DOI: http://dx.doi.org/10.7551/978-0-262-31050-5-ch012

Architectures for Self-reproduction: Abstractions, Realisations and a Research Program

Barry McMullin

Dublin City University, Ireland
barry.mcmullin@dcu.ie

Abstract

It is well recognised that von Neumann's seminal abstraction of machine self-reproduction can be related to the reality of biological self-reproduction — albeit only in very general terms. On the other hand, the most thoroughly studied artificial evolutionary systems, incorporating meaningful self-reproduction, are the coreworld systems such as Tierra, Avida etc.; and these, in general, rely on a purely "self-inspection" mode of reproduction (or, more simply, "replication"). To the extent that the latter has any direct biological analog it would appear to be with molecular level reproduction and evolution in the hypothesised RNA-world. In this paper I review the details and distinctions between these modes of reproduction. I indicate how the abstract von Neumann architecture can, in fact, be readily realised in coreworld systems; and outline the research program that flows from this. Finally I attempt to make more precise the resulting analogies with molecular biology, at least up to the (prokaryotic) cell level.

Background: von Neumann's Problem

As is well known, in the late 1940s and early 1950s John von Neumann conducted an investigation into problems of understanding the evolutionary growth of complexity (Burks, 1966; McMullin, 2000). In particular, he wondered how it could be possible for a machine, of a given level of complexity, to construct an offspring machine of greater complexity than itself. *Prima facie*, from an engineering point of view, this seems like a paradox: surely any machine capable of constructing other machines must, in some sense, already contain the design of those machines within itself; and therefore must already be more complex that any such offspring machine. And yet, if the theory of biological evolution (at least of evolutionary descent from one, or a small number, of primordial ancestor species) is correct, and if biological organisms are some (possibly very special) kind of "machine", then this sort of constructive increase in complexity must not just be possible, but must happen repeatedly, indeed almost continuously, over evolutionary time. As a special (edge) case, even without considering evolutionary *growth* of complexity, viewing organisms as machines at all, and recognising their universal capacity for *self-reproduction*, implies that essentially arbitrarily complex machines can construct offspring of at least *equal* complexity to themselves. So, *inter alia*, von Neumann wondered how it can be that arbitrarily complex machines can be capable of such self-reproduction.

The formulation (and thus solution) of these problems may seem to hinge critically on what we mean by "complexity"; but for the immediate purposes of this paper it will suffice to adopt von Neumann's own, vague and qualitative, definition that complexity means the ability to "... do very difficult and involved things" (Burks, 1966, p. 78). Specifically, the machines under consideration must exist in a universe in which they can "do" more or less "difficult and involved" things, ideally with no obvious upper bound.

With this starting point, von Neumann proceeded to formulate a completely general and abstract machine architecture whereby such machines could:

- be arbitrarily complex (i.e., the set of such machines would span whatever range of complexity is possible at all within their universe),
- be capable of self-reproduction,
- be capable of undergoing spontaneous "mutation", giving rise to offspring which are different in kind, but this difference is retained through further cycles of reproduction — i.e., the differences "breed true",
- and where the entire set of such machines is connected under mutation.

That is, starting with an arbitrarily simple machine having such an architecture, there would exist sequences of possible mutations leading to machines of the highest complexity possible in the particular universe, where all of these machines share the same architecture and all are also capable of self-reproduction.[1]

[1] The bare existence of such sequences does not guarantee that any would ever be followed. That would depend on quite separate factors — most critically, what ecological and selectional interactions arise between mutationally distinct machine lineages. However, such questions will fall outside our immediate scope here.

Having formulated the general architecture, von Neumann went on to consider its realisation in particular "model" universes. He first imagined an abstract, but still quite physically motivated, "kinematic automaton" universe — not unlike a modern "physics engine" world such as commonly applied in certain forms of computer gaming and animation. His most detailed elaboration was in the case of a two dimensional, homogeneous, "tesselation automaton", or "cellular automaton" (CA) universe as it would now be called. He did successfully demonstrate, in this particular CA universe, the essential detailed design of a particular "seed" machine that had his abstract architecture (Burks, 1966). This was sufficient to establish the principle that there could also exist, in this universe, arbitrarily complex machines which would also be logically capable of self-reproduction, and that this entire (infinite) set of self-reproducing machines would be connected under mutation.

The Abstract von Neumann Architecture

The abstract architecture described by von Neumann consists of a complete machine, denoted M, having a highest level decomposition into an active, functional, component, labelled P, and a relatively passive component labelled G. We write $M = (P + G)$.

