Models and Mechanisms for Artificial Morphogenesis

Bruce J. MacLennan

Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.cs.utk.edu/∼mclennan

Abstract. Embryological development provides an inspiring example of the creation of complex hierarchical structures by self-organization. Likewise, biological metamorphosis shows how these complex systems can radically restructure themselves. Our research investigates these principles and their application to artificial systems in order to create intricately structured systems that are ordered from the nanoscale up to the macroscale. However these processes depend on mutually interdependent unfoldings of an information process and of the "body" in which it is occurring. Such *embodied computation* provides challenges as well as opportunities, and in order to fulfill its promise, we need both formal and informal models for conceptualizing, designing, and reasoning about embodied computation. This paper presents a preliminary design for one such model especially oriented toward artificial morphogenesis.

Keywords: Algorithmic assembly, embodied computation, embodiment, embryological development, metamorphosis, morphogenesis, nanotechnology, post-Moore's Law computing, reconfigurable systems, self-assembly, self-organization.

1 Background

1.1 Embodied Computation

Pfeifer, Lungarella, and Iida [29, p. 1088] provide a concise definition of *embodiment*: "the interplay of information and physical processes." Hence, *embodied computation* may be defined as computation whose physical realization is directly involved in the computational process or its goals. It includes computational processes that directly exploit physical processes for computational ends and those processes in which information representation and computation are implicit in the physics of the system and its environment, which effectively represent themselves. It also includes computational processes in which the intended effects of the computation include the growth, assembly, development, transformation, reconfiguration, or disassembly of the physical system embodying the computation. Embodied computation is based on some of the insights from *embodied cognition* and *embodied artificial intelligence* [6,7,13,28], but extends them to all computation [21,25].

F. Peper et al. (Eds.): IWNC 2009, PICT 2, pp. 23–33, 2010.

The most common model of computation (binary digital logic) is far removed from the physical processes by which it is implemented, and this has facilitated a beneficial independence of computer design from device technology. Thus our technological investment in computer design has been preserved through several generations of device technology (from relays to ICs). However, as in the coming decades we enter the era of post-Moore's Law computing, increasing the density and speed of computation will require a greater assimilation between computational and physical processes [21,22,24,25]. In part this will be accomplished by developing new physical systems and processes, but the other half of the equation is to develop new models of computation that are closer to the laws of physics. The challenge is to identify models that are sufficiently low level to be readily implementable and sufficiently high level to be relatively independent of particular implementation technologies.

One of the advantages of embodied computing is that many computational processes are performed by physical systems "for free." For example, neural networks often make use of universal approximation theorems based on linear combinations and simple sigmoidal functions [12, pp. 208–94]. However, sigmoidal behavior is a common consequence of many physical processes, in which some resource is exhausted, and therefore sigmoids are available "for free" without additional computation. Similarly, negative feedback often arises from the natural degradation or dissipation of physical substances, and so these processes are directly available for embodied computation. Similarly, stochastic effects, which are unavoidable in many physical systems, especially at the nanoscale, can be exploited profitably in algorithms such as stochastic resonance [4] and simulated annealing [14]. Finally, in embodied computation the concurrency typical of physical, chemical, and biological processes can be directly exploited to achieve parallelism "for free." There are many other examples.

Typically, the structure of a computational system governs its function, that is, the computation it performs. Conversely, in *algorithmic assembly* a computational process governs the assembly of some physical structure [3,15,16,30,31]. However, some embodied systems integrate these two processes. For example, in embryological morphogenesis, the physical structure of the embryo governs the very computational processes that create the physical structure of the embryo. Structure governs function, and function creates structure.

Further, since embodied computation systems are potentially capable of modifying their own structure, they can be naturally adaptive. Beyond this, they may be "radically reconfigurable," that is, able to reorganize their physical structure to adapt to changing circumstances and objectives [18,19,20,23]. A related, but very important, property is self-repair, since acceptable configurations often can be defined as stationary states to which the system reconfigures after damage. Finally, embodied computation systems can be designed for self-destruction, which is especially important for nanoscale systems.

