The Creation of Novelty in Artificial Chemistries

Dominique Groß[1] **and Barry McMullin**[2]
[1]University of Bergen, Senter for Vitskapsteori, 5007 - Bergen, Norway
Dominique.Gross@svt.uib.no
[2]Dublin City University, School of Electronic Engineering, Dublin 9, Ireland
mcmullin@eeng.dcu.ie

Abstract

We observe that the world surrounding us perpetually creates novelty. The question we examine in this article is whether it is possible to build computer models that are similarly creative. The discussion focuses specifically on artificial chemistries.

Introduction

At a first glance the problem of novelty creation in computers actually does not appear to be a problem at all. Computer models are nothing but algorithms and as such unable to transcend the boundaries of their own world which has been created by the modeller. However, the general feeling is that we can simulate with arbitrary degree of precision the objects we find in nature; then, assuming that the latter is capable of creating novelty, we can ask: "Why should it not be possible to create novelty in computers if nature can do it?" One possible answer might be that novelty creation critically depends on the complexity of nature; this complexity, one might argue, cannot be mirrored in computer models because the time needed to implement and run such a complex model would exceed all limits of available human and financial resources—see (Gross & Jefferies 2001).

Although this answer might be valid for some types of problems, it potentially misses an important point. Nature might be complex on some levels, but is often assumed to be simple at the level of elementary particle physics and chemistry which are the most fundamental levels. This assumption of, and quest for, simplicity of the most fundamental laws of nature was introduced by Galilei. Free after Galilei, one might now look for artificial (= computer simulated) worlds which are both simple (in the sense that they function according to only a few comprehensive laws), but at the same time are able to create complexity of arbitrary degree. *Prima facie* then, there is no reason why this should not be possible using today's sophisticated technology and methodological tools.

The question of the creation of novelty in computer models is a central one for Artificial Life. If ever credible life-forms or evolutionary processes *in silico* are to be realised, then at least they have to have the capability to generate novelty. The question is also interesting on a more philosophical level: if we consider life as an essentially algorithmic process, then it should be possible to mimick its most important features in a computer-based model; we have only to figure out the "right" algorithm(s). On the other hand, it might also be that life is fundamentally non-algorithmic—see for example (Rosen 1991).

The concept of novelty we have in mind in this paper is clearly closely related to *complexity* in the informal sense formulated by John von Neumann (von Neumann 1949, p. 78); and the *creation* of novelty then corresponds to the (evolutionary?) *growth* of complexity (McMullin 2000). Nonetheless we deliberately choose the term "novelty" in this case, so that its natural vagueness and subjectivity may serve to emphasise the ill-defined nature of the problems under discussion.

Agent Based Models (ABMs)

Many ALife systems and simulations take the generic form of *Agent Based Models* or ABMs. At its most abstract, an ABM consists of a collection of primitive computational entities, called *agents*. We can, without loss of generality, assume that agents are implemented in the form of computational *objects*.[1] Thus an agent consists essentially of a package of *state variables* and *methods*. The methods are distinct computations which have been programmed for the agent, and which can be invoked by that agent or, indeed, other agents in the system.

Agents are organised into distinct *classes*, where a class is associated with a specific set of methods. A particular agent is then said to be an *instance* of its class. All agents of the same class share the same methods; but each individual instance has its own distinct state variables.

The computations realised by the agent methods may, in general, involve (pseudo-)stochastic elements; corre-

[1] That is, they are implemented in some *object oriented* programming environment such as *Java* or *C++* etc. Technically a single agent may consist of a number of distinct objects, but that will not affect the presentation here.

spondingly, the entire system behaviour may be stochastic.

Agents generally have some "locality" of interaction. That is, a given agent can only ("instantaneously") interact with a subset of the other agents, thought of as those in its *neighborhood*.

Both the collection of agents, and the neighborhood relations between them, may be static. For example, a Cellular Automaton (CA) can be viewed as an ABM in this sense. However, more usually, the collection of agents, and the neighborhood relations between them, is *dynamic*; which is to say that agents can be *moved*, *created* and *destroyed*. These dynamics in the constitution of, and relations between, agents are typically mediated through an encompassing "world"—which itself can be formally represented by one or more additional agents, albeit of particularly distinctive classes.

Conceptually, agents are normally regarded as *concurrent* computational units. However, ABMs are typically realised on serial computers such that, in fact, only one agent executes at a time. Some mechanism for scheduling CPU time, and providing for synchronisation, between the agents is therefore necessary; and the resulting system behaviours may be quite different depending on the details of how this is implemented. One simple approach is that an `update()` method is invoked, and allowed to execute to completion, on each agent in turn (in either a fixed or a stochastic sequence). Of course, in the course of executing its `update()` method a particular agent may invoke methods on other agents (within its neighborhood): so even with this apparently simple sequencing approach, agent activities may be interleaved in very complicated ways.