G consists of a linear chain of sub-components. Each sub-component can be chosen from some finite set of possible component types. We assume there are at least two such distinct types or configurations. G can be of arbitrary length, and the different component types must be "compatible" in the sense that the chain can be constructed with an arbitrary sequence of these allowed types. G can then be thought of as roughly analogous to the *tape* of a Turing machine, effectively acting as an information storage system. The particular choice of component at a particular position corresponds to, or represents, the discrete "symbol" recorded in a single "square" of a Turing machine tape. G has no activity or functionality in its own right: it just retains the specific sequence of its components, which is to say, the specific information stored in it.[2]

P consists of four distinct, but functional and interacting, "sub-machines" or "sub-assemblies":

A **:** This is a "general constructive automaton" or "programmable constructor".[3] By hypothesis, A can interpret the information stored in the chain G (in the specific sequence of its sub-components) as representing the description of some essentially arbitrary configuration or assembly of whatever types of machine components are made available in the particular universe. That is to say, A can construct any (largely) arbitrary target (sub-)machine. For the moment we label the latter as X: so we can say that G contains a description of some machine X (relative to A's particular interpretation). We denote this by saying that G stores the information $\phi(X)$ or simply $G = \phi(X)$ (where the function ϕ is determined by the detailed design of A). Once activated, A will scan G and proceed to construct an instance of X.[4]

B **:** This is a "general tape copier". Once activated, it will scan G and construct another, separate, chain, with exactly the same sequence of sub-components (i.e., storing the same "information").

C **:** This is a "controller", which primarily acts to sequence the operations of other sub-assemblies of P. It operates in a cyclical manner. It activates A and B in turn (the order does not matter in general), or possibly even concurrently (provided they will not interfere with each other — in particular in their access to G). It then connects together the two newly constructed sub-assemblies — the sub-machine X and the copy of G — forming $(X + G)$ and releases this into the surrounding environment, as a new, discrete and separate, machine. In general this releasing step is also thought of as incorporating an initial "activation" of the new, offspring, machine. C then starts the cycle again from the beginning, and repeats this indefinitely (for as long as M remains functional or "alive").

D **:** This denotes an arbitrary assembly of the components allowed within the universe, i.e., it effectively represents an arbitrary (sub-)machine in its own right, with arbitrary functionality (and thus "complexity"). D can be supposed to operate autonomously of, and indeed concurrently with, the other sub-assemblies, with the one constraint that it must not compromise or interfere with the functionality already described for A, B and C. If necessary, it can be assumed that in addition to sequencing A and B, C also co-ordinates with, or even controls, D to assure this independence of operation.[5]

[2] As an aside, for our particular purposes here (and unlike a conventional Turing machine tape) it is not essential that each individual "square" of G be capable of being "rewritten" with an arbitrarily different (allowed) symbol. It is sufficient if each position can be made to hold an arbitrary symbol at the time the chain is initially constructed. In this sense, G is more like a "read only memory" than a conventional, Turing tape-style, "read write memory".

[3] A is commonly referred to as a "universal constructor". However, that usage is somewhat problematic, and I will avoid it here.

[4] In the general case, this process may "fail" for various reasons. G may not be correctly structured as a "tape" at all (it may have the wrong morphology — not be a linear chain — or have incorporated components not recognised as denoting symbols etc.). There may be configurations for G that are correctly structured (as sequences of symbols) but which do not describe any machine relative to A. There may be possible machines X which A cannot construct (i.e., for which there is no corresponding description $\phi(X)$ relative to A). However, I will not consider these issues further here.

[5] von Neumann referred to D as "ancillary" machinery, but this is somewhat misleading. In the general case, D may represent the great bulk of the physical constitution of the overall machine M and, in any case, determines any and all of its *distinctive* functionality (i.e., over and above self-reproduction *per se*).

In summary then, a generic behaviour of any machine having this architecture is to repeatedly construct "offspring" machines of the form $(X + G)$, which is to say $(X + \phi(X))$, for arbitrary X. But since X is arbitrary, we can now take the final step of *stipulating* that $X = P$.[6] Thus, in designing the machine M we would first design P as already described above and then deliberately arrange that the information recorded into G is just that sequence which, under the function $\phi()$, represents P itself, i.e., $G \triangleq \phi(P)$. With this identification, we have $M = (P + \phi(P))$ and the offspring of M is precisely:

$$\begin{aligned}(X + G) &= (X + \phi(X)) \\ &= (P + \phi(P)) \\ &= M\end{aligned}$$

i.e., M is now self-reproducing. This is true of the entire family of machines represented by M, having arbitrary (and thus arbitrarily complex) sub-systems ("ancillary machinery") D.