We are used to programming in an idealized world of perfect logic, independent of physical realization, and therefore embodied computing presents unfamiliar challenges, for we have to pay more attention to the physical realization of an embodied computation and its environment. However, embodied computation is not simply applied physics, for although embodied computation makes more direct use of physical processes than does conventional computation, its focus is on processes that are relatively independent of specific physical systems, so that they can be applied in a variety of physical substrates. For example, reaction-diffusion systems can be instantiated in a wide variety of media [1]. Further, since natural computation is embodied, we often can look to natural systems to get ideas for implementing artificial embodied computing systems.

1.2 Artificial Morphogenesis

Our research is currently focused on a particular application of embodied computation that is of central importance in nanotechnology: artificial morphogenesis [22]. While a variety of self-assembly processes have been applied to nanomaterials, these have not solved the problem of assembling systems that are hierarchically structured from the nanoscale up to the macroscale. Some of the issues are being addressed in the context of research in programmable matter and reconfigurable robotics, but we believe that these efforts would be improved by a more morphogenetic approach, for embryological morphogenesis is the best example that we have of successful self-assembly of physical systems structured from the nanoscale up to the macroscale.

Morphogenesis has a number of special characteristics that distinguish it from most other self-organizing processes, and we believe that these characteristics will be important in embodied computation. For example, we commonly think of computation as taking place in a fixed substrate, and many self-assembly processes also assume a fixed substrate or matrix in which agents move. In morphogenesis, in contrast, the computational medium is assembled by the computational process, as a zygote develops into a hollow blastula and then into a more complex structure of tissues, which govern the information and control processes in the medium. Generally speaking, in nature self-organization proceeds without the benefit of fixed, predefined reference frames and coordinate systems, which is one source of the robustness of these processes.

In morphogenesis, *tissues* (groups of cells with a common function) form and reform under control of their inherent self-organizing processes. We think of the embryo as solid, but most of the tissues are elastic, at least during development, and elastic properties influence the forms that develop [35, ch. 6]. In other cases, tissues behave more like viscous fluids, perhaps percolating through a more solid matrix, and this fluid motion is essential in cell migration [5][9, pp. 92–4][34]. Non-cellular substances, such as morphogens and other signaling chemicals diffuse like gases through anisotropic media, but cells also exhibit facilitated diffusion [9, pp. 13–15, 156, 252]. In many cases, tissues occupy a middle ground, with viscoelastic properties [9, pp. 21–2, 133]. In general terms, morphogenesis takes place in the relatively unexplored realm of *soft matter* [8][9, p. 2], and our theories of embodied computation, at least as applied to morphogenesis, need to take account of its characteristics.

Developmental biologists have identified about twenty fundamental processes involved in morphogenesis [9, pp. 158–9][32]. If indeed, these processes are sufficient to produce complex, hierarchically-structured systems, such as vertebrate organisms, then they define an agenda for embodied computation applied to artificial morphogenesis.

2 Requirements for a Formalism

The goal of our research is to develop and evaluate formal methods for embodied computation oriented toward artificial morphogenesis. This implies several requirements.

First, the goals of embodied computation are best served by a continuous perspective, since the laws of physics are primarily continuous. This is especially true in our intended application area, morphogenesis, for tissues and their environments are naturally treated as continua (e.g., epithelia, mesenchyme, blood).

Nevertheless, the two perspectives — the discrete and the continuous — are complementary. In many applications of embodied computation, especially when the computational process is implemented by very large numbers of computational elements, we will want to be able to move fluently between the two perspectives. This is especially the case in morphogenesis, where in the early stages of development we are faced with discrete phenomena — 1, 2, 4, 8, etc. cells with specific shapes and arrangements — whereas in later stages (when there are more than $\sim 10\,000$ cells) it is more convenient to treat the cell masses as viscoelastic tissues and apply continuum mechanics [9]. Therefore a formalism for embodied computation should support a systematic ambiguity between discrete and continuous models. Thus, the formalism should support complementarity by treating bodies, tissues, and other macroscopic masses as comprising an indefinitely large number of *elements*, which we interpret ambiguously as infinitesimal points in a continuum or as members of a finite, discrete set of units.

To maintain these complementary interpretations, the formalism should treat the elements as having a small, indeterminate size, and therefore the formalism should describe the properties of elements in terms of *intensive quantities*, such as number density and mass density, which do not depend on size, in preference to *extensive quantities*, such as volume and mass, which do.

