Nidus Design Document

Tim Taylor
Department of Artificial Intelligence
University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL, U.K.
`timt@dai.ed.ac.uk`
`http://www.dai.ed.ac.uk/daidb/people/homes/timt/`

14th June 1998

1 Introduction

This document describes Nidus[1], a model (actually, a family of models) designed to investigate the essential components and interactions required to support the origin and evolution of living organizations. Sections 2–3 describe the motivations for developing the model, and Sections 4–5 describe more specific considerations. The model, research programme and relation to other work are described in Sections 6–9. In a bit more detail, the organization of this document is as follows: In Section 2 a description is given of what is meant by the term 'living organization' in this work, or, in other words, what exactly it is we are hoping to produce. Next, Section 3 contains a brief description of two alternative approaches commonly taken to explaining the origin of life. Section 4 lists the basic components and interactions that are taken as being essential ingredients of any system capable of supporting the origin and evolution of living organizations, and Section 5 lists some other criteria that were important when developing the Nidus design. The model itself is described in Section 6, followed by a mapping between Nidus and the real world in Section 7 and an overview of a programme of research in Section 8. The document concludes in Section 9 with a description of how Nidus relates to other work on the origin and evolution of living organizations.

2 Living Organization

A major motivation for developing Nidus was the realisation that the previous artificial evolutionary system developed by the author, called Cosmos [47], lacked careful consideration of exactly *what* was being evolved. It became clear that such considerations are vital if we wish to model the origin and evolution of *life*[2], rather than evolutionary processes in a more general sense. Indeed, this shortcoming is not specific to Cosmos, but is characteristic of Tierra, the artificial life platform upon which Cosmos was based [40], and all other Tierra-like platforms that have been developed elsewhere.

[1] "*nidus* noun, (1) a nest or breeding place... (2) a place where something originates, develops..." *Webster's Dictionary*

[2] Most Artificial Life researchers working on evolutionary models *do* wish to do this, as is either implicit or, more usually, explicit, in their publications.

Evolution is a process of *change*. It tells us something about the *trajectory* of reproducing entities through their space of possible forms, and explains how reproducing entities become adapted to their environment. However, it *assumes* the existence of reproducing entities to begin with, and does not specify what sort of entities they should be, other than that they must be able to reproduce. Similarly, it does not specify that any particular sort of *environment* is necessary—evolution is a very general phenomenon.

A model in which a population of integers reproduce with occasional mutation, and differential survival based, perhaps, upon how large the integer is, will exhibit evolution, but it will never produce anything more than just integers. To take a more familiar example, Genetic Algorithms satisfy the basic requirements for the evolution of the individual 'chromosomes', but all that is generally evolving is the encoded solution to some predetermined problem[3]. Thus it is clear that if we are interested in modelling the evolution of life, we must:

1. have a clear idea of what sorts of functions or roles a reproducing entity must fulfill if we are to consider it alive, i.e. a definition of life (we do not specify that the definition has to be 'correct' or universally agreed upon, but it *does* have to be explicitly stated so that it is clear what phenomena are under investigation), and

2. include in our model explicit components and interactions not only to allow for an evolutionary process to emerge, but also to allow for the existence of entities that fulfill any other functions or roles that we have specified as necessary for life.

In other words, evolution is not sufficient to explain life; we also require a *theory of living organization* (i.e. a working definition of life), and a *theory of the kind of worlds which are capable of supporting such organization*. We must incorporate all of these considerations into any A-life model designed to investigate the emergence and evolution of life.

To be fair, Tom Ray, the designer of Tierra, did offer a definition of life in his work. Ray says "I would consider a system to be living if it is self-replicating, and capable of open-ended evolution" [40] (p.372). However, many people would complain that self-replication[4] is not sufficient to define life, not least because the notion of *self*-replication, in contrast to any other type of replication, is problematic in itself. The issue of self-replication in the context of the evolution of life is discussed in more detail in Section 9.2.2. Another (related) problem with Ray's definition of life is that it does not define what sorts of environments might support life, or the sorts of ecological interactions which should be available. I think it is therefore fair to say that designers of Tierra-like models typically do not explicitly offer an adequate definition of living organization, and do not provide an explicit list of assumptions about the type of environment required to support the origin and evolution of such organizations.

In the remainder of this section I will first give an informal description of my definition of living organizations and the type of environment in which they exist (Section 2.1), and then present a number of more precise definitions (Section 2.2). In Section 4 I make an initial attempt at enumerating the basic components and interactions necessary for a world to support the origin and evolution of these organizations.

[3] For more discussion on self-reproduction, see Section 9.2.2.

[4] I use the terms 'replication' and 'reproduction' more or less interchangeably, although the former tends to refer to copying of a single molecule or component, and the latter to the copying of a whole organism or complex organization.

2.1 An Informal Description

I will first describe the sort of *environment* in which organisms[5] exist:

> Organisms exist in a world in which novel compound components can be built up by a process of construction from a set of atomic components (and these structures can also be broken down to their atomic components). Some reactions require an input of energy, and others release energy. There is a flux of energy through the system which allows for a basic set of interaction classes to occur spontaneously. Compound components can have novel functions (i.e. functions not available in the original set of atomic components), which may operate on specific classes of other components.

Now, the sort of things that organisms *are*:

> Existing in the sort of environment just described, organisms are self-maintaining (self-producing and self-repairing) organizations that have achieved a high degree of entification from their environment by defining their own boundary, within which they can control and regulate components and processes. Being dynamically self-maintaining they are nonequilibrium structures, requiring a continual input of energy and matter in order to maintain their organization[6]. Organisms can, in some sense, be seen as *traders*, engaging in exchanges of matter and energy in a *market* comprising their biotic and abiotic environment. Indeed, these exchanges of matter and energy are probably a primary driving force behind most types of coevolution of organisms. Higher degrees of living organizations exhibit higher degrees of autonomy from matter and energy input.

2.2 Attempts at a More Precise Description

A number of people have proposed somewhat more precise definitions of living organizations. Humberto Maturana and Francisco Varela's concept of *autopoiesis* [30, 49] was, to my knowledge, the first attempt to capture a precise definition of what an organism actually *is* and *does*, without regard to how it came to be how it is (i.e. they deliberately avoided an evolutionary explanation). Informally,

> "an autopoietic machine is a homeostatic (or rather a relations-static) system that has its own organization (defining network of relations) as the fundamental invariant." [49] (p.13)

The formal definition of autopoiesis is as follows[7]:

> "An autopoietic system is organized (defined as a unity) as a network of processes of production (transformation and destruction) of components that produces the components that: (1) through their interactions and transformations continuously regenerate and realise the network of processes (relations) that produced them; and (2) constitute it (the machine) as a concrete unity in the space in which they exist by specifying the topological domain of its realisation as such a network." [49] (p.13)

[5] I use the terms 'living organization' and 'organism' more or less interchangeably, although I will tend to use the former to emphasise the sorts of spatial and causal relationships that exist between parts of a living entity.

[6] We might also add here the requirement that the organism must be able to reproduce. See Section 2.2 for arguments for and against including reproduction and evolution in the definition of living organizations.

[7] Unfortunately, like much of Maturana and Varela's writing in English editions, the definition is somewhat opaque.

Autopoiesis is at one end of the spectrum of definitions of life, where different definitions place different emphasis on the relative importance of the *metabolic* (or *ecological* or *economic*) aspects versus the *evolutionary* aspects of organisms. While I agree with Maturana and Varela that evolution, being solely a mechanism of *change*, cannot describe what a living organization actually *is*, I think that autopoiesis misses (or at least fails to make explicit) an important aspect of the concept of life, namely that organisms are exquisitely *adapted* to a particular ecological niche in a particular environment. That is, organisms engage in exchanges of matter and energy with their environment (biotic and/or abiotic), as members of an ecological food web. In biological life, this adaptation to a particular niche comes about by the process of evolution.

Federico Morán and colleagues state a definition of life that is effectively the same as the definition of autopoiesis, but with the proviso that the viability of an organization is based on the reproduction of information (i.e. organisms evolve and therefore become adapted):

> "... one can define living organization as the result of a process of temporally recursive networks of component production, self-closed by a physical border generated by the system itself, whose viability is based on informational mechanisms of self-reproduction." [34] (p.220)

One could argue that adaptation could be achieved by some mechanism other than evolution, and propose an appropriately modified version of the above definition. However, the fact remains that all examples of biological life have achieved adaptation by evolution, and definitions which allow for adaptation by other mechanisms are generalised to cater for purely hypothetical examples of life.

In their paper, Morán et al. go on to point out that a definition such as this, based on the idea of self-maintenance, can be viewed in two different ways: functional and energetic. From the functional point of view, we can think of a living organization as maintaining its informational components by processes which they themselves codify. However, from the energetic point of view, the idea of self-maintenance "lies in the mutual relation between components and energetic couplings: the network generates those components that allow mechanisms of energetic coupling which generate the very organization that produces them recursively" (p.220). Morán et al. then discuss living organizations as dissipative structures, open to energy and matter and requiring a continual supply of extra energy to maintain themselves. Biological dissipative structures are different to physical dissipative structures because they participate directly in their own self-maintenance, and they also act upon their environment, transforming it. They then refine the definition given above as follows:

> "Life is based on a self-sustaining chemical organization able to ensure its own energetic autonomy. The ensemble of these processes is called metabolism." [34] (p.221)

I shall stick with Morán et al.'s definition, with this extra refinement that we consider energetic as well as functional self-maintenance, when referring to 'living organizations' in Nidus.

3 The Origin of Life

In the previous section, the distinction was made between definitions of life which emphasise the evolutionary perspective and those which emphasise the metabolic perspective. A similar distinction exists in models put forward to explain the *origin* of life on Earth[8]. The two approaches can be called the *replicator-first* approach and the *metabolism-first* approach.

The replicator-first model assumes that the original seed for life was the existence of some sort of material that could exist in a large or infinite variety of forms, could reproduce more or less faithfully without the assistance of complicated machinery, and had some mechanism whereby specific other reactions or processes could become associated with specific forms of the material. Examples include various RNA-world models (for references see, for example, [37]), and Cairns-Smith's clay model [9]. The presence of a simple self-reproducer of this nature is enough to begin a process of evolution. The idea is that some forms of the material may be such that they have processes associated with them (e.g. they may act as a catalyst for some reaction) that act to stabilise the material. Such forms will be favoured by natural selection (precisely because they are more stable), and evolution proceeds by selecting reproducers that catalyse more and more reactions that are beneficial to the stability of the replicator. At some point the reactions will effectively give the replicator complete control over the composition of its local environment, at which stage the network of reactions will probably fulfill our criteria for being a living organization. It seems that most prominent evolutionary biologists and chemists favour this approach, e.g. John Maynard Smith [31], Richard Dawkins [11] and Graham Cairns-Smith [9].

In contrast, the metabolism-first approach assumes that self-maintaining organizations were the seed of life. These models assume that the world is such that self-maintaining (collectively autocatalytic) organizations of chemical reactions occur spontaneously with reasonable probability. Being self-maintaining, they persist for reasonable durations. Another consequence of being self-maintaining is that they produce all of the components from which they are composed, so it is easy to imagine scenarios by which some organizations of this type might reproduce (e.g. by splitting in two). Such self-reproducing organizations will become more abundant, and will replace non-reproducers if there is competition for resources. With self-reproduction comes evolution, so any variations of these self-reproducing and self-maintaining organizations that make them more stable will be selected for. By this process, the idea is that a genetic representation will emerge by natural selection to give the organization a high degree of stability. This approach to the origin of life is favoured by Maturana and Varela (see, e.g., Chapter 5 of [49]), and variations have also been suggested by, among others, Freeman Dyson [14] and Stuart Kauffman [27].

A problem with the metabolism-first approach is that there is no guarantee that a collectively autocatalytic reaction set will, in general, support a large variety of mutations. That is, there may only be a small number of variations of the set that are reachable by a series of single mutations which retain the property of being collectively autocatalytic (self-maintaining). Moreover, if we have a self-reproducing organization, even if a mutation does produce a variant set that is still self-maintaining, the variation will not necessarily be passed on to its offspring. In other words, to use the terminology of Maynard Smith and Szathmáry [31] (pp.41–44), self-maintaining reaction sets without genetic information will almost certainly be 'simple replicators' or maybe 'limited hereditary replicators', but not, as is required for sustained evolution, 'indefinite hereditary replicators'. For these reasons, I will use the replicator-first scenario, initially at least, as my working hypothesis in Nidus, and incorporate features into Nidus to model the conditions considered necessary for this scenario to develop.

[8] Here I am only talking about models that assume a terrestrial origin of life.

4 Essential Components and Interactions

The methodological approach adopted in Nidus has been to:

1. Consider biological life, and popular theories that have been put forward to explain its origin and evolution.

2. Try to abstract the fundamental aspects of the physical (and chemical) world that make it capable of supporting life, and ignore all other aspects. This leads us to propose a tentative list of the fundamental types of components and interactions that would be required by *any* system if it is to be capable of supporting the origin and evolution of living organizations. This list is presented in the rest of this section.

3. From the proposed list of fundamental components and interactions, design a minimal computer model which incorporates all of them. This is the Nidus design, described in Section 6.