Mutation

As already noted, an essential aspect of von Neumann's problem was that he wanted to understand or model a general mechanism for *mutational change*, that is, spontaneous or accidental modifications of a (self-reproducing) machine's structure that would nonetheless *breed true*. With respect to his abstract architecture we will now systematically explore the possibilities.

First consider modifications to the structure of $P = A + B + C + D$. In the case of changes to any of A, B or C the expected outcome is simply that the reproductive functionality will be broken — the machine will cease to construct any further offspring, or any offspring it does construct will be malformed, probably without function, and certainly no longer identical to the parent. So such changes cannot breed true.

In the case of an alteration to D, say from D to D', then, as long as this does not compromise the ongoing operation of $A + B + C$, the machine will continue to produce offspring. But since the P part of these offspring is constructed by A interpreting (or "decoding") $G = \phi(P)$, these offspring will revert to the form $P + \phi(P) = (A + B + C + D) + \phi(A + B + C + D)$ and not the altered form $P' + \phi(P) = (A + B + C + D') + \phi(A + B + C + D)$ that the parent now has. So again, such changes will not breed true.

[6] There are subtleties here. In particular, it must be the case that each particular P (i.e., defined by $A + B + C$ combined with a particular D) is indeed constructable by A. As already noted, this is not necessarily the case in general; but for my purposes here I shall simply assume that this condition can be satisfied with "sufficient" generality (i.e., for a sufficiently wide definition of D).

Now let us consider a modification to the structure of G, changing it to G'. In the first instance we now expect that the active machinery of M (P) will continue in operation, so it will continue to produce offspring. Further, since the B component simply copies whatever chain or information sequence it is presented with, this offspring will contain G' rather than G; but this will not be the only difference in the offspring. In general the offspring P part will also be different, as it now results from A decoding G' rather than G. Assuming that, relative to A, G' codes for any functional machinery at all, we can denote this as P' where $G' = \phi(P')$. So now the produced offspring have the form $M' = (P' + \phi(P'))$. The subsequent behaviour of these offspring — and in particular whether the change will now breed true — will depend on the exact nature of the differences between P and P'.

Note that even though (by hypothesis) we are considering only a local ("point") change from G to G', it does not follow that the change from P to P' will be similarly localised: this depends entirely on the nature of the decoding function ($\phi^{-1}()$) implemented by A. In the general case, $P' = \phi^{-1}(G')$ might be arbitrarily different from P. Nonetheless, for our current purposes we will suppose that the effect is at least approximately local; in particular, let us consider the case where the difference between P and P' is localised to just one of the components A, B, C or D.

The simplest case to understand is where only D is affected, changing to D', i.e., $P' = A + B + C + D'$. In this case, assuming that D' does not interfere with the operation of A, B or C, then $M' = (A + B + C + D') + \phi(A + B + C + D')$ has exactly the same abstract architecture as the original M and therefore will successfully self-reproduce. Thus, any changes of this sort — changes in G that affect only the D part and function of the offspring — will indeed subsequently breed true. Such changes fully qualify as *heritable mutations* in the normal biological sense. Further, if it so happens (as it sometimes may) that D' is even slightly more complex (more complicated in its behaviour) than D, then this will be an example of an incremental evolutionary growth in machine complexity. Ultimately the entire set of machines of the form $M' = (A + B + C + D') + \phi(A + B + C + D')$ for arbitrary D' are all individually self-reproducing and are fully connected together through this network of possible heritable mutations — spontaneous variations in G that do not affect A, B or C in the offspring.

At this point we can recognise that decomposition of M into P and G is closely analogous to the biological decomposition of an (individual) organism into its *phenome* and *genome* respectively. Broadly speaking, modifications to the phenome may or may not impair the ability of the organism to reproduce; but if it *can* still reproduce then the offspring will not inherit this purely phenotypic change to the parent. Conversely, at least some genotypic changes to the parent

will both be expressed in the offspring phenome and subsequently breed true. And indeed if, as previously mentioned, the D component forms the bulk of the machine M, and, correspondingly, the bulk of G codes for D, then we can expect that the bulk of possible changes to G will, if they result in viable offspring at all, fall into this category of giving rise to heritable mutations. Further, we can now recognise the "decoding" function ϕ^{-1}, implemented by A, as corresponding to what would be called a *genotype-phenotype* mapping in biology. For notational convenience below, we will also denote this mapping by $\psi \triangleq \phi^{-1}$.