In all ABMs, each agent has a well defined instantaneous *agent state*, represented by the values of its state variables; and a dynamics, represented by its methods. By definition, the former can, and generally do, change during the execution of the model; whereas the latter cannot.[2]

Correspondingly, the entire system has an instantaneous system-wide state. This is, in effect, the array or vector of agent states (including any "world" or "environmental" agent states) for all agents instantiated at any given instant. Now the set of all possible states for any single agent (its phase space) is determined by its class specific set of state variables, and is thus fixed. However, in the general case where agents may be dynamically created or destroyed, the phase space of the system as a whole will be dynamic. We may say that it is of infinite (or, at least, indefinite) dimensionality; but that, at each time-step only a sub-space of dimensionality $\sum_{i=0}^{N} \mu_i^t d_i$ is occupied, where μ_i^t is the multiplicity of agents of class i at time-step t and d_i is the fixed dimensionality of agent of class i; altogether there are N different classes of agents.

At any given time then, the state of the system is uniquely identified by its position in the system phase space. We call this completely detailed system state the *micro-state*. The key feature is that the trajectory of the micro-states, including any changes in the dimensionality of the (micro-state) phase space, is completely entailed (whether deterministically or stocastically) by the micro-states themselves. In some sense, the micro-state *exhausts* the entailment of the model.

Now given such a model, with a particular repertoire of classes, scheduling and synchronization mechanisms etc..., this will give rise to a particular micro-state dynamics, or system behaviour, which must be examined and analysed. One possibility to do this is to look directly at the micro-state trajectory—but in general this is not useful. Instead, one is usually interested in the behaviours and interactions of agents at a coarser level of resolution. One might take averages over state variables of some or all agents, count associations between agents, or simply interpret some subjective *gestalt* patterns in graphical representation(s) derived from the micro-state.

In any case, the model is analysed at the level of some kind of user-defined description which (it is hoped) grasps the essential features of interest. At this level the history of the model is characterised by the succession of some *macroscopic* states, or macro-states, $\{F_{\theta=0}, F_{\theta=1}, \ldots, F_{\theta=m}\}$; these are determined by the micro-states together with some more or less arbitrary analysis rules which transform the micro-states into the macro-states. Even the timescale of the macro-states is not necessarily the same as that of the micro-states (reflected in the notation above by the letter θ as time index). Certainly, knowledge of macro-state alone is not generally sufficient to deduce the micro-state; thus the transition from the micro- to the macro-state generally (indeed, deliberately) involves some loss of information.

Note that we use both "micro-state" and "macro-state" here with senses that are derivative of, but still somewhat distinct from, the corresponding terms in statistical mechanics. Specifically, as we have noted, while the phase space of individual agents is fixed, the phase space of the whole system is not: its very dimensionality changes as agents are created or destroyed. Similarly, the macro-states of statistical mechanics are formally defined ensembles or equivalence classes of micro-states; whereas, in our case macro-states are rather informal, possibly even subjective, observables. They are not even necessarily features "of the system" (i.e., as a whole); this is a point we shall return to later.

[2] Except in the degenerate sense of individual agent creation and destruction; but even then, the repertoire of available *classes*, and their associated methods, is taken to be static and immutable.

On Novelty

In this section we would like to discuss the problem of novelty creation in computational systems in general and ABMs in particular.

The notion of novelty is a very difficult one. What is genuinely novel and what is just a different manifestation of something old is often hard to decide and the qualification depends largely on personal experience, insight and knowledge. To some degree, novelty is in the eye of the beholder, because something can only be novel relative *to* something or somebody. We shall not therefore pretend to be able to explicitly write down conditions under which we would accept a phenomenon (in the real or artificial worlds) as novel.

Of course, in a philosophical sense, it is sometimes contested whether there is "real" novelty at all in the world. But insofar as we are willing to stipulate that some interesting form of novelty does arise in nature, and continues to do so on an ongoing basis, and that this is exemplified in the phenomena of life itself, then it is reasonable to look for artificial or synthetic systems that can be at least *comparably* creative.

As a starting point, we might say that a "novel" state is reached whenever a system moves to a new (not previously occupied) point in its micro-state phase space. A slightly stronger form of novelty would be when the micro-state phase space itself changes—due to the creation or destruction of agents. However, given that, as we have already noted, each micro-state completely entails the next micro-state we can hardly identify anything expressed purely in these terms as "novel" in our sense.

Thus, for example, a trivial ABM might consist of a single agent whose only behaviour is to create new agents of the same class. This would result in a system with a constantly (indeed, exponentially) expanding—and thus "novel"—micro-state phase space; but we would clearly want to exclude this as an interesting or substantive example of the creation of novelty. Indeed, we might say that this is precisely the source for the intuition that computer models cannot, in general, exhibit novelty of any worthwhile kind.

Having said this, it seems that novelty should be looked for on some meso- or macro-scale. The attentive reader will already have noticed the somewhat paradoxical situation we find ourselves in. On the one hand, in some sense, the micro-state of the model contains **all** there is to know about the model's state; but on the other hand this undigested information is not adequate to judge whether something interestingly novel has happened in the system. In order to decide this we must first employ the selective analysis rule(s) which leads us to the macro-scale. It is clear that novelty at the micro-scale does not translate *necessarily* into a novel macro-state or macroscopic behaviour; depending on the rules which leads from micro-state to macro-state, the latter might even be completely unaffected by "novelty" of the former. To put it another way, in the absence of the rules that lead from micro-state to macro-state it is impossible to decide whether a certain novel micro-state is also macroscopically novel. Novelty creation on the macro-scale thus appears to depend as much on the micro/macro relationship as on the micro-state itself.