The proposed list of essential fundamental components and interactions necessary for supporting the origin and evolution of living organizations is shown in Figure 1. I have arranged the list hierarchically: in this way, I hope that it is easier to see how I have arrived at the more specific requirements from the more general. I would expect (and hope) that the requirements appearing at higher levels in this hierarchy are the least controversial, but more specific items will be more open to debate and revision in the light of experimental results.

The items in this list are explained in the rest of this section, together with the reasons why I have included them in the list. It is emphasised that this is a proposal for a *minimal* list of interactions and components that a world must contain if it is to support the origin and evolution of living organizations. Particular worlds may contain many other types of component and interaction (e.g. environmental carriers of information, such as light, sound etc.), but the claim is that they must contain *at least* what is listed in Figure 1.

The list may be viewed as my proposed *definition* of what it takes for a world to support the origin and evolution of living organization. As such, it is subject to empirical investigation, and is open to criticism and discussion. This list may appear somewhat vague, but I have tried to keep it as general as possible. Results may lead us to refine or change it.

4.1 The environment

This is my proposed definition of the sort of world in which living organizations may arise. The fundamental features of the world are *matter* and *energy*. These are dealt with separately below, but they are closely related, as is explained.

4.1.1 Self-organization of matter

Living organizations arise when evolution acts in a world in which matter has an inherent ability to self-organise into compounds and interacting sets of compounds. Natural selection favours some of these self-organised forms over others. To build such a world, we need:

Aggregative matter. The fundamental **components** in the world are called **atoms**. Atoms are the building blocks from which larger components can be constructed. The number of atoms in the world is always conserved during reactions. These components are the operands of the world (the things which get acted upon), and can also act as operators (intrinsically determining

- **The environment**
  - **Self-organization of matter**
    - **Aggregative matter**
    - **Reactions**
      - **Aggregative**
      - **Degradative**
    - **Conservation of matter**
    - **Specificity of reaction**
    - **Control of reaction**
  - **Spatial representation**
  - **Diffusion**
  - **Energy**
    - **Forms of energy**
      - **Environmentally conveyed energy**
      - **Energy associated with matter**
    - **Energy transduction during reactions**
      - **Exothermic reactions**
      - **Endothermic reactions**
    - **Conservation of energy**
    - **Entropy increase in a closed system**
    - **External energy source**
    - **Energy coupling**
      - **Coupling external energy source with other forms of energy**
      - **Coupling energy transduction between components**
- **Biogenesis**
  - **Indefinite hereditary auto-replicators with catalytic activity**
  - **Control of local environment**

Figure 1: The Essential Components and Interactions

reactions between other components—see *Control of Reaction*). As new operators can appear in the world by the aggregation of existing operators, this is the sort of system that Walter Fontana and Leo Buss call a constructive dynamical system [17]. At the risk of getting a bit ahead of myself, the following quotation from Chris Langton demonstrates the relevance of using an aggregative system when studying the evolution of life:

> "The molecular logic of life is a *dynamic distributed* logic. An initial set of operators and operands goes to work producing more operators and operands, which immediately enter into the ongoing logical 'fray'. Some of these new operators and operands are distributed as new initial sets in the process of self-reproduction. This dynamical character of the molecular logic of life is unlike a typical formal logic which, although it provides an initial set of operators and primitive operands, has no internal dynamics of its own." [29] (p.122)

Reactions. All allowed reactions may occur spontaneously, but the rate at which they occur can be increased in the presence of catalysts (see *Control of Reaction*). Energy is the primary mechanism for determining which reactions are allowed at any given time (see *Energy*). To implement reactions in our world, we need:

- **Aggregative** reactions for building larger components from smaller components.
- **Degradative** reactions for breaking larger components into smaller components.

Conservation of matter. Atoms are the fundamental building blocks of the world, and can never be created or destroyed during reactions.

Specificity of reaction. Each component can only react with a particular set of other components, so that specific reactions pathways can emerge.

Control of reaction. Some reactions can only occur in the presence of compounds that act as catalysts but do not participate in the net reaction. Such reactions are therefore controlled by the presence or absence of these catalysts. In the real world, living organization is achieved largely via the control of reactions by catalysts (enzymes). Another important aspect of this control of reactions by enzymes is the ability of many of them to act as *switches*, so that their activity can be turned on or off by the presence or absence of a specific regulatory molecule.

4.1.2 Spatial representation

Spatial representation provides the notion of individuality to specific components within the world. Without such a representation, in a world where any component could react with any other component, the resulting behaviour will be statistically based on the global concentrations of different kinds of components and the notion of individual organisms has no meaning. In nature, spatial isolation is one of the prime mechanisms by which speciation occurs. Speciation, in turn, alters the ecological niches available for other species to fill in a given ecosystem. It is therefore likely that a spatial representation can facilitate the emergence of complex ecosystems in a world of evolving species.

4.1.3 Diffusion

The components in the world must be able to move around (passively, and maybe actively as well), so that they come into contact with different neighbouring components over time (i.e. their local environment changes). In this way, any component can engage in a variety of reactions over time.

4.1.4 Energy

Energy is the primary mechanism for determining whether a reaction between particular components is possible, and reactions generally entail an exchange of energy between the reactants and the environment. Energy is therefore the driving force behind activity in the world—without it, all activity in the system eventually dies out.

Nearly all previous models of the origin of life ignore energetic considerations. However, I agree with the following quote from Federico Morán et al. that energetic considerations are an essential part of a satisfactory description of what living organisms *do*:

> " [previous models] work just as well without any coupling to exergonic and endergonic metabolic reactions. They are provided by their investigators with whatever nutrients and event probabilities are needed. For the theoretical purposes these models are meant to serve, this is entirely justified. The authors are dealing, after all, with the *logic* of virtual life, not with its implementation in some particular material basis. In computer science one can design or analyze logical circuitry without regard to the reality of the underlying hardware, and this holds also for the formal treatment of possible biochemical systems. Accordingly these authors need not consider that in implementing such systems, one must include a good deal of ancillary machinery which has no interpretation in the logical design. In computer engineering, side effects of the hardware include power supply, capacitive leakage, current induction, switch bounce, generation of heat, and many others whose correction vastly complicates the task. Likewise in metabolism, the collection of nutrients, excretion of waste products, correction of errors, defensive and competitive strategies, and so forth all require ancillary metabolic machinery which seems to us almost to embody the essence of living systems." [34] (p.220)

Having said this, a claim in this work is that, for living organizations, it is *catalysts* which play the predominant role in *controlling* which reactions occur (see *Control of Reaction*) (i.e. catalysts play the important *logical* role). Living organizations have to ensure they have a constant supply of energy exchanges occur in all reactions, but it is the catalysts that control *which* reactions are occurring at any given time. However, this basic need for a continual input of energy still has important consequences for the sorts of things that living organizations may be expected to do. It is proposed that the important features of the world with regard to energy from the point of view of explaining biological organization are as follows:

Forms of energy. It is proposed that two forms of energy are fundamental:

- **Environmentally conveyed energy**. Energy can be transmitted in the environment, independent of matter.

- **Energy associated with matter**. Every component in the world does, however, have a certain minimum amount of energy associated with it (which is generally different for different components). A new component can only be created if there is sufficient energy locally available to achieve this level. The energy associated with components, and the amount of energy available locally, therefore determine whether a particular reaction can occur.

Energy transduction during reactions. When components undergo reactions, the amount of energy associated with the products of the reaction may show a net increase or decrease when compared to that of the reactants. In energetic terms, there are therefore two types of reaction:

- **Exothermic reactions**. The total energy associated with the final products is less than that associated with the reactants, so energy is released during the reaction.

- **Endothermic reactions**. The total energy associated with the final products is more than that associated with the reactants, so an source of energy is required for the reaction to occur.

Conservation of energy. During any sort of reaction, energy is never created or destroyed but only converted from one form into another.

Entropy increase in a closed system. If the world is closed to energy and matter (i.e. there is no flow of either of these into or out of the world), then the positional and thermal disorder (i.e. the entropy) will tend to increase. The world will therefore tend to homogenise.

Any reaction involves the exchange of energy between a component and its local environment. The local environment consists of environmentally conveyed energy, and possibly of other local components which may also act as a source or sink for energy.

External energy source. Given that the world will tend to homogenise if it is closed to energy, we require an external source of energy to keep the system away from equilibrium and, ultimately, to enable it to support living organization.

Energy coupling. To transfer energy from the external source (environmentally conveyed energy) to individual components (energy associated with matter), and between individual components, forms of coupling are required. Specifically, we need:

- Mechanisms for **coupling the external energy source with other forms of energy** (e.g. energy associated with matter).

- Mechanisms for **coupling energy transduction between components**. Some reactions may result in energy being released from a component (which may or may not cause that component to break down). The world must have a mechanism by which this energy can be used to drive another reaction which requires energy.

4.2 Biogenesis

There are a couple of additional specific considerations that are required if the sort of world described above is to support the *origin* and *evolution* of living organizations.

4.2.1 Indefinite hereditary auto-replicators with catalytic activity

As mentioned in Section 3, sustained evolution requires indefinite hereditary replicators. In other words, we must have replicators which can exist in an indefinite number of distinct forms, and which, when they reproduce, produce new structures which resemble the old ([31] pp.41–44). These replicators must be relatively stable so that they can reliably pass information from one generation to the next. Additionally, if these replicators are to act as a seed for evolution in the world (i.e. if we are taking a replicator-first approach to the origin of life), they must be able to replicate by themselves, without the aid of complicated support machinery (i.e. they should be able to replicate without being a member of a large autocatalytic reaction set). To use the terminology introduced later in the document (Section 9.2.2), the replicators must therefore be *indefinite hereditary auto-reproducers*. For further discussion of this, see Section 9.2.2. A further requirement of these replicators is that they have catalytic activity, or in other words, specific configurations of the replicators should facilitate specific other reactions which therefore become associated with the replicator. If a replicator happens to be associated with reactions that in any way make it a better replicator, that replicator will be selected for in favour of others. In this way, there is a selection pressure for replicators which catalyse reactions which tend to improve their ability to replicate, so a process of evolution can begin. This requirement for catalytic activity is discussed further in Sections 8 and 9.2.2.

4.2.2 Control of local environment

This requirement comes from the fact that the definition of living organization we have adopted includes the idea of a 'physical border'. What is generally meant by this is that organisms are enclosed in *membranes* which allow selective diffusion both in and out. Slightly more generally, we can say that an organism at least must have *some* mechanism by which it can control its local chemical environment.

5 Other Design Criteria

Nidus has been designed as a system that implements all of the features enumerated in Section 4. On top of these, there were a few other criteria that guided the design process:

- **Compact representation**
  Because we are representing the world at a low level (a single living organization will be a collection of interacting components—probably a fairly large collection), we would like our design to have a very compact representation in terms of memory usage, to give us the potential of looking at the evolution of large numbers of organisms. Ideally, we would like to be able to run a world containing something in the order of millions of components on a single computer. For this to be possible, each atom can only take up a very small number of bytes of memory.

- **Low computational cost for interactions**
  Similarly, the faster the implementation is, the longer the evolutionary runs are that we will be able to complete in a given period of time. Therefore, none of the processes involved in running the model should be computationally expensive.

6 Nidus Design

With the points raised in Sections 4 and 5 in mind, the design of Nidus is as follows. The approach can be viewed as trying to embed a Tierra-like population of evolving programs on top of an artificial chemistry. The molecules (components) formed by the artificial chemistry are treated as the programs to be interpreted. All molecules therefore obey the fundamental reaction rules of the world (the defined 'physics'), but larger components may also have a wide range of emergent properties which come about through their being interpreted as programs. The analogy is that biomolecules have diverse properties defined by their precise three-dimensional arrangement (e.g. their secondary or tertiary structure etc.), which are in practice unpredictable from their primary structure. This part of the design is only partially comparable with Tierra, however, because self-reproduction in Nidus does not come about through evolving programs which explicitly encode a self-reproduction *algorithm*. On the contrary, in Nidus the idea is that self-reproduction is achieved by *fundamental reactions* in the artificial chemistry which copy certain types of component by *template replication*. In a sense, therefore, self-reproduction happens 'for free', but evolution proceeds by selecting the particular forms of these template-replicators that happen to encode for emergent functionality which has the effect of making the templates more stable or making the replication more reliable or quicker.

I will first describe the environment and basic components of the system, followed by the fundamental reaction rules (the 'primitive operators'). The interpretation of components as programs ('emergent operators') is then described. The description concludes by describing the top-level algorithm used to run Nidus, and a list of the system parameters.

6.1 Environment and Components

Nidus has a two-dimensional environment, divided into discrete squares. Each square may contain zero or one **atoms**, and may also carry a certain amount of environmentally-conveyed **energy** (see Section 6.1.1 for further details about energy in the environment). The world wraps around in both directions to give the environment a toroidal topography, thereby avoiding boundary effects.

Each atom is a member of a particular **class**. The number of different classes in the world is defined by the system parameter $N_{classes}$ (see Section 6.5). Strong **links** can form between atoms to form larger **components**, but only *between atoms of the same class*, and only *along one dimension* (designated the x direction). See Section 6.2 for the conditions under which links may be made or broken.

