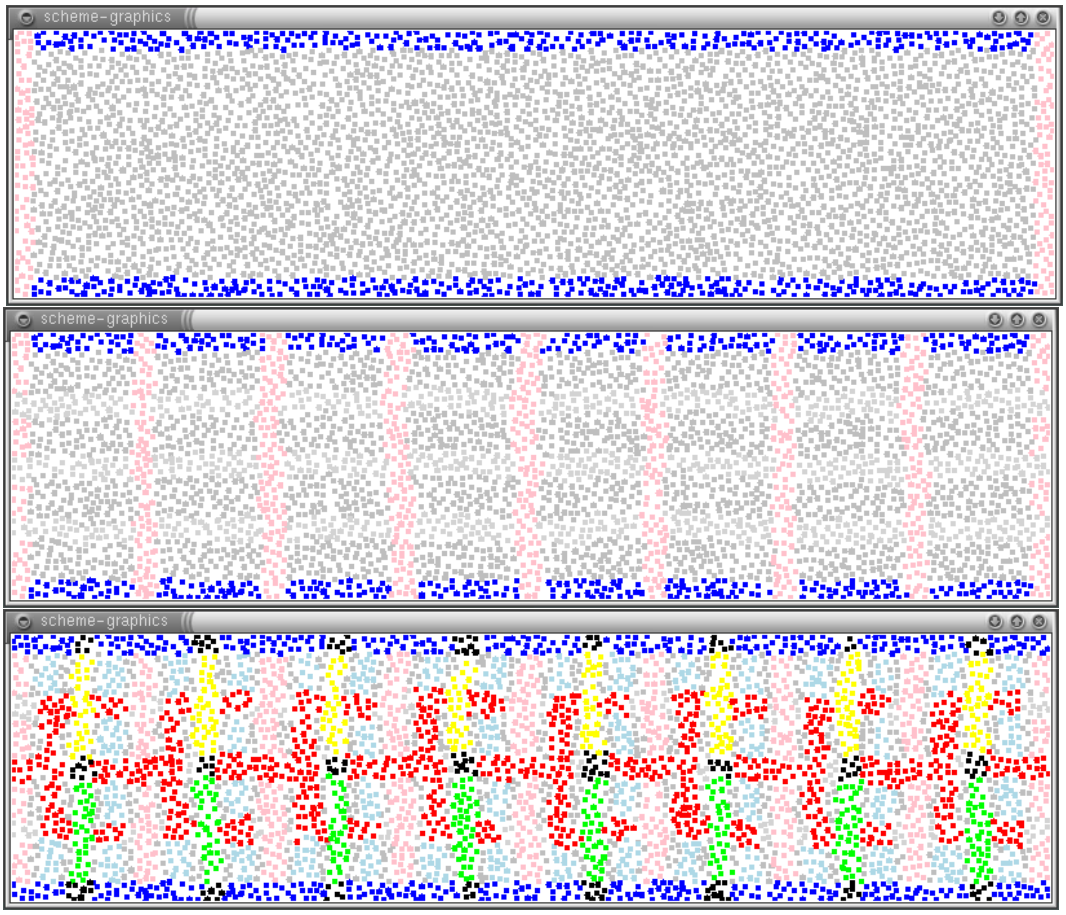

Figure 1: Differentiation into an inverter chain pattern. All elements have the identical program; initially only elements on the border have special state. Elements communicate only locally. The elements coordinate to first divide the region into 8 segments and then the elements within each segment differentiate to form an inverter pattern within each segment. Non-repeating circuit patterns can be created by differentiating based on the segment number.

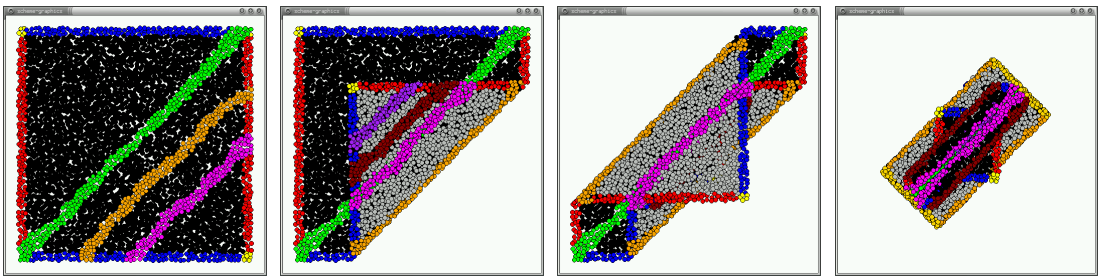

Figure 2: Folding an envelope structure. The constraints on the elements are similar to Figure 1; in addition the elemnts can coordinate to fold the sheet using an actuation model based on epithelial cell morphogenesis.

original axes of the embryo [34]. There are no global coordinates, no external beacons, no globally unique identifiers, no global clock, and the elements are randomly distributed. Yet the elements reliably differentiate to form an inverter pattern. The overall *global* view of this local process is that the elements coordinate to first divide the region into 8 segments and then the elements within each segment differentiate to form an inverter pattern within each segment. Non-repeating circuit patterns could be created by cells differentiating differently based on the segment number. Locally elements use techniques based on cell differentiation theories in the *Drosophila* and sea urchin embryos [47, 26].

