

Classifying and Relating Affordance-based Models of Reproduction

Matt Webster* and Grant Malcolm†

Abstract

We present the first full formalisation of our approach to reproduction models and classification using Gibson's theory of affordances. Our formal reproduction models incorporate a labelled transition system that describes how a reproductive system changes over the course of reproduction. The actors in the system are represented by a set of entities together with a relation describing the states in which those entities are present, and a function mapping actions to sets of entities which enable those actions to be performed. Finally, a sequence of actions uniquely identifies the path taken by a single entity, the reproducer, during reproduction. We show how reproduction models can be classified based on whether the reproducer is assisted or unassisted in reproduction, and whether or not the reproducer is active during reproduction. We prove that all assisted and unassisted reproduction models have a related model which has the opposite classification, and generalise the classifications based on self-description and reproductive mechanism given in earlier work to the general case of classification by aspects. We discuss the relevance to the field of artificial life, give a potential application to the fields of computer virology, and demonstrate reproduction modelling and classification in action using examples.

Keywords: reproduction, formal model, classification, affordances, aspects.

*Corresponding author. Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK. Email: matt@csc.liv.ac.uk. Tel: +44 (0) 151 795 4239.

†Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK. Email: grant@csc.liv.ac.uk. Tel: +44 (0) 151 795 4244.

1 Introduction

The ability to reproduce is at the heart of what it means to be alive, both for biological and artificial life systems (see, e.g., [29, 21]). Indeed, a great deal of the early work in artificial life, such as von Neumann’s reproducing automaton [28] and Langton’s loops [13], focuses precisely on reproduction.

Defining the boundary between animate and inanimate is difficult: on the one hand, a definition should be liberal enough to admit all instances that are generally held to be living; on the other hand, it should be strict enough to rule out instances that display many life-like characteristics, yet stretch intuitions about what life — artificial or biological — really is. For example, biological viruses are often argued to be on the boundary between animate and inanimate [27].

In contrast to this, reproduction seems comparatively straightforward to define. There are many clear and paradigmatic examples of reproducers, both artificial and biological: biological organisms and the genes that control them [5], von Neumann’s reproducing automaton [28], computer viruses [3] and other forms of reproducing malware [6], and so forth. However, there are other examples that stretch intuitive definitions of reproduction: photocopies [12], gliders in Conway’s Game of Life [8], seeding crystals, fixed points of mathematical functions, or even a pen on a desk which, being in a stable state, ‘reproduces’ from one instant to the next thanks to the physical laws of the universe.

An ideal definition of reproduction would rule out such rogue examples; an alternative approach would be to accept a *spectrum* of reproduction, and structure the space of reproducers in such a way that trivial and rogue examples of reproducers inhabit a clearly defined region that is separated from the region containing the paradigmatic examples. In this paper, we adopt the latter approach, distinguishing between paradigmatic and rogue examples according to whether or not the reproducer appears to rely on external agency. We consider ‘entity-based’ models of reproduction, in which the reproducer is one entity interacting with others in its environment. Our notion of external agency is based on Gibson’s theory of affordances, which describes the functional relationships between an animal and its environment [9]. For an animal, an affordance is an ‘opportunity for action.’ For example, a piece of food affords an animal nourishment, a tree affords it the ability to climb to safety, and a cave affords shelter. We use affordances to form a classification of reproducers based on the functionality that they afford to themselves and the functionality that their environment affords them, with respect to the process of reproduction.

This gives a basic classification between *assisted* and *unassisted* models

of reproduction, which is presented in Section 2.1 below, along with an orthogonal division into ‘trivial’ and ‘non-trivial’ models. However, we want to do more than simply divide the class of reproducers: we are interested in this class as a *structured space*; i.e., we would like to be able to capture similarities and uniformities between particular models. To this end we introduce a notion of *refinement* between models that captures the intuition that two models may describe the same reproductive process at different levels of abstraction. We present some fundamental results relating refinement to our basic classifications: non-trivial models cannot be refined by trivial models, and trivial assisted models cannot be refined by trivial unassisted models.

The structured space of reproducers is further elaborated in Section 3, where we present our main results: every assisted reproduction model is refined by an unassisted model and, conversely, every non-trivial unassisted model refines an assisted model. These results reflect different ways of viewing the reproductive process. In the case of an assisted reproducer, it is possible to view its reproduction as unassisted if one considers the entities that afford reproduction to be a single, collaborating whole. In the case of an unassisted reproducer, it is still possible to view its environment as affording reproduction; for example, von Neumann’s reproducing automaton can be viewed as being dependent on the transition function of the cellular automaton in which it is realised.

Our classification into assisted and unassisted models is based on earlier work [32, 30], where we presented four main classes of reproduction model, which we called Types I, II, III, and IV. These classes were based on predicates concerning whether the sets of actions corresponding to the self-description and reproduction mechanism of the reproducer were assisted by entities other than the reproducer, or not. The case where both were unassisted corresponded naturally to unassisted reproduction; the case where the self-description was unassisted and the reproductive mechanism was assisted we called Type II; the case where the self-description was assisted and the reproductive mechanism was unassisted we called Type III; and the case where both were assisted we called Type IV. In Section 3.3 we show that this type of classification can be generalised for arbitrary predicates on the actions in the reproduction model using ‘aspect-based’ classification.

Throughout Sections 2 and 3 we illustrate our formal affordance-based models, and their classification and refinement, with worked examples based on biological and computer viruses.

In the concluding section we compare our work with other approaches, including the work of Freitas & Merkle [7], Taylor [26], McMullin [15], Luksha [14], Rosen [19, 20] and Chaitin [2]. We discuss the applicability of our results to different artificial life forms, as well as a practical application to

the problem of computer virus detection. Finally, we give some directions for future research.

2 Formal Models of Reproduction

Our goal is to classify and study the relationships between models of reproductive processes in a rigorous way. In this section we specify precisely what we mean by a model of a reproductive process: on the one hand, we want our notion of model to be general enough to cover as many examples as possible, while on the other hand, we want the notion to have enough structure to allow us to capture relevant similarities and differences between specific models. Since any model of reproduction necessarily identifies some *thing* that reproduces, it seems reasonable to take an individual-based approach to modelling reproduction, and we will assume that reproductive models identify a collection of individuals that play some role in the reproductive process, and that the reproducer itself is a particular individual in this collection. We also assume that the reproductive model specifies a state space and the events that occur to move from one state to another.

A state space together with events or ‘actions’ that move the system from one state to another form a *labelled transition system*, which consists precisely of a set S of states, a set A of actions, and a ternary relation $\mapsto \subseteq S \times A \times S$ specifying the transitions between states. Given such a labelled transition system, we usually write $s \xrightarrow{a} s'$ instead of $(s, a, s') \in \mapsto$, to indicate that action a may move the system from state s to state s' . As an example, consider the life cycle of a bacteriophage virus, which consists of five stages: (i) **a**ttachment of the virus to the host cell; (ii) **i**ntroduction of the virus’s genome to the interior of the cell; (iii) **s**ynthesis of new virus parts; (iv) **m**aturation of these parts into mature offspring; and (v) **r**elease of these offspring back in the environment. At this schematic level, there are five actions: $A = \{\mathbf{a}, \mathbf{i}, \mathbf{s}, \mathbf{m}, \mathbf{r}\}$, and the state space has six states: $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$, where s_1 represents the initial state before the bacteriophage attaches to the cell, s_2 represents the state after attachment, and so forth. The labelled transition system as a whole can be pictured thus:

$$s_1 \xrightarrow{\mathbf{a}} s_2 \xrightarrow{\mathbf{i}} s_3 \xrightarrow{\mathbf{s}} s_4 \xrightarrow{\mathbf{m}} s_5 \xrightarrow{\mathbf{r}} s_6 \quad (1)$$

In this schematic model of the bacteriophage, we might posit just two entities: the cell, which is present in all states except the final state, and the bacteriophage, which, as the reproducer, is present in at least the first state s_1 , and final states s_5 and s_6 . This gives us a very simple model of bacteriophage reproduction, in which we identify the bacteriophage with its own

offspring. Note that Cohen [4] adopts such an identification in modelling computer viruses that may change their source code from generation to generation, introducing the terminology ‘viral set’ for the set of all instances of the virus code that might be so generated. We might take an even more abstract view of identity, and identify the bacteriophage with its genome, in which case we would have a model where the bacteriophage is present in all states.