Weak **associations** may also form between atoms. These may form between atoms of any class, and may form along both dimensions (i.e. in both the x and y directions). A group of components coupled by associations is called a **compound**. See Section 6.2 for the conditions under which associations may be made or broken.

Components move randomly throughout the world by **diffusion**, as explained in Section 6.2. A compound moves as a single object. **Mutations** may also affect the state of some components, as explained in Section 6.2.4.

Components of length greater than or equal to a threshold specified by the system parameter $N_{eop\_threshold}$ are decoded as programs, and may therefore perform a wide variety of tasks. This aspect of Nidus is explained in Section 6.3.

An illustration of a typical section of the Nidus world is shown in Figure 2.

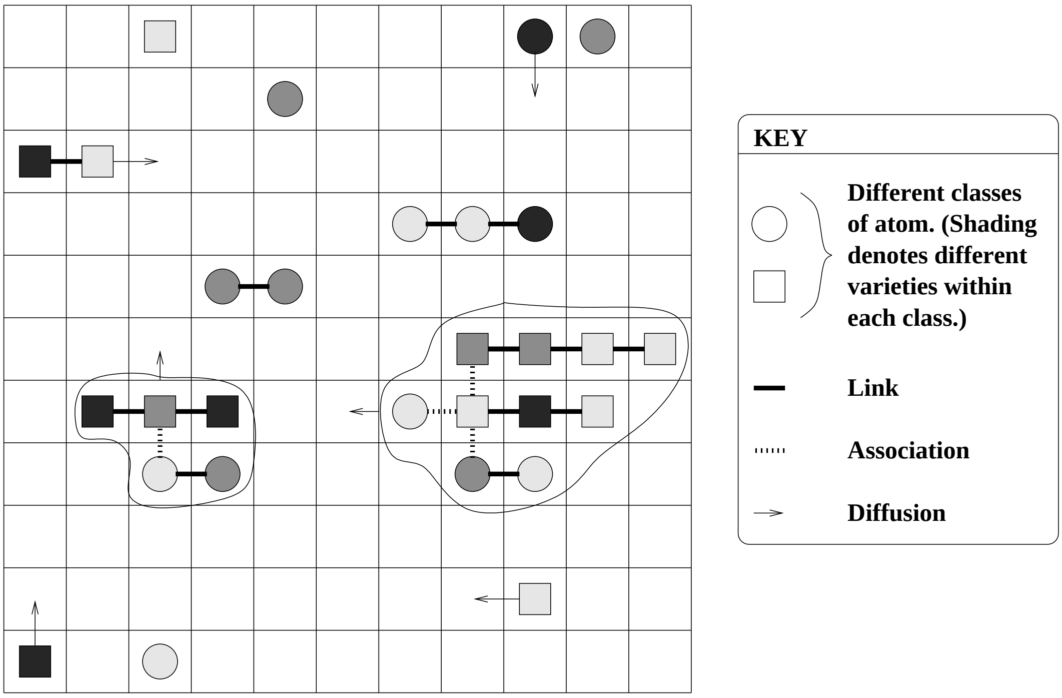

KEY
Different classes of atom. (Shading denotes different varieties within each class.)
Link
Association
Diffusion

Figure 2: The Nidus Environment

6.1.1 Energy flux

It has already been said that each square in the Nidus environment can carry a certain amount of environmentally-conveyed energy. This energy is supplied by an external source, at an amount determined by the function $F_{N\_energy\_flux}$ at each time step (see Section 6.5 for details of this function). Each square actually has a queue of length $N_{energy\_decay\_period}$ to record how much energy is available there. At each time step, the new input of energy from the external source is pushed onto this queue, and the oldest member of the queue is popped off. The queue therefore provides a rolling window to record energy input over the last $N_{energy\_decay\_period}$ time steps, or in other words each input of energy decays after $N_{energy\_decay\_period}$ time steps. When energy is required from a square to drive reactions, it is extracted by reducing the energy stored at each position in the queue, starting with the oldest member and continuing with each successively more recent member until the required amount has been supplied (or all of the energy has been exhausted).

6.2 Primitive Operators (P-OPs)

There are a number of operators which may act on components in the system at any time, for making and breaking links and associations, and for implementing diffusions and mutations.

6.2.1 Links

The procedure by which links may be formed spontaneously between neighbouring components is shown in Figure 3, and that by which they may spontaneously break is shown in Figure 4. Note that energy is required from the environment in order to make a link, and energy is released when a link is broken. The functions $F_{P\_make\_link}$ and $F_{P\_break\_link}$ are explained in Section 6.5.

```
for each atom:
if
    • there is an immediately neighbouring atom in the positive $x$ direction, and
    • both are of the same class $i$, and
    • the atoms are not already linked or associated, and
    • the combined environmental energy of the two squares $> E_{class_i}$
then, with probability given by the function $F_{P\_make\_link}$
    • make a link between the two atoms
    • remove any associations that the current atom might have
    • reduce locally-available environmental energy by $E_{class_i}$
```

Figure 3: The *Link Formation* P-OP

```
for each atom:
if
    • atom has a link to a neighbouring atom in the positive $x$ direction
then, with probability given by the function $F_{P\_break\_link}$
    • remove the link between the two atoms
    • release energy $E_{class_i}$ into the local environment
```

Figure 4: The *Link Destruction* P-OP

6.2.2 Associations

The procedure by which associations may be formed spontaneously between neighbouring components is shown in Figure 5, and that by which they may spontaneously break is shown in Figure 6. The functions $F_{P\_make\_assoc}$ and $F_{P\_break\_assoc}$ are explained in Section 6.5. Note that the formation or destruction of associations do not, in themselves, involve input or output of energy. Note also that the spontaneous formation of associations only occurs in components belonging to a restricted set of classes $\mathcal{S}$ (a parameter of the system), and an atom spontaneously forming an association in this manner can only do so with another atom of the same class. Also, the associations can only happen in the y direction. Additionally, the function $F_{P\_make\_assoc}$ is defined in such a way (Section 6.5) that there is a much higher chance of associations forming between atoms of the same *variety* than between those of different varieties. Other associations, not restricted to components belonging to the set of classes $\mathcal{S}$, and not necessarily even forming between components of the same class, may also be formed between a component acting as a catalyst (an E-OP) and other components specified by the catalyst, in both the x and y directions—see Section 6.3.

```
for each atom:
if
    • there is an immediately neighbouring atom in the positive $y$ direction, and
    • both are of the same class, and
    • class $\in \mathcal{S}$, and
    • the neighbouring atom has no existing links or associations
then, with probability given by the function $F_{P\_make\_assoc}$
    • make an association between the two atoms
```

Figure 5: The *Association Formation* P-OP

```
for each atom:
if
    • atom has an association with a neighbouring atom in the positive $x$ or $y$ directions
then, with probability given by the function $F_{P\_break\_assoc}$
    • break the association between the two atoms
```

Figure 6: The *Association Destruction* P-OP

The action of the *association formation* P-OP, together with the fact that the *link formation* P-OP breaks associations on the focal atom when a link is formed, means that classes with an E_{class_i} which is in the order of the amount of energy available in the environment (so that there is a fairly high chance of links forming spontaneously), should exhibit a simple form of self-replication by template copying. In other words, they will be indefinite hereditary auto-replicators, which is one of our requirements for biogenesis (Section 4.2). This process is illustrated in Figure 7.

6.2.3 Diffusion

Components (or more accurately, compounds, as associated groups of components move as a single unit) move randomly around the environment under the action of the *Diffusion* P-OP, shown in Figure 8.

6.2.4 Mutation

The *Mutation* P-OP, shown in Figure 9, affects a component's state vector. An explanation of the state vector is given in Section 6.3.1. The precise action of mutation may be slightly different for each item in the state vector, but generally has the effect of changing the current value of that item to a randomly chosen new value.

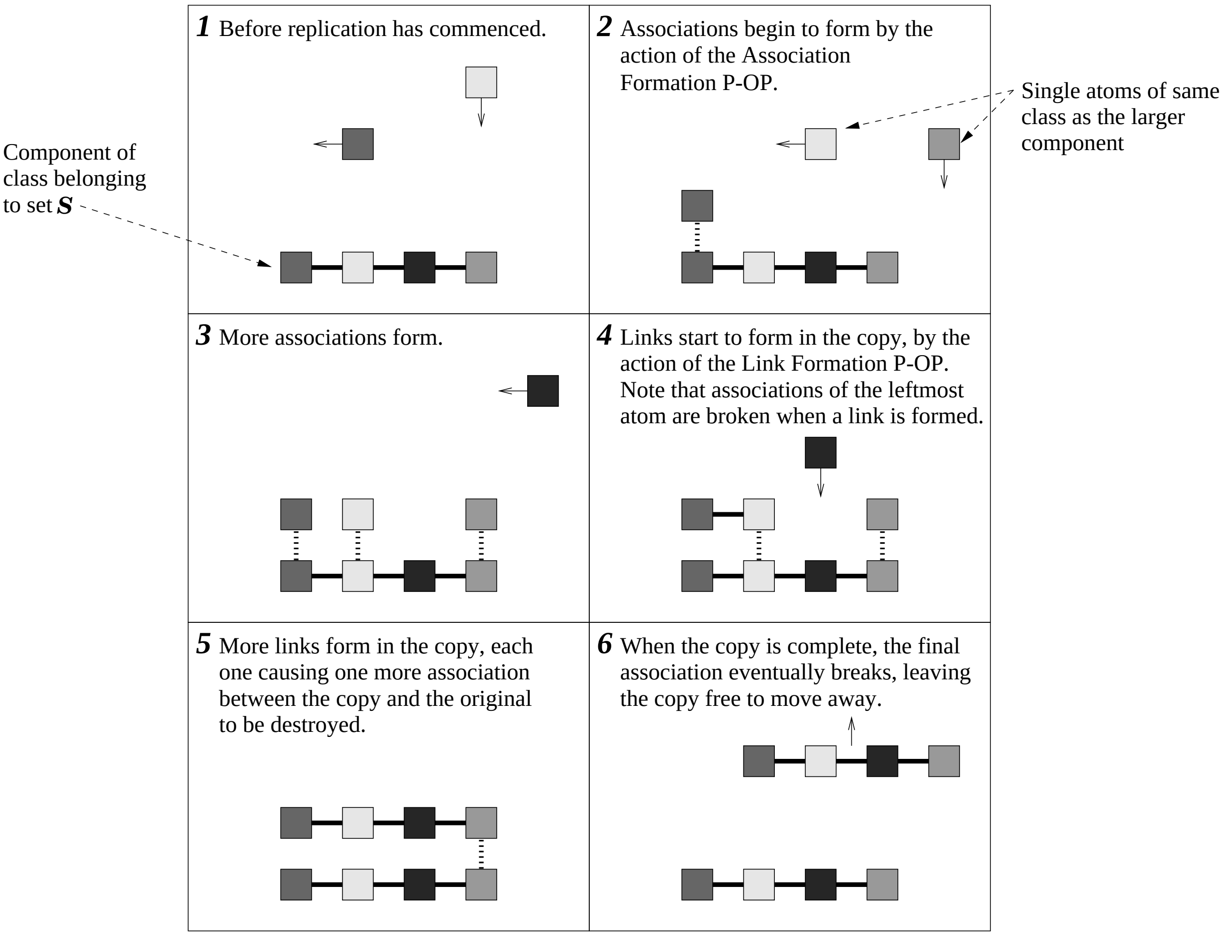

1 Before replication has commenced.
2 Associations begin to form by the action of the Association Formation P-OP.
Single atoms of same class as the larger component
Component of class belonging to set S
3 More associations form.
4 Links start to form in the copy, by the action of the Link Formation P-OP. Note that associations of the leftmost atom are broken when a link is formed.
5 More links form in the copy, each one causing one more association between the copy and the original to be destroyed.
6 When the copy is complete, the final association eventually breaks, leaving the copy free to move away.

Figure 7: Indefinite Hereditary Auto-Replication by Template Copying

```
for each compound:
with probability given by the function $F_{P\_diffusion}$
    • select a direction from {up, down, left, right}, each with probability 0.25
    • if
        – there is free space in the selected direction to enable to compound to move one square in that direction
    • then
        – move the compound one square in the selected direction
```

Figure 8: The *Diffusion* P-OP

```
for each component:
if
    • number of atoms in component $\geq N_{eop\_threshold}$
then, with probability given by the function $F_{P\_mutation}$
    • select an item from the component's state vector, each with equal probability
    • mutate the selected item
```

Figure 9: The *Mutation* P-OP

6.3 Emergent Operators (E-OPs)

As mentioned earlier, as well as the operation of the P-OPs, action may also come about in the world by components being interpreted as programs. In other words, components not only act as *operands*, but can generally also act as emergent *operators* (E-OPs).

Specifically, any component (a linked string of atoms of the same class) of length greater than or equal to a threshold specified by the system parameter $N_{eop\_threshold}$ is considered an E-OP. Every E-OP in the world is given its own state vector (Section 6.3.1), and at each step of the Nidus top-level control loop, the next N instructions are decoded from the component and executed, where N is given by the function $F_{N\_insts\_per\_timestep}(E_{class_i}, g)$[9]. Nidus has been designed so that the functionality available to components acting as E-OPs is easily extendible should that be required (see Section 6.3.3). The details of E-OPs are explained below.