But let us now return to the final remaining cases of possible machine modifications; namely a change to G to G' (i.e., still a "genotypic" change) where this results in a change of (one of) A, B or C in the offspring. At first sight, one is inclined to suppose that the earlier analysis of *direct* changes to A, B or C in the parent machine can be immediately reapplied to this (first generation) offspring, and conclude that all such mutant offspring will in fact be sterile (so that these cases will also have no evolutionary significance). And indeed, in von Neumann's original presentation that is exactly the position he adopted. However, in fact, this overlooks additional possibilities which do merit deeper consideration. Viable changes to B or C should surely not be absolutely ruled out; but for our particular purposes here we will choose to focus specifically on the role of changes localised to subsystem A, the programmable constructor.

Let us consider then the behaviour of a first generation offspring machine of the form $M' = (A' + B + C + D) + \phi(A'+B+C+D)$. If A' is completely broken, then, indeed, M' will be sterile, as von Neumann supposed. But what if A' is still essentially functional in the sense that it does "decode" $G' = \phi(A'+B+C+D)$ to produce the component P' of its (now second generation) offspring; *but* the exact decoding function implemented by A' has changed — it is no longer $\psi()$ but some more or less modified function $\psi'()$?

In this case the outcome depends critically on the detailed behaviour of this modified decoding function; indeed, it depends on the behaviour of this function precisely for the particular element of its domain represented by $G' = \phi(A' + B+C+D)$. Probably the most common case will be that this decodes as some arbitrary $P' = \psi'(\phi(A'+B+C+D))$. No general further analysis of that case is possible, but we can reasonably expect that that (second generation) offspring will be largely or wholly non-functional i.e., we again conclude that M' is sterile.

But there is still the logical possibility that, even though $\psi \neq \psi'$ (by hypothesis) we might still have $\psi'(\phi(A' + B + C + D)) = (A' + B + C + D)$, i.e., for that particular G' the two functions might still co-incide. We would say that ψ' (or A') is *backwards compatible* with ψ (or A) for that particular value of their common domain. In that (special, but conceivable) case, the machine M' can be equally well represented as having the structure $A' + B + C + D + \phi'(A' + B + C + D)$. By the logic of the abstract architecture, this is indeed self-reproducing, so we again have an example of a heritable mutation; but of a quite different order to the "normal" case of a simple change affecting D'. In fact, this would correspond to *a mutation in the genotype-phenotype mapping itself*. In one mutational step, we would have moved from exploring the space of machines $M = (A + B + C + D) + \phi(A + B + C + D)$, with arbitrary D and all sharing a common genotype-phenotype map ψ defined by A, to exploring a different space of machines $M' = (A' + B + C + D) + \phi'(A' + B + C + D)$, still with arbitrary D, but now sharing a different genotype-phenotype map ψ'. This may not seem like a big change at first sight — as the space of machines D, which putatively defines all of the "interesting" variation in complexity, may still be essentially identical in both cases. But, the topology of the mutational connections — and thus the dynamics of evolutionary change, and the ultimate complexity which emerges — may be radically different.

The very least we might conclude is that any system that has this possibility for evolutionary modification of the genotype-phenotype map has, at the very least, some additional degrees of evolutionary freedom that would not otherwise be available. This will depend, of course, on the space of "possible" decoding functions $\psi()$ and on the subset of these that might be mutationally accessible from any particular starting point (the particular $\psi()$ implemented by some particular, initial, seed, machine). It seems very difficult to make any general statement about this; but one particular case would be where the potential machinery represented by A is capable of incorporating, as part of the decoding process, any arbitrary Turing computation (i.e., the available configurations for A encompass the flexibility of universal computation). This would at least guarantee that the space of possible genotype-phenotype mappings (and corresponding evolutionary dynamics) spans a very wide range of possibilities. However, even this still leaves completely open the question of how richly connected this mutational space might be; i.e., how common would be the "backwards compatible" mutations that actually allow ψ to successfully mutate?

Finally for this discussion we should note again that this possibility of a mutable genotype-phenotype mapping relies on a peculiarly tangled loop of inter-dependency. For self-reproduction to work in von Neumann's architecture, the programmable constructor (A) must always be described or encoded into the genome under a mapping ($\phi()$) which precisely inverts the mapping that that constructor itself physically realises ($\psi()$), at least for that one particular description represented by genome G. Howard Pattee in particular has especially drawn attention to this aspect of von Neumann's architecture, and distinguished it with the term *semantic closure* (Pattee, 1982).