In the second example in figure 2, the identically-programmed, flexible elements coordinate to fold the sheet into an envelope-like structure. The constraints on the elements and the initial conditions are similar; one addition is that each cell has an apical-basal polarity. At the local level the elements use similar cell differentiation based mechanisms, however the cells can also coordinate their actuation to fold the sheet in a manner based on folding in epithelial tissues. In addition the new sheet configuration implicitly affects the cell behavior through sensing of contact. Cells can communicate through contact. In both of the examples, the desired goal pattern and shape are specified at an abstract level on a continuous sheet, with no notion of cells and local behavior.

This work presents a novel approach to the the design of self-assembling systems and has many important features.

1. **Global-to-local Compilation:** The cell program is directly compiled from the global shape description. This differs significantly from approaches based on cellular automata where local rules are constructed empirically or evolutionary approaches where the local-to-global relationship is not well-understood. The compilation process confers significant advantages: not only can the desired global goal be described at an abstract level, but results from geometry and paper-folding mathematics can be used to reason about what kinds of shapes can and cannot be self-assembled by the cells.

2. **General-purpose Local Primitives:** Our work has so far revealed a small set of powerful but simple primitives for local organization, based on biological mechanisms. The global shape specification compiles into cell programs based on only five primitives: gradients, neighborhood query, polarity inversion, cell-to-cell contact and flexible folding. These primitives are inspired by biologists' understanding of how pattern and morphology appear in the development of embryos such as the *Drosophila* and sea urchin [34, 26, 35]. Each primitive is simple and general, but can be composed to produce complex prespecified behavior. Complexity comes from the effect of the interactions between the elements and their interactions with the environment.

   For example, a gradient is analogous to the gradient of a chemical concentration from a source. This primitive can be used for measuring distance, ascertaining local direction (tropism), and creating regions. It exploits the spatial locality of communication to make crude geometric inferences. Gradients themselves can be quite complex based on the location of sources, however a cell program for creating and propagating gradients is very simple. Gradients are an extremely robust mechanism in the face of individual cell errors and failures. Gradients are also quite general, and we have used them for topological pattern formation language [8], the design of ad-hoc networks for distributed sensors, and discovery of global coordinates without the use of external beacons [33, 31, 9].

3. **Robustness:** The cell programs are robust, without relying on regular cell placement, global coordinates, or synchronous operation, and can tolerate a small amount of random cell death.

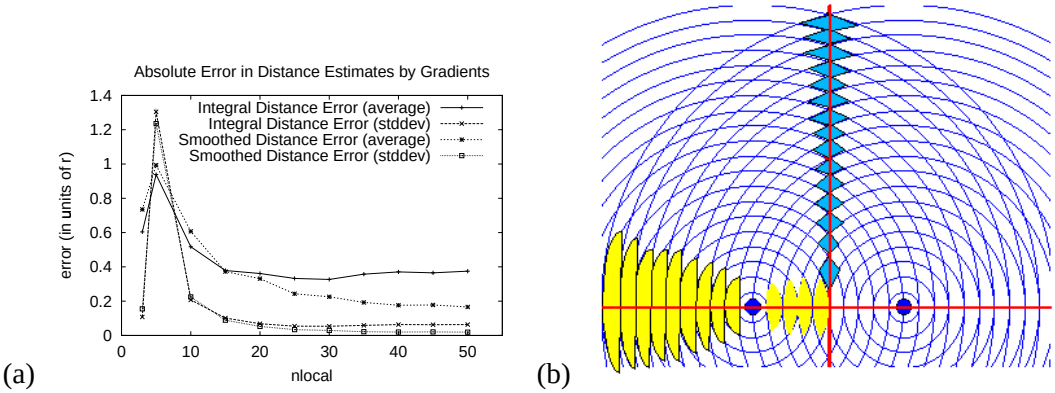

Figure 3: (a) Robustness of gradients as a function of the average number of neighbors of an individual cell (nlocal). The graph shows that an expected average of 15 is sufficient to produce small errors in gradients, especially in combination with local averaging (smoothing). (b) When multiple gradients are composed the error varies with respect to the sources of the gradients. The error is minimum along the bisection. This technique allows us to predict the error produced by the composition of gradients as well as design algorithms that take advantage of the low error regions.

Instead, robustness is achieved by depending on large and dense cell populations, using average behavior rather than individual behavior, trading off precision for reliability, and avoiding any centralized control. For example, there are no global coordinates: instead cells "discover" positional information as and when needed, using primitives inspired by the early segmentation of the *Drosophila* embryo. We have analyzed the behavior of the above primitives, through a combination of theoretical and experimental analysis that leverages traditional techniques from distributed computing, probabilistic analysis, and graph theory, as well as results from packet radio networks and planar geometry. Nagpal has shown that an average local neighborhood of 15–20 cells is sufficient to reliably control the self-assembly of shapes and achieve geometric faithfulness on randomly distributed cells. We can not only analyze the behavior of gradients but also the effect of composing gradients (figure 3). These properties are extremely attractive from an engineering perspective and the concepts are likely to have general applicability to programming large self-reconfiguring robots and smart matter applications.

4. **Analogies to Biology:** The language provides many insights into the relationship between local behavior and global morphology, that have analogies to biological systems. One of the key goals of this research will be to further explore the relationship between the models provided by this work and similar systems in biology.