The formalism should be suitable for describing processes in all kinds of media, including solids, liquids, gases, and physical fields. Further, since viscoelastic media are important in morphogenesis (for example, mesenchyme is best characterized as viscoelastic [9, p. 98][10]), we expect to be dealing with materials of differing viscosity and elasticity. Finally, some of these materials may be anisotropic (e.g., epithelium), and our formalism must accommodate that possibility.

The formalism must be capable of describing active elements, such as living cells and microrobots, as well as passive elements, such as diffusing chemicals, fluid media, and solid substrates. In particular, the formalism should be applicable to living agents as well as to nonliving ones, for embryological morphogenesis, which is the inspiration for this technology, is based on living cells, and we also want to address artificial morphogenetic processes based on genetically engineered organisms and other products of synthetic biology.

Energetic issues are critical to embodied computational systems, which must be maintained in a nonequilibrium state either for a definite duration or indefinitely. Therefore the formalism needs to be able to describe and define the flow of matter and energy through the system. Dissipation of energy is especially critical for microscopic nonequilibrium systems, and should be addressed by the formalism. Similarly embodied computation must address the disposal of other waste products, such as unrecyclable chemical reaction products.

Descriptions of morphogenetic and developmental processes in terms of partial differential equations are naturally expressed relative to a fixed, external three-dimensional reference frame. There are two problems with this. First, the natural reference frame is the developing body itself, which might not have any significant relationship to a fixed frame. Second, since active elements (such as migrating cells) are responding to their local environments, it it natural to describe their behavior in terms of their intrinsic coordinates or their immediately local frame. Therefore tensors seem to be the best way to describe the properties and behaviors of elements.

Biologists often find it convenient to express regulatory networks qualitatively, using *influence models* to indicate that one gene product enhances or represses the expression of another gene [36]. These qualitative regulatory relationships are an important tool for conceptualizing control processes in which the quantitative relationships are unknown or are not considered critical. Therefore, the formalism should permit the expression of qualitative control relationships.

In embodied computation, especially as it pushes towards the nanoscale, noise, error, uncertainty, defects, faults, and other sources of indeterminacy and unpredictability are unavoidable [17]. Therefore a formalism for embodied computation should be oriented toward the description of stochastic processes. Further, since many self-organizing processes depend on stochasticity and sampling effects for symmetry breaking, the formalism should admit them.

3 Approach

Having outlined the requirements for a formalism for embodied computation, we present a preliminary definition.

Substances. One of the central concepts in the proposed formalism is a *substance*, which refers to a class of phenomenological continua with similar properties, especially from a computational perspective (e.g., cohesion, viscosity, behavioral repertoire). They are defined by (perhaps fuzzy) ranges of parameter values, ratios of parameters, etc. Substances are naturally organized in a hierarchy, from the most general classes (e.g., solid, liquid, gas) to specific physical substances (e.g., liquid water, oxygen gas, mesenchyme, a specific extracellular matrix). The more generic classes are more useful from a computational perspective, since they have the potential of being realized by a greater variety of specific substances. For example, for the purposes of embodied computation it may be sufficient that a substance diffuse and degrade at certain relative rates, and be producible and detectable by other particles, but the choice of specific substance might be otherwise unconstrained.

Bodies. In object-oriented programming the instances (members) of classes are discrete atomic objects, which frequently act as software agents. In our formalism for embodied computing, in contrast, instances of substances are called *bodies* or *tissues*, which are phenomenological continua of elements (discrete *particles* or infinitesimal *material points*). Several bodies may occupy the same region of space and interact with each other. For example, the same region may be occupied by a volume of diffusing chemical and by a mass of cells following the chemical gradient. The kind and degree of interpenetrability possible, as well as other interactions, are determined by the definitions of the substances. Some bodies may be quite diffuse (e.g., disconnected cells migrating through mesenchyme).

Mathematically, a body is defined to be an indefinitely large set $\mathcal{B}$ of elements (particles or material points). At any given time t each element $P \in \mathcal{B}$ has a location in a region of three-dimensional Euclidean space $\mathcal{E}$ defined by a vector $\mathbf{p} = C_t(P)$, where C_t is a continuous function that maps $\mathcal{B}$ into this region. As in continuum mechanics, C_t defines the *configuration* of $\mathcal{B}$ at time t, and therefore C_t reflects the *deformation* of the body as a consequence of its own internal dynamics and its interaction with other bodies.