This very abstract and schematic model of bacteriophage reproduction gives an example of

Definition 1. *A basic reproduction model is a tuple*

$$(S, A, \mapsto, Ent, r, \varepsilon, p) ,$$

where

- (S, A, \mapsto) is a labelled transition system;
- Ent is a set of ‘entities’ with $r \in Ent$ the particular entity that reproduces in the model;
- $\varepsilon \subseteq Ent \times S$ is a binary relation, with $e \varepsilon s$ indicating that entity e is present in the state s ;
- p is a path through the transition system representing the reproduction of r , i.e., p consists of a sequence $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n$ with $s_{i-1} \xrightarrow{a_i} s_i$ for $0 < i \leq n$, and with $r \varepsilon s_0$ and $r \varepsilon s_n$.

The last item in this definition states that there is at least one path through the transition system that shows that the reproducer does actually reproduce. In the bacteriophage example, the path is the entire system, as pictured in (1).

The model of bacteriophage reproduction described above is very much idealised, and *schematic* insofar as the states are just abstract labels for idealised stages in the life cycle of a prototypical bacteriophage. In this regard, it is very similar to the diagrams that can be found in introductory textbooks on virology. We shall present more concrete models later on, but even at this schematic level of abstraction, we can see that the bacteriophage’s life cycle involves an essential interaction between the bacteriophage and a — similarly prototypical — cell. Indeed, it is clear that the bacteriophage requires a cell for all stages in its life cycle, placing it at one end of a spectrum of reliance upon external agency; reproducers at the other end of the spectrum are those that are able to reproduce without the help of other entities: these might

include single-celled organisms, or von Neumann’s reproducing automaton, for example.

In order to capture a notion of assistance within our models, whereby certain actions happen as a result of the presence of one or more entities, we postulate a function Aff that assigns to any action a the set of entities, $Aff(a)$, that mutually *afford* the action a , i.e., those entities without whose presence a could not be performed. We make this formal in the following

Definition 2. *An affordance-based reproduction model is a tuple*

$$(S, A, \vdash, Ent, r, \varepsilon, p, Aff) ,$$

where $(S, A, \vdash, Ent, r, \varepsilon, p)$ is a basic reproduction model, and

$$Aff : A \rightarrow \mathcal{P}(Ent)$$

such that, for all states s , if a is possible in s (i.e., $s \xrightarrow{a} s'$ for some state s'), then $e \in s$ for all e in $Aff(a)$.

For example, in our model of the bacteriophage life cycle, given the set of entities $Ent = \{b, c\}$, with b the bacteriophage, which is present in all states, and c the cell, present in all states except s_6 , we might set

$$Aff(a) = \{b, c\}$$

for all actions a . We could read this as saying that all actions are afforded by the cell to the bacteriophage, or that all actions require both the cell and the bacteriophage to be present. Note, however, that this is simply one way of modelling the bacteriophage’s lifecycle, and so depends entirely upon the intentions of the modeller. It would be just as acceptable, on a formal level, to set

$$\begin{aligned} Aff(\mathbf{a}) &= Aff(\mathbf{i}) = Aff(\mathbf{r}) = \{b, c\} \\ Aff(\mathbf{s}) &= Aff(\mathbf{m}) = \{c\} \end{aligned}$$

which would imply that only the cell is necessary for synthesis and maturation. Yet again, we might set

$$Aff(a) = \{b\}$$

for all actions a , stating that the bacteriophage relies only on itself for reproduction, giving, once more, a different model to the previous ones, perhaps reflecting a modeller’s assumption that, say, cells are a freely available resource.

We shall revisit in Section 2.2 the issue of different models of ‘the same’ reproductive process, but with the notion of affordance we now have sufficient theoretical machinery in place to model a wide variety of reproductive systems, and turn now to the question of classifying such systems according to their reliance on external agency.

2.1 Classifying Reproduction Models

An obvious next step is to classify reproduction models according to whether or not the reproducer is assisted in its act of reproduction. We shall say that those reproduction models in which reproduction is not assisted by any other entity are *unassisted*, and those in which reproduction is assisted are, naturally, *assisted*.

In order to simplify the definition of assistance, it is useful to think of the set of all entities which aid the act of reproduction. We call this set the *ecology* of a model.

Definition 3. *The ecology of a model M , $E(M)$, is the union of the sets $Aff(a_i)$ for a_i in the path p .*

We can then classify a model M as assisted or unassisted, depending on the ecology of M .

Definition 4. *An affordance-based reproduction model M can be classified as unassisted iff there is no entity e , different from the reproducer r , in $E(M)$. Conversely, a model M is assisted iff there is some entity e different from r in $E(M)$.*

As illustrated in this definition, for brevity we often refer to an affordance-based reproductive model simply as ‘a model’.

As well as classifying models based on assistance, we can classify based on whether the reproducer participates in its own reproduction or not. The idea of a reproducer not participating in its own reproduction might seem paradoxical, at first, but is quite natural in some circumstances. For instance, a photocopy does little in the act of its own reproduction. It is likely that many such “trivial” examples of reproduction involve reproducers that do not seem to participate in their own reproduction, e.g., gliders within Conway’s Game of Life. We summarise this distinction in

Definition 5. *A model M is trivial iff the reproducer $r \notin E(M)$. Conversely, M is non-trivial iff $r \in E(M)$.*

We now demonstrate how a reproduction model for a copier computer virus can be defined and classified.

Example 1. *The following copier computer virus reproduces when it is executed by the Bourne Again Shell (‘Bash’) interpreter in Unix:*

```
cp $0 $0.copy
```

The command `cp` takes as arguments two filenames, and copies the contents of the first file, if it exists, into the second file, which will be created if it does not already exist. The expression `$0` is a special variable that is set to the name of the shell script that is currently running.

In general, for a computer virus, we would like to base a reproductive model on an operational semantics [17] for the programming language in which the virus is written. That is, the labelled transition system has programs as labels, and the states are those of an abstract machine that executes the language. An operational semantics formally specifies the transition relation $s \xrightarrow{p} s'$ by specifying which states s' may be reached by executing program p in starting state s .

In this case, we can represent the state of a computer running the Bash interpreter as a tuple $\text{FS} \mid \text{B} \mid \text{CS}$, where FS represents the filestore, B represents the state of the Bash interpreter, and CS represents a sequence of shell script commands that are to be executed. For the sake of simplicity, we will assume that the filestore is just a sequence of shell scripts, and we will represent each script as $[\text{FH} : \text{CS}]$, where FH is the name of the script and CS is the sequence of shell-script commands in the script. The state of the Bash interpreter would consist of variable–value pairs for all of Bash’s environmental variables; since for our example we are interested only in the variable `$0`, we will represent the state of the interpreter simply as $\$0 : \text{FH}$, where FH is the value of the variable `$0`. As for the commands, we will restrict attention to names of shell scripts and commands of the form `cp E1 E2`, where E1 and E2 are expressions. Thus, for example,

$[\text{virus} : \text{cp } \$0 \$0.\text{copy}] \mid [\$0 : \text{null}] \mid \text{virus}$

represents a state where the only script in the filestore is the copier virus, the variable `$0` has value `null`, and the command about to be executed is a call of the shell script `virus`.