   For example, the cell program is scale independent which implies it can create the same shape at many different scales without modification, simply by increasing the size of the cell sheet. Scalability is an extremely attractive feature: not only does the cell program work correctly over varying numbers of cells, but the function (goal) itself scales with the number of cells. This system can also generate many different related shapes without any modification to the cell program. The method by which both scale-independence and related shape are achieved is by recursively subdividing the space relative to the boundary. Each operation (fold) is relative to the current boundary and creates new "boundaries". Thus complexity is generated by recursively applying simple operations, while still remaining relative to the

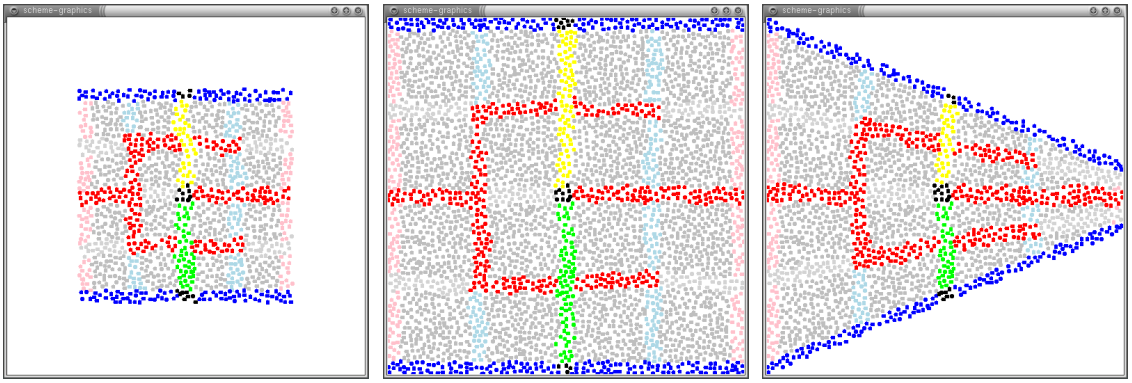

Figure 4: If the same inverter cell program is run on a larger number of elements (same density per unit area and same neighborhood size) then the pattern formation automatically scales to the number of elements. For an asymmetric sheet the pattern scales asymmetrically in a predictable way. This is achieved without any change to the program.

original boundary. This suggests a mechanism for achieving shape transformations in the manner of D'Arcy Thompson's famous examples [42]. In figure 4, the inverter pattern scales not only according to the number of elements but also the shape of the sheet.

This has significant implications for biology: if the cell programs do not differ, then methods such as genetic analysis are not likely to reveal much information. Instead observations of the process of development and comparisons between processes in closely related species may give us better insights into how morphology is created.

B.4 Proposed Research

The goal of the research described in this proposal is twofold: to extend our programming models to new domains, and to investigate the connection of our models to biological processes. We will design new languages that describe the formation of shape by exploiting replication (growth), motility, and deletion (cell death). Just as paper folding is a natural process for shape formation on a sheet, we will need to discover new models to describe shape formation in these new domains.

Our previous work has exhibited several similar traits to biological systems and provided us with insights into how these systems may work. We will exploit these insights to design biological experiments, then identify one or more colleagues in the Biology department and collaborate with them to perform these experiments. Specifically, we will investigate scale independence and morphological differences in closely-related species. We will also investigate the connection between the kinds of failures that we have observed in the simulated morphogenesis processes and those that occur in naturally-occurring birth defects, concentrating on failures of geometry, topology and synchronization.

B.4.1 Programmable Self-assembly Based on Growth, Motility and Cell Death

Our preliminary work has focused on 2D pattern formation and shapes produced through epithelial cell based folding. However this only scratches the surface of what is possible. Biology suggests many different metaphors for constructing complex structure: growth, cell motility, cell death, material deposition, etc. Our goal is to use a similar programming languages approach to the design of

artificial systems that form 3D volumetric structures based on:

1. **Growth:** Growth is fundamental to the creation of structure in embryos, from folding in the chick eye to limb generation [2]. Growth is important not only for creating structure but also for self-repairing structure, such as regeneration of limbs [26].

2. **Limited Motility:** Examples of this are epithelial cell migration for wound repair, neural cell separation in *Drosophila* embryos, and migration of mesenchymal cells during early gastrulation [48, 36, 24, 23]. This plays an important role in achieving robustness and self repair and is often seen in conjunction with cell growth.

3. **Cell Death:** Programmed cell death (Apoptosis) plays an important role in the formation of structures such as the *Drosophila* eye, where the exact hexagonal pattern (neurocrystal) is produced by programmed cell death of intervening cells [26]. In many cases, such as the formation of fingers in embryos, cell death plays a complementary role to growth in removing material between scaffolds [2].

Some of the specific tasks involved are:

1. We will construct detailed computational models and frameworks for cell actuation and sensing behavior, based on current studies of cell growth, motility and apoptosis. Already there are several sources of mathematical and mechanical models for such systems [2, 36, 24, 23].

2. We will extend current primitives to the new sensing/actuation models as well as develop new primitives. There are strong reasons to believe that many of the current primitives will apply in these new regimes. Gradients for example are known to play a role in regulating growth; grafting experiments on cockroach legs suggests that gradients play a role in regulating the length of the leg and also regeneration in severed legs [26]. In many cases sensing and actuation affects the primitives implicitly, for example in the case of sheet folding, cells form new local neighborhoods through contact which allows gradients to seep through multiple layers of the sheet as if it were one fused layer.

3. We will develop robust general-purpose algorithms and general methodologies for the analysis of biologically-inspired algorithms. Special attention will be paid to robust design: principles that do not rely on synchronous operation or regular grids of cells, and which can tolerate cell death. We expect that many of the techniques developed in prior work will be adaptable to these new systems. As in previous work, we expect the analysis to leverage not only traditional distributed and parallel-computing ideas, but also results from computational geometry and other areas.