Another possible approach to the question of novelty focuses on the issue of *deterministic* versus *stochastic* dynamics. That is to say, the unsatisfying or uncreative nature of deterministic computational models is attributed precisely to the fact that the micro-state completely or *uniquely* determines the next (micro-)state (and thus, given the analysis rule, of the macro-state). Of course, the opposite extreme would be a "completely" stochastic dynamics—where there is no correlation between current state and next state at all. Such a model will then, at each time-step, take a completely random position in phase space. No doubt each microstate is then "novel", in the sense of "unpredictable"; but again, this is hardly an any more interesting form of novelty. Such a random state trajectory has—with overwhelming probability—no interesting intrinsic regularity; neither would there be a sensible analysis rules which could lead to interesting, (in the sense that it would not teach us anything interesting about the world) macro behaviour. We will therefore exclude this trivial kind of complete stochastic variation as an example of genuine novelty. But this still leaves us with the possibility of novelty generation in the intermediate case of deterministic dynamics leavened with some stochastic elements.

ABMs do often combine deterministic rules with stochastic elements; amongst other things, this allows the implementation of evolutionary models. Many will be intuitively inclined to regard some of those systems as novelty producing. However a closer look suggests that, if novelty is produced at all, it is rather strictly limited; see in this context, for example, (Bedau *et al.* 1997; Bedau & Brown 1997).

A first hint in this direction is the fact that repetitive runs of the system, with the same initial conditions, but different pseudo-random seeds typically lead to qualitatively similar behaviours.[3] For example in the `Tierra` system (Ray 1996), starting with the original ancestor organism, the phenomena of "parasites" and "hyperparasites" will essentially always emerge. Furthermore, very often the analogue is true for a variation of the initial conditions. This indicates already in those examples that it is not really the randomness that introduces the putatively novel phenomena into the system; the stochastic elements only change details of the occurrence of these phenomena. If the randomness really were the source of

[3] If this were not the case then the model might even be regarded as rather useless; see (Mitchell 2000).

novelty, then a different seed of the pseudo-random number generator should presumably (nearly always) give rise to qualitatively different behaviour of the model.

Thus it seems that in most ALife systems the stochastic elements fulfil the function of perturbation from an otherwise deterministic attractor of the microscopic dynamics—potentially allowing ongoing transitions between these attractors, which would not be possible in a purely deterministic system. Transitions to (previously unobserved) attractors may then be identified with "novelty". But the randomness is not in itself the source of novelty. It is merely helping to make manifest the variety in the potential attractors of the underlying deterministic dynamics; which is to say, establishing a long term probability distribution over the occupancy of these attractors. In Karl Popper's words, we must be inclined to conclude that "... indeterminism is not enough" (Popper 1973).

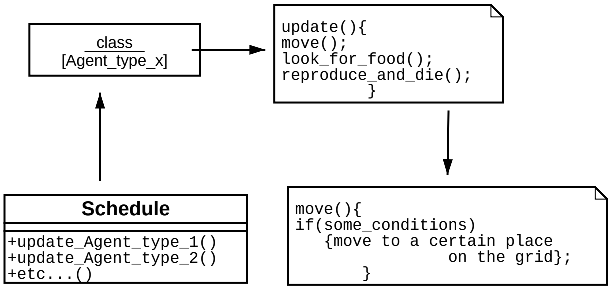
Of course, arguments for and against novelty-creation in existing ALife systems can go on forever. In order to avoid lengthy and unfruitful discussions, we may suggest an additional, albeit very informal, criterion: that novelty should be produced **perpetually**. In practice perpetual novelty creation does not need to mean *eternal* novelty creation, but only that the macroscopic behaviour does not quickly settle on one or a few attractors, but goes on exhibiting novelty for much longer than the typical simulation-time.

In the following sections we will look at the issue of novelty more concretely by considering two specific categories of ABM, featuring what we shall call *closed* versus *open* agents. There is not a sharp demarcation between these: but, roughly speaking, closed agents have a relatively small variety of *pre-conceived* (and pre-programmed) behaviours; whereas open agents can explore an indefinitely large variety of behaviours, which are not pre-conceived by the designer.

ABMs with Closed Agents

We will first investigate the potential for novelty creation in those "classical" ABMs having closed agents.

As already outlined, this might be typically implemented according to the scheme depicted in figure 1. Its basic underlying idea is clear: there is a central scheduler which invokes the `update()` method for each individual agent, according to some scheduling algorithm. In each case, the `update()` method invokes other appropriate methods (on the agent itself or on other agents in its neighborhood) in order to implement the agents' behavioural rules. Where the agents are thought of as modelling organisms then these behaviours may include such things as "movement", "feeding", "reproduction", "death" etc. The distinct "types" of agents correspond to different classes. The repertoire of classes is, however, fixed, and usually comparatively small.



```
update(){
move();
look_for_food();
reproduce_and_die();
            }
```

```
move(){
if(some_conditions)
   {move to a certain place
               on the grid};
      }
```

Figure 1: Schematic outline of a typical ABM with "closed" agents. The agents' `update()` methods are called according to a schedule. The update usually consist of a series of more basic actions to be taken, such as movement, uptake of food and reproduction and death. How exactly these actions are performed and under which conditions they are called may be variable—within some pre-defined limits—which enables evolutionary agent adaptation.