Self-Reproduction in Coreworlds

As noted, the most detailed proposal for an artificial realisation of the von Neumann self-reproduction architecture was in his two dimensional CA universe. Von Neumann himself, having set aside this work in the form of a planned, but unfinished manuscript, did not have the opportunity to return to it before his untimely death in 1957. In any case, the detail and required scale of the model was such that it is only very recently that it has become technically feasible to build a practical implementation[7]. More importantly, and quite independently of scaling issues, that particular realisation was never expected or intended to support investigation of any substantive evolutionary processes, because the machines are intrinsically fragile with no ability to interact (even with their own offspring) without causing catastrophic breakdown in organisation (McMullin, 2000).

By contrast, so-called *coreworld* systems do allow practical investigation of relatively large populations of self-reproducing agents over sufficient time to allow significant evolutionary change. These agents can be conceived as as small machine code programs, each occupying a block of allocated memory and executed by a dedicated processor (CPU). Memory blocks and CPUs are managed and allocated from a common pool.

The roots of the coreworld concept can be traced back to Nils Barricelli, who was invited to visit the Institute for Advanced Studies (IAS) in Princeton by von Neumann in the early 1950s (Barricelli, 1957).[8] However, the more recent and most extensively studied systems of this type are Tierra (Ray, 1992, 1994) and Avida (Adami and Brown, 1994; Adami, 1998).

While these systems do involve self-reproduction of a sort, to date this has not been implemented using the von Neumann architecture. Instead, reproduction is achieved by direct "self-inspection".[9] That is, these universes are designed in such a way that machines (realised as software agents or processes) of arbitrary complexity (within the context of the particular universe) can successfully examine their own detailed internal structure (their own memory image) without disruption; and can copy this structure into a newly allocated memory block, allocate a separate CPU, and then release an essentially identical offspring machine into the (shared) environment. In terms of the von Neumann architecture it is as if the G and P components are combined in one single, active, structure which also serves as its own description; accordingly, it need only be copied — no "decoding" step is needed as the copied structure is already directly constructed in the required "functional" form. But viewed as P we can still decompose it into a part concerned with self-inspection/copying (corresponding to von Neumann's B) and a part corresponding to all machinery not directly concerned with reproduction (von Neumann's D). There is no A or C, and no separate G. Instead, we have just $G \equiv P = (B + D)$

Heritable mutations are still perfectly possible. Any modification anywhere in the memory image which does not impair the inspection (reproduction) functionality (i.e., almost any change to part D, and typically at least some changes to part A) will result in an offspring which is still capable of self-reproduction, preserving that modification — i.e., the modified/mutated machine will breed true.

Accordingly, in this architecture there is no decomposition into a separate "phenome" and "genome". In effect, the whole machine (or at least its tape/memory image) is simultaneously both phenome and genome.

Despite this seemingly radical simplification, a family of such self-inspectors (defined by a specific B and arbitrary D) still meets all the desiderata to address von Neumann's problem as it was expressed earlier.

Now this kind of pure self-inspection architecture need not be feasible in general. Thus, in von Neumann's own CA based model universe, it is not possible for an arbitrary machine to completely inspect its own internal structure without disruption — it would effectively have to disassemble (i.e., destroy) itself in the process. Similarly, in the physical-chemical world of real biology, there are serious limitations to the possibilities for self-inspection. In such universes, the von Neumann architecture, with its separation of the functional, dynamic, phenome from a static, linear (completely open to non-destructive inspection) genome clearly enables a qualitatively richer set of self-reproducing machine configurations, and thus a qualitatively richer potential for evolutionary exploration. But conversely, in universes such as coreworlds, which have been specifically engineered to support comprehensive, non-destructive, inspection of arbitrary machine configurations, it would seem that all conceivable evolutionary phenomenology must already be available via machines with the self-inspection architecture; so even if machines with the full von Neumann architecture *could* be realised in such universes, there would arguably be no point or interest in doing so.

Nonetheless, there are grounds for suggesting that this issue should not yet be completely closed. In particular, note that if we restrict attention to self-inspectors, not only is there no decomposition into genome (G) and phenome (P); there is also no property of semantic closure (in Pattee's sense) and no *mutable mapping from genotype space to phenotype space*. Given the discussion of the previous

[7] It is reported that "... in 2008, the hashlife algorithm was extended to support the 29-state and 32-state rulesets ... [and] On a modern desktop PC, replication now takes only a few minutes" (Wikipedia contributors, 2012).