6.3.1 The State Vector of a Component

Every E-OP has a vector associated with it, which is primarily used to provide it with some state. This means that the E-OP can perform actions based upon the results of previous operations. The contents of the state vector are listed in Figure 10. An explanation of each item in the state vector is given below.

[9] g is the Current Instruction Group, as specified by the State Vector. See Sections 6.3.1 and 6.3.3.

- **Instruction Pointer** (`int`)
- **Current Instruction Group** (`int`)
- **Flag** (`bool`)
- **Current Stack** (`int`)
- **Stack 1: Shape Specification** (`stack<bool>`, $length = N_{shape\_spec\_max\_length}$)
- **Stack 2: Class Specification** (`stack<bool>`, $length = n$)
- **Stack 3: Direction Specification** (`stack<bool>`, $length = 2$)
- **Energy Level** (`int`)

Figure 10: The State Vector

Instruction Pointer (`int`)
Indicates the position along the component which will be decoded to get the next instruction. After an instruction has been decoded, this pointer is incremented to point to the next unread position. If the pointer is incremented past the end of the component, it is reset to zero. Initial value is zero.

Current Instruction Group (`int`)
Indicates which Instruction Group is currently in use. Can be changed with the reserved instruction `switch_inst_grp` (see Section 6.3.3). Initially points to the Catalysis instruction group.

Flag (`bool`)
Generally used to indicate the successful completion of an instruction. Initial value is `false`.

Current Stack (`int`)
Indicates which of the three stacks listed below is currently active. Instruction Groups which make use of the stacks will contain commands to point *Current Stack* to any of the three stacks (see instructions in Table 1). Any command that attempts to perform an operation involving some other component will look at one or more of these stacks to obtain a specification of which other components it may act upon. Initially points to Stack 1.

Stack 1: Shape Specification (`stack<bool>`, $length = N_{shape\_spec\_max\_length}$)
An E-OP can push bits onto this stack to build up a binding site specification which will match a set of components. (A component matches the specification if its 'shape' string contains an identical substring to the specification—see Section 6.3.2). This stack is initially empty.

Stack 2: Class Specification (`stack<bool>`, $length = n$)
An E-OP can push bits onto this queue to build up a binary representation of a number which will specify one of the available classes. The length of the queue, n, is the smallest length such that $2^n \geq N_{classes}$.

Stack 3: Direction Specification (`stack<bool>`, $length = 2$)
An E-OP can push bits onto this queue to build up a binary representation of a direction (00, 01, 10 and 11 representing *up*, *left*, *down* and *right* respectively). This stack is also used to specify one of the four binding sites (labelled 0–3) for a catalyst E-OP (see Table 2).

Energy Level (`int`)
Each component has a certain minimum amount of energy associated with the links between its constituent atoms (E_{class_i} units of energy for each link). In addition, each component also has the capacity to store extra energy on top of this, up to some limit defined for each component (see 'Energy Storage' in Section 6.3.3). The current energy level of the component in excess of the minimum level associated with its links is stored in this state variable. Initial value is zero.

6.3.2 Component Representation, Decoding and Shape

A single atom is represented by a pair of numbers, one indicating the class to which it belongs, and the other indicating the particular variety within that class that it is an instance of.

When larger components form by atoms of the same class linking together (remember, this can only happen in the x direction), the string of numbers representing the varieties of the constituent atoms, starting with the leftmost atom and working rightwards, is called the component's '**shape**' string. The individual numbers in the shape string are actually represented in binary (in $N_{bits\_per\_inst}$ bits), and these are all concatenated. The shape string is therefore actually a single (possibly fairly long) binary string.

For decoding an E-OP, its *Instruction Pointer* points to a particular bit in its shape string. The next instruction is read off the E-OP simply by reading the next $N_{bits\_per\_inst}$ bits from the state string[10], and using a lookup table for the *Current Instruction Group* to ascertain which instruction this substring represents. When an instruction has been read, the *Instruction Pointer* is automatically incremented by $N_{bits\_per\_inst}$, and reset to zero if its value exceeds the maximum length of the state string. More is said about the decoding of E-OPs in Section 6.3.3.

The shape string is also used to determine whether a particular component can bind to a catalyst. This is explained in Section 6.3.3.

6.3.3 Instruction Groups

It has already been explained how an E-OP is decoded as a list of instructions to be executed. In this section I will say more about the available instructions themselves, and how they are arranged into Instruction Groups.

The design was motivated by the desire that the functional space available to the E-OPs through the instructions provided should be easily expandable. In other words, we should be able to add more instructions to the available set (or remove existing ones) very easily, without having to alter any other parts of the system.

In the design as it stands, *each variety of atom* effectively encodes *a single instruction.* If we were to straightforwardedly specify that each new instruction added to the instruction set had to be encoded for by one new variety of atom, then we have the problem that the number of varieties of atom must necessarily grow as the functional space is increased. Such an increase in the number of varieties may or may not have important consequences for the performance

[10] As the variety of each individual atom in the component is actually represented as an $N_{bits\_per\_inst}$-bit binary number, all we are doing during the decoding process is reading off a number representing the *variety of the next atom.* In other words, at least for this basic Nidus system, each atom encodes a single instruction.

of various aspects of the system, so I tried to devise a scheme whereby this necessary linkage between number of instructions and number of varieties could be avoided. The solution adopted was to divide the instructions into a number of *groups*, where only one group is active in a given E-OP at any one time. The following paragraphs explain this in more detail.

Before a Nidus run commences, the system is configured with a number of different instruction groups. Three groups are supplied by default (Catalysis, Control of Local Environment, and Energy Storage—all described below), but some of these may be removed, or more groups may be added. The groups are numbered consecutively from 0 to $N_{grps} - 1$ (where N_{grps} is the total number of groups available in the current configuration—three by default).

For each instruction group, a lookup table must be provided, mapping each of the numbers 1 to $(2^{N_{bits\_per\_inst}} - 1)$ to a single instruction. This may be a one-one or a many-one mapping (i.e. more than one number may represent the same instruction), but *every* number in the range must map to some instruction. Therefore, any single group may contain up to $(2^{N_{bits\_per\_inst}} - 1)$ different instructions. The lookup tables for the three default instructions groups are given in Appendix A.

None of the instruction groups may use the number 0 to represent an instruction, as this is reserved for the special instruction `switch_inst_grp` which is available no matter which instruction group is currently being used. This instruction will be explained shortly.

Each E-OP keeps a record (in its state vector) of which instruction group it is currently using. While it is being decoded one atom at a time as explained above, as long as the number read off its shape string is not 0, the lookup table of the current instruction group is used to decide which instruction the number represents.

If the number read off the shape string is 0, decoded as `switch_inst_grp`, then the behaviour is as follows. The next N bits are read from the shape string, where N is the smallest integer such that $2^N \geq N_{grps}$. These bits are interpreted as the binary representation of a group number. If this number is less than N_{grps}, then the *Current Instruction Group* member of the E-OP's state vector is updated to this new group number, and the Instruction Pointer is incremented in steps of size $N_{bits\_per\_inst}$ until it has passed the region on the shape string that was just used to determine the new group number. (If the number read off the state string after the `switch_inst_grp` instruction is greater than or equal to N_{grps}, then the Current Instruction Group remains unchanged.) Decoding of the shape string then proceeds as normal, using the appropriate lookup table for the new instruction group.

As mentioned earlier, three groups are supplied by default. These are described below:

- **Catalysis**

  In Section 6.2.2 it was explained how components belonging to the set of classes $\mathcal{S}$ can spontaneously form associations with other components. This spontaneous activity is the mechanism by which template replication is achieved.

  In addition to this P-OP, components of *any* class can also form associations with other components if they are E-OPs and they are using the *Catalysis* group of instructions. The way this can happen is as follows. Using the subgroup of instructions shown in Table 1, the E-OP can specify a particular set of components upon which it can operate. Specifically, the contents of Stack 1 (Shape Specification) is a bit string that is matched against every other component of class specified by the contents of Stack 2 (Class Specification) in the locality of the E-OP (in a direction specified by the contents of Stack 3). A match is successful if the other component contains the Shape Specification as a substring of its shape string.

`push_00`	push two 0s onto current stack
`push_01`	push a 1 followed by a 0 onto current stack
`push_10`	push a 0 followed by a 1 onto current stack
`push_11`	push two 1s onto current stack
`clear_stack`	erase contents of current stack
`stack_1`	stack 1 is now current stack
`stack_2`	stack 2 is now current stack
`stack_3`	stack 3 is now current stack

Table 1: The *Component Specification* Instruction Subgroup

An E-OP can actively form an association with a component specified in the manner just described, by issuing a `bind` instruction. If a component matching the specified class and shape is found in the locality, then an attempt is made to bind it to one of four possible binding sites on the E-OP (illustrated in Figure 11), as specified by the contents of Stack 3 (Direction Specification). If there is enough room to accommodate the component in the specified position, then it is moved into place and an association is made between a single atom in the E-OP (the leftmost atom for directions 0, 2 and 3, or the rightmost atom for direction 1) and a single atom of the other component (the leftmost atom for directions 0, 1 and 2, or the rightmost atom for direction 3).

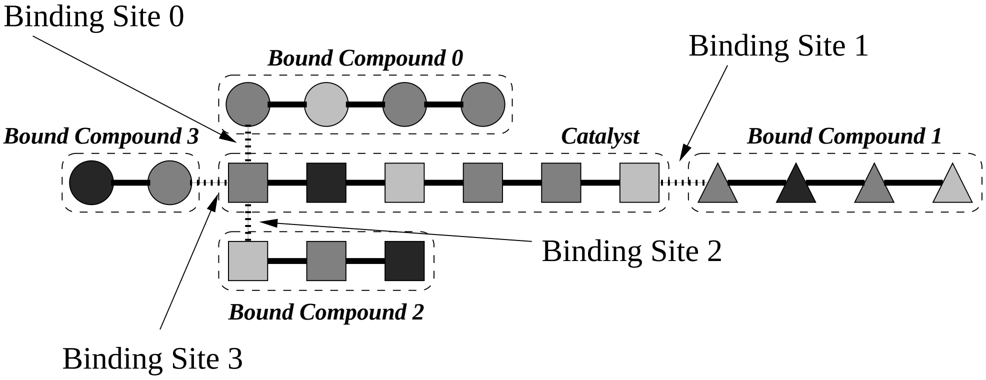
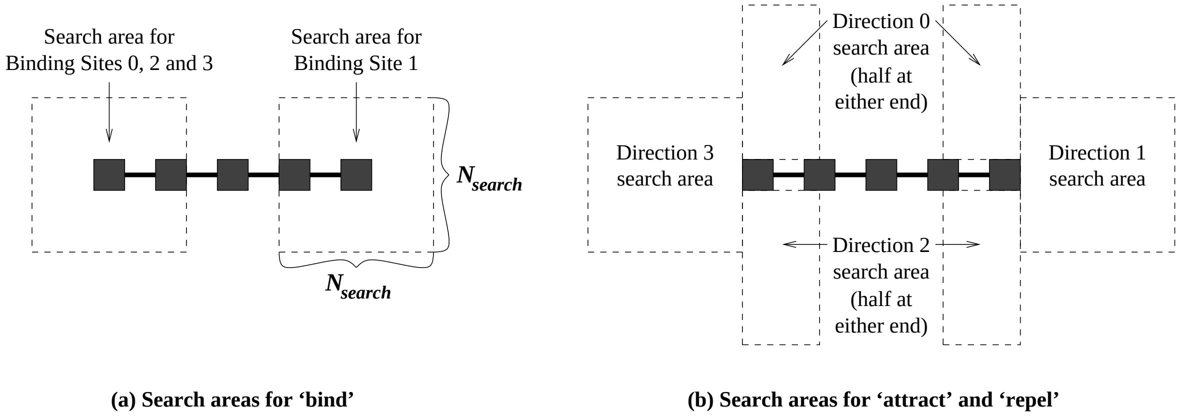


Figure 11: Available Binding Sites for a Catalyst E-OP

An E-OP using the Catalysis Group of instructions can then perform various operations on its associated components, using the instructions shown in Figure 2. Note that in contrast the the Association Formation P-OP, an E-OP acting as a catalyst can form associations with components of *any* class it specifies. Another important point to note is that the definition of 'locally available energy' for the link command does not just refer to local environmentally-conveyed energy, as it does for the Link Formation P-OP. Local environmentally-conveyed energy is the first source that is used to attempt to make the new link. However, if the environment does not contain enough energy, and if another component is bound to the catalyst in the binding site next to the two components trying to be linked (in a clockwise direction), then it attempts to donates the required energy to enable the link reaction to proceed. This is accomplished by first donating any excess energy it may have, and then, if this is insufficient, by breaking its own internal links one by one to release more energy. In an extreme case, this other energy-donating component may


Search area for Binding Sites 0, 2 and 3
Search area for Binding Site 1
N_{search}
N_{search}
Direction 0 search area (half at either end)
Direction 3 search area
Direction 1 search area
Direction 2 search area (half at either end)

(a) Search areas for 'bind'

(b) Search areas for 'attract' and 'repel'

Figure 12: Search Areas for `bind`, `attract` and `repel` E-OP commands

completely disintegrate, and still not be able to provide enough energy for the reaction. If this is the case, then the link attempt finally fails (but the donor component remains disintegrated).