By definition, a closed agent can only perform those tasks, or exhibit those behaviours, that are pre-coded in its methods. Consequently, any variations in agent behaviour in the course of model execution have to have been preconceived, at some level, by the programmer. The available classes (which is say the class methods) are not subject to variation.

Variability is firstly limited to varying populations of the particular set of implemented classes; and within the instances of a particular class, variability is limited to *state* variations—i.e., of the values of the state variables. However, it is common in ALife models to partition the state variables between some which can vary during the normal, somatic life-time of an agent; and others which vary only when new agents are being created (i.e., at reproduction). The latter may be regarded as behavioural *parameters* or, in biological terms, represent agent *genotype.*

In this category of model then, the scope of evolution is either to select among the specific, pre-programmed, classes (if these come into darwinian competition) or to select among lineages within those classes created by parameter variation ("mutation").

This can, of course, provide useful and interesting tests of selectional conjectures; but insofar as the whole scheme boils down to the exploration of pre-determined variants of a small, fixed, set of agent classes, it seems it cannot qualify as generating novelty in our sense—*at least, not at the level of the individual agents.* It remains conceivable that genuinely novel phenomena may arise at some more macroscopic, collective, level, involving groups (colonies?) of agents. We shall see later (section *Novelty in Artificial Chemistries*) that this concept of novelty creation at the level of the collective behaviour of multiple agents—in the absence of novelty creation at the level of agents themselves—is a key motivation for a particular form of ABM known as an *Artificial Chemistry* (AC).

ABMs with Open Agents

First, however, we turn in rather a different direction—to consider systems where the *agent* level constraints assumed above are relaxed; that is, where the agents are *open.* By this we mean that the potential variability in agent behaviours is very large (if not infinite); but this variability is not merely parametric modulation of still fixed and stereotyped behaviours, but rather is of a scope which exceeds the capability of a designer to pre-conceive (never mind design) it.

At heart this means that among the methods (perhaps the only method) of an agent there is a *general purpose (Turing complete) computer*; and among the state variables (perhaps the only variable), there is a *programme.* On one level, the agents are still as rigid as ever—the available agent methods cannot be varied. Yet, on another level, there is literally unlimited scope for variation in agent behaviours. Indeed, this behavioural variation will even be formally *unpredictable* (in the sense of Turing halting).

Perhaps the best known example of this *genre* is `Tierra` (Ray 1996); beside this original there are a number of related ALife systems with a similar basic idea such as for example the `Avida` systems (Adami 1998). We will henceforth refer to those systems collectively as "`Tierra`-like systems".[4]

Whenever, as is the case in `Tierra`-like systems, the "code" which *essentially* defines the behaviour of the agents is created at run-time and not at compile time, then it seems clear that many of the restrictions which limited novelty creation in ABMs with closed agents should disappear. In particular, if the conditions exist for darwinian selection (the agents can reproduce, but with population growth limited by competition), and the embedded code in the agent state variables is subject to mutation, then it seems that there should be scope for continuing, open-ended ("perpetual") evolutionary creation of novelty. `Tierra`-like systems offer exactly this. The agents and their behaviours are not bound by the pre-conceiving imagination of the modeller/programmer, but can freely emerge from the evolutionary process.

There is no doubt that `Tierra` and similar systems do in fact show interesting evolutionary phenomena; but as already noted in the section *On Novelty*, it is also a well-known fact that the creativity of the evolution in those systems seems to be limited and cannot be sustained for a long period of time. Attempts to build `Tierra-like` systems with a more long-term evolution have largely failed.

The nature of these limitations is still very much an open question; however we can consider at least a few specific conjectures as to the explanation.

A first possible factor may be the rather small size of the artificial worlds which were initially studied. These allowed only a comparatively small number of creatures to exist simultaneously. This hypotheses is easily testable. The rapid increase in available computing power means that, even after a relatively short waiting time, the implementation of much larger experimental worlds has already become possible. However, at least for the case of `Tierra,` these larger worlds do not seem to have led to a significantly more interesting phenomenology.

A second possible explanation for the rather limited phenomenology of `Tierra` is that it is due to a lack of structure and complexity of the *environment* of the digital organisms. The environment as experienced by the agents may simply lack features to be adapted *to.* An interesting study of the dependence of adaptability on environmental complexity in ALife systems is provided by (Fletcher, Bedau, & Zwick 1997).

In the case of `Tierra`-like systems this points at a dilemma. The more complex their artificial worlds are designed, the more potential niches they offer to the agents. This should prolong the evolutionary process and increase the potential for the evolutionary development of interesting agent behaviours. On the other hand, the appeal and elegance of `Tierra`-like systems is exactly the simplicity of their initial world. To much designed and programmed complexity would to some degree defeat the purpose of those systems.

In the original `Tierra` the environment in which the agents live is quite deliberately minimalist. It essentially consists only of a "memory" space which the digital creatures inhabit; the "slicer" which allocates CPU time; and the "reaper" which kills off agents to ensure a continuing supply of free memory for further reproduction (and thus—hopefully—evolution).