[8] Barricelli actually did his earliest work of this type on the "IAS Machine", the stored programme electronic digital computer built at Princeton to von Neumann's original design and under his direction.

[9] We focus here on coreworld systems; but note that the self-inspection mechanism has been proposed in other frameworks also (e.g., Laing, 1977; Morita and Imai, 1996).

section we would like to hold open at least the possibility that even though the von Neumann architecture is not *necessary* for "general purpose" self-reproduction in coreworlds, nonetheless it may be *useful*; it may give rise to evolutionary dynamics and phenomenology (including exploration or complexification of the genotype-phenotype mapping) that would not otherwise arise.

So, can we say whether it is possible in principle to embed agents having the von Neumann architecture in coreworld systems?

The general, in principle, answer is "yes". A coreworld already provides facilities for a machine (software agent) to allocate an additional memory block, configure it (write into it) as desired, allocate a CPU to start execution on that memory block, and release this functioning assembly into the environment as a separate, autonomous, agent. As it is possible for an agent to inspect and copy its entire memory image into the offspring memory block, then it should surely be possibly to copy only some part of the image (to be identified as G). The remaining part of the offspring can be separately populated by executing a more or less arbitrary "decoding" process on the content of G. Assuming the coreworld instruction set is Turing complete this decoding can, in principle, involve any arbitrary Turing computation.[10] The P part of the agent is the only part that is directly executed during this process; and it will decompose straightforwardly into program sections having the functionality of the von Neumann components A, B and C; any remaining part of P, which is functional and allows the agent to perform behaviours unrelated to reproduction, will correspond to von Neumann's D.

Experimental investigation of the implementation and evolutionary behaviour of such agents is currently underway in both Tierra and Avida. For my immediate purposes here I simply mention some of the specific questions to be studied:

- How frequently are viable mutations of the genotype-phenotype map ($\psi()$) observed? How does this vary with the specific initial choice of map?
- Is the structure of the von Neumann architecture (the decomposition of M into $G + P$, and the further decomposition of P into $A + B + C + D$) stable under evolution? Or do the roles of these components become blurred?
- More particularly, can the reproduction architecture revert back to simple self-inspection?
- To the extent that the reproduction architecture does *not* revert back to self-inspection, does the seeding with a von Neumann ancestor alter the typical evolutionary dynamics of these systems? For example, does the typical parasite (hyperparasite etc.) phenomena of Tierra still occur?

[10] In the case of Tierra, Turing completeness was originally demonstrated by Maley (1994), albeit it turned out to be extremely clumsy due to the lack of instructions to directly move data between memory and CPU registers. A similar limitation affects the default configuration of Avida. However, in both cases it is straightforward to enable additional instructions to remove these limitations.

Minimal Biological Self-Reproduction

Having now articulated the two clearly distinct abstract approaches to self-reproduction — the von Neumann indirect, genotype-phenotype architecture, and the approach of comprehensive self-inspection and inspection — let us consider in more detail the relation between these and biological self-reproduction. I will focus on primitive, minimal, forms of biological self-reproduction on the basis that these give the best possibility of identifying analogues to these abstract mechanisms.

Von Neumann's work was presumably grounded in some prior biological knowledge — including the conceptual distinction between genotype and phenotype, the known fact that the genotype had a material basis in the chromosome(s), and that it admitted of some degree of linear decomposition in the form of linkage maps among discrete, heritable, "genes". He was likely also aware of Schrödinger's specific suggestion that the chromosomes involved some kind of "aperiodic crystal" structure (Schrödinger, 1944). However, von Neumann still devised his abstract architecture around five years[11] before the detailed molecular structure of DNA was first identified (Watson and Crick, 1953). It was striking then that DNA turned out to be a polymer with a linear primary structure, where there are four distinct possible monomers that can be used at each position, in arbitrary sequence, while still being compatible with the overall structure. Further, during (cellular) reproduction, this sequence is precisely copied to a separate DNA molecule that is distributed to the offspring. Subsequent work in molecular biology demonstrated further that there exists active molecular machinery which can "decode" ("translate") the information sequence from the DNA to produce all the key functional and structural components of the cell, primarily in the form of proteins. In particular, there is a very well defined (and almost universal) coding between DNA sequence and protein sequence, which is *prima facie* strongly reminiscent of the $\psi()$ mapping in the von Neumann architecture.