4. We will develop new high-level languages for the specification of shape. This is an important piece of the research — the choice of shape specification language determines to a large degree the decomposition of the problem. For example, paper folding is a natural process for describing shape formation on a sheet and the axioms provide the decomposition. Choosing the right decomposition will be necessary in order to translate abstract notions of shape to local processes. We will need to discover new models to describe shape formation in these new domains, and this will require cross-disciplinary and non-traditional thinking.

This research will play a complementary approach to other research in self-assembly, such as the work on structural and thermodynamic self-assembly [22, 18, 11], in exploring the space of shape

formation. Structural or thermodynamic self-assembly focuses on the design of materials with properties that cause them to assemble into the desired structure when mixed together. The work takes inspiration from molecular and chemical self-assembly. By contrast our approach is that of "programmed" self-assembly, where the elements contain instructions for their behavior. In nature we see structures formed by both methods, e.g. crystal formation and embryogenesis. Explorations of these areas will give us further idea of which methodology is appropriate for a given problem. So far the work on self-assembly has explored a very limited space. Reconfigurable robot research has focused on the formation of structure through coordination of independent mobile units, which are independent or forced to remained connected [15, 50, 19]. Our previous work uses a new metaphor, that of a flexible sheet, and shows that such a substrate can be very versatile for creating shape. The proposed research will explore new metaphors for creating shape, and extend our knowledge of how to self assemble different structures.

B.4.2 Investigating the Relationship with Biological Systems

Our current understanding of the generation of structure in biological systems is limited. We lack insight into how global structure emerges from local interactions between cells. Our preliminary work has focused on exactly this area, but in the context of computational elements. Our systems exhibit structural traits that have strong parallels to those seen in biological systems. Also, because the relationship between global and local behavior in our systems is well understood, we can use these systems as a model to design biological experiments.

One such trait is scale independence, also known as size invariance, which refers to the appearance of the same structure at a variety of scales. Some examples of this trait include:

1. Embryos of the sea urchin develop normally over ten-fold size variations, and in the case of the hydra, a fragment one hundredth the volume can give rise to a complete animal [47].

2. Species of the *Drosophila* vary over five-fold in size, but the shape and proportion of the body parts is highly conserved, as is their DNA [26].

3. Extremely complex structures such as lungs, kidneys, and the digestive system appear in a wide range of sizes.

Many of these cases suggest (especially in the case of the embryos) that the processes for forming morphology are capable of scaling to different sizes, without modification of the DNA. However biologists currently have very little understanding of how this is achieved at the cell level.

One way in which biologists are attacking this problem is through genetic analysis. But our work has shown that visually dramatic global differences do not necessarily translate to large differences in a cell program, which implies that methods like genetic analysis are not likely to reveal much information. Instead observations of the process of development and comparisons between processes in closely related species may give us better insights into how the morphology is created.

Wolpert proposed the idea of positional information and "balancing" gradients to explain scale independence in the initial patterning of the hydra and sea urchin [47]. One problem with this model is that generating complex scalable patterns requires very complex comparison functions and significant precision in the original coordinate system. Our systems exploit many of the same techniques used by Wolpert, but combine them with the idea of recursive decomposition. This has the effect of simplifying the comparison functions and reducing the precision constraints. Consequently, we are able to produce complex structure from very simple parts.

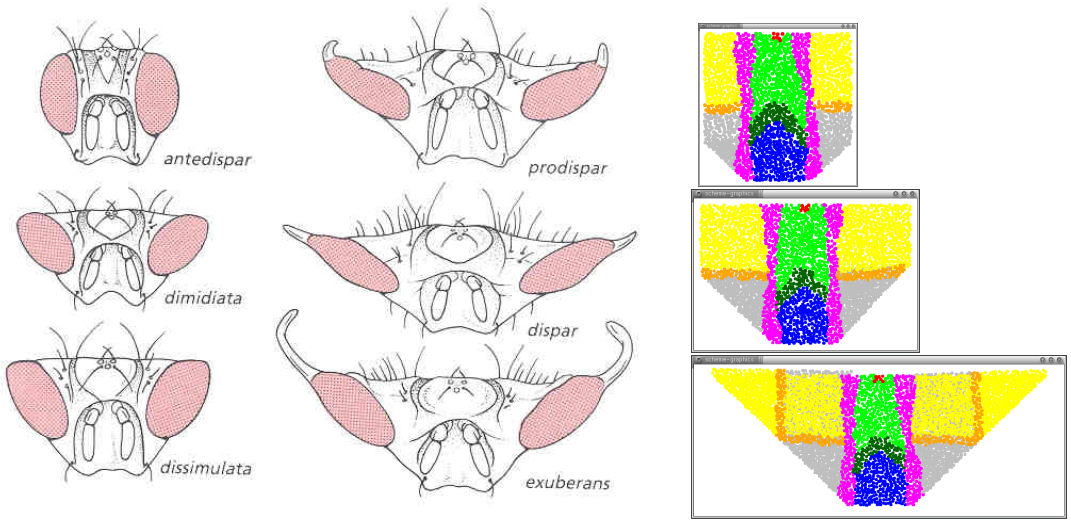

Figure 5: On the left are the heads of closely related species of *Drosophila* (from *The Making of the Fly* [26]). On the right is a caricature of how the *Drosophila* head proportions vary. In each case the regions are created by the same cell program, the scaling occurs automatically as the initial size and shape of the head is varied. Patterns within each region would also scale accordingly. This demonstrates how visually dramatic global differences do not necessarily translate to large differences in a cell program.