We will not spell out the details of the operational semantics for this simplified Bash interpreter here; the interested reader can find a formal description written in the specification language Maude [16] online [34]. For our present purposes, it is sufficient to note that the operational semantics permits the following path:

$$\begin{array}{l} [\text{virus} : \text{cp } \$0 \$0.\text{copy}] \mid [\$0 : \text{null}] \mid \text{virus} \\ \xrightarrow{\text{get}} \\ [\text{virus} : \text{cp } \$0 \$0.\text{copy}] \mid [\$0 : \text{virus}] \mid \text{cp } \$0 \$0.\text{copy} \\ \xrightarrow{\text{subst}} \\ [\text{virus} : \text{cp } \$0 \$0.\text{copy}] \mid [\$0 : \text{virus}] \mid \text{cp } \text{virus } \text{virus}.\text{copy} \\ \xrightarrow{\text{cp}} \end{array}$$

```
[virus.copy : cp $0 $0.copy] [virus : cp $0 $0.copy]
| [$0 : virus] |
```

which shows that executing the virus causes its code to be reproduced.

In this model we identify three entities: the copier computer virus (*cv*), the reproducer in this model; the string rewriting agent (*sra*), which rewrites $\$0$ to the name of the script currently executing; and the *cp* command, which creates the copy of the virus. Therefore $Ent = \{cv, sra, cp\}$. Since substitution for $\$0$ and copying file contents are basic functions of the Bash interpreter, we let *sra* and *cp* be present in all states — this is a reasonable choice for our simplified operational semantics; in a more detailed semantics, we might, for example, specify that *cp* is not present in certain ‘error’ states arising from hardware or software failures that make the filestore unavailable. We further specify that *cv* is present in all states where the command *cp \$0 \$0.copy*, or the result of substituting for $\$0$ in this, is present either in the filestore or as a command about to be executed by the interpreter.

Thus far, we have defined a basic reproduction model; we make this an affordance-based model by specifying:

$$\begin{aligned} Aff(subst) &= \{sra\} \\ Aff(cp) &= \{cp\} \\ Aff(get) &= \{cv\} \end{aligned}$$

It is readily checked that these equations satisfy the constraint of Definition 2, and that the result is a non-trivial assisted model.

Several variations on this model may be given by changing the definition of the function *Aff*. For example, if one feels that substitution for $\$0$ is a freely available resource that can be taken for granted, one may set $Aff(subst) = \emptyset$. Similarly, one may have $Aff(cp) = \emptyset$ if one feels that copying files may be taken for granted. Together, these two changes would give a model in which the entities *sra* and *cp* may be considered surplus to requirements and dropped from the set *Ent*. This model would then be a non-trivial unassisted model.

2.2 Refinement of Reproduction Models

It should be clear from the preceding examples that we are not classifying reproducers *per se*; rather, we are classifying *models* of reproducers, and we allow for the possibility that a reproductive process may be modelled in many different ways. It is possible that this permissiveness might seem inappropriate. After all, it might be argued, the primary goal of a model is

verisimilitude: things are one way or another, and the obligation on a model is to say which way things are; so if there are two different models of the same thing, then at least one model is wrong.

Our view, which may show a bias towards practices in Computer Science, is that it is often useful to allow different models of the same process, perhaps at different levels of abstraction, or reflecting different states of understanding of the process being modelled. In software engineering, for example, it is common to start with a very abstract specification of a system, and repeatedly refine this by adding more details and constraints, until a final, very concrete specification is reached. Each version of the specification can be seen as a model of the not-yet-realised system, at varying levels of abstraction. The important relationship between the different models is a form of consistency: the more concrete models impose more constraints on admissible behaviours, or every behaviour allowed by the concrete models is also allowed in the more abstract models. In this section we present a notion of refinement for affordance-based reproduction models that captures the idea that one model provides a more concrete view of the same process modelled by another. We will then show that it is possible to freely move between viewing a process as assisted or unassisted.

Consider the schematic model of a bacteriophage’s life-cycle presented in Section 2; the following gives a more concrete version of this lifecycle.

Example 2. *We use terms to represent individual cells, bacteriophages, and bacteriophage RNA. For example, we use `b-rna` to name a particular bacteriophage RNA sequence, and write `T4[b-rna]` to denote an individual T4 bacteriophage with that sequence (we are not concerned with any specific mechanics of RNA reproduction in this model, so we need do no more than name the RNA here). Similarly, we write `Cell[]` for an individual cell, and we denote states where several individuals coexist by simply juxtaposing the terms for the individuals; thus, for example,*

`Cell[] T4[b-rna] Cell[] T4[b-rna] T4[b-rna]`

denotes a state containing two cells and three bacteriophages. We consider this state to be equivalent to any permutation of its constituent entities. Technically, we mean that juxtaposition is an associative and commutative operation; semantically, we think of this state as a ‘soup’ in which the constituent entities can ‘move around’ in order to interact with one another. We also allow a similar sort of soup to exist within a cell’s membrane; for example,

`Cell[b-rna T4[b-rna] b-rna b-rna]`

denotes a single cell that contains three bacteriophage RNA strands, and one mature bacteriophage.

Such a situation can come to pass by a bacteriophage attaching to a cell and injecting its RNA. We write $C-T$ for a cell C with attached bacteriophage T , and postulate two rewrite rules that allow attachment and injection of RNA (we omit the labels of the actions):

$$\begin{aligned} C T &\mapsto C-T \\ \text{Cell}[S]-T4[R] &\mapsto \text{Cell}[S R] \end{aligned}$$

which say, respectively, that bacteriophage T can attach to cell C , and that when a bacteriophage with RNA R is attached to a cell that contains ‘internal soup’ S , the RNA R can be injected into that internal soup. Similarly, we give rewrite rules that allow bacteriophage RNA to be replicated inside a cell, and that allow bacteriophage RNA to mature into T_4 bacteriophages:

$$\begin{aligned} \text{Cell}[R S] &\mapsto \text{Cell}[R S R] \\ \text{Cell}[\text{b-rna } S] &\mapsto \text{Cell}[S T_4[\text{brna}]] \end{aligned}$$

Finally, in the life-cycle of the bacteriophage, we allow cells to rupture, releasing matured bacteriophages into the environment:

$$\text{Cell}[S] \mapsto S$$

Clearly, these five rewrite rules correspond to the five schematic stages of the bacteriophage life-cycle. Moreover, we can see these stages applied to individuals, as in the following example:

$$\begin{aligned} &T_4[\text{b-rna}] \text{Cell}[] T_4[\text{b-rna}] \\ &\mapsto \\ &T_4[\text{b-rna}] \text{Cell}[]-T_4[\text{b-rna}] \\ &\mapsto \\ &T_4[\text{b-rna}] \text{Cell}[\text{b-rna}] \\ &\mapsto \\ &T_4[\text{b-rna}] \text{Cell}[\text{b-rna b-rna}] \\ &\mapsto \\ &T_4[\text{b-rna}] \text{Cell}[\text{b-rna b-rna}] \\ &\mapsto \\ &T_4[\text{b-rna}] \text{Cell}[T_4[\text{b-rna}] \text{b-rna}] \\ &\mapsto \\ &T_4[\text{b-rna}] T_4[\text{b-rna}] \text{b-rna} \end{aligned}$$

which shows a $T4$ bacteriophage attaching to a cell, injecting its RNA, that RNA being copied, maturing, and then being released as the cell ‘ruptures’ (albeit after minimal reproduction and maturing of the $T4$ RNA).

The states of this model are the terms of sort ‘soup’, and the rewrite rules given above determine the actions and transitions. We can postulate entities comprising a cell, \mathbf{Cell} , which is present in a state iff that state has a subterm of the form $\mathbf{Cell}[\dots]$, and bacteriophages, present in a given state iff the RNA, $\mathbf{b-rna}$ occurs as a subterm. Furthermore, we can take the condition in Definition 2 to define affordances, which gives us that all the actions (attachment, injection, etc.) are afforded jointly by the cell and the bacteriophages (and hence this model is an assisted reproduction model).