By default, the *Current Instruction Group* member of an E-OP's state vector points to the Catalysis Group.

- **Control of Local Environment**
  Recall from Section 2.2 that our definition of life requires that a living organization is able to create a 'physical border' to distinguish itself from the environment. In Section 4.2 we relaxed this requirement slightly by saying that a living organization must at least have mechanisms by which it can control its local environment. This control is achieved in Nidus by the `attract` and `repel` instructions, shown in Table 3.

- **Energy Storage**
  Every component has a minimum energy level which corresponds to the number of links it contains multiplied by the energy required to make each link (E_{class_i}). Energy may be extracted from a component at its minimum energy level, but only by breaking links within it, and thereby breaking the component into smaller pieces. However, an E-OP may store a certain amount of energy in addition to its minimum level. The amount it can store is determined by the number of 1s in the binary representation of its shape string—one additional unit of energy may be stored for each 1 appearing in the shape string. The current amount of energy that an E-OP has stored in excess of its minimum level is recorded in its state vector member *Energy Level.* The E-OP can collect energy from its local environment using the `energy_collect` command, shown in Table 4. If this E-OP then binds to a another E-OP with catalytic activity, this excess energy is then available to drive reactions specified by the catalyst, without necessarily leading to the component's disintegration (as long as it's *excess* energy level remains above zero).

6.4 Top-Level Algorithm

To summarise the important events that occur when Nidus is running, the top-level algorithm that runs the main control loop is shown in Figure 13.

Component Specification Subgroup, plus:	
`bind`	1. clear flag 2. search for a component matching class specified by Stack 2 and shape specified by Stack 1, in direction specified by Stack 3 (see Figure 12 for details of area searched) 3. *if* matching component found, *and* there is space for it to bind at binding site specified by Stack 3, *then* move component to binding site, *and* form a single association between the catalyst and the component, *and* set flag
`release`	1. clear flag 2. *if* a component is bound at the binding site specified by Stack 3, *then* break the association *and* set flag
`link`	1. clear flag 2. *if* a component, A, is bound at the binding site specified by Stack 3, *and* a component, B, is bound at the next binding site (in clockwise direction), *and* A and B are of the same class, *and* local energy E_{class_i} is available to form a link (from local environment or from another bound component—see text), *and* there is room for B to link onto end of A furthest from A's point of attachment to catalyst, *then* break associations between A and catalyst and between B and catalyst, *and* move B to end of A furthest from A's point of attachment to catalyst, *and* form a new link between A and B, *and* subtract energy E_{class_i} from local environment, *and* set flag
`break`	1. clear flag 2. *if* a component is bound at the binding site specified by Stack 3, *then* break link in specified component halfway along its length, *and* release energy E_{class_i} to local environment, *and* break association between component *and* catalyst, *and* set flag
`if_bound`	*if* a component is *not* bound at the binding site specified by Stack 3, *then* increment Instruction Pointer by $N_{bits\_per\_inst}$
`ifn_bound`	*if* a component *is* bound at the binding site specified by Stack 3, *then* increment Instruction Pointer by $N_{bits\_per\_inst}$
`wait`	*if* flag is clear, repeat previous instruction, *else* do nothing

Table 2: The *Catalysis* Instruction Group

Component Specification Subgroup, plus:	
`attract`	1. clear flag 2. search for a component matching class specified by Stack 2 and shape specified by Stack 1, in direction specified by Stack 3 (see Figure 12 for details of area searched) 3. *if* matching component found, *and* there is space for it to move one square towards E-OP (in x or y direction), *then* move component one square towards E-OP, *and* set flag
`repel`	1. clear flag 2. search for a component matching class specified by Stack 2 and shape specified by Stack 1, in direction specified by Stack 3 (see Figure 12 for details of area searched) 3. *if* matching component found, *and* there is space for it to move one square away from E-OP (in x or y direction), *then* move component one square away from E-OP, *and* set flag

Table 3: The *Control of Local Environment* Instruction Group

`energy_collect`	*if* a unit of energy is available in the local environment, *and* component's energy level is not at its maximum, *then* remove one unit of energy from the local environment *and* increment the component's excess Energy Level by 1

Table 4: The *Energy Storage* Instruction Group

```
Initialise
currentTimeStep = 1
while (currentTimeStep < TimeStepLimit)
{
    For each *square* in the environment
      update energy from external source

    For each *component*
      if (length $\geq N_{eop\_threshold}$)
        execute N more instructions ($N = F_{N\_insts\_per\_timestep}(E_{class_i}, g)$)
        perform mutations
      endif

    For each *atom*
      perform primary operations relating to links and associations

    For each *compound*
      perform diffusions

    currentTimeStep += 1
}
```

Figure 13: The Top-Level Algorithm

6.5 Parameters

The parameters of the system are summarised in the following list. To keep the design as general as possible, many of these parameters are expressed as functions. In practice, the definitions of many of these functions may be very simple, and may not even use all of the information supplied to them.

- Fundamental Parameters:

  1. $\mathbf{N_x}, \mathbf{N_y}$
     The dimensions, expressed in number of squares, of the environment.
  2. $\mathbf{N_{classes}}$
     The number of classes. Each class is assigned an index value i, between 0 and $(N_{classes} - 1)$, and has two parameters associated with it:
     - $\mathbf{E_{class_i}}$
       The energy required to form a link between atoms of class i.
     - $\mathbf{N_{atoms_i}}$
       The number of atoms of class i in the environment.
  3. $\mathcal{S}$
     A set of class index numbers representing the classes upon which the Association Formation P-OP operates. See Section 6.2.2.
  4. $\mathbf{F_{P\_make\_link}}(E_{class_i}, E_{compound\_1}, E_{compound\_2}, E_{local\_env})$
     A function which when given the linkage energy associated with the class i of two components that are being considered by the Link Formation P-OP, together with

the total energy level of both of the components, and the total amount of energy available from the local environment, returns a number which is the probability that a link is formed between the two components. See Section 6.2.1.

5. $\mathbf{F_{P\_break\_link}}(E_{class_i}, E_{compound\_1}, E_{compound\_2}, E_{local\_env})$
As $F_{P\_make\_link}$, but for Link Destruction. See Section 6.2.1.

6. $\mathbf{F_{P\_make\_assoc}}(E_{class_i}, E_{compound\_1}, E_{compound\_2}, E_{local\_env}, variety1 == variety2)$
As $F_{P\_make\_link}$, but for Association Formation. See Section 6.2.2. This function takes an additional fifth argument, which is a boolean value to indicate whether the two atoms under consideration are of the same variety or not. For the indefinite hereditary auto-replication scheme to work, the probability returned by this function should be very high if the two atoms are of the same variety, and very low otherwise.

7. $\mathbf{F_{P\_break\_assoc}}(E_{class_i}, E_{compound\_1}, E_{compound\_2}, E_{local\_env})$
As $F_{P\_make\_link}$, but for Association Destruction. See Section 6.2.2.

8. $\mathbf{F_{P\_diffusion}}(x, y, t)$
A function which when given the coordinates (x,y) of a square in the environment, and a time step number t, returns the probability of diffusion for an atom at that place and time. When used by the Diffusion P-OP, x and y refer to the position of the top-leftmost atom in the compound under consideration. See Section 6.2.3.

9. $\mathbf{F_{P\_mutation}}(x, y, t)$
A function which when given the coordinates (x,y) of a square in the environment, and a time step number t, returns the probability that an E-OP with its leftmost atom at that place and time will suffer a mutation. See Section 6.2.4.

10. $\mathbf{F_{N\_energy\_flux}}(x, y, t)$
A function which when given the coordinates (x,y) of a square in the environment, and a time step number t, returns the amount of environmentally conveyed energy available at that place and time.

11. $\mathbf{N_{energy\_decay\_period}}$
The number of time steps that a given input of energy will survive for in a square in the environment before decaying (unless it has already been used to drive a reaction within this period). See Section 6.1.1.

12. $\mathbf{F_{N\_insts\_per\_timestep}}(E_{class_i}, g)$
A function which when given the linkage energy of class i and an instruction group number g, returns the number of instructions to be executed per time step for a single E-OP of class i with Current Instruction Group g.

13. $\mathbf{N_{bits\_per\_inst}}$
This parameter specifies how many bits are required to encode a single instruction. This determines how many varieties of atom exist in each class, $N_{varieties} = 2^{N_{bits\_per\_inst}}$. The number of entries in the lookup table for each Instruction Group is also equal to $N_{varieties}$.

14. $\mathbf{N_{eop\_threshold}}$
Components with this number of atoms or more are considered as E-OPs. Components of shorter length are not decoded, and do not have a state vector. See Section 6.3.

- Parameters associated with E-OP instructions:

1. $\mathbf{N_{shape\_spec\_max\_length}}$
   Specifies the maximum length of Stack 1 (Shape Specification) in the State Vector (see Section 6.3.1).
2. $\mathbf{N_{search}}$
   Determines the search area for the `bind`, `attract` and `repel` commands. See Section 6.3.3 and Figure 12.

7 A Mapping Between Nidus and the Real World

At this stage, it is worth stepping back from the details of the design, and remembering the original objectives that Nidus was intended to fulfill. The objectives were discussed in Section 4, and summarised in Figure 1. The way in which Nidus fulfills each of these objectives is summarised in Table 5, which shows, for each 'leaf node' in Figure 1, how it is fulfilled in Nidus, and also how it is fulfilled in the real world. This table therefore also gives a rough idea of the appropriate level of analogy to be drawn between Nidus and the real world for each of the features. Note however that the analogies are not exact; biomolecules, for example, are not the fundamental elements of the real world, but from the point of view of biological organization it is perhaps not too unreasonable to regard them as such.

8 Research Objectives

The major question that I wish to address with Nidus initially is whether the theory of the replicator-first evolution of living organization is supported by my assumptions listed in Section 4. The basic claims I am making are as follows:

if

- indefinite hereditary auto-replicators exist in an environment, and
- they have some catalytic activity,

then

- replicators which catalyse reactions which have the effect of making the replicator either (a) more stable, or (b) more accurate at replicating, or (c) faster at replicating, will be selected for. In particular, in the kind of environment described in Section 4,
  - the exchanges of energy and matter will lead to open-ended, hierarchical evolution, and
  - at some point during the course of evolution, the self-replicators will achieve a self-maintaining organization which satisfies my definition of life.

Some open questions, suggesting further directions for research with Nidus, include:

- To what extent are externally induced environmental perturbations necessary for the continued evolution of the system?
- To what extent is a heterogeneous environment (e.g. regions of different diffusion rates representing movement in fluids and on surfaces) required for the evolution of living organization?

Feature	**Nidus**	**Real World**
Aggregative matter	Yes (Atoms, non-atomic Components and Classes)	Yes (Biomolecules, Macromolecules and Classes (proteins, nucleic acids, polysaccharides, lipids etc.))
Aggregative reactions	Yes (Link Formation P-OP and `link` instruction for E-OPs)	Yes (Constructive biochemical reactions)
Degradative reactions	Yes (Link Destruction P-OP and `break` instruction for E-OPs)	Yes (Degradative biochemical reactions)
Conservation of matter	Yes	Yes
Specificity of reaction	Yes (P-OPs relating to Links and Associations only occur between components of same class. Reactions driven by E-OPs only occur between components matching the E-OPs' binding specifications)	Yes (Not all reactions are possible, determined largely by energy considerations)
Control of reaction	Yes (E-OPs acting as catalysts can increase the rate of specific reactions, and can also function as switches such that their action is conditional on some other compound being bound, or not being bound, at the same time)	Yes (Enzyme mediated reactions—which are play are far more important role than spontaneous reactions in biochemical systems. Enzymes act as catalysts for specific reactions, and can also act as switches such that their action is conditional on the presence or absence of some other molecule)
Spatial representation	Yes (Discrete, two dimensional)	Yes (Continuous, three dimensional)
Diffusion	Yes (Diffusion P-OP)	Yes (Tides, wind currents, etc.)
Environmentally conveyed energy	Yes	Yes (Electromagnetic radiation)
Energy associated with matter	Yes (Link energy, plus excess Energy Level in E-OPs)	Yes (Internal energy of molecules, charges associated with ions, etc.)
Exothermic reactions	Yes (All degradative reactions)	Yes
Endothermic reactions	Yes (All aggregative reactions)	Yes
Conservation of energy	Yes	Yes
Entropy increase in a closed system	Yes (All constructive reactions are endothermic, and all degradative reactions are exothermic)	Yes
External energy source	Yes (Flux of environmentally conveyed energy)	Yes (Sun)
External energy coupling	Yes (Environmentally conveyed energy may be used to form links, both with the Link Formation P-OP, and with the `link` command in Catalysis E-OPs)	Yes (Photosynthesis)
Internal energy coupling	Yes (Catalyst E-OPs may use energy stored in a bound component to drive reactions between other bound components)	Yes (Reactions associated with the ATP-ADP cycle, etc.)
Indefinite hereditary auto-replicators	Yes (Components belonging to classes in the set $\mathcal{S}$, which are subject to the Association Formation P-OP)	Yes (various hypotheses: RNA?, PNA?, clay?)
Control of local environment	Yes (E-OPs using the `attract` and `repel` commands)	Yes (Semi-permeable membranes)

Table 5: Mapping between Nidus and the Real World

- If there is a limit on the information carrying capacity of a single component in the system (e.g. if components above a certain length are unstable, or cannot be copied reliably), can evolution proceed by forming stable hypercycles, as suggested by Eigen and Schuster [15]?
- Some basic general laws have been proposed with respect to the emergence of energy currencies in living organizations [34]. Nidus can be used to test whether these really are general laws.
- One of the features of the Nidus design is that it allows the functional space available to the components in the system to be extended in a simple and incremental manner. A series of experiments can be conducted in which various features are added to the functional space (e.g. allowing environmentally transmitted information (c.f. light, sound etc.), active motility, etc.). There are many questions that could potentially be addressed with this kind of approach.