It seems that we can therefore safely discount the emergence of, for example, predation in `Tierra`—that is, agents which kill other agents to exploit their resources. As `Tierra` is implemented, there is simply no mechanism for one organism to attack or kill another; and, in any case, the resources of a killed organism go back to a common pool and are immediately available for the next organism which attempts to reproduce—they would not be preferentially available to a (nascent) lineage of predators.

In order for predation to emerge, there would have to be generic mechanisms—presumably implemented by some "world" agent(s)—for redistribution of resources between agents. But the `Tierran` world agents (mem-

[4] `Tierra` is not, in fact, implemented with the strict, object-oriented, ABM architecture we have presented in the section *Agent Based Models (ABMs)*; however, this is a technicality which does not affect the arguments we present.

ory, slicer, reaper) are *closed*; they are *not* capable of evolving, but simply offer a fixed, stereotyped repertoire of interactions for the organism agents to engage with.

Of course, in a suitably enhanced and modified version of `Tierra` we may certainly observe more phenomena than in the original system. Specifically, one might add precisely the sorts of attack and resource redistribution capabilities that could underpin predation phenomena. Presumably one might then observe classical evolutionary arms races between predators and prey, with elaboration and refinement of more or less sophisticated strategies for both.

But of course, this all misses the point. Again, this world will ultimately be limited by the closed set of interaction possibilities. So, to pursue the example even further, we could certainly *still* not expect to see organisms burying themselves in "sand" in order to avoid detection by predators—simply because there is nothing of the nature of "sand" in this world, nor any mechanism for it to spontaneously appear!

It should be clear that there are two, at least, conceptually distinct kinds of limitation being identified here:

- The fact that some (key) agents are still closed; worse, it is not at all apparent how one might even *conceive* of "opening up" these particular agents in an effective way.

- Even more fundamentally, there is the fixed nature of the agent *interfaces*. Although the agents may be *internally* capable of implementing arbitrary (Turing) computations, the evolutionary significance of this will be sharply limited if it can only impact on other agents in severly restricted, and fixed, ways. Or to put it another way, it is like imagining the evolution of organisms with general purpose, programmable, nervous systems—but imprisoned within completely immutable sensor and effector systems (Cariani 1991).

In summary then, the properties of the world in which the artificial organisms live is crucial to the evolutionary potential of the model-system; its "physics" and "chemistry", the ways in which organisms may be born, die and receive resources all constrain its evolutionary potential. Yet engineering a complex—but still fixed—world is no solution: it merely delays the still inevitable plateau of evolutionary exhaustion.

We conclude that, for our purposes at least, current ALife models with open agents (in the sense of evolutionarily programmable) turn out to be, at best, inadequate—and, at worst, a dead end. The challenge, of course, is to formulate a system where the world itself, including the very mechanisms or interfaces for agent interaction, is open and indefinitely mutable and creative. We see this as a major challenge ahead.

Novelty in Artificial Chemistries

We now turn our attention to a rather different kind of ABM: the *Artificial Chemistries* (ACs). For examples, see (Rasmussen *et al.* ; McMullin 1997a; 1997b; Fontana 1991; Dittrich, Ziegler, & Banzhaf 2001).

At first this will seem like a retrograde step, because ACs fall into the category of ABMs with *closed* agents. In fact, ACs do not normally even have the most limited evolutionary dynamics which we discussed in the section *ABMs with Closed Agents. Prima facie* then, the prospects for such systems to exhibit perpetual novelty seem extremely weak; but we will pursue the analysis nonetheless.

We must first characterize what we mean by ACs, and how they relate to the systems already described. As mentioned, ACs are ABMs with closed agents, and thus conform to the general framework shown in figure 1. However, the interpretation of the agents is now rather different. Instead of thinking of the agents as representing organisms, in ACs they are taken to denote *atoms* or (small) *molecules* (we will use the generic term "particles"). The agent classes now represent *elements* and/or *molecular species*. The interactions between agents are motivated as—usually highly abstract—models of chemical reactions. In simple cases, this can result in the replacement of the reacting agents with different agents representing the appropriate reaction products. In more complex cases, there will be at least one kind of agent interaction which attempts to explicitly model chemical *bonding*: the effect is to establish agents into more or less stable *aggregations*—corresponding to larger scale macromolecules or molecular complexes.

Now, in constrast to the situation of the previous section, where the emphasis was on novelty on the part of individual agents, in ACs the agents are actually *designed* to have entirely immutable behaviours (methods); and, further, to have only a small number of distinct varieties (classes, types of agents). However, the new prospect is of novelty on a higher, or more macroscopic level: the level of the macromolecules or agent-aggregates.

This deliberate limitation of artificial chemistries at the level of agent variety is of course an attempt to mimic fundamental processes in nature which seem to operate exactly this way. The variety of chemical elements is essentially static, and relatively tiny compared to the stunning variety of higher level aggregates (molecules, cells, organisms, colonies etc.) which they give rise to—in processes which therefore surely have to be regarded as novelty producing. An ultimate goal of research with artificial chemistries is therefore to exhibit similar higher order structures and novelty creating processes in computer based systems.