In order to relate this model (call it $T4Cell$) to the schematic model of Section 2, note first that we can map states of that model to states of $T4Cell$ as follows:

$$\begin{aligned}
s_1 &\mapsto f(s_1) = \mathbf{Cell} [] \ \mathbf{T4}[\mathbf{b-rna}] \\
s_2 &\mapsto f(s_2) = \mathbf{Cell} [] \text{-}\mathbf{T4}[\mathbf{b-rna}] \\
s_3 &\mapsto f(s_3) = \mathbf{Cell} [\mathbf{b-rna}] \\
s_4 &\mapsto f(s_4) = \mathbf{Cell} [\mathbf{b-rna} \ \mathbf{b-rna}] \\
s_5 &\mapsto f(s_5) = \mathbf{Cell} [\mathbf{b-rna} \ \mathbf{T4}[\mathbf{b-rna}]] \\
s_6 &\mapsto f(s_6) = \mathbf{b-rna} \ \mathbf{T4}[\mathbf{b-rna}]
\end{aligned}$$

This maps the path of the schematic model to the path of $T4Cell$; i.e., it preserves transitions, which have the same actions in both models. The schematic model had two entities, b and c , representing the bacteriophage and the cell, respectively. We can map these to entities $h(b) = \mathbf{T4}[\mathbf{b-rna}]$ and $h(c) = \mathbf{Cell}$, respectively, noting that this preserves occurrences in states: if $e \varepsilon s$, then $h(e) \varepsilon f(s)$. Moreover, since all actions are afforded jointly by the cell and the bacteriophage in both models, affordances are clearly preserved as well.

We generalise this example in

Definition 6. For basic reproduction models $M = (S_M, A_M, \mapsto_M, Ent_M, r_M, \varepsilon_M, p_M)$ and $N = (S_N, A_N, \mapsto_N, Ent_N, r_N, \varepsilon_N, p_N)$, a refinement $M \rightarrow N$ is a triple of functions (f, g, h) , where $f : S_M \rightarrow S_N$, $g : A_M \rightarrow A_N$, and $h : Ent_M \rightarrow Ent_N$, such that

1. $s \xrightarrow{a}_M s'$ implies $f(s) \xrightarrow{g(a)}_N f(s')$,
2. $e \varepsilon_M s$ implies $h(e) \varepsilon_N f(s)$, and

3. $h(r_M) = r_N$.

Moreover, if M and N are affordance-based models, then $(f, g, h) : M \rightarrow N$ is a refinement iff, in addition, $h(\text{Aff}_M(a)) \subseteq \text{Aff}_N(g(a))$ for all actions $a \in A_M$ (note we write $h(X)$ for the set resulting from applying h to every element of the set X).

Intuitively, a refinement $M \rightarrow N$ indicates that M and N model the same process, but N provides a more detailed or concrete model, i.e., N refines M . Since transitions, occurrences and affordances are all preserved, all of the behaviour, entities, and affordances described in M also occurs in N , although N may provide more states, actions, and entities than figure in M .

Despite this intuitive image of a refinement as an inclusion of one model within another, there are many interesting refinements where one or more of the component functions is not injective. In particular, models may be refined by ‘amalgamating’ two or more entities. We will see in the following section that such refinements can move freely between assisted and unassisted reproductive models.

Example 3. We give a refinement of the copier computer virus model M_{cv} from Example 1. Let N_{cv} be as follows:

- $S_{N_{cv}}, A_{N_{cv}}, \vdash_{N_{cv}}, r_{N_{cv}} (= cv)$ and $p_{N_{cv}}$ are identical to those in M_{cv} ;
- $\text{Ent}_{N_{cv}} = \{cv, sra+cp\}$;
- $\text{Aff}_{N_{cv}}(\text{subst}) = \text{Aff}_{N_{cv}}(cp) = \{sra+cp\}$; and
- $sra+cp \in_{M'_{cv}} s$ iff $sra \in_{M_{cv}} s$ or $cp \in_{M_{cv}} s$.

It is readily checked that the identity functions $1_{S_{M_{cv}}} : S_{M_{cv}} \rightarrow S_{M_{cv}}$ and $1_{A_{M_{cv}}} : A_{M_{cv}} \rightarrow A_{M_{cv}}$, together with the function mapping cv to cv and both sra and cp to $sra+cp$, give a refinement $M_{cv} \rightarrow N_{cv}$. Note that in this case, both M_{cv} and N_{cv} are assisted reproductive models.

2.3 Allowed Refinements of Reproduction Models

The definitions of assistance and triviality given earlier are independent dichotomies, meaning that we can divide the space of reproduction models into four disjoint parts, depending on the assistance and triviality of a reproduction model. It is interesting to note that refinement between these four parts is limited in certain directions, which implies a structure of the space of reproduction models. Firstly, there are no refinements from non-trivial models to trivial models.

Proposition 1. *If M and N are models, and there is a refinement $M \rightarrow N$, then M being non-trivial implies that N is non-trivial.*

Proof. If M is non-trivial then we know that $r_M \in \mathbf{E}(M)$. By Definition 6, we know that $h(r_M) = r_N$ and $h(\text{Aff}_M(a)) \subseteq \text{Aff}_N(g(a))$. Therefore $r_N \in \mathbf{E}(N)$ and N is non-trivial, as desired. \square

Secondly, there are no refinements to trivial, unassisted models.

Proposition 2. *For all trivial, unassisted affordance based models, N , there is no refinement $M \rightarrow N$, where M is trivial and assisted.*

Proof. Proof is by contradiction. Suppose that a refinement $M \rightarrow N$ exists. Since M is assisted, then by Definition 4 there must be some entity $x \in \text{Ent}_M$, different from the reproducer r , such that $x \in \text{Aff}_M(b)$ for some action b in the path. By Definition 6 we know that $h(\text{Aff}_M(b)) \subseteq \text{Aff}_N(g(b))$ and therefore $h(x) \in \text{Aff}_N(g(a))$. However, $\text{Aff}_N(b') = \emptyset$ for all actions b' in the path, because N is trivial and unassisted. Therefore, there can be no such function h , and therefore the refinement cannot exist, as desired. \square

It is straightforward to demonstrate that refinements are allowed in all other directions, and therefore these proofs are omitted. The resulting structure of the space of reproduction models is summarised in Figure 1.

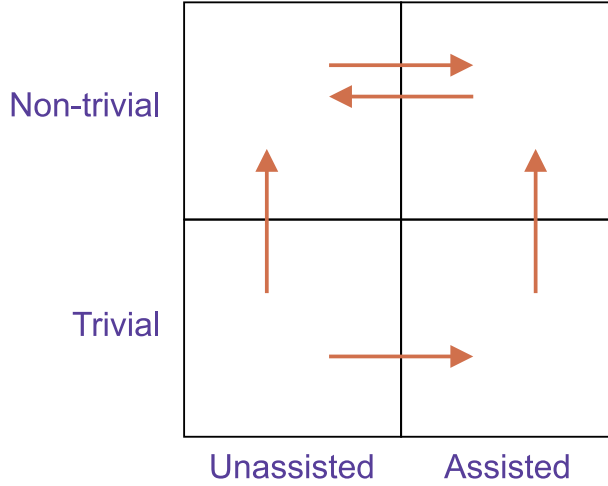


Figure 1: Allowed refinements between classes of affordance-based models.

3 The Unassisted and Assisted Reproduction Theorems

In this section we show that the classification into assisted or unassisted models does not reflect an intrinsic property of the process being modelled, but rather reflects the decisions made in constructing a particular model. We show that every model can be refined by an unassisted model, and, dually, every model refines an assisted model.