In more detail then, it is possible to identify specific molecular analogues of several elements of von Neumann's architecture:

- G: Corresponds to DNA; in the case of bacteria, literally a single DNA chromosome.
- P: All the active molecular machinery, coded for by the DNA genome. Largely made up of structural and functional proteins. These can be partially refined as:

[11] First publicly presented at the 1948 Hixon Symposium (von Neumann, 1951).

- A: The protein synthesis machinery, particularly the ribosomes, but also including the DNA to RNA transcription enzymes, and a number of other essential components.
- B: The DNA replication enzymes (DNA polymerase and others).
- C: DNA transcription regulators etc.
- D: All remaining protein (and RNA) components, not directly involved in reproduction.

While these analogies are striking, they should not be overstated either. There are also many points of contrast:

- DNA is double stranded, and this is an essential feature of the molecular mechanism for sequence replication ("inspection").
- While bacteria do have just a single DNA molecule, this is circular rather than strictly linear. More complex organisms have a genome organised into multiple chromosomes; and may retain multiple versions of each (diploidy or polyploidy).
- Biological reproduction can and does commonly involve transfer of DNA between separate individuals (horizontal gene transfer, sexual recombination).
- The von Neumann self-reproduction cycle is largely sequential; whereas, while some cellular processes do have a somewhat sequential pattern (e.g., DNA replication and transcription, mRNA translation), most cellular processes proceed concurrently and asynchronously.
- While there is a clear and well established molecular mapping between DNA sequence and protein sequence, the latter alone certainly does not represent a complete specification of even phenotype (morphology, behaviour etc.) even at the bacterial level; so von Neumann's abstract mapping $\psi()$ encompasses much more than just the so called "genetic code" — the latter is, at best, one initial step in this overall mapping.
- While proteins play a dominant functional role in the cell, RNA molecules also have some critical and pervasive functions. This is significant in terms of the architectural analogy in that RNAs are only transcribed from DNA — not *translated* from it. They form an intermediary in the translation to proteins (as messenger- or mRNAs) but also play active roles as enzymes (so-called ribozymes) including as important components in the ribosomes, and in the form of the transfer- or tRNAs which mediate the translation of mRNA codons (triplets of bases) into specific amino acids (protein monomers) according to the genetic code. There is no direct analog in the von Neumann architecture for the RNAs.

Prima facie the role of RNAs in the translation process might call into question whether even the genetic code component of the genotype-phenotype mapping is itself "encoded" into the genome in a self-referential manner; i.e., whether this arrangement satisfies the Pattee criterion of semantic closure. However, it turns out that the key determinants of the specific coding relationships are not RNAs but protein enzymes (the aminoacyl tRNA synthetases). These are, therefore, coded for in the genome according to the decoding they themselves specify, so this subtle aspect of the von Neumann architecture actually does hold reasonably well (cf., Hofstadter, 1985).

Finally, let us consider whether or to what extent the pure self-inspection mechanism of self-reproduction, traditionally employed in coreworld systems, has any direct biological analog. It is clear that no cellular organism has this self-inspection character. Viruses appear like a better candidate (and, indeed, this underlies the use of the term "computer virus" for malware which functions to reproduce itself "in the wild" of computer networks).

However, biological viruses rely on exploitation of molecular machinery in host cells for replication (copying) of the viral genome, and generally for transcription and/or translation to produce additional functional components (protein sheaths etc.) required for effective propagation. So biological viruses cannot reasonably be said to *self*-reproduce in the manner of traditional coreworld reproducers. It might be argued here that even coreworld reproducers need access to *some* external resources to function — but these are just the primitive, unstructured, "raw materials" of memory and CPUs provided by the coreworld universe. This is not comparable or analogous to the reliance of a biological virus on exploiting the already structured, complex, machinery of host cells.

A better analogy would appear to be to the *in vitro* molecular evolution systems, in which "naked" RNA molecules replicate by direct template inspection, and can give rise to perfectly Darwinian evolutionary dynamics (Joyce, 2007). Admittedly, this is no longer a "natural" biological system, but it is derived from real biological materials. This does compare well to the coreworld self-inspectors in one important respect: the single sequence of monomers in the molecules essentially functions simultaneously as both genome and phenome: the complete sequence is replicated (copied), and also directly determines (via the folded three-dimensional structure) the relevant enzymatic (ribozyme) activity. But the analogy is still weak in that such systems require the external provision not just of (already quite complex!) "raw materials" in the form of activated nucleotides, but also very complex protein enzymes to mediate the replication itself (e.g., $Q\beta-$replicase).