There are considerable problems with systems such as Nidus relating to analysing the resulting behaviour. This is particularly difficult because we are generally not interested in the presence of single components, but rather in the *interactions* that emerge between groups of components. It is difficult to devise an automated procedure for tracking such interactions. A number of different approaches are available to side-step this problem. For example, in McMullin's work on the α-Universes [32], he tracked the concentration of one specific component that was an essential part of the organization he was interesting in observing. It is also possible to track macroscopic measures of the whole system which might be expected to give some indication of interesting behaviour at the microscopic level. Dittrich and Banzhaf use this approach in their work [13]. Both of these approaches will be employed in Nidus to search for the most useful measures.

9 Relation to Other Work

A growing number of people have started looking at 'artificial chemistries' over the last five years or so, for a number of different reasons. Nidus bears some similarity to many of these, and also to work on artificial evolution, but there are also some important differences. These differences are due largely to different questions that each system has been designed to address, and also to the other design criteria (Section 5) that were important for Nidus.

In this section I list the work which I have come across which seems to be most closely related to Nidus, and for each one I give a brief indication of the important similarities and differences. The list is divided into work relating to software implementations of artificial worlds, and work of a more theoretical nature. I start the list with Nidus itself, to say a few words about what I think are the most important aspects of the system.

9.1 Software Implementations of Artificial Worlds

Taylor (Nidus) Nidus is based around a proposed list of the minimum properties that a world must satisfy if it is to be capable of supporting living organization. The list includes aspects of both matter and energy, and the definition of living organization adopted in the work regards cycles of both matter and energy as vital aspects of living organizations. The Nidus world satisfies all of the properties on the list, and will first be used to study replicator-first scenarios for the evolution of life (Section 8). This is, to my knowledge, the first system that will be used to look at how self-maintaining organizations may be built around an indefinite hereditary auto-replicator with catalytic activity as a result of the replicator *evolving* to become more stable—most of the other systems described here look for the *spontaneous emergence* of self-maintaining networks. Every attempt has been made to be explicit about all of the assumptions going into Nidus, so that they are open to discussion and criticism by others.

Barricelli Nils Aall Barricelli was, to my knowledge, the first person to actually run artificial evolution experiments on computers. He conducted a series of experiments, starting in 1953, to investigate the evolution of life (and not, as many others after him, purely to investigate the use of evolution as an optimisation technique) [4, 5, 6]. Barricelli did explicitly discuss in his work what it would take in addition to reproduction and random variation in order for us to consider his evolved organisms 'alive'. In fact, he decided that the term 'life' was too poorly defined to be of use in the context. Barricelli introduced the concept of symbiogenesis[11] in his work as an additional requirement for his organisms (which he then called 'symbioorganisms'), and, instead of asking whether these symbioorganisms were alive, asked the reciprocal question of "whether the objects we are used to call[ing] living beings are a particular class of symbioorganisms" [6] (p.7). This is a useful tactic, and in fact is the same tactic that I am using in my work. However, my 'minimal requirements' (Section 4) are somewhat more extensive than Barricelli's, as I believe that it is not possible to study many of the interesting question of life from within the symbioorganism framework[12]. Having said this, the design of Nidus has certainly been influenced by Barricelli's attempts to at least *propose* an explicit list of the assumptions behind the model, and also by the simplicity of the model he developed. Also, Barricelli conducted some experiments in which the individual symbioorganisms were also decoded into a strategy for playing a simple game (i.e. they had a phenotype) [6]. When two symbioorganisms were competing to reproduce into the same space, they played the game according to their individual strategies, and the winner was allowed to reproduce. This idea of decoding the reproducing organisms, or giving them a phenotype, can be seen as a forerunner to the idea of E-OPs in Nidus.

Conrad and Pattee Michael Conrad and H.H. Pattee described an early model in which individual organisms, with a genotype representation and a phenotype obtained by interpreting the genotype as instructions, compete in a one-dimensional world for the possession of 'chips' which they use for self-repair and reproduction [10]. It is closely related in many ways to Ray's later Tierra model, but with a smaller number of instructions representing a limited set of possible interactions between organisms, rather than a computationally complete instruction set as in Tierra (although even in Tierra there is still only a limited number of types of interaction between organisms). Also, it has a notion of conservation of matter, lacking in Tierra, to model ecosystem interactions. Having said this, there is only one type of matter in the model (a 'chip'), and it has a fairly arbitrary connection to the structure of an organism. For example, an organism's genome is represented as a string of 'states' rather than a string of matter—an organism's store of chips is only used to determine when it can repair itself and when it can reproduce. The major consequence of disassociating the structure of the organism from the

[11] According to Barricelli, the symbiogenesis theory claims that if genes are to evolve into "relatively higher forms of life" [4] (p.145), they must only be able to reproduce through a symbiotic relationship with other genes. I think that there is an interesting parallel between this theory and Eigen and Schuster's hypercycle model [15].

[12] But I do regard the interactions between components, and the exchanges of matter and energy between organizations and their environment, as being of fundamental importance to the evolution of life. In some ways, it could be said that I am trying to make the symbiogenesis model more explicit, or to 'flesh out' the details with specific regard to the origin and evolution of living organization.

‘matter’ in the world is that the structure must therefore be predefined and is not able to evolve, whereas, had it been embedded in the material world, new organism structures could emerge from new organizations of the matter. This problem of predefining a non-material structure for organisms is shared by Tierra, ECHO and other models (see below), and was one of the main objectives in the design of Nidus was to avoid this (see Section 4).

Holland and McMullin (α-Universes) John Holland proposed the α-Universes as a suitable environment in which to study the spontaneous emergence of self-reproducing systems [20]. Holland’s original work was based upon a mathematical analysis of phenomena that he expected to emerge in the system. Fifteen years later, the α-Universes were implemented as a computer program by Barry McMullin [32]. McMullin found a number of problems with the design that were not anticipated by Holland and which meant that it did not produce the ‘life-like’ behaviour that he postulated. As many of the problems were ultimately due to components in the world being unable to control their local environment, McMullin has subsequently gone on to investigate software implementations of autopoiesis [33]. The design of Nidus owes a considerable amount to Holland’s original design and McMullin’s subsequent investigations. Indeed, even the terminology of ‘primary operators’ and ‘emergent operators’ has been borrowed. Differences between Nidus and the α-Universes include: Nidus has the concept of two sorts of bond (associations and links) rather than just one; the emergent operators in Nidus can perform a much wider variety of tasks than they can in the α-Universes, because they are translated as programs rather than having to match a limited number of predefined ‘templates’, and; Nidus includes emergent operators for allowing components to control their local environment (this could have been added to the α-Universes, but was not included in the original formulation).

Holland (ECHO) After the α-Universes, Holland developed the ECHO model of complex adaptive systems [21, 22, 23]. Although Nidus is more similar to the α-Universes than it is to ECHO, some of the ideas behind Nidus and ECHO are the same. Most important is the notion, shared by both models, that it is the ‘market’ that emerges from exchanges of resources between individual agents that is the source of much of the interesting behaviour of a complex system. However, ECHO models individual agents at a somewhat higher level than does Nidus: in ECHO agents have a predefined structure and can participate in a limited set of interactions with other agents; in Nidus, an equivalent ‘agent’ might be a self-maintaining set of components. As such, is not restricted to an organization which has been predefined by the designer, and the interactions which may emerge between ‘agents’ in Nidus is also less restricted.

Ray (Tierra) As explained in Section 2, Nidus was developed to overcome some of the problems I perceived with my earlier work with Cosmos [47], which was itself a development of Tom Ray’s Tierra model [40]. The idea in Nidus of treating components in the system as computer programs, and running a few instructions on each component at each time step, comes from Cosmos and Tierra. By only running a limited number of instructions from each component at each time, we do not run into the halting problem, which is a potential problem for other kinds of program evolution system. Many of the differences between Tierra and Nidus have already been discussed (Section 2). Perhaps the most important differences are the following: In Nidus, self-reproduction comes about by the operation of basic rules (the Association Formation P-OP, Section 6.2.2) whereas in Tierra a program must encode a self-reproduction *algorithm.* There is no reason to suppose that a step-by-step mutational path generally exists from one self-reproduction algorithm to another, where each step encodes an algorithm which is better (or at least as good) as the previous one (see Section 9.2.2). Also, unlike in Nidus, Tierra does not have the notion of conservation of matter—a program can write a copy of itself into another part of the computer's memory without first having to 'collect' the individual instructions that make up the copy from other parts of the memory. There is therefore no competition for materials, so that the very notion of a Tierran 'organism' being self-maintaining or autopoietic is problematic. The only competition that exists in Tierra is for the globally-available resources of memory and CPU-time. The expectation is that by introducing the concept of local competition for materials and energy in Nidus, if living organizations do emerge then food webs, trophic levels and hierarchical evolution will also emerge in the long run.

Fontana, Buss et al. (AlChemy etc.) Walter Fontana and colleagues have produced some of the best recent work on artificial chemistry [16, 18, 17]. They argue that a formalism is needed in biology (and other areas) for constructive systems (i.e. those where the components are objects whose structure can change as the result of interactions). This should be coupled with classical dynamic systems approaches to form a *constructive dynamic systems theory*. In their work, chemical molecules are represented using a formalism such as λ-calculus or linear logic. Although these representations (especially molecules as proofs in linear logic) turn out to have many desirable properties, they are not particular compact representations, and the processes involved in deciding the products of a reaction may be somewhat long-winded (e.g. involving a procedure for normalisation or cut elimination on a term). With Nidus the emphasis has been on designing a system with a compact representation and simple operations, so that it is feasible to study large numbers of components interacting over long periods of time. Other differences include the lack of spatial representation in Fontana et al.'s work (the reactions occur in a 'well-stirred tank'), no explicit representation of energy, and, in their work with the λ-calculus at least, the lack of conservation of matter. The lack of spatial representation in this work, as in most of the other work described in this section, means that there can be no notion of *individuality* in the organizations which emerge. The system can therefore not be used, at least in its present form, to model individual organisms in an evolving population.

Banzhaf, Dittrich et al. (BinSys etc.) Wolfgang Banzhaf, Peter Dittrich and colleagues are working on a catalytic self-organising reaction system of binary strings [3, 13]. Their most recent work involves the decoding of these strings as programs which determine how one string reacts with another, which is a very similar concept to the idea of E-OPs in Nidus. However, the decoded binary string performs operations that determine the product of the reaction by directly changing bits within it, rather than in Nidus where an E-OP acting as a catalyst can just make or break links between specific components. This means that Dittrich and Banzhaf's work does not have the same idea of conservation of matter as does Nidus, because a reaction can produce an arbitrary new binary string which is not necessarily of simple composition of the reactants (i.e. there are no indestructible 'atoms'). Most of their published work also differs from Nidus in that it does not have a spatial representation (the reactions occur in a well-stirred tank). An interesting discussion has been presented about adding an efficient topological structure to the model using a hashing algorithm [12], although this method is not ideal for my purposes. For example, components can still be grouped together in bins, so they still do not necessary have a unique position, and the hashing function presented is based upon the binary strings themselves, so that any particular binary string must always be placed in the same bin (if it is not full) rather than having the freedom to move around.

Ikegami and Hashimoto (Machine-Tape Coevolution) Takashi Ikegami and Takashi Hashimoto have described a system [25, 26] which is somewhat similar to that of Dittrich and

Banzhaf [13] in that a population of binary strings are evolving which are translated into machines. A major difference is that the binary strings which are treated as machines evolve in a separate population to the binary strings which are treated as tapes (i.e. input for the machines). There is no spatial representation in the model, and all machines can potentially interpret all tapes (if they fulfill a match criterion). Also, as in Dittrich and Banzhaf's work, the notion of conservation of matter is not as strong as it is in Nidus.

Yamamoto and Kaneko (Tile Automaton) The 'Tile Automaton' has been introduced as a "model for the origin of life" by Tomoyuki Yamamoto and Kunihiko Kaneko [50]. Their model (specifically the spatial version of it) is basically similar to a cellular automata (CA), but with the extra concept of each non-empty state (tile) having a velocity which determines its motion. Colliding tiles react according to predefined rules (c.f. the rule table of CAs). This model therefore does have a spatial representation, but, in common to many CA-based models, does not have the concept of conservation of matter (the number of cells in non-quiescent states, i.e. the number of tiles, is not constant).