So what specific types of novel behaviour may we hope to produce in ACs? We will want the AC to have the potential to dynamically create new macromolecular species, with novel functions and properties. In a sense, of course, the concept here is a *generalisation* of the original motivation underlying `Tierra`: starting from a relatively simple "ur-chemistry," consisting of a few primitive agent classes (elements, primitive molecular species) only, we would hope that the complexity in such a system increases over time through the progressive emergence of new types of macromolecules, with new behaviours *and which can relate and interact in new ways.*

Note that while this "ur-chemistry" should ideally be maximally simple, some initial degree of complexity must presumably be supplied if certain functionalities are to emerge. For example, it is hard to imagine how enzymatic interactions could arise in a purely two dimensional chemistry.

In any case, it is clear that the implementation of *bonding* in an AC must be critical to the prospects for this open ended novelty at the macromolecular level. Let us therefore consider this in a little more detail.

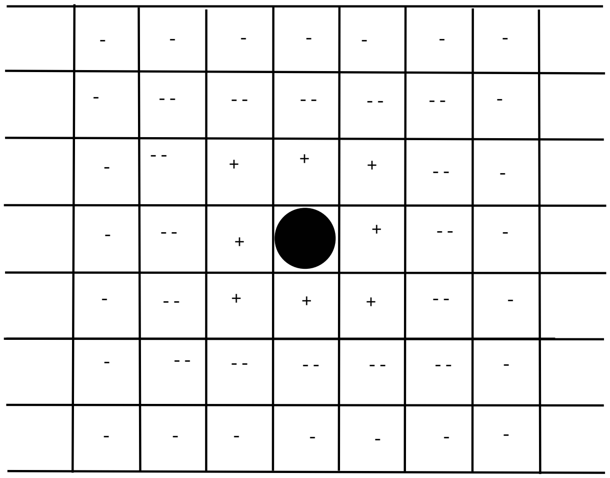
A common, and conceptually simple, way to implement bonding in ACs is by specific and explicit state changes of the respective agents (McMullin 1997b). Thus, particle A registers in its state variables that it is now bonded with particle B, and particle B records a precisely complementary relationship in its state variables. Bonded particles are then not different from unbonded particles in any other respect than that they are "tagged" in this way as being bonded; but, of course, the methods for implementing particle motion are explicitly programmed in such a way as to respect this bonding (i.e., to maintain the spatial juxtaposition of the bonded particles within some specified constraints). Similarly, the exact conditions for bond formation and rupture are explicitly specified in advance, within the various particle methods.

However, we would like to suggest that this form of explicitly programmed bonding seems unlikely to support the sorts of emergent novelty which we are in search of. The "new" higher level particle formed as a result of this kind of bonding is little more than a collection of its primitive components plus a few motion constraints *which have been pre-conceived and explicitly pre-programmed.* In this sense bonds—their formation, behaviour, rupture—are completely pre-specified, *bottom-up* phenomena; they exclusively stem from the explicitly programmed properties of the bonded particles, rather than being emergent effects of the molecular configuration per se. This seems distinctly unlike the formation of new molecular species in nature and it also restricts the potential for novelty creation in the model.

It seems that bonding in ACs might be much more interesting (with respect to the creation of novelty) if it were emergent, or a product of constraints that are *top-down.* We mean by this that bonding would not stem from explicit, pre-programmed, properties of the particles, but rather from the macroscopic or collective properties of the particle configurations.This is also the way nature seems to work. Bonding between particles would then not simply amount to tagging the participating particles and invoking extra rules which implement the constraints of bonded particles. If the top-down philosophy is followed, then bonding becomes a phenomenon that emerges from the interaction of particles, not the other way round, i.e., bonding is possible because there are constraints on the single particles. The collective should restrict the behaviour of the constituents to such a degree that they form a new whole. If the kind of constraint the whole exerts on the parts is very sensitive to the configuration of the parts, then it seems that this may be a potential source of variety and novelty.

The question is now whether strictly agent-based ACs with this property can be constructed. The snake has to bite its own tail. Is it possible to generate top-down constraints in bottom-up models? Note that in real chemistry exactly these kinds of top-down constraints arise and are one of the sources of novelty. Proteins, for example, are only composed of a handful of different types of atoms, yet are very different from one another. Size and configuration matters in the real world, and is a genuine source of novelty. If we manage to implement a similar kind of mechanism in an AC, then we might also hope for similar creation of novelty.

In order to better illustrate the idea of top-down constraints we will schematically outline the description of a minimal AC which exhibits them. Note that this is an imaginary toy-model devised just to illustrate this particular point. We do not suppose that it would, in fact, demonstrate any continuing creation of novelty (not, at least in this minimal form).



-	-	-	-	-	-	-
-	--	--	--	--	--	-
-	--	+	+	+	--	-
-	--	+	●	+	--	-
-	--	+	+	+	--	-
-	--	--	--	--	--	-
-	-	-	-	-	-	-

Figure 2: Potential field established by an isolated particle.

The AC is defined on a 2-dimensional square lattice and consists of only one type of particle (primitive agent). The particle exerts an attracting force in its immediate (Moore) neighbourhood, and a repelling force in the successive (Moore-like) neighbourhoods at distances 2 and 3. The corresponding potential field established by an isolated particle is indicated schematically in figure 2. The gradient of this field at any site would indicate the (vector) force imposed on a particle at that site. Potential fields from multiple particles are assumed to superpose linearly, as indicated in figure 3.