3.1 The Unassisted Reproduction Theorem

One potential application of refinement is to identify the entities that assist the reproducer in the act of reproduction, and conglomerate them with the reproducer to form a ‘super-entity’. Of course, the resulting aggrandised reproduction model will then be an unassisted model irrespective of whether the original reproduction model was assisted or unassisted. In the following Propositions 3–6, and culminating in Theorem 1, we will show how for any reproduction model, M , there is a corresponding unassisted reproduction model $M^\#$ — in which states, actions, action successions, and the reproducer’s path remain unchanged — for which there is a refinement arrow $M \rightarrow M^\#$. The first step is to define $M^\#$ in

Definition 7. *Given a reproduction model $M = (S, A, \mapsto, Ent, r, \varepsilon, p, Aff)$, we define*

$$M^\# = (S, A, \mapsto, Ent^\#, r, \varepsilon^\#, p, Aff^\#)$$

where

1. $Ent^\# = (Ent \setminus \mathbf{E}(M)) \cup \{r\}$;
2. $r \varepsilon^\# s$ iff $e \varepsilon s$, for some entity $e \in \mathbf{E}(M) \cup \{r\}$; and for all $e \in Ent \setminus \mathbf{E}(M)$, $e \varepsilon^\# s$ iff $e \varepsilon s$; and
3. $Aff^\#(a) = h(Aff(a))$, where $h : Ent \rightarrow Ent^\#$ maps $c \in \mathbf{E}(M)$ to r , otherwise $h(e) = e$.

Proposition 3. *For any model M , $M^\#$ is an affordance-based reproduction model.*

Proof. By Definition 2, we require that for all $e \in Aff^\#(a)$, and for all states s with a is possible in s then $e \varepsilon^\# s$. Suppose that action a is possible in state s , and $e \in Aff^\#(a)$. By Definition 7(3), $e = h(e_0)$ for some $e_0 \in Ent$, and because M is an affordance-based reproduction model, it follows that

$e_0 \varepsilon s$. If $e_0 \in \mathbf{E}(M)$, then $e = h(e_0) = r$ and $e \varepsilon^\# s$ by Definition 7(2); if $e_0 \notin \mathbf{E}(M)$, then $e = h(e_0) = e_0 \varepsilon s$ and so $e \varepsilon^\# s$ as desired. \square

Now that we have established that both M and $M^\#$ are models, we must check that $M^\#$ is in fact unassisted.

Proposition 4. *For any model M , $M^\#$ is unassisted.*

Proof. The only entities which afford reproductive actions (i.e., those in p) to r in M are those in $\mathbf{E}(M)$. Therefore, for any a_i in p , if $e \in \text{Aff}^\#(a_i)$, then $e = h(c)$ for some $c \in \mathbf{E}(M)$ and so $e = r$. \square

Next we show that $M^\#$ refines M .

Proposition 5. *For all models M , there is a refinement $M \rightarrow M^\#$.*

Proof. The refinement $M \rightarrow M^\#$ consists of the triple $(1_S, 1_A, h)$, where 1_S and 1_A are identities on states and actions, and $h : \text{Ent} \rightarrow \text{Ent}^\#$ is as defined in Definition 7. Clearly, transitions are preserved (Definition 6(1)), and preservation of occurrences (Definition 6(2)) follows immediately from Definition 7(2), and we need show only $h(\text{Aff}(a)) \subseteq \text{Aff}^\#(a)$ for all actions a , but this is immediate from Definition 7(3). \square

This gives us our main result for this section:

Theorem 1 (Unassisted Reproduction Theorem). *Every reproduction model can be refined by an unassisted reproduction model.*

In other words, for any reproduction model, be it assisted or not, there is another model which captures the same reproductive process but with modified entities, which is classified as unassisted. Therefore, all models of assisted reproduction can be refined so that they instead capture unassisted reproduction. This tells us that apparent unassisted reproduction is simply a consequence of the way a particular reproductive process is modelled, since it can also be modelled as unassisted reproduction.

In order to demonstrate Theorem 1, we give an example of a model $M_{cv}^\#$ that is a refinement of the copier computer virus model M_{cv} (cf. Examples 1 and 3), and is an unassisted model.

Example 4. *Let $M_{cv}^\#$ be constructed from the model M_{cv} of Example 1, as in Definition 7. This gives us:*

- $S_{M_{cv}^\#}, A_{M_{cv}^\#}, \vdash_{M_{cv}^\#}, r_{M_{cv}^\#} (= cv)$ and $p_{M_{cv}^\#}$ are identical to those in M_{cv} ;
- $\text{Ent}_{M_{cv}^\#} = \{cv\}$

- $cv \varepsilon_{M_{cv}^\#} s$ for all $s \in S$
- $Aff_{M_{cv}^\#}(a) = \{cv\}$ for all actions a .

The refinement $M_{cv} \rightarrow M_{cv}^\#$ consists of the identity functions on states and actions, and the function that maps all entities to cv .

If a model M is an unassisted model, then our construction of $M^\#$ just yields the original model M :

Proposition 6. *If M is unassisted, then $M = M^\#$.*

This is proved by inspection of Definition 7, noting that if M is unassisted, then $\mathbf{E}(M) \subseteq \{r\}$. A slightly stronger statement says that the construction of $M^\#$ is the smallest change to M that is needed to obtain an unassisted reproduction model. Consider the situation in Figure 2, where the arrow along the top is the refinement of Proposition 5. If there is some other

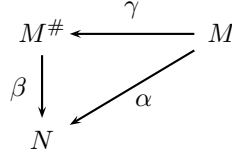


Figure 2: Refinement arrows between M , $M^\#$ and N .

unassisted model N that refines M , then it makes a larger change than $M^\#$ does, and it refines $M^\#$ as well; moreover, it does so in a unique way. This is stated formally in the following

Proposition 7. *For all reproduction models M , $M^\#$ is the least unassisted refinement of M .*

Proof. Suppose $(\alpha_f, \alpha_g, \alpha_h)$ is a refinement of M by an unassisted reproduction model $N = (S_N, A_N, \vdash \rightarrow_N, Ent_N, r_N, \varepsilon_N, p_N, Aff_N)$.

We show that there is a unique refinement $\beta = (\beta_f, \beta_g, \beta_h) : M^\# \rightarrow N$ such that α is the composition of γ and β , where $\gamma = (1_S, 1_A, h)$ is the refinement $M \rightarrow M^\#$ of Proposition 5. Since γ is the identity on states and actions, we clearly require $\beta_f = \alpha_f$ and $\beta_g = \alpha_g$, and all that remains is to define β_h . By the definition of refinement, we require $\beta_h(r_{M^\#}) = r_N$, and for all other $e \in Ent^\#$ (i.e., $e \in Ent \setminus \mathbf{E}(M)$), we set $\beta_h(e) = \alpha_h(e)$. This clearly satisfies the uniqueness constraint on β , and we need only show that β does indeed exist; i.e., we need to show that $\alpha_h(e) = r_N$ for all $e \in \mathbf{E}(M)$. If $e \in \mathbf{E}(M)$, then there is some a_i in p_M with $e \in Aff_M(a_i)$, and so $\alpha_h(e) \in Aff_N(\alpha_g(a_i))$, but since N is unassisted, this must mean that $\alpha_h(e) = r_N$, as desired. \square

3.2 The Assisted Reproduction Theorem

From Theorem 1 we know that all reproduction models can be refined by an unassisted reproduction model, and therefore all reproduction models can be viewed as unassisted. In this subsection we show that the converse is also possible, that for all reproduction models, there is a corresponding assisted reproduction model. We define this corresponding model in

Definition 8. *Given a reproduction model, $M = (S, A, \mapsto, Ent, r, \varepsilon, p, Aff)$, we define*

$$M_{\#} = (S, A, \mapsto, Ent_{\#}, r, \varepsilon_{\#}, p, Aff_{\#})$$

where

- $Ent_{\#} = Ent \cup \{G\}$;
- $Aff_{\#}(a) = Aff(a)$ if $r \notin Aff(a)$, and $Aff_{\#}(a) = Aff(a) \setminus \{r\} \cup \{G\}$ if $r \in Aff_M(a)$;
- for all states s , $G \varepsilon_{\#} s$ iff $r \varepsilon s$;
- $e \varepsilon_{\#} s$ iff $e \varepsilon s$ for all entities $e \neq G$.