The last analogy we might consider is with the molecular replicators of the so-called RNA-World hypothesis (Gilbert, 1986). In this case it is assumed that there must have ex-

isted at least some these RNAs with the ability to function as RNA-polymerases (ribozyme RNA-replicases). Since these would then be able to make copies of themselves, it would seem that these would be quite precisely analogous to coreworld self-inspectors. Admittedly, this would still be somewhat hypothetical: no such general purpose ribozyme RNA-replicases have been identified to date. But more importantly, even here, there is still one crucial *dis*-analogy. Even if ribozyme RNA-replicases are possible, it is not supposed that one individual molecular could literally function to replicate *its own* sequence. Rather, it is assumed that it could replicate the sequence of another already existing copy of itself. This may seem like a minor detail; but in fact the selectional dynamics of such systems would be very different indeed (showing hyperbolic rather than exponential growth, and a typical phenomenon not of Darwinian "survival of the fittest", but "survival of the most common"). Consequently, the resulting evolutionary phenomena would also be expected to be very different indeed from anything normally occurring in coreworld systems.

Conclusion

This paper has set out to review and compare the two "canonical" forms of machine self-reproduction that have been proposed in the field of Artificial Life. Of these, the simpler, self-inspection, mechanism has been fully realised in the so-called coreworld systems, and the resulting evolutionary dynamics have been the subject of extensive experimental investigations. The other, the von Neumann architecture, has largely been studied in the context of cellular automaton universes. It has only recently been realised in any practical experimental system; and only in forms where, quite aside from very high computational demands, significant evolutionary dynamics are impossible even in principle due to the intrinsic fragility of the reproducing agents. It has been pointed out, nonetheless, that there are grounds for supposing that the von Neumann architecture may facilitate certain kinds of evolutionary exploration that are not possible at all in systems based exclusively on self-inspectors; and that moreover, while both these forms of self-reproduction are very abstract compared to any real biological reproducers (or even hypothetical ones such as the replicating molecules of the RNA-world), the von Neumann architecture does admit significantly more and deeper points of analogy with real molecular biology than self-inspectors do. An outline has been given of how von Neumann architecture reproducers could, in fact, be realised in coreworld systems; and some specific questions have been formulated for study in such a research program.

Acknowledgements

This work was supported by Complexity-NET Project EvoSym, via the Irish Research Council for Science, Engineering and Technology (IRCSET).

References

Adami, C. (1998). *Artificial Life, An Introduction*. Springer.

Adami, C. and Brown, C. T. (1994). Evolutionary learning in the 2D artificial life system "Avida". In Brooks, R. A. and Maes, P., editors, *Proc. Artificial Life IV*, pages 377–381. MIT Press.

Barricelli, N. (1957). Symbiogenetic evolution processes realized by artificial methods. *Methodos*, 9(35-36):143–182.

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

Gilbert, W. (1986). Origin of life: The RNA world. *Nature*, 319:618.

Hofstadter, D. (1985). The genetic code: Arbitrary. In *Metamagical Themas: Questing for the Essence of Mind and Pattern*, page 671–699. Penguin Books, London.

Joyce, G. F. (2007). Forty years of in vitro evolution. *Angewandte Chemie (International Ed. in English)*, 46(34):6420–6436. PMID: 17634987.

Laing, R. A. (1977). Automaton models of reproduction by self-inspection. *Journal of Theoretical Biology*, 66:437–456.

Maley, C. C. (1994). The computational completeness of Ray's Tierran assembly language. In Langton, C. G., editor, *Artificial life III*, pages 503—514. Addison-Wesley.

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

Morita, K. and Imai, K. (1996). A simple self-reproducing cellular automaton with shape-encoding mechanism. In Langton, C. G. and Shimohara, K., editors, *Artifical Life V*, page 489–496. MIT Press, Nara, Japan.

Pattee, H. (1982). Cell psychology: an evolutionary approach to the symbol-matter problem. *Cognition and Brain Theory*, 5(4):325–341.

Ray, T. S. (1992). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artifical Life II*, volume X, pages 371–408. Addison-Wesley Publishing Company, Inc., Redwood City, California.

Ray, T. S. (1994). An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life*, 1:179–209.

Schrödinger, E. (1944). *What Is Life?* Cambridge University Press, 1st edition edition.

von Neumann, J. (1951). The general and logical theory of automata. In Jeffress, L. A., editor, *Cerebral Mechanisms in Behavior—The Hixon Symposium*, pages 1–41. John Wiley and Sons.

Watson, J. D. and Crick, F. H. C. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738.

Wikipedia contributors (2012). Von neumann universal constructor. Page Version ID: 487564813.