.

The behavior of our systems suggests many possible biological experiments. In the *Drosophila melanogaster*, Nusslien-Volhard and Weischaus showed that the head, thorax and abdomen segmentation boundaries are affected by thresholds of the *bicoid* protein generated from source cells (maternally determined) at the anterior pole [34, 26]. However *Drosophila* species occur over wide range of sizes. It is likely that the initial patterning proteins and genes are conserved across many of these species. If one could compare two species of different sizes, one could ask several questions:

1. Are the embryos the same size when initial patterning occurs?
2. If not, do the segments scale proportionally with the embryo size?
3. How does the extent and concentration of the *bicoid* gradient compare in the different species and do segments occur at relative or absolute concentrations of *bicoid*?

Answers to each of these questions would shed light on how initial pattern formation happens and how it relates to theories of scale independence.

Our work also demonstrates that many related shapes can be created without any change in the cell program. Experiments on two closely related Hawaiian *Drosophila* species, with very different looking heads, has shown that there are only a small number of genes, probably 10, that are responsible for the difference in head shape [43]. However there is no understanding of what these "shape genes" control. Figure 5 shows the heads of several related species of *Drosophila* on the left [26]. The proportions of the center region and eyes scale in different ways. On the right is a caricature of the heads, generated by our system. In each case the pattern is generated by the same cell program, the only difference being the shape of the initial sheet. In the program the width

of the center region is proportional to the height of the sheet and scales similarly. By a recursive application of this idea, any pattern formed within one of the major regions scales automatically as one scales the overall sheet. This demonstrates how significantly different shapes can be produced by the same cell program, and suggests that one possible function of these genes may be to determine the growth along the major axes of the head.

Our models provide predictions about specific types of failures in topology, geometry, and synchronization. For example, work by Coore and Nagpal presented two different mechanisms, both based on gradients, that represent fundamentally different strategies of generating structure — the first favors topology while the second favors geometry [8, 32]. As a result these mechanisms behave differently in the presence of failures. At different times during development, either topology or geometry may be more important, and the relative importance is partly revealed by the strategy chosen as well as the ways in which failures occur.

We can draw a parallel between this behavior of our systems and that of cells. Both strategies imply different cell behavior — in one case cell behavior is affected by its local neighborhood, in the other it is strictly autonomous. For instance during gastrulation the gut grows from one end to the other guaranteeing connectivity between the mouth and anus. Similarly the formation of neurons may involve growing towards a source. However during the segmentation of the body into regions, elements may autonomously decide which region they belong to, since connectivity is not an issue. The way in which these systems respond to regional cell failures reveals insights into the original function as well as the extent to which each cell is affected by its neighbors.

We will exploit this parallel behavior by finding examples of natural systems that both exhibit these traits, and are amenable to experimentation and observation of the developmental process. We will collaborate with biologists who are working on such systems, and jointly design and execute these experiments. Based on our previous collaborations and associations with professors at Whitehead Institute and other biology laboratories in the MIT area, we believe we are well placed to establish these collaborations.

Initially we will focus on the drosophila examples presented. There is a significant amount of data available on early *Drosophila* embryogenesis and also on the Hawaiian species of *Drosophila* [16]. In addition, we will explore other species that exhibit similar properties, such as scale independence. There are several well-studied candidate systems, for example, zebrafish, lung formation in the mouse, and sea urchin embryogenesis.[41, 2].

Although this work is in the beginning stages, results in this area have the potential to have a significant impact on our understanding of developmental processes and creation of morphology. Not only will this promote the use of artificial systems for understanding systems level biology, but it will also increase collaboration between computer science and biology in this area, where focus has traditionally been very narrow.

B.5 Long-term Goal: Programming In Vitro Computation

The work presented in this proposal is part of a larger effort, called Amorphous Computing [1], that is focused both on programming and building new computational environments and substrates, which will embed computation into the physical world. We are also engineering computation within living *e. Coli* cells by implementing engineered genetic regulatory networks [25, 45, 44]. Knight and Weiss have successfully engineered several gates within *e. Coli* cells, as well as communication between cells based on autoinducers. Ultimately, however, we believe that the real advances will not be from single cells but from programming ensembles of cells. The research proposed here will be key to our ability to control the behavior of aggregates. Many of the primitives proposed naturally translate to cell capabilities. An example application of such a system would be patterning

in living biofilms, as an approach to nanoscale engineering and molecular electronics. The ability to organize cells into precise patterns and to cause cells to secrete chemical components could be the foundation for the engineering construction of complex extracellular structures and precise control of fabrication at the sub-nanometer level to replace lithography.

B.6 Research, Education and Societal Impact

In addition to our research goals we are also deeply committed to education, interdisciplinary research, and undergraduate research experience. We are also committed to the wide and open dissemination of our research ideas, educational content, and software.

One of the key requirements for promoting this type of research is the training and education of students to think across discipline boundaries. Our group has a long history of supporting interdisciplinary thought and also of advocating interdisciplinary education for students. Understanding how to make connections between different bodies of knowledge can lead to significant insights. We are therefore committed to training graduate students in interdisciplinary research. We also believe that the undergraduate research experience is vital to the dissemination of new visions of computing, as well as an integral part of undergraduate education; a part of the grant budget is earmarked for the support of two undergraduate students each year.

Our research agenda will foster collaborations between computer scientists and biologists, not just at the level of tools but at the level of concepts. We strongly believe that both sides have much to gain from this collaboration.