The model $M_{\#}$ introduces a new entity, G , and ascribes to it all the actions afforded by r in M . We might think of this new entity as the ‘Laws of Nature’, which make possible all the actions viewed (in M) as under the control of the reproducer; such a change of viewpoint might be seen in viewing, on the one hand, organisms as actively reproducing through their own actions, and viewing them, on the other hand, merely as phenotypes of genes that persist, or not, under the action of natural selection. Another example of such a change of viewpoint would be to view reproducers such as Langton’s loops [13] as, on the one hand, reproducing by means of extending a process that loops back on itself, or, on the other hand, as just phenomena that emerge from the iterative application of the evolution rule of the cellular-automata grid in which the loops are realised. In this example, the entity G is the evolution rule; in the previous example, it is natural selection. We might say that the Unassisted Reproduction Theorem of the previous section represents an ecological view of reproduction, in which a number of separate entities that collaborate in a process of reproduction can be viewed as a entity in itself, whereas the construction of this section represents a reductionistic approach, in which any behaviour (though especially reproduction for our purposes) can be viewed as a manifestation of physical or computational laws.

Proposition 8. *For all reproduction models M , $M_{\#}$ is a reproduction model and there is a refinement $M_{\#} \rightarrow M$.*

Proof. It is clear from the construction that $M_{\#}$ is an affordance-based reproduction model whenever M is. The refinement $M_{\#} \rightarrow M$ consists of the triple $(1_S, 1_A, h)$, where 1_S and 1_A are identities, and h is defined as follows: $h(G) = r$, and $h(e) = e$ for $e \in Ent$. We must check that the conditions from Definition 6 hold. Condition (1) holds trivially, because S , A and \mapsto are identical in M and $M_{\#}$. Condition (2) holds by construction of $M_{\#}$, as does condition (3). The final condition, that $h(Aff_{\#}(a)) \subseteq Aff(a)$, holds because $Aff_{\#}$ replaces r by G and h maps G to r . \square

The Assisted Reproduction Theorem follows from this, with one proviso: that the original model M is *non-trivial* (cf. Definition 2).

Theorem 2 (Assisted Reproduction Theorem). *Every non-trivial reproduction model M refines an assisted reproduction model.*

Proof. This follows directly from Proposition 8, noting that if $r \in Aff(a_i)$, then $Aff_{\#}(a_i) = Aff(a_i) \setminus \{r\} \cup \{G\}$, so $r \neq G \in Aff_{\#}(a_i)$, and therefore $M_{\#}$ is an assisted model. \square

The requirement that M be non-trivial is from Definition 8, as $M_{\#}$ is assisted if M is non-trivial.

We illustrate this construction by revisiting the copier virus M_{cv} of Example 1, an assisted reproducer that was rendered unassisted in Example 4, giving the model $M_{cv}^{\#}$. We now apply the Assisted Reproduction Theorem to give

Example 5. *Let $(M_{cv}^{\#})_{\#}$ be constructed from $M_{cv}^{\#}$ following Definition 8. We have*

- $Ent_{(M_{cv}^{\#})_{\#}} = \{cv, G\};$
- $G \varepsilon_{(M_{cv}^{\#})_{\#}} s$ for all $s \in S$
- $Aff_{(M_{cv}^{\#})_{\#}}(a) = \{G\}$ for all actions a .

In other words, G is omnipresent and omnipotent in that it alone affords all the actions of the copier virus's reproductive cycle.

These examples show that our approach does not say that assisted and unassisted models are the same thing: the constructions of Definitions 7 and 8 are not bijections, as M_{cv} and $M_{\#}$ are different. What our approach *does*

say is that a reproductive process may be viewed qualitatively in different ways. Moreover, the notion of refinement serves to rank these qualitatively different approaches to modelling the same process: ecological approaches are more refined than reductionistic approaches, or, following Example 2, more *schematic*.

3.3 Further Classification Using Aspects

In earlier work [32, 30], we classified reproducers into four categories, which we called Types I, II, III, and IV. These categories were based on identifying which actions in the model were concerned with obtaining a self-description (SD) of the reproducer, and which actions were concerned with the actual machinery (RM) of constructing a copy of the reproducer. Models where all actions in both groups were unassisted were categorised as Type I, which in the present paper we have simply called ‘unassisted’; the remaining types subdivide the assisted class of models. The case where SD was unassisted and RM was assisted we called Type II; the case where SD was assisted and RM was unassisted we called Type III; and the case where both were assisted we called Type IV.

As we argued in [32, 30], a self-description and its use in the construction of a copy of the reproducer plays a central role in a great number of reproduction models. However, there may be applications where other aspects of the reproductive process play a more prominent role. For example, many computer viruses contain code that is designed to prevent the infection of already-infected files, or code that is intended to avoid detection by — or even to actively attack — anti-virus software, and the ecology or dependencies of these code fragments are therefore of interest to the producers of anti-viral software. As another example, in biological organisms, sexual reproduction, random mutations in DNA, and the interactions of the organism itself with its environment, including, for example, predators or potential mates, are the aspects of the reproductive cycle where natural selection plays a role: it would be hard to explain a peacock’s tail without a notion of mate-selection. In this section we shall generalise this mode of classification to arbitrary predicates on the actions in the reproduction model, which we call *aspects*.

Formally, an aspect is just a name, such as ‘self-description’, ‘detection-avoidance’, or ‘mate-selection’. As such, it has no intrinsic formal meaning, and its application to reproduction models depends on the intentions of the modeller. It applies to a particular model as a predicate on the actions of that model, saying which actions are concerned with that aspect. For example, the actions of the T4 bacteriophage life-cycle (see Section 2) that are concerned

with reproductive machinery would be synthesis of the bacteriophage RNA and its maturation. The actions of the copier computer virus (see Example 1) concerned with the same aspect would be the call of the `cp` function; the actions in this model concerned with obtaining a self-description would be the substitution of the virus’s file name for the variable `$0` — but again, there is freedom for interpretation on the part of the modeller: it would be perfectly acceptable to also include the call of `cp` as concerned with obtaining a self-description, which would be reasonable, as this is where the actual viral code is accessed in the file store.

Of course, we may be interested in studying reproduction models with regard to a particular aspect, or with regard to several.

Definition 9. *Given an aspect α , an α -model consists of an affordance-based model M together with a predicate M_α on the actions of M . If C is a set of aspects, a C -model is a model with a predicate M_α for each $\alpha \in C$.*

We can relativise the model theory of the preceding sections to aspects or sets of aspects.

Definition 10. *For aspect α and α -model M , we define the α -ecology of M , $E_\alpha(M)$, to be the union of all the sets $Aff_M(a_i)$ for a_i in the path of M for which M_α holds: i.e., all entities that afford α -actions in M .*

Definition 11. *We say M is α -unassisted iff $E_\alpha(M) \subseteq \{r_M\}$, and M is α -assisted iff $E_\alpha(M)$ contains some entity other than r_M . Similarly, M is C -unassisted iff it is α -unassisted for each $\alpha \in C$, and C -assisted iff it is α -assisted for some $\alpha \in C$.*

Moreover, it is clear that the constructions of the Assisted and Unassisted Reproduction Theorems can be relativised to arbitrary sets of aspects. Thus, for any α -model M , there is an α -unassisted model that refines M , and (for non-trivial M) an α -assisted model that is refined by M .

Any unassisted model will be α -unassisted, for any aspect α , and in this sense, aspects serve to subdivide the space of assisted models. The basic distinction between assisted and unassisted could be viewed as arising from an aspect that holds for all actions in a reproduction model’s path, but in general, n aspects give 2^n categories of assisted reproduction. For example the four-fold categorisation (Types I–IV) described above arises from the two aspects of self-description and reproductive-machinery. Conversely, ‘enough’ aspects serve to recapture the absolute notion of ‘unassisted’:

Proposition 9. *If M is C -unassisted and the set of actions in p_M is covered by the aspects in C , i.e., each a_i in p_M satisfies some M_α for $\alpha \in C$, then M is unassisted.*

Proof. For any a_i in the path of M , there is at least one aspect $\alpha \in C$ for which M_α holds, and since M is α -unassisted, $Aff_M(a_i) \subseteq \{r_M\}$, and so $E(M) \subseteq \{r_M\}$. \square

In summary, aspects provide a generalisation of our basic distinction between assisted and unassisted models of reproduction, and allows finer-grained distinctions between assisted models.