Boerlijst and Hogeweg This work is based on a CA model of pre-biotic spatial self-structuring among self-replicative models that are linked cyclically by catalysis [7]. It is a simple and elegant model of hypercycles, but, from my point of view, is lacking a number of features to make it useful for a more general study of the origin and evolution of life (but of course this was not what it was designed to do). Most importantly, it is not a *constructive* system, i.e. elementary atoms cannot combine to perform operations not available in the original set.

Yoshii, Inayoshi and Kakazu (Atomoid) Atomoid [51] is a reaction-formation model which shares the basic design concepts of Nidus, to model the essential properties of the real world in order to study self-organization and evolution. Atomoid, like Nidus but unlike most other systems, also models energy transformations. However, the representation of atoms in Atomoid is somewhat different, as the analogy is with *atoms* in the real world, rather than with *biomolecules* as in Nidus. A big difference between the two systems is that Atomoid tends to model reaction rules and energy transformations in a way that much more literally resembles real chemical reactions, whereas in Nidus it is claimed that many of these details are not important for the sorts of questions it was designed to investigate, and more weight has been given to producing a compact representation and avoiding time-consuming calculations during reactions.

Suzuki, Tsumoto and Tanaka (ARMS) This work proposes an Abstract Rewriting System on Multisets (ARMS) as a framework for investigating the emergence of reaction cycles [46]. In ARMS, unlike in Nidus, the allowable reactions are explicitly provided by the designer in the form of a collection of rewrite rules. There is also a strict order in which the rules are applied to the components in the system. In contrast, only the basic reactions are defined in Nidus, but E-OPs can evolve to perform many other tasks. Also, ARMS does not have a spatial representation and there is no notion of conservation of matter.

9.2 Theory

9.2.1 Cellular Automata and Self-Reproduction

Nearly all of the practical implementations of artificial worlds just described are related in some way to the seminal work of John von Neumann. In the late 1940s and early 1950s, von

Neumann devoted considerable time to the question of how complicated machines could evolve from simple machines. Some of his work on this subject appears in [2]. The original model considered by von Neumann was a constructive system, which Burks has called both the 'robot model' and the 'kinematic model' [2] (p.374), which in many respects is similar to much of the recent work on constructive systems described above (although somewhat more complicated). However, von Neumann decided that the system was too complicated to capture in a set of rules that were both simple and enlightening, so he turned his attention to developing the cellular automata (CA) framework with Stanislaw Ulam. With this framework, he used the idea of a Universal Constructor (a machine embedded within a CA which, when fed a tape containing suitable instructions, could construct any other cell assembly) to investigate self-reproduction and evolution.

Von Neumann's work concentrated on the *logic* required for a self-reproducing machine to be able to evolve increased complication. He therefore did not specifically deal with various biological concerns, most notably concerns of energy. Burks says of the kinematic model that "von Neumann intended to disregard the fuel and energy problem in his first design attempt. He planned to consider it later, perhaps by introducing a battery as an additional elementary part" [2] (p.485). Another major difference between biological organisms and von Neumann's self-reproducers is the capacity of the former, but not the latter, for self-maintenance in the face of environmental perturbations. Alvy Ray Smith has pointed out [44] that some of the CA-based self-reproduction models developed by von Neumann and more recently by others are very non-biological in other ways as well (e.g. reproduction in CA models does not occur by development from an 'egg', most models suffer from 'overcrowding' such that an individual self-reproducer can only reproduce once (or a small number of times) before it runs out of space in which to place its offspring, etc.).

Much of the recent work concerning self-reproduction in CAs (e.g. [28, 24, 48, 36, 39]) also has this 'non-biological' character. In fact, as pointed out by Barry McMullin, much of the recent work in this area does not even seem to share von Neumann's concern with the evolution of increased complication, but addresses the 'problem' of self-reproduction in and of itself [32]. On top of this, these studies do not generally consider the ability of the automata to actively maintain its own structure in the face of environmental perturbations. This deficiency has certainly been recognised for a long time (e.g. [1] and, more recently, [32]), but very little work has so far been done to create more robust self-reproducing CAs. Only when such considerations are included in our models can we expect there to be selection pressure for self-reproducers with the ability of self-maintenance, leading to the evolution of living organization.

9.2.2 The 'problem' of trivial self-reproduction

As hinted at in Section 2, the concept of *self*-reproduction as distinct from other sorts of reproduction is not as well defined as it might seem. In fact there are at least three, fairly independent ways of categorising any process of reproduction, as I will describe in this section. The term 'self-reproduction' is used in a number of different ways, and often blurs the distinction between these methods of categorisation.

John von Neumann's work on self-reproduction in CAs concerned the question of how machines might be able to evolve increased complication in order to perform any conceivable kind of computation. This is why his design for a self-reproducing machine had to be capable of universal construction (and therefore universal computation), and why it was designed in such a way that it could withstand some kinds of mutation so that it could evolve into a machine that performed other tasks as well as universal construction.

Much of the recent work on self-reproduction in CAs, starting with [28], has dropped this requirement for universal construction, but, as mentioned above, at the price of the reproducing machines being able to evolve by stepwise mutations into an unlimited variety of viable forms. Also, having dropped the requirement for universal construction, it is hard to offer a specification for what the self-reproducing should be like in order to avoid the 'problem of trivial self-reproduction'. Trivial self-reproduction occurs when reproduction of a particular sort of configuration happens purely due to the rules of the system rather than to anything explicitly encoded in the configuration itself. For example, a CA with a transition rule such as "if this cell is empty (in the quiescent state) and one of its neighbouring cells in in state $\mathcal{A}$, then change the state of this cell to $\mathcal{A}$" is an environment in which the state $\mathcal{A}$ trivially self-reproduces.

However, I think that with regard to the evolution of life, the issue of trivial reproduction is something of a red herring. When looking at any sort of reproduction, I think it is useful to look at the process by which reproduction is accomplished in three different ways:

1. The degree to which the *algorithm* for reproduction (the way in which the process is controlled) is *explicitly encoded* on the configuration being reproduced, rather than being *implicit in the physical laws* of the world (e.g. the transition table in a CA).

2. Whether reproduction happens purely by the action of the physical laws of the world on the configuration to be reproduced (*auto*-reproduction), or whether it requires auxiliary physical machinery as well (*assisted*-reproduction).

3. The number of different configurations that exist, connected by mutational pathways, that are capable of reproducing their specific form (i.e. the distinction between *limited hereditary* reproducers and *indefinite hereditary* reproducers).

The distinction between auto- and assisted-reproduction is a dichotomy, but the other two distinctions each define a spectrum of possibilities. The distinctions are generally independent of each other, although the more explicitly encoded the reproduction algorithm is, the less likely it is to be an indefinite hereditary reproducer (because of the decreasing number of mutational pathways from one viable reproduction algorithm to another).

In Figure 14 I have categorised some of the reproducers that have been discussed so far according to each of these three distinctions. Some points to note about this figure are:

- Tierran organisms and von Neumann's Universal Constructor are placed midway along the limited–indefinite hereditary scale because, although both representations are *capable* of supporting universal computation in principle, only mutations which retain the ability to reproduce will be *viable*,

- Trivial CA self-reproducers are, in general, limited hereditary reproducers, because even though a single state may be able to reproduce, a compound set of states will usually not be able to reproduce as a whole,

- The distinction between trivial and non-trivial self-reproduction that preoccupies some recent CA work is a distinction on the implicit–explicit axis, but from the point of view of the evolution of living organization, I think that the other two axes are more important:-

- Most importantly, I have placed the desirable 'seed for evolution of life' in the auto–implicit–indefinite hereditary corner of the space. The seed should be auto-reproducing (i.e. not rely upon auxiliary machinery) if it is to have a reasonable chance of spontaneously emerging, and it should be an indefinite hereditary reproducer (and therefore not explicitly

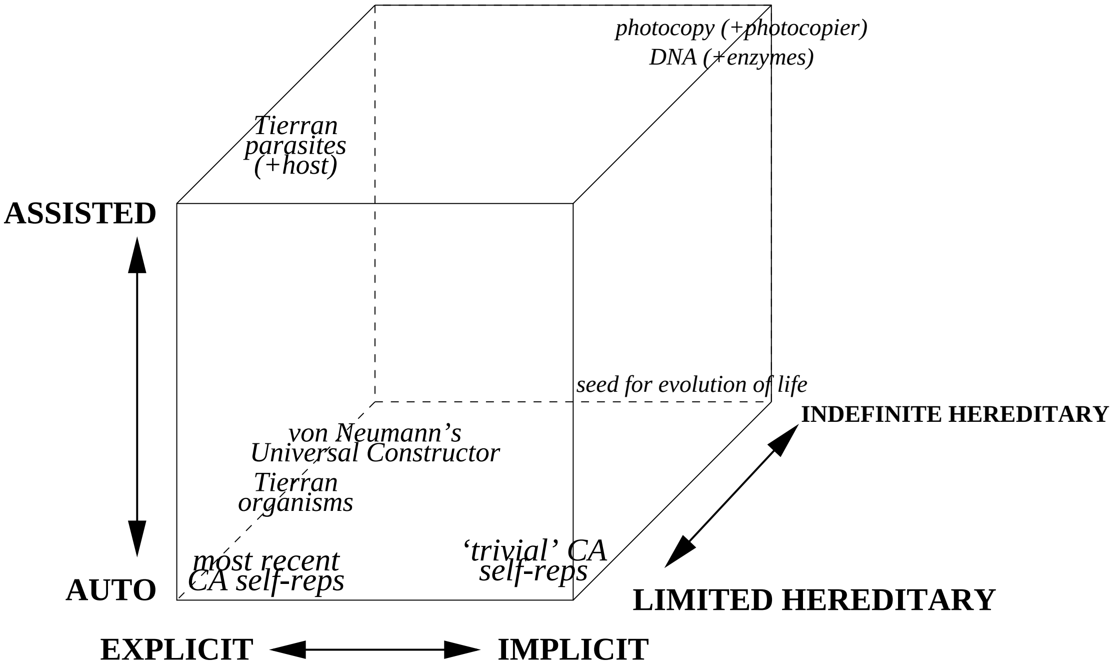


Figure 14: Categorisation of Reproducers

encoded) to support an on-going, open-ended evolutionary process. Of course, the claim that this is the sort of seed we require for the origin of life is controversial (see Chapters 2–5 of [31] for arguments for and against this sort of idea), but it is my intention with Nidus to investigate whether or not it is *possible* that such a seed could have supported the evolution of living organization.

Although my claim is that the seed should not be an explicit encoding of the self-reproduction algorithm, I am not saying that it should be inert. On the contrary, to reiterate the requirements listed in Sections 4.2.1 and 8, the seed *must* also have some *catalytic activity.* Thus, in common with von Neumann's treatment of self-reproduction, in this scenario a self-reproducer acting as a seed for the evolution of life has two aspects: (1) it gets copied (during which process it is treated as *data*), and (2) it has specific activity associated with it (i.e. it is also treated as a machine or *operator*). The difference is that von Neumann-style self-reproduction requires that the action of the self-reproducer as an operator must include explicit instructions for its own reproduction, whereas the kind of seed I am talking of can encode *any* sort of catalytic activity, because it gets reproduced implicitly. The idea is that reproducers that encode catalytic activity that help them become better reproducers *in any way at all* will be selected for. As evolution proceeds, a growing phenotype of associated, catalysed processes will become connected to the reproducers, as selection picks out the variants that are best able to compete, survive and reproduce.

The scenario just described, in which a population of operators (programs) which can perform arbitrary tasks are being reproduced implicitly by the laws of the world, is a fairly accurate description of Genetic Algorithms or Genetic Programming. The difference is that for the evolution of living organization, the sorts of processes that can become associated with (catalysed by) the reproducers should be processes of aggregation and degradation of other components, energy transductions and so on, or in other words, the sorts of processes listed in Section 4. This again demonstrates why it is important when building models of the origin and evolution of life to describe the sorts of processes and interactions that can happen in the world, and not simply the mechanisms for evolution.

9.2.3 Living organization

Chris Langton has argued that CAs, or, more generally, aggregate systems, can indeed be useful for studying most of the other important functions performed by living organisms, and lists what sorts of functions these are [29] (although Langton again ignores energetic considerations, concentrating on the 'logic of life'). However, there is still no simple, principled way to program the transition table of a CA in such a way that all of these higher-level phenomena can emerge. Some recent work has been published on *evolving* the transition tables in a non-uniform CA [43], or, in other words, of automatically programming it. If we consider that Nidus is an aggregate system not too distantly related to a non-uniform CA, then the emergence and selection of E-OPs in Nidus can similarly be viewed as a process of (at least partially) evolving the transition tables.

More recently, Federico Morán, Alvaro Moreno and colleagues have published some very interesting work on the question of providing a realistic description of the essence of life [35, 34]. This work has already been referred to in Sections 2.2 and 8. In [34] a useful comparison is presented of various 'origin of life' models (Rosen's (M,R)-system, Kauffman's autocatalytic networks, Eigen's hypercycle, and Maturana and Varela's autopoietic model). They point out that each model relies "on several types of biochemical interaction, specifically those of reaction, diffusion, catalysis, and template replication" (p.217). Furthermore, catalysis and template replication may be considered as special types of reaction, so each model can be compared directly "by reducing the complex events in each to elementary reaction and diffusion events" (p.217). Nidus differs from the models discussed in [34] in that it has both a spatial representation, and also models energetic coupling and transductions within and between organisms and their abiotic environment.