References

[1] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, R. Nagpal, E. Rauch, G. Sussman, and R. Weiss. Amorphous computing. *Communications of the ACM*, 43(5), May 2000.

[2] J. Bard. *Morphogenesis.* Cambridge University Press, U.K., 1990.

[3] R. Beckers, O. Holland, and J. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. *Artificial Life*, 4, 1994.

[4] A. Berlin. *Towards Intelligent Structures: Active Control of Buckling.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, May 1994.

[5] A. Berlin and Gilbert. Collaboration and coordination in massively distributed smart matter. *Workshop on Amorphous Computing*, September 1999.

[6] W. Butera and V. Bove. Literally embedded processors. In *Proc. of SPIE Media Processors*, 2001.

[7] P. Cheung, A. Berlin, D. Biegelsen, and W. Jackson. Batch fabrication of pneumatic valve arrays by combining mems with printed circuit board technology. In *Proc. of the Symposium on Micro-Mechanical Systems, ASME Intl. Mech. Engineering Congress and Exhibition*, Dallas, TX, November 1997.

[8] D. Coore. *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, February 1999.

[9] D. Coore, R. Nagpal, and R. Weiss. Paradigms for structure in an amorphous computer. AI Memo 1614, MIT Artificial Intelligence Lab, 1997.

[10] J. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The dynamics of collective sorting: robot-like ants and ant-like robots. *Simulation of Adaptive Behavior: from animals to animats*, 1990.

[11] R. D'Souza. *Macroscopic order from reversible and stochastic lattice growth rules.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, September 1999.

[12] C. Folk and C-M. Ho. Micro-actuators for control of delta wing with sharp leading edge. In *39th AIAA Aerospace Sciences Meeting and Exhibit*, January 2001.

[13] S. Forrest and S. Hofmeyr. Immunology as information processing. *In Design Principles for the Immune System and Other Distributed Autonomous Systems*, 2000.

[14] S. Forrest and M. Mitchell. What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. *Machine Learning*, 13:285–319, 1993.

[15] T. Fukuda and S. Nakagawa. Approach to the dynamically reconfigurable robot systems. *Journal of Intelligent and Robotic Systems*, 1:55–72, 1988.

[1]Papers by the authors and members of the Amorphous Computing Group are available online at `http://www.swiss.ai.mit.edu/projects/amorphous/paperlisting.html`

[16] D. Grimaldi. Phylogenetics and taxonomy of *zygotherica* (diptera:drosophila). *Bulletin of the Museum of American Natural History*, 186:103–268, 1987.

[17] S. Hall, E. Crawley, J. Howe, and B. Ward. A hierarchic control architecture for intelligent structures. *Journal of Guidance, Control and Dynamics*, 14(3):503–512, 1991.

[18] J. Heath, P. Kuekes, G. Snider, and R. Williams. A defect-tolerant computer architecture: Opportunities for nanotechnology. *Science*, 280:1716–1721, 1998.

[19] T. Hogg, H. Bojinov, and A. Casal. Multiagent control of self-reconfigurable robots. In *4th International Conference on Multi-Agent Systems (ICMAS)*, July 2000.

[20] H. Huzita and B. Scimemi. The algebra of paper-folding. In *First International Meeting of Origami Science and Technology*, Ferrara, Italy, 1989.

[21] D. Ingber. The architecture of life. *Scientific American*, January 1998.

[22] R. Jackman, S. Brittain, A. Adams, M. Prentiss, and G. Whitesides. Design and fabrication of topologically complex, three-dimensional microstructures. *Science*, 280:2089–2091, 1998.

[23] A. Jacobson and R. Gordon. Changes in the shape of the developing vertebrate nervous system analyzed experimentally, mathematically and by computer simulation. *Journal of Experimental Zoology*, 197:191–246, 1976.

[24] A. Jacobson, G. Oster, G. Odell, and L. Cheng. Neurulation and the cortical tractor model for epithelial folding. *Journal of Embryology and Experimental Morphogenesis*, 96:19–49, 1986.

[25] T. Knight and G. Sussman. Cellular gate technology. In *Proceedings of the First International Conference on Unconventional Models of Computation (UMC98)*, January 1998.

[26] P. A. Lawrence. *The Making of a Fly: the Genetics of Animal Design*. Blackwell Science Ltd, Oxford, U.K., 1992.

[27] F. Leighton. *Introduction to Parallel Algorithms and Architectures*. Morgan Kaufmann, San Mateo, CA, 1992.

[28] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, Wonderland, 1996.

[29] M. Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16((2-4)):321–331, December 1995.

[30] M. Mitchell, J. Holland, and S. Forrest. When will a genetic algorithm outperform hill climbing? *Advances in Neural Information Processing Systems*, pages 285–319, 1994.

[31] R. Nagpal. Organizing a global coordinate system from local information on an amorphous computer. AI Memo 1666, MIT Artificial Intelligence Lab, 1999.

[32] R. Nagpal. *Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, June 2001.

[33] R. Nagpal and D. Coore. An algorithm for group formation in an amorphous computer. In *Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, October 1998.

[34] C. Nusslein-Volhard. Gradients that organize embryo development. *Scientific American*, August 1996.

[35] G. Odell, G. Oster, P. Alberch, and B. Burnside. The mechanical basis of morphogenesis: 1. epithelial folding and invagination. *Developmental Biology*, 85:446–462, 1981.

[36] G. Oster, J. Murray, and A Harris. Mechanical aspects of mesenchymal morphogenesis. *Journal of Embryology and Experimental Morphogenesis*, 78:83–125, 1983.