An embodied computation system comprises a finite and fixed number of bodies (or tissues), each having properties that allow it to be classified as one substance or another. The behavioral dynamics of these bodies, in mutual interaction, defines the dynamics of the embodied computation system, but it must be prepared in some appropriate initial state. This is accomplished by specifying that a particular body of a defined substance occupies a specified region of Euclidean space and by specifying particular values for the variable properties of the substance throughout that region. While the full generality of mathematics may be used to define the initial bodies, we are most interested in physically feasible preparations, but this depends on the specifics of the embodied computation system.

Elements. A substance is defined in terms of the properties and behaviors of its elements, and so we need to consider how they may be defined consistently with the requirements of Sec. 2. For most purposes the formalism makes use of *material* (*Lagrangian, reference*) descriptions of these properties and behaviors, rather than a *spatial* (*Eulerian*) reference frame. This means that we consider a physical quantity Q as a time-varying function of a fixed particle $P \in \mathcal{B}$ as it moves through space, $Q(P,t)$, rather than as a time-varying property $q(\mathbf{p},t)$ at a particular fixed location in space, $\mathbf{p} \in \mathcal{E}$. In effect, the use of material variables in preference to spatial variables is a particle-centered description, and is a more object-oriented or agent-based way of thinking, in that we can think of the particles as carrying their own properties with them and behaving like well-defined entities.

As explained in Sec. 2, in order to maintain independence of the size of the elements, so far as possible all quantities are *intensive.* This is one of the characteristics that distinguishes this model of embodied computing from ordinary mathematical descriptions of physical phenomena. For example, instead of dealing with N, the number of particles corresponding to an element, we deal with its *number density* n, the number of particles per unit volume. Note that if the number density becomes very small, then the dynamics will be subject to small sample effects, which are often important in self-organization.

The requirement for complementary perspectives implies that the properties of elements be treated as bulk quantities, that is, as the collective properties of an indeterminate ensemble of "units" (e.g., cells or molecules). This creates special requirements when the units constituting an element might have differing values for an attribute. For example, cells, molecules, and microrobots have an orientation, which is often critical to their collective behavior. In some cases all the units will have the same orientation, in which case the orientation can be treated as an ordinary intensive property of the element. In most cases, however, we must allow for the fact that the units may have differing orientations (even if a consequence only of defects or thermal agitation). Therefore, we have to consider the *distribution* of orientations; each orientation (represented by a unit vector $\mathbf{u}$) has a corresponding probability density $\Pr\{\mathbf{u}\}$. If n is the number density of an element, then $n \Pr\{\mathbf{u}\}$ is the number density of units with orientation $\mathbf{u}$. Equivalently, the orientation may be interpreted as a vector-valued random variable, $\mathbf{U}$.

Morphogenesis in development sometimes depends on cell shape; for example, a change from a cylindrical shape to a truncated cone may cause a flat tissue to fold [9, pp. 113–16][27]. Generally a shape can be expressed as a vector or matrix of essential parameters that define relevant aspects of the shape. Two complexities arise in the expression of shape in this formalism for embodied computing. First, the shape must be expressed in a coordinate-independent way, which means that we are dealing with a *shape tensor.* Second, since an element represents an indefinite ensemble of units, shape must be treated as a probability distribution defined over shape tensors, or as a tensor-valued random variable.

Behavior. The behavior of the elements (particles, material points) of a body is defined by rules that describe the temporal change of various quantities, primarily intensive tensor quantities. Such changes might be expressed in continuous time, by ordinary differential equations, e.g., $\mathrm{D}_t X = F(X, Y)$, or in discrete time by finite difference equations, e.g, $\Delta_t X = F(X, Y)$. Generally, these would be *substantial* or *material* (as opposed to *spatial*) time derivatives, that is, derivatives evaluated with respect to a fixed particle: $\mathrm{D}_t X = \partial X(P, t)/\partial t|_{P\,\mathrm{fixed}}$.