4 Conclusion

Extending our earlier work [32, 33, 30], we have given the first full account of formal affordance-based reproduction modelling, classification and refinement. We have shown that we can specify reproduction systems using models in which discrete time processes are modelled using labelled transition systems. We can define the entities present in a model, and specify in which states they are present. When entities are present in a state, they can afford actions which move that state to another state. We can divide these models by classification as ‘unassisted’ or ‘assisted’ reproduction, the former describing a state of affairs where no actions in the reproducer’s path are afforded by entities other than the reproducer, and the latter describing there is at least one action in the path that is afforded by an entity other than the reproducer. We can also classify models as ‘non-trivial’ or ‘trivial’, dependent on whether or not there are actions in the path that are afforded by the reproducer. Our examples included biological and computer viruses. Biological viruses are interesting phenomena at the boundary of accepted definitions of life [27], and computer viruses are a form of artificial life [23] — the practical problems of detecting and classifying computer viruses was the focus of much of our earlier work referred to above.

Clearly, models of classic artificial life examples are also possible; e.g., reproduction in cellular automata, such as the systems of von Neumann [28] and Langton [13] are easily modelled using labelled transition systems. Indeed, a forthcoming paper [31] gives an account of Langton’s loops in which the reproducing loop is modelled as a composite entity built from its constituent ‘data-paths’.

The main novel contribution of the present paper is to present affordance-based reproduction models as a structured space. Models of the same reproductive process at different levels of abstraction are related by refinement, and we have presented some fundamental results on refinements between trivial and non-trivial, and between assisted and unassisted models. In particular, our Assisted and Unassisted Reproduction Theorems show that it is possible

to model the same process in different ways so that it can be viewed as either assisted or unassisted, and that unassisted models, where the reproducer is viewed as part of an ecology, are more refined than unassisted models, where reproduction is viewed as a result of computational or physical laws.

There is an interesting parallel between this latter case and work by Rosen [18] on a paradox arising from reproduction models where an automaton reproduces: such an automaton can be viewed as a mapping $f : A \rightarrow B$, from inputs A to outputs B , so if f reproduces, then $f \in B$, but the mapping f cannot be defined before its output set B is. Rosen notes that the paradox is avoided in cases where reproduction is mediated: his example is of a mapping $F : B \times (A \times B) \times A \rightarrow (A \times B) \times (A \times B)$ such that $F(y, (x', y'), x) = ((x', y), (x, y'))$, so that for components a , and b of a structure (a, b) , we have $F(b, (a, b), a) = ((a, b), (a, b))$ — i.e., the structure (a, b) is reproduced from its components. Rosen comments ‘the structure to be duplicated is not possessed of an active role in the duplication process but is utilized, as far as the theory of automata is concerned, in a relatively passive manner’ (*op. cit.*, p. 391). In our terminology, F *affords* reproduction to the structure.

In earlier work [32, 33] we described how reproduction models could be classified according to two separate dichotomies based on whether or not the actions corresponding to the reproductive mechanism and self-description of a reproductive process were afforded by entities other than the reproducer. In Section 3.3 we generalised classifications of this sort using aspects: predicates on the actions in a reproduction model. Since each aspect is either unassisted or assisted, a number n of aspects gives up to 2^n different classes of reproduction models.

Classification of reproduction based on the reliance of a reproducer on the environment can also be found in the works of Freitas & Merkle (ch. 5, [7]), Taylor [26, 25] and Luksha [14]. Freitas and Merkle give categories for the location of replication information (i.e., self-description) and replicator parasitism (i.e., reliance on external agency for the reproductive machinery). Taylor divides the reproducer space into two: reproduction occurs either with or without reliance on external agency (auto- and assisted-reproduction respectively). Luksha offers a categorisation of reproducers based on the relative complexities of the reproducer and its environment, from ‘quasi-self-reproducers’ with no formal reproductive structures ‘fully capable reproducer’ which reproduces independently.

The distinction between assisted and unassisted reproduction is, therefore, a recurrent theme in reproduction classification. Our approach offers formal models of reproduction systems, and reduces the problem of classifica-

tion to a purely formal exercise. Of course, the process of encoding a reproduction system within a reproduction model remains subjective; however, with the Unassisted and Assisted Reproduction Theorems we have shown that unassisted and assisted models are related in refinements, and therefore any bias is mitigated by the fact that any unassisted model has a corresponding model which describes essentially the same reproduction system, but is actually classified as assisted, and vice versa. In this way we have given a formal first step towards eliminating bias in the subjective act of reproduction modelling.

Formal models of reproduction have also been given by McMullin [15] and Rosen [19, 20], who given formalisation of the act of reproduction based on mathematical functions. Chaitin [2] works towards a formal definition of life based on algorithmic information theory. The works of von Neumann, Codd, Langton and others [22] on cellular automaton-based formulations of reproductive behaviour are formal, but are largely constructive proofs of the existence of purely formal descriptions of reproduction. In contrast, we have presented a means of creating formal models of reproduction at the ecological level, in which the reliance of the act of reproduction on entities external to the reproducer can be specified, and classification and refinement of models is possible.

Our approach to reproduction modelling is based on an assumption that entities are recognisable in order to generate a set of symbols representing them, but this is not necessarily the case. In particular, Henz & Misra [11] have given an interesting approach to recognising evolving artificial life forms based on the recurrence of similar patterns due to evolutionary reproductive behaviour.

4.1 Future Work

We conclude by sketching out some areas for future research.

4.1.1 Metrics for Reliance on External Agency

In Section 3.3 we described how reproduction models can be classified as assisted or unassisted, or classified using arbitrary predicates on the actions in the reproduction model (i.e., aspects). However there may be further opportunities to create metrics to compare unassisted reproduction models, based on other factors. For example, if we see the act of reproduction as a computational process of a certain minimal complexity, then if the actions that a reproducer affords itself together are less than the complexity of the whole reproductive process, then there must necessarily be some other (external)

entity that compensates for this. Therefore, when comparing two unassisted reproduction models that require a similar environment (e.g., two computer viruses), then we can compare their reproductive reliance on external agency by comparing the complexity of those reproductive actions that are afforded to the computer viruses, or those that the computer viruses afford themselves. For example, we could assume that the more complex a particular reproducer's self-afforded actions, the less the reliance on external agency. Of course, this presupposes the existence of some level of abstraction at which we can compare the complexity of actions, but in several cases, such as computer viruses, Tierran organisms, cellular automaton reproducers, etc., such a comparison would seem possible. Different methods of complexity could be used, e.g., space/time complexity, or the Kolmogorov complexity of the reproducer itself. Our recent work applying affordance-based classification to computer viruses gave simple examples of how such metrics might be used [30].

There is also empirical evidence of differing degrees of reliance on external agency with respect to biological viruses. It is known that, '[viruses] with large genomes depend less on host functions than those with small genomes' [10]. This effectively states that the information content in the self-description (genome) is related to the reliance on external agency (the host cell). Another possible extension of this work would be use the methods described above to formalise this statement within our ontology.

4.1.2 Strategies for Reproduction

In December 2000, a relatively unprolific virus on the Windows 32-bit platform was able to infect executable files containing relatively prolific network worms [24]. The destructive payload of the virus combined with the infectiousness of the worms created dangerous hybrids that were not predicted by the vendors of anti-malware software. These hybrids were an emergent property of a complex 'ecology' of reproducers, in which reproduction processes could overlap.

A useful extension of this work would be to be able to analyse these ecologies of reproducers, i.e., systems where more than one reproducer is present. Such ecologies could be constructed using affordances common between entities, for example, a bacterium might afford a site of infection for a bacteriophage virus, without necessarily specifying which virus might infect the bacterium. The labelled transition systems of the different reproductive processes could be combined using techniques such as those developed in process algebra [1]. In real-life ecologies, reproducers are capable of inter-

esting behaviours such as crossing a species gap (e.g, biological viruses), or spontaneous virus–worm hybridisation (see above). In being able to build models of ecologies of reproducers by combining their models in a formal way, we could begin to analyse and predict interesting emergent properties of multi-reproducer systems.