[37] A. Pamecha, I. Ebert-Uphoff, and G. S. Chirikjian. Useful metrics for modular robot planning. *IEEE Transactions on Robotics and Automation*, 13(4), August 1997.

[38] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5), May 1999.

[39] M. Resnick. *Turtles, Termites and Traffic Jams*. MIT Press, Cambridge, MA, 1994.

[40] L. Segel and R. Lev Bar-Or. On the role of feedback in promoting conflicting goals of the adaptive immune system. *Journal of Immunology*, 163:1342–1349, 1999.

[41] J. Slack. *From Egg to Embryo, second edition.* Cambridge University Press, U.K., 1991.

[42] D Arcy Thompson. *On Growth and Form, abridged edition.* Cambridge University Press, U.K., 1961.

[43] F. Val. Genetic analysis of the morphological differences between two interfertile species of hawaiin drosophila. *Evolution*, 31:611–629, September 1977.

[44] R. Weiss, G. Homsy, and T. Knight. Toward in vivo digital circuits. In *Dimacs Workshop on Evolution as Computation*, January 1999.

[45] R. Weiss, G. Homsy, and R. Nagpal. Programming biological cells. In *8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '98), Wild and Crazy Ideas Session*, October 1998.

[46] B. C. Williams and P. P. Nayak. Immobile robots: AI in the new millennium. *AI Magazine*, 1996.

[47] L. Wolpert. Positional information and the spatial pattern of cellular differentiation. *Journal of Theoretical Biology*, 25:1–47, 1969.

[48] L. Wolpert. *Principles of Development.* Oxford University Press, U.K., 1998.

[49] M. Yim. *Locomotion With A Unit-Modular Reconfigurable Robot.* PhD thesis, Stanford, Department of Computer Science, 1994.

[50] M. Yim, J. Lamping, E. Mao, and J. G. Chase. Rhombic dodecahedron shape for self-assembling robots. Spl techreport p9710777, Xerox PARC, 2000.

C Biographical Sketches

Gerald Jay Sussman

a. Education:

School	*Degree*	*Date*
MIT	S.B. (Mathematics)	1968
MIT	Ph.D. (Mathematics)	1973

b. Professional Appointments:
All at the Massachusetts Institute of Technology

1991–present	Matsushita Professor of Electrical Engineering
1984–1991	Full Professor of Electrical Engineering
1977–1984	Associate Professor of Electrical Engineering
1973–1977	Assistant Professor of Electrical Engineering

Awards and Honors:

2000	Elected Member of the National Academy of Engineering
1996	Elected Fellow of the American Academy of Arts and Sciences
1994	Elected Founding Fellow of the ACM
1992	Amar G. Bose award for Engineering Education
1991	ACM Karl Karlstrom Outstanding Educator Award
1990	Elected Founding Fellow of the AAAI
1987	Hertzberg Lecturer – Carlton University, Canada.
1981	IJCAI Computers and Thought Lecturer

c (i) Selected publications relevant to this proposal:

1. "Amorphous Computing," Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F. Knight Jr., Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss, in *Communications of the ACM*, **43**, 5, May 2000.

2. "Cellular Gate Technology," Thomas F. Knight and Gerald Jay Sussman, *Proc. UMC98, First International Conference on Unconventional Models of Computation*, Auckland, NZ, January 1998.

3. *Structure and Interpretation of Computer Programs*, Harold Abelson, Gerald Jay Sussman and Julie Sussman, MIT Press and McGraw-Hill, 1985, (published translations in French, Japanese, and German). Second Edition, 1996.

4. "Intelligence in Scientific Computing," Harold Abelson, M. Eisenberg, M. Halfant, J. Katzenelson, E. Sacks, G.J. Sussman, J. Wisdom, K. Yip, *Communications of the ACM*, **32**, no. 5, May 1989.

5. "A Computational Model for the Acquisition and Use of Phonological Knowledge," Kenneth Yip and Gerald Jay Sussman, *Proc. of National Conference on Artificial Intelligence*, July 1997. Also MIT Artificial Intelligence Memo 1575, March 1996.

c (ii) Selected other publications:

1. *Structure and Interpretation of Classical Mechanics*, Gerald Jay Sussman and Jack Wisdom, with Meinhard Mayer, MIT Press, 2001.

2. "The first report on Scheme revisited," Gerald Jay Sussman and Guy L. Steele Jr., *Higher-Order and Symbolic Computation*, **11**, No.4, pp. 399-404, 1998.

3. "Spin-induced Orbital Precession and its Modulation of the Gravitational Waveforms from Merging Binaries," T.A. Apostolatos, C. Cutler, G.J. Sussman, and K.S. Thorne, *Phys. Rev. D.*, 15 June 1994.

4. "Increasing the Compressive Strength of a Column via Active Control", A. Berlin and G.J. Sussman, *Proceedings of the Third International Conference on Adaptive Structures*, Oct 1992.