In order to maintain complementarity between discrete and continuous time, the proposed formalism expresses such relationships in a neutral notation:

$$\text{Đ}X = F(X, Y).$$

This *change equation* may be read, "the change in X is given by $F(X, Y)$." The rules of manipulation for the Đ operator respect its complementary discrete- and continuous-time interpretations.

A particle-oriented description of behavior implies that in most cases active particles (e.g., cells, microrobots) will not have direct control over their position or velocity. Rather, particles will act by controlling local forces (e.g., adhesion, stress) between themselves and other particles in the same body or in other bodies. Therefore, "programming" active substances will involve implementing change equations that govern stress tensors and other motive forces associated with the particles.

As discussed in Sec. 2, noise, uncertainty, error, faults, defects, and other sources of randomness are unavoidable in embodied computation, especially when applied in nanotechnology. Indeed, such randomness can often be exploited as a computational resource in embodied computing. Therefore, the behavior of elements will often be described by stochastic differential/difference equations. To facilitate the complementary continuous/discrete interpretations, it is most convenient to express these equations in the Langevin form:

$$\text{Đ}X = F(X, Y, \ldots) + G_1(X, Y, \ldots)\nu_1(t) + \cdots + G_n(X, Y, \ldots)\nu_n(t),$$

where the ν_j are noise terms, but we must be careful about the interpretation of equations of this kind.

To see why, consider a stochastic integral, $X_t = \int_0^t H_s \mathrm{d}W_s$, where W is a Wiener process (Brownian motion). In differential form this is $\mathrm{d}X_t = H_t \mathrm{d}W_t$. To maintain complementarity, this should be consistent in the limit with the difference equation: $\Delta X_t = H_t \Delta W_t$. This implies that the stochastic integral is interpreted in accord with the Itō calculus. The corresponding stochastic change equation in Langevin form is $\text{Đ}X_t = H_t \text{Đ}W_t$. Interpreted as a finite difference equation it is $\Delta_t X_t = H_t \Delta_t W_t$, which makes sense. However the corresponding differential equation, $\mathrm{D}_t X_t = H_t \mathrm{D}_t W_t$, is problematic, since a Wiener process is nowhere differentiable. Fortunately we can treat $\mathrm{D}_t W_t$ purely formally, as follows. First observe that $\Delta W_t = W_{t+\Delta t} - W_t$ is normally distributed with zero mean and variance Δt, $\Delta W_t \sim \mathcal{N}(0, \Delta t)$. Therefore $\Delta_t W_t = \Delta W_t / \Delta t$ is normally distributed with unit variance, $\Delta_t W_t \sim \mathcal{N}(0, 1)$, and $\Delta_t W_t$ can be treated as a random variable with this distribution. To extend this to the continuous case, we treat $\text{Đ}W_t$ as a random variable, $\text{Đ}W_t \sim \mathcal{N}(0, 1)$. Therefore the stochastic change equation $\text{Đ}X = H\text{Đ}W$ has consistent complementary interpretations. Similarly, for an n-dimensional Wiener process $\mathbf{W}^n$, we interpret $\text{Đ}\mathbf{W}^n$ to be an n-dimensional random vector distributed $\mathcal{N}(0, 1)$ in each dimension.

As explained above (Sec. 2), one requirement for the formalism is that it be able to express qualitative behavioral rules corresponding to the influence diagrams widely used in biology. Therefore, we define the notation $\text{Đ}X \sim -X, Y, Z$ (for example) to mean that the change of X is "repressed" (negatively regulated) by X and "enhanced" (positively regulated) by Y and Z. We have been calling such a relationship a *regulation* and read it, "the change in X is negatively regulated by X and positively regulated by Y and Z." Formally, $\text{Đ}X \sim -X, Y, Z$ is interpreted as a change equation $\text{Đ}X = F(-X, Y, Z)$ in which F is a function that is unspecified except that it is monotonically non-decreasing in each of its arguments. (Thus the signs of the arguments express positive or negative regulation.)