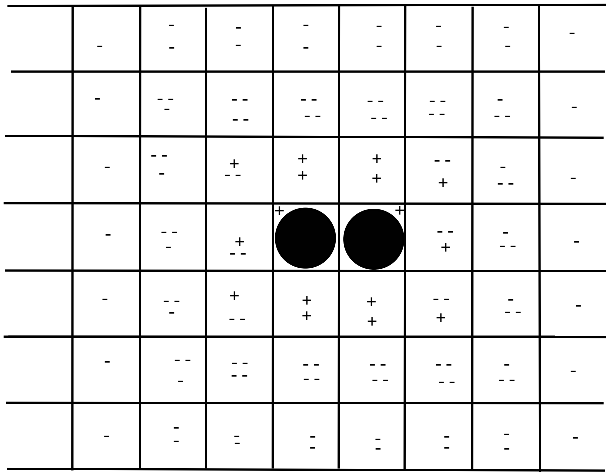


Figure 3: Overlapping potential fields superpose linearly.

The underlying particle motion is imagined to be (approximately) newtonian—i.e., a conservative "billiard ball" mechanics.[5] Thus given the right amount of energy a particle may penetrate the "distant" potential barrier of another particle, enter its "near" potential well, and the two particles will become "bonded". The configurations of figures 3 and 4 may represent such bonded particles (depending on the respective particle energies). If so, then this new (macro-)molecular configuration establishes a new collective potential field, quite different from that of either isolated particle. For example, it will offer distinctive, emergent, preferences for formation of further bonds.

This is already an example of a top-down constraint in our sense. Two or more particles which are trapped in each others potential wells become "bonded". Depending on the configuration, the (macro-)molecule might form further bonds with other particles or molecules. Which particles can bond with one another depends critically on the configuration of the whole. Likewise, the stability of the macromolecule will critically depend on how it is arranged, which itself increases the potentential for the emergence of novel higher-level particles (macromolecules). Note that the bonding rules, the allowed configurations or the conditions under which bonds break are nowhere explicitly defined, but are emergent properties of the underlying mechanics of the AC and the distinctive potential field of the particles. A macromolecule will thus have properties which are not, in effect, already pre-figured in programmed bonding properties of the constituent particles. Even in this toy version, it seems that a two particle molecule may be capable of forming a further bond with a third particle; such bonding may, in turn, alter the collective potential field, so that the bonding between all three particles is *stronger*.

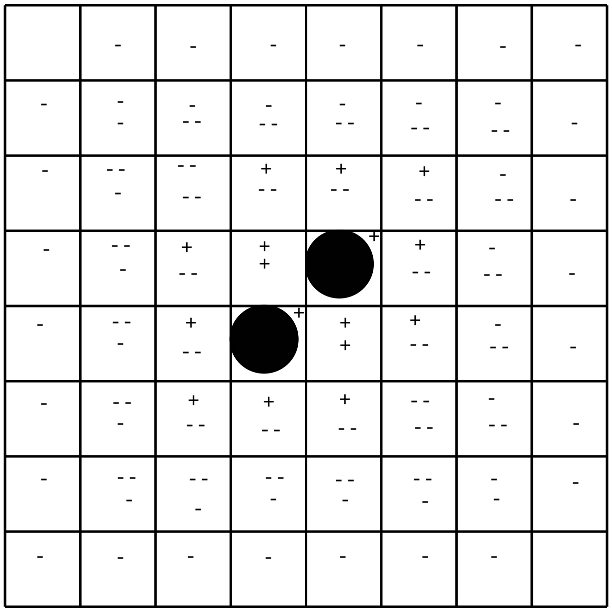


Figure 4: (Alternative) bond configuration.

We thus see that bonding—and consequently higher order "chemical" properties—in such a model will not be pre-conceived or pre-programmed in any substantial way at the agent level, but rather emerge in a highly dynamic way. Already in this toy model we may observe a significant variety of macromolecular configurations and behaviour. The richness of the phenomenology of the model would, of course, be enhanced by introducing different classes of particles with different fields; this might then be a viable way to introduce novelty on an ongoing basis, through the creation of new types of macromolecules.

In some respects, of course, the avenue suggested here is fundamentally close to that of (Rasmussen *et al.*); but it is also still different in its suggested degree of abstraction and in its purpose. We do not propose attempting to model any *particular* chemical species and the described toy-model explicitly renounces any realism; this shift of attention allows to explore the generative power of more generalised artificial chemistries, but still with

[5] A number of technical implementation issues would arise here, particularly due to the discrete nature of space and time. These will not be addressed in detail here; but see also, for example, (Rasmussen *et al.*).

emergent bond formation. The emphasis is thus mainly on methodological aspects: The question at this point is not how a specific phenomenon can be modelled, but more general, how ACs with high or even perpetual novelty creation might be constructed.

At this stage it is unclear whether these approaches will ever lead to a model which supports the perpetual creation of novelty (the toy model certainly would not), but the *ansatz* at least seems to merit investigation to show whether it is fruitful or not.