5. "Chaotic Evolution of the Solar System," Gerald Jay Sussman and Jack Wisdom, *Science*, **257**, 3 July 1992.

d. Synergistic Activities:

Sussman has been deeply involved in education, Artificial Intelligence and interdisciplinary research at MIT. He is a coauthor of the introductory Computer Science textbook used at MIT, which has been translated into French, German, Chinese, and Japanese. As a result of his contributions to computer science education he recieved the ACM's Karl Karlstrom Oustanding Educator Award and the Amar G. Bose teaching award. Sussman and his former student, Guy L. Steele Jr., invented the Scheme programming language in 1975. Recently Sussman developed an innovative course on classical mechanics and published a textbook, with Prof. Jack Wisdom, that uses programming in Scheme as a tool for teaching and understanding mechanics. Sussman was the principal designer of the Digital Orrery. This machine was used to discover numerical evidence of chaotic motions in outer planets; it is now retired at the Smithsonian. Sussman is a Member of the National Academy of Engineering, a Founding Fellow of the Association for Computing Machinery (ACM), a Fellow of the American Academy of Arts and Sciences (AAAS), and a Founding Fellow of the American Association for Artificial Intelligence (AAAI). He is also a senior member of the IEEE.

e. Collaborators and Other Affiliations:

(i) Collaborators: Prof Harold Abelson (MIT), Don Allen (MIT), Dr. Drew Andy (Molecular Sciences Institute), Dr Roger Brent (Molecular Sciences Institute), Dr Thomas F. Knight Jr. (MIT), Prof Jack Wisdom (MIT).

(ii) PhD thesis supervisors: Prof Marvin L. Minsky and Seymour A. Papert.

(ii) In the past five years Sussman has supervised the PhD theses of Panayotis Skordos, Thanos Siapas, Brian LaMacchia, Daniel Coore, Radhika Nagpal, and Ron Weiss. He has supervised the PhD theses of 32 students.

Radhika Nagpal

a. Education:

School	*Degree*	*Date*
MIT	S.B. and S.M. (Computer Science)	June 1994
MIT	Ph.D. (Computer Science)	June 2001

b. Professional Appointments:

present	Postdoctoral Lecturer, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
1994-1995	Member of Technical Staff, Bell Laboratories, Murray Hill, NJ.

Awards:

AT&T/Lucent Bell Labs Graduate Research Program for Women (GRPW) Fellowship Recipient, 1994.

c (i) Selected publications relevant to this proposal:

1. "Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics", Radhika Nagpal, PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technlogy, June 2001.

2. "Amorphous Computing," Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F. Knight Jr., Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss, in *Communications of the ACM*, **43**, 5, May 2000.

3. "Organizing a Global Coordinate System from Local Information on an Amorphous Computer", R. Nagpal, MIT Artificial Intelligence Memo 1666, 1999.

4. "An Algorithm for Group Formation in an Amorphous Computer", R. Nagpal, D. Coore, Proceedings of the 10th International Conference on Parallel and Distributed Computing Systems (PDCS'98), 1998.

5. "Programming Biological Cells", R. Weiss, R. Nagpal, G. Homsy, 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '98), Wild and Crazy Ideas Session, 1998.

c (ii) Selected other publications:

1. "Implementing Browsing Operations in MPEG", R. Nagpal, H. Kanakia, 7th International Workshop on Packet Video, Australia 1995.

2. "Implementing Single-cycle Store Instructions in Write-through, Write-back and Set-associative Caches", R. Nagpal, AT&T Technical Memo, 1127-950117-01TM, 1994

d. Synergistic Activities:

Nagpal has supervised many undergraduate students in the Undergraduate Reaserch Program at MIT (UROP). She has also been a long time mentor in the KEYs (Keys to Empowering Youth) program for encouraging high-school girls in Math and Science, through programs at MIT.

e. Collaborators and Other Affiliations:

(i) Collaborators: Prof Harold Abelson (MIT), Dr Thomas F. Knight Jr. (MIT), Ron Weiss (MIT).

(ii) PhD thesis supervisors: Prof Gerald Jay Sussman and Prof Harold Abelson.

Chris Hanson

a. Education:

School	*Degree*	*Date*
MIT	S.B. (Computer Science)	1980

b. Professional Appointments:

1993–present	Principal Research Scientist, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
1982–1993	Research Scientist, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
1980–1982	Software Engineer, Data Translation Inc.

c (i) Selected publications relevant to this proposal:

1. "Amorphous Computing," Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F. Knight Jr., Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss, in *Communications of the ACM*, **43**, 5, May 2000.

2. "The MIT Scheme programming environment", Chris Hanson, Guillermo Rozas, James Miller, et. al., software system under development from 1982 to present, `http://www.swiss.ai.mit.edu/proje`

c (ii) Selected other publications:

1. "The revised^4 report on the algorithmic language Scheme", J. Rees and W. Clinger, eds., in *Lisp Pointers* 4(3), ACM, 1991.

2. *IEEE Standard for the Scheme Programming Language—IEEE Std 1178-1990*, David Bartley, Chris Hanson, and James Miller, eds., IEEE Computer Society, December, 1990.

3. "Efficient Stack Allocation for Tail-Recursive Languages," Chris Hanson, in *Proceedings, ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.

4. "The Scheme-81 Architecture—System and Chip," John Batali, Edmund Goodhue, Chris Hanson, Howie Shrobe, Richard M. Stallman, and Gerald Jay Sussman, *Proceedings, Conference on Advanced Research in VLSI*, Cambridge, MA, January 1982.

d. Synergistic Activities:

Hanson is the principal author of the MIT Scheme, the programming environment used in MIT's introductory programming course from 1982 to the present. More recently, MIT Scheme has been used as the substrate for an innovative course in classical mechanics developed by Sussman and Prof. Jack Wisdom. This free software is distributed on the internet and is used both for course work and commercial software development.

e. Collaborators and Other Affiliations:

(i) Collaborators: Prof. Harold Abelson (MIT), Don Allen (MIT), Prof. Gerald Jay Sussman (MIT).