4 Example

The following example, which illustrates our formalism, extends the model developed by Frederico Bussolino and his colleagues of vasculogenesis, the early stages of formation of capillary networks from dispersed endothelial cells [2,11,33] (cf. [26]). Aggregation is governed by a morphogen that is released by the cells and that diffuses and degrades.

```
substance morphogen:
    scalar fields:
        $C$                    || concentration
        $S$                    || source
    order-2 field $\mathbf{D}$     || diffusion tensor
    scalar $\tau$                 || degradation time constant

    behavior:
    $ĐC = \triangle(\mathbf{D}C) + S - C/\tau$    || diffusion + release − degradation
```

($\triangle = \nabla \cdot \nabla$ is the Laplacian for tensor fields.) The cells produce morphogen at a rate α and follow the gradient at a rate governed by attraction strength β. Cell motion is impeded by dissipative interactions with the substrate, which are measured by an order-2 tensor field γ. Since cells are filled with water, they are nearly incompressible, and so they have a maximum density n_0, which influences cell motion; to accommodate this property (which will also apply to microrobots), the model uses an arbitrary function, ϕ, that increases very rapidly as the density exceeds n_0.

```
substance cell-mass is morphogen with:
    scalar fields:
        $n$                    || number density of cell mass
        $\phi$                 || cell compression force
    vector field $\mathbf{v}$      || cell velocity
    scalars:
        $n_0$                  || maximum cell density
        $\alpha$               || rate of morphogen release
        $\beta$                || strength of morphogen attraction
    order-2 field $\gamma$     || dissipative interaction

    behavior:
  $S = \alpha n$    || production of morphogen
                    || follow morphogen gradient, subject to drag and compression:
$Đ\mathbf{v} = \beta \nabla C - \gamma \cdot \mathbf{v} - n^{-1} \nabla \phi$
$Đn = -\nabla \cdot (n \cdot \mathbf{v}) + \mathbf{v} \cdot \nabla n$    || change of density in material frame
  $\phi = [(n - n_0)^+]^3$                    || arbitrary penalty function
```

References

1. Adamatzky, A., Costello, B.D.L., Asai, T.: Reaction-Diffusion Computers. Elsevier, Amsterdam (2005)
2. Ambrosi, D., Gamba, A., Serini, G.: Cell directional persistence and chemotaxis in vascular morphogenesis. Bulletin of Mathematical Biology 67, 1851–1873 (2004)
3. Barish, R.D., Rothemund, P.W.K., Winfree, E.: Two computational primitives for algorithmic self-assembly: Copying and counting. Nano Letters 5, 2586–2592 (2005)
4. Benzi, R., Parisi, G., Sutera, A., Vulpiani, A.: Stochastic resonance in climatic change. Tellus 34(10-16) (1982)
5. Beysens, D.A., Forgacs, G., Glazier, J.A.: Cell sorting is analogous to phase ordering in fluids. Proc. Nat. Acad. Sci. USA 97, 9467–9471 (2000)
6. Brooks, R.: Intelligence without representation. Artificial Intelligence 47, 139–159 (1991)
7. Clark, A.: Being There: Putting Brain, Body, and World Together Again. MIT Press, Cambridge (1997)
8. de Gennes, P.G.: Soft matter. Science 256, 495–497 (1992)
9. Forgacs, G., Newman, S.A.: Biological Physics of the Developing Embryo. Cambridge University Press, Cambridge (2005)
10. Fung, Y.C.: Biomechanics: Mechanical Properties of Living Tissues. Springer, New York (1993)
11. Gamba, A., Ambrosi, D., Coniglio, A., de Candia, A., Di Talia, S., Giraudo, E., Serini, G., Preziosi, L., Bussolino, F.: Percolation, morphogenesis, and Burgers dynamics in blood vessels formation. Physical Review Letters 90(11), 118101-1 – 118101-4 (2003)
12. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, Upper Saddle River (1999)
13. Johnson, M., Rohrer, T.: We are live creatures: Embodiment, American pragmatism, and the cognitive organism. In: Zlatev, J., Ziemke, T., Frank, R., Dirven, R. (eds.) Body, Language, and Mind, vol. 1, pp. 17–54. Mouton de Gruyter, Berlin (2007)
14. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
15. MacLennan, B.J.: Universally programmable intelligent matter: Summary. In: IEEE Nano 2002 Proceedings, pp. 405–408. IEEE Press, Los Alamitos (2002)
16. MacLennan, B.J.: Molecular combinatory computing for nanostructure synthesis and control. In: IEEE Nano 2003 Proceedings, p. 13_01. IEEE Press, Los Alamitos (2003)
17. MacLennan, B.J.: Natural computation and non-Turing models of computation. Theoretical Computer Science 317(1-3), 115–145 (2004)
18. MacLennan, B.J.: Radical reconfiguration by universally programmable intelligent matter. In: Air Force Reconfigurable Electronics Workshop, Santa Fe, NM, May 17-18. Air Force Research Laboratory (2004)
19. MacLennan, B.J.: Radical reconfiguration of computers by programmable matter: Progress on universally programmable intelligent matter — UPIM report 9. Technical Report CS-04-531, Dept. of Computer Science, University of Tennessee, Knoxville (2004)
20. MacLennan, B.J.: Highly programmable matter and generalized computation: Research in reconfigurable analog and digital computation in bulk materials. In: 1st AFRL Reconfigurable Systems Workshop, Albuquerque, NM, Feburary 14-15, pp. 14–15. Air Force Research Laboratory (2007)