4.1.3 Computer Virology: Anti-Virus Applications

In computer virology, computer viruses and network worms spread within computer systems whilst anti-virus software scans for suspect behaviours typical of reproducing malware in a process known as dynamic analysis [6]. As discussed in Section 2.1, we can classify reproduction models as assisted or unassisted, depending on how we model the entities and affordances in those models. In recent work [30] we have shown that a practical application of such classification is in prioritisation of dynamic analysis on systems where resources are limited, e.g., on PDAs, smartphones, PCs, or other pervasive computing applications. Dynamic analysis depends on the ability of anti-virus software to intercept communications between reproducing malware and external entities such as the operating system, daemons/services, the filestore, network protocols, etc. Malware typically must enlist the help of these other entities in its reproductive process. Antivirus software is able to analyse this behaviour and flag it as suspicious in order to detect files infected by malware. In order to apply our ontology, we can say that the act of dynamic analysis by the antivirus software places constraints on the reproduction model that we construct. For example, if the antivirus software is able to intercept calls by a computer virus to the file store (during disk input/output operations, for example), then it is logical to classify the virus and the file store as separate entities. If the anti-virus software cannot intercept calls to the operating system, then effectively it cannot ‘distinguish’ between the virus and the OS, and within the reproduction model we should treat them as one entity. So, when malware is afforded an action by an external entity, and the anti-virus software is able to detect this, the anti-virus scanner has a better chance of detecting the malware than if it could not detect this behaviour.

By classifying malware in this way, we will know that the most difficult malware to detect at run-time will be those whose reproduction models are classified as unassisted, because the anti-virus software cannot detect the behaviour of these viruses and worms, because it cannot intercept the calls made by the virus to external entities. The viruses whose reproduction models are assisted will have behaviours that are detectable by the anti-virus software. So, on a system where resources are limited, the anti-virus analysis

scanner can focus its static analysis attentions on the ‘unassisted’ malware, because this cannot be detected at run-time, and should therefore be high priority.

Acknowledgements

The authors wish to thank the reviewers and participants of The First IEEE Symposium on Artificial Life (CI-ALife 2007, Honolulu, USA) and the 2nd International Workshop on the Theory of Computer Viruses (TCV 2007, Nancy, France) for their useful questions and comments.

References

- [1] Baeten, J. (2005) A brief history of process algebra. *Theoretical Computer Science*, 335, 131–146.
- [2] Chaitin, G. J. (1979) Toward a mathematical definition of “life”. Levine, R. D. & Tribus, M. (eds.), *The Maximum Entropy Formalism*, pp. 477–498, MIT Press.
- [3] Cohen, F. (1987) Computer viruses — theory and experiments. *Computers and Security*, 6, 22–35.
- [4] Cohen, F. (1989) Computational aspects of computer viruses. *Computers and Security*, 8, 325–344.
- [5] Dawkins, R. (1990) *The Selfish Gene*. Oxford University Press, USA, first published 1976. ISBN: 0192860925.
- [6] Filiol, E. (2005) *Computer Viruses: from Theory to Applications*. Springer, ISBN 2287239391.
- [7] Freitas Jr., R. A. & Merkle, R. C. (2004) *Kinematic Self-Replicating Machines*. Landes Bioscience, ISBN 1570596905.
<http://www.molecularassembler.com/KSRM.htm>. Accessed 26th March 2008.
- [8] Gardner, M. (1970) Mathematical games: The fantastic combinations of John Conway’s new solitaire game ‘life’. *Scientific American*, 223, 120–123.
- [9] Gibson, J. J. (1977) The theory of affordances. *Perceiving, Acting and Knowing: Toward an Ecological Psychology*, pp. 67–82.

- [10] Granoff, A. & Webster, R. G. (eds.) (1999) *Encyclopedia of Virology*, vol. 3, pp. 1414–15. Academic Press, entry on “Virus Multiplication Cycle”.
- [11] Henz, M. & Misra, J. (2007) Towards a framework for observing artificial life forms. *Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007)*, pp. 23–30, IEEE Press.
- [12] Hofstadter, D. R. (2000) *Gödel, Escher, Bach: an Eternal Golden Braid*, chap. 16, p. 499. Penguin.
- [13] Langton, C. G. (1984) Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10, 135–144.
- [14] Luksha, P. O. (2003) The firm as a self-reproducing system. *Proceedings of 47th International System Science Society Conference*.
- [15] McMullin, B. (2000) John von Neumann and the evolutionary growth of complexity: Looking backwards, looking forwards. . . . *Artificial Life*, 6, 347–361.
- [16] Meseguer, J. & Roşu, G. (2007) The rewriting logic semantics project. *Theoretical Computer Science*, 373, 213–237.
- [17] Plotkin, G. D. (2004) A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60–61, 17–139, DOI:10.1016/j.jlap.2004.03.002.
- [18] Rosen, R. (1959) On a logical paradox implicit in the notion of a self-reproducing automaton. *Bulletin of Mathematical Biophysics*, 21, 387–394.
- [19] Rosen, R. (1991) *Life Itself*. Columbia University Press.
- [20] Rosen, R. (1999) *Essays on Life Itself*. Columbia University Press, ISBN: 978-0231105118.
- [21] Schrödinger, E. (1944) *What is Life?*. Cambridge University Press.
- [22] Sipper, M. (1998) Fifty years of research on self-replication: An overview. *Artificial Life*, 4, 237–257.
- [23] Spafford, E. H. (1994) Computer viruses as artificial life. *Journal of Artificial Life*, 1, 249–265.

- [24] Symantec Press Centre (2000), Symantec warns computer users of destructive Christmas Day virus/worm mutation. http://www.symantec.com/region/reg_ap/press/my_001219b.html, accessed 26th March 2008.
- [25] Taylor, T. (1999) On self-reproduction and evolvability. Floreano, D., Nicoud, J.-D., & Mondada, F. (eds.), *Proceedings of the Fifth European Conference on Artificial Life (ECAL99)*, Springer-Verlag.
- [26] Taylor, T. J. (1999) *From Artificial Evolution to Artificial Life*. Ph.D. thesis, University of Edinburgh, <http://www.tim-taylor.com/papers/thesis/>. Accessed 26th March 2008.
- [27] Villarreal, L. P. (2004) Are viruses alive? *Scientific American*, 291, 100–105.
- [28] von Neumann, J. (1966) *Theory of Self-Reproducing Automata*. University of Illinois Press, edited by Arthur W. Burks.
- [29] Weaver, N., Paxson, V., Staniford, S., & Cunningham, R. (Spring 2006) Life. Zalta, E. N. (ed.), *The Stanford Encyclopedia of Philosophy*, <http://plato.stanford.edu/archives/spr2006/entries/life/>. Accessed 26th March 2008.
- [30] Webster, M. & Malcolm, G. Classification of computer viruses using the theory of affordances. *Journal in Computer Virology*, to appear. DOI: 10.1007/s11416-007-0079-4.
- [31] Webster, M. & Malcolm, G. Hierarchical components and entity-based modelling in artificial life, submitted to The Eleventh International Conference on the Simulation and Synthesis of Living Systems (ALIFE XI).
- [32] Webster, M. & Malcolm, G. (2007) Reproducer classification using the theory of affordances. *Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007)*, pp. 115–122, IEEE Press.
- [33] Webster, M. & Malcolm, G. (2007) Reproducer classification using the theory of affordances: Models and examples. *International Journal of Information Technology and Intelligent Computing*, 2.
- [34] Webster, M. & Malcolm, G. (2008), Classifying and relating formal affordance-based models of reproduction — Unix Bash script virus specification. <http://www.csc.liv.ac.uk/~matt/pubs/maude/3/>.