Summarising and Concluding Remarks

We have considered several different types of ABM models, relevant to ALife, from the perspective of *novelty*.

The first type were generic ABMs with closed agents, but capable of exhibiting evolutionary dynamics. However, by the very definition of closed agents, these will have severely limited potential for novel behaviour—at the agent level. In essence, the programmer must prespecify and code the complete range of possible variations the agents can undergo. In spite of the limited potential for novelty creation, this type of model is often very powerful and can provide and has already provided a number of important insights into various theoretical and practical problems of selection and evolutionary dynamics. However, it is not fruitful with respect to its ability to create novelty.

This limitation can partly be overcome by a modification to the agent-based paradigm. If the agents are open—if they have embedded general purpose computing capability—then there is obviously a greater potential for spontaneous creation of novelty. Examples of existing artificial life systems which follow this principle are `Tierra` or `Avida`. However, these models seem ultimately to be disappointing; evolutionary development (of the agents) reaches a plateau and effectively ceases. How the phenomenology of these models can be substantially improved upon is an open research question of fundamental importance to the understanding of ALife modelling.

In real life, the emergence of new types of proteins with new functions is, in an evolutionary perspective, an example of a major source of perpetual novelty. The characteristic feature of real chemistries is that macromolecules and molecular complexes show fundamentally novel behaviours relative to the constituting particles; we say that the macromolecule constitutes a new entity which constrains its parts from the "top-down". We propose that similar mechanisms, implemented in ACs, may substantially enhance the creation of novelty in such models.

References

Adami, C. 1998. *Introduction to Artificial Life.* New York: Springer-Verlag.

Bedau, M., and Brown, T. 1997. Visualizing Evolutionary Activity of Genotypes. *Artificial Life.* To be published.

Bedau, M.; Snyder, E.; Brown, T.; and Packard, N. 1997. A Comparison of Evolutionary Activity and Artificial Evolving Systems in the Biosphere. In Husbands, P., and Harvey, I., eds., *Proceedings of the Fourth European Conference on Artificial Life.*

Burks, A. W., ed. 1966. *Theory of Self-Reproducing Automata [by] John von Neumann.* Urbana: University of Illinois Press.

Cariani, P. 1991. Emergence and artificial life. In Langton et al. (1991), 775–797.

Dittrich, P.; Ziegler, J.; and Banzhaf, W. 2001. Artificial Chemistries—A Review. *Artificial Life* 7:225–275.

Fletcher, J.; Bedau, M.; and Zwick, M. 1997. Dependence of Adaptability on Environmental Structure in a Simple Evolutionary Model. *Adaptive Behavior* 4(3):283–315.

Fontana, W. 1991. Algorithmic Chemistry. In Langton et al. (1991), 159–209.

Goldstein, J. 2001. Emergence, Radical Novelty, and the Philosophy of Mathematics. In Sulis, W., and Trofimova, I., eds., *Nonlinear Dynamics in the Life and Social Sciences*, NATO Science Series A: Life Sciences Vol. 320. Amsterdam: IOS Press. 133–153.

Gross, D., and Jefferies, D. 2001. Complexity Beyond Agent-Based Models. *Complexity International.* forthcoming.

Langton, C.; Taylor, C.; Farmer, J.; and Rasmussen, S., eds. 1991. *Artificial Life II*, volume X of *Santa Fe Institute Studies in the Sciences of Complexity.* Reading: Addison-Wesley.

McMullin, B. 1997a. Modelling autopoesis: Harder than it might seem! Dublin City University, School of Electronic Engineering, Technical Report: bmcm9704, URL: http://www.eeng.dcu.ie/ alife/bmcm9704/.

McMullin, B. 1997b. SCL: An artificial chemistry in Swarm. Dublin City University, School of Electronic Engineering, Technical Report: `bmcm9702` and Santa Fe Institute Working Paper: `97-01-002`.

McMullin, B. 2000. John von Neumann and the evolutionary growth of complexity: Looking backwards, looking forwards. *Artificial Life* 6(4):347–361.

Mitchell, M. 2000. Life and Evolution in Computers . Technical Report 00-01-001, Santa Fe Institute.

Popper, K. R. 1973. Indeterminism is not enough. *Encounter* 40(4):20–26. A revised version of this essay appears as *Addendum 1, pages 113–130, of:* (Popper 1988).

Popper, K. R. 1988. *The Open Universe: An Argument for Indeterminism.* London: Hutchinson. From the *Postscript to the Logic of Scientific Discovery*, edited by W.W. Bartley, III. First edition published 1982.

Rasmussen, S.; Baas, N.; Mayer, B.; Nilsson, M.; and

Oleson, M. Ansatz for Dynamical Hierarchies. *Artificial Life* 7(4):329—354.

Ray, T. 1996. An Approach to the Syntheses of Life. In Boden, M., ed., *The Philosophy of Artificial Life*, Oxford Readings in Philosophy. Oxford: Oxford University Press. 111–145.

Rosen, R. 1991. *Life Itself.* New York: Columbia University Press.

von Neumann, J. 1949. Theory and organization of complicated automata. In Burks (1966). 29–87 (Part One). Based on transcripts of lectures delivered at the University of Illinois, in December 1949. Edited for publication by A.W. Burks.