21. MacLennan, B.J.: Embodiment and non-Turing computation. In: 2008 North American Computing and Philosophy Conference: The Limits of Computation, Bloomington, IN, July 10-12. The International Association for Computing and Philosophy (2008), http://www.cs.utk.edu/~mclennan/papers/AEC-TR.pdf
22. MacLennan, B.J.: Computation and nanotechnology (editorial preface). International Journal of Nanotechnology and Molecular Computation 1(1), i–ix (2009)
23. MacLennan, B.J.: Configuration and reconfiguration of complex systems by artificial morphogenesis. In: Reconfigurable Systems Workshop 2009: Discovery Challenge Thrust, Santa Fe, NM, July 20-22. Air Force Office of Scientific Research (2009)
24. MacLennan, B.J.: Super-Turing or non-Turing? Extending the concept of computation. International Journal of Unconventional Computing 5(3-4), 369–387 (2009)
25. MacLennan, B.J.: Bodies — both informed and transformed: Embodied computation and information processing. In: Dodig-Crnkovic, G., Burgin, M. (eds.) Information and Computation. World Scientific, Singapore (in press)
26. Merks, R.M.H., Brodsky, S.V., Goligorksy, M.S., Newman, S.A., Glazier, J.A.: Cell elongation is key to in silico replication of in vitro vasculogenesis and subsequent remodeling. Developmental Biology 289, 44–54 (2006)
27. Odell, G.M., Oster, G., Alberch, P., Burnside, B.: The mechanical basis of morphogenesis. I. Epithial folding and invagination. Developmental Biology 85, 446–462 (1981)
28. Pfeifer, R., Bongard, J.C.: How the Body Shapes the Way We Think — A New View of Intelligence. MIT Press, Cambridge (2007)
29. Pfeifer, R., Lungarella, M., Iida, F.: Self-organization, embodiment, and biologically inspired robotics. Science 318, 1088–1093 (2007)
30. Rothemund, P.W.K., Papadakis, N., Winfree, E.: Algorithmic self-assembly of DNA Sierpinski triangles. PLoS Biology 2(12), 2041–2053 (2004)
31. Rothemund, P.W.K., Winfree, E.: The program-size complexity of self-assembled squares. In: Symposium on Theory of Computing (STOC), New York, pp. 459–468. Association for Computing Machinery (2000)
32. Salazar-Ciudad, I., Jernvall, J., Newman, S.A.: Mechanisms of pattern formation in development and evolution. Development 130, 2027–2037 (2003)
33. Serini, G., Ambrosi, D., Giraudo, E., Gamba, A., Preziosi, L., Bussolino, F.: Modeling the early stages of vascular network assembly. The EMBO Journal 22(8), 1771–1779 (2003)
34. Steinberg, M.S., Poole, T.J.: Liquid behavior of embryonic tissues. In: Bellairs, R., Curtis, A.S.G. (eds.) Cell Behavior, pp. 583–607. Cambridge University Press, Cambridge (1982)
35. Taber, L.A.: Nonlinear Theory of Elasticity: Applications in Biomechanics. World Scientific, Singapore (2004)
36. Tomlin, C.J., Axelrod, J.D.: Biology by numbers: Mathematical modelling in developmental biology. Nature Reviews Genetics 8, 331–340 (2007)