9.2.4 Hierarchical evolution

During the course of my work, my ideas of how the evolution of biological life has proceeded on Earth have themselves evolved somewhat. The sort of picture I now have is of an oscillation between 'divergent' and 'convergent' phases (very similar to a model proposed by Werner Schwemmler [42]). During the divergent phase, there is a rapid expansion of new forms of organism, as previously unoccupied ecological niches are filled. Then, as suggested by Stanley Salthe, "the production of specific kinds proceeded apace in the process we call organic evolution until it approached saturation of number of kinds... when the emphasis of the process of complication shifts towards coevolutionary elaborations of pairs, guilds, and even more complex symbioses" [41] (pp.252–253). This is what I call the convergent phase. The convergent phase continues until a symbiotic relationship is discovered of such a nature that a number of individual replicators become so closely related that they lose the ability to replicate individually but must do so as a whole. This assembly therefore collectively becomes a new unit of selection for evolution, of hierarchically greater complexity than the previous units of selection. Having made this 'major transition' [31], the new organism can then seed a new phase of divergent evolution, rapidly filling novel niches associated with its new way of life. I have come to this picture of evolution largely through reading the work of Maynard Smith and Szathmáry [31], Salthe [41], Buss [8], O'Neill et al. [38], Gould [19] and Schwemmler [42], and I agree with Stewart [45] that the synthesis of these major evolutionary transitions is a major challenge for artificial life. I believe that the sort of constructive reaction-diffusion system described in this document should be capable of supporting this type of evolutionary process.

References

[1] Michael A. Arbib. Self-reproducing automata—some implications for theoretical biology. In C.H. Waddington, editor, *Towards a Theoretical Biology*, volume 2, pages 204–226. Edinburgh University Press, 1969.

[2] W. Aspray and A. Burks, editors. *Papers of John von Neumann on Computing and Computer Theory.* MIT Press, 1987.

[3] Wolfgang Banzhaf. Self-organisation in a system of binary strings. In R. Brooks and P. Maes, editors, *Proceedings of Artificial Life IV*, pages 109–118, Cambridge, MA, 1994. MIT Press.

[4] Nils Aall Barricelli. Symbiogenetic evolution processes realized by artificial methods. *Methodos*, 9(35-36), 1957.

[5] Nils Aall Barricelli. Numerical testing of evolution theories. Part I. Theroetical introduction and basic tests. *Acta Biotheoretica*, XVI(1/2):69–98, 1962.

[6] Nils Aall Barricelli. Numerical testing of evolution theories. Part II. Preliminary tests of performance. Symbiogenesis and terrestrial life. *Acta Biotheoretica*, XVI(3/4):99–126, 1963.

[7] M.C. Boerlijst and P. Hogeweg. Spiral wave structure in pre-biotic evolution: Hypercycles stable against parasites. *Physica D*, 48:17–28, 1991.

[8] Leo W. Buss. *The Evolution of Individuality.* Princeton University Press, 1987.

[9] A.G. Cairns-Smith. *Seven Clues to the Origin of Life.* Cambridge University Press, 1985.

[10] Michael Conrad and H.H. Pattee. Evolution experiments with an artificial ecosystem. *Journal of Theoretical Biology*, 28:393–409, 1970.

[11] Richard Dawkins. *The Selfish Gene.* Oxford University Press, Oxford, 2nd edition, 1989.

[12] Peter Dittrich and Wolfgang Banzhaf. A toplogical structure based on hashing - emergence of a 'spatial' organisation. Poster presented at the Fourth European Conference on Artificial Life (ECAL97), Brighton, U.K., 1997.

[13] Peter Dittrich and Wolfgang Banzhaf. Self-evolution in a constructive binary string system. Preprint, University of Dortmund, Department of Computer Science, April 1998.

[14] Freeman J. Dyson. *Origins of Life.* Cambridge University Press, 1985.

[15] Manfred Eigen and Peter Schuster. *The Hypercycle.* Springer, Berlin, 1979.

[16] Walter Fontana. Algorithmic chemistry. In CG Langton, C Taylor, JD Farmer, and S Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 211–254, Redwood City, CA, 1991. Addison-Wesley.

[17] Walter Fontana and Leo W. Buss. The barrier of objects: From dynamical systems to bounded organizations. In J. Casti and A. Karlqvist, editors, *Boundaries and Barriers*, pages 56–116. Addison-Wesley, 1996.

[18] Walter Fontana, Günter Wagner, and Leo W. Buss. Beyond digital naturalism. *Artificial Life*, 1:211–227, 1994.

[19] Stephen Jay Gould. *Wonderful Life: The Burgess Shale and the Nature of History.* Penguin Books, 1989.

[20] John H. Holland. Studies of the spontaneous emergence of self-replicating systems using cellular automata and formal grammars. In A. Lindenmayer and G. Rozenberg, editors, *Automata, Languages, Development*, pages 385–404. North-Holland, New York, 1976.

[21] John H. Holland. Echo: Explorations of evolution in a minature world. In J.D. Farmer C.G. Langton and J. Doyne, editors, *Proceedings of the Second Conference on Artificial Life*, Redwood City, CA, 1990. Addison-Wesley.

[22] John H. Holland. *Hidden Order: How Adaptation Builds Complexity.* Addison-Wesley/Helix Books, 1995.

[23] Peter T. Hraber, Terry Jones, and Stephanie Forrest. The ecology of Echo. *Artificial Life*, 3(3):165–190, 1997.

[24] Jesús Ibáñez, Daniel Anabitarte, Iker Azpeitia, Oscar Barrera, Arkaitz Barrutieta, Haritz Blanco, and Francisco Echarte. Self-inspection based reproduction in cellular automata. In F. Morán, A. Moreno, J.J. Merelo, and P. Chacón, editors, *Advances in Artificial Life: Third European Conference on Artificial Life*, Lecture Notes in Artificial Intelligence, pages 564–576, Berlin, 1995. Springer.

[25] Takashi Ikegami and Takashi Hashimoto. Coevolution of machines and tapes. In F. Morán, A. Moreno, J.J. Merelo, and P. Chacón, editors, *Advances in Artificial Life: Third European Conference on Artificial Life*, Lecture Notes in Artificial Intelligence, pages 234–245, Berlin, 1995. Springer.

[26] Takashi Ikegami and Takashi Hashimoto. Replication and diversity in machine-tape coevolutionary systems. In C.G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, Cambridge, MA, 1997. MIT Press.

[27] Stuart A. Kauffman. Autocatalytic sets of proteins. *Journal of Theoretical Biology*, (119):1–24, 1986.

[28] Christopher G. Langton. Self-reproduction in cellular automata. *Physica D*, (10):135–144, 1984.

[29] Christopher G. Langton. Studying artificial life with cellular automata. *Physica D*, (22):120–149, 1986.

[30] Humberto R. Maturana. The organization of the living: A theory of the living organization. *International Journal of Man-Machine Studies*, 7:313–332, 1975.

[31] John Maynard Smith and Eörs Szathmáry. *The Major Transitions in Evolution.* W.H. Freeman, Oxford, 1995.

[32] Barry McMullin. *Artificial Knowledge: An Evolutionary Approach.* PhD thesis, Department of Computer Science, University College Dublin, October 1992. URL: `ftp://ftp.eeng.dcu.ie/pub/alife/bmcm_phd/`.

[33] Barry McMullin and Francisco J. Varela. Rediscovering computational autopoiesis. In P. Husbands and I. Harvey, editors, *Fourth European Conference on Artificial Life*, pages 38–47. MIT Press/Bradford Books, 1997.

[34] Federico Morán, Alvaro Moreno, Eric Minch, and Francisco Montero. Further steps towards the realistic description of the essence of life. In C.G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 216–224, Cambridge, MA, 1997. MIT Press.

[35] Alvaro Moreno, Arantza Etxeberria, and Jon Umerez. Universality without matter? In R. Brooks and P. Maes, editors, *Artificial Life IV*, pages 406–410. The MIT Press, 1994.

[36] Kenichi Morita and Katsunobu Imai. A simple self-reproducing cellular automaton with shape-encoding mechanism. In C.G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 450–457, Cambridge, MA, 1997. MIT Press.

[37] Juan C. Nuño, Pablo Chacón, Alvaro Moreno, and Federico Morán. Compartimentation in replicator models. In F. Morán, A. Moreno, J.J. Merelo, and P. Chacón, editors, *Advances in Artificial Life: Third European Conference on Artificial Life*, Lecture Notes in Artificial Intelligence, pages 116–127, Berlin, 1995. Springer.

[38] R.V. O'Neill, D.L. DeAngelis, J.B. Waide, and T.F.H. Allen. *A Hierarchical Concept of Ecosystems.* Princeton University Press, Princeton, NJ., 1986.

[39] Jean-Yves Perrier, Moshe Sipper, and Jacques Zahnd. Toward a viable, self-reproducing universal computer. *Physica D*, (97):335–352, 1996.

[40] Thomas S. Ray. An approach to the synthesis of life. In C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 371–408. Addison-Wesley, Redwood City, CA, 1991.

[41] Stanley N. Salthe. *Evolving Hierarchical Systems: Their Structure and Representation.* Columbia University Press, New York, 1985.

[42] Werner Schwemmler. *Symbiogenesis: A Macro-Mechanism of Evolution.* Walter de Gruyter, Berlin, 1989.

[43] Moshe Sipper. Co-evolving non-uniform cellular automata to perform computations. *Physica D*, (92):193–208, 1996.

[44] Alvy Ray Smith. Simple nontrivial self-reproducing machines. In C.G. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 709–725. Addison-Wesley, Redwood City, CA, 1991.

[45] John Stewart. Evolutionary transitions and artificial life. *Artificial Life*, 3(2):101–120, 1997.

[46] Yasuhiro Suzuki, Shusaku Tsumoto, and Hiroshi Tanaka. Analysis of cycles in symbolic chemical system based on abstract rewriting system on multisets. In C.G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 521–528, Cambridge, MA, 1997. MIT Press.

[47] Tim Taylor. The COSMOS artificial life system. Working Paper 263, Department of Artificial Intelligence, University of Edinburgh, October 1997. Available from `http://www.dai.ed.ac.uk/daidb/people/homes/timt/papers/`.

[48] Gianluca Tempesti. A new self-reproducing cellular automata capable of construction and computation. In F. Morán, A. Moreno, J.J. Merelo, and P. Chacón, editors, *Advances in Artificial Life: Third European Conference on Artificial Life*, Lecture Notes in Artificial Intelligence, Berlin, 1995. Springer.

[49] Francisco J. Varela. *Principles of Biological Autonomy.* North Holland, Amsterdam, 1979.

[50] Tomoyuki Yamamoto and Kunihiko Kaneko. Igniting the cycle of creation - an approach to create metabolism with Tile Automaton. In C.G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 399–406, Cambridge, MA, 1997. MIT Press.

[51] Shinichiro Yoshii, Hiroaki Inayoshi, and Yukinori Kakazu. Atomoid: A new prospect in reaction-formation system. Spontaneous hypercycles guided by dissipative structural properties. In C.G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 418–425, Cambridge, MA, 1997. MIT Press.

A Appendix: Lookup tables for default Instruction Groups

00000	switch_inst_grp	01000	bind	10000	push_00	11000	push_00
00001	bind	01001	release	10001	push_01	11001	push_01
00010	release	01010	link	10010	push_10	11010	push_10
00011	link	01011	break	10011	push_11	11011	push_11
00100	break	01100	if_bound	10100	clear_stack	11100	clear_stack
00101	if_bound	01101	ifn_bound	10101	stack_1	11101	stack_1
00110	ifn_bound	01110	wait	10110	stack_2	11110	stack_2
00111	wait	01111	wait	10111	stack_3	11111	stack_3

Table 6: Lookup table for the *Catalysis* Instruction Group

00000	switch_inst_grp	01000	attract	10000	push_00	11000	push_00
00001	attract	01001	repel	10001	push_01	11001	push_01
00010	repel	01010	attract	10010	push_10	11010	push_10
00011	attract	01011	repel	10011	push_11	11011	push_11
00100	repel	01100	attract	10100	clear_stack	11100	clear_stack
00101	attract	01101	repel	10101	stack_1	11101	stack_1
00110	repel	01110	attract	10110	stack_2	11110	stack_3
00111	attract	01111	repel	10111	stack_3	11111	stack_3

Table 7: Lookup table for the *Control of Local Environment* Instruction Group

00000	switch_inst_grp	01000	energy_collect	10000	energy_collect	11000	energy_collect
00001	energy_collect	01001	energy_collect	10001	energy_collect	11001	energy_collect
00010	energy_collect	01010	energy_collect	10010	energy_collect	11010	energy_collect
00011	energy_collect	01011	energy_collect	10011	energy_collect	11011	energy_collect
00100	energy_collect	01100	energy_collect	10100	energy_collect	11100	energy_collect
00101	energy_collect	01101	energy_collect	10101	energy_collect	11101	energy_collect
00110	energy_collect	01110	energy_collect	10110	energy_collect	11110	energy_collect
00111	energy_collect	01111	energy_collect	10111	energy_collect	11111	energy_collect

Table 8: Lookup table for the *Energy Storage* Instruction Group