Contents lists available at SciVerse ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Research Article

R-Models: A mathematical framework for capturing notions of abstraction and assistance in reproductive systems

Matt Webster[a], Grant Malcolm[b,*]

[a] Virtual Engineering Centre, Daresbury Laboratory, Warrington WA4 4AD, UK
[b] Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK

ARTICLE INFO

Article history:
Received 30 May 2012
Received in revised form 25 July 2012
Accepted 30 July 2012

Keywords:
Modelling
Reproduction
Classification
Affordances

ABSTRACT

R-Models are an approach to capturing notions of assistance and abstraction in reproductive systems, based on labelled transition systems and Gibson's theory of affordances. *R*-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 an affordance-modelling function mapping actions to sets of entities which enable those actions to be performed. We show how *R*-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 *R*-models have a related *R*-model which has the opposite classification. 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.

1. Introduction

In this paper we introduce affordance-based reproduction models, or *R*-models, as a way of capturing notions of assistance and abstraction in reproductive systems. 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., Weber, 2006; Schrödinger, 1944; Koonin, 2012). Indeed, a great deal of the early work in the field of artificial life, such as von Neumann's reproducing automaton (von Neumann, 1966) and Langton's loops (Langton, 1984), focuses precisely on reproduction.

There are many clear and paradigmatic examples of reproducers, both artificial and biological: biological organisms and the genes that control them (Dawkins, 1990), von Neumann's reproducing automaton (von Neumann, 1966), computer viruses (Cohen, 1987) and other forms of reproducing malware (Filiol, 2005), and so forth. There are other examples that stretch intuitive definitions of reproduction: photocopies (Hofstadter, 2000), gliders in Conway's game of life (Gardner, 1970), 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.

The wide variety of reproducing systems has led many researchers to develop taxonomies and classifications of reproduction, e.g., (Sipper, 1998; Freitas and Merkle, 2004; Taylor, 1999, 1999; Luksha, 2003). A recurring theme in these classifications is how much the reproducer relies on its surroundings, the environment, in order to reproduce. For example, Freitas and Merkle categorise reproducers based on whether information for reproduction (such as a self-description in the case of a von Neumann self-replicating automaton) is based within the reproducer or somewhere in its environment, and the degree to which a reproducer is "parasitic", i.e., the degree to which it is reliant on external agency in order to complete the act of reproduction (Freitas and Merkle, 2004, Chapter 5). In another example, Taylor divides the reproducer space into two: reproduction occurs either with or without reliance on external agency (Taylor, 1999, 1999). A third example comes from Luksha's categorisation of reproducers, in which the relative complexities of the reproducer and its environment come into play (Luksha, 2003).

However, such classification can be confusing as in many cases it looks as though there can be more than one classification for a given reproducer. For example, we could say that von Neumann's self-reproducing automaton could not operate without the cellular automaton environment in which it exists. Alternatively, we can take the environment for granted if we decide that its involvement is unimportant, and then we can attribute to the reproducer everything that the environment previously did for the reproducer. In this case reproduction occurs simply as a result of the reproducer's

* Corresponding author.
E-mail addresses: mattpwebster@gmail.com (M. Webster), grant@liv.ac.uk (G. Malcolm).

0303-2647/$ – see front matter © 2012 Published by Elsevier Ireland Ltd.
http://dx.doi.org/10.1016/j.biosystems.2012.07.003

own agency. (In the first case we could say that von Neumann's automaton is "assisted" by its environment, and in the latter case it is "unassisted".) Similar confusion can occur in other examples of reproduction, from bacteria to Langton's loops to computer viruses. We call this the "assistance problem" of reproductive system classification.

Another point of contention when classifying and comparing reproduction systems is the "abstraction problem". It is possible to talk about a reproducer at many different levels of abstraction. For example, we could say that bacteriophage viruses reproduce in a 5-step process corresponding to the stages of attachment to the host cell, insertion of the viral RNA, synthesis of various parts that will make up the offspring, maturation of the offspring and finally release of the offspring into the environment. Alternatively, we could view the bacteriophage's reproduction in terms of the biochemical reactions taking place during the reproductive process, from the use of enzymes to inject viral DNA to the creation of necessary proteins. This could involve many thousands of different reproductive steps. We could also say that the bacteriophage reproduces in one step, that the entire reproduction process is simply a single indivisible event—we may use this view if, for example, we are more interested in how quickly the virus spreads through a population of host cells. All three of these abstraction levels are different, yet they all describe the same reproductive system. In what ways (if any) are these three views of the same reproductive process related?

In this paper we hope to clarify what is meant by "assistance" and "abstraction" using *R*-models, and therefore help to tackle the assistance and abstraction problems for reproduction classification. We introduce the *R*-model mathematical framework to compare different "views" of the same reproductive act, e.g., bacteriophage virus reproduction and computer virus reproduction, and show that different views of the same reproductive process correspond to different *R*-models. Our main results suggest that the notion of assistance is a property not of a reproductive system, but of a *model* of that system. We are able to show that for every *R*-model in which an entity's reproduction is unassisted, there is an *R*-model of the same system in which the entity's reproduction is assisted, and vice versa.

1.1. Structure of the paper

In Section 2 we begin by introducing two of the main ideas on which *R*-models are based: labelled transition systems and affordances. Next we give a formal definition of *R*-models, and show how they can be used to describe different types of assistance and abstraction for reproductive systems. We introduce "refinements" as a way of relating *R*-models to each other, and give examples showing how *R*-models can be constructed and related. In Section 3 we explore the "space" of *R*-models in more detail by describing a number of theoretical results on *R*-models, the most important of which lets us find corresponding assisted *R*-models for any unassisted *R*-model and vice versa (the significance and meaning of these terms will be given in the appropriate sections). We follow this with some practical examples of how these formal results can be applied. We conclude with comparisons with related work, and give directions for future work.

2. Formal models of reproduction

As stated in Section 1, our goal is to be able to relate different views of a reproductive act in a rigorous way, in order to clarify what is meant by "abstraction" and "assistance" within reproductive systems. In this section we specify precisely what we mean by an *R*-model: on the one hand, we want *R*-models 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 *R*-models. Since any model of reproduction necessarily identifies some reproducer that reproduces, it seems reasonable to take an entity-based approach to modelling reproduction, and we will assume that reproductive models identify a set of entities that play some role in the reproductive process, and that the reproducer itself is a particular entity in this set. We also require that an *R*-model specifies a state space and the events that occur to move from one state to another. We now examine in detail the theoretical tools used to specify reproduction in terms of state spaces and entities, namely labelled transition systems and affordances.

2.1. Labelled transition systems

A set of states 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 $s \overset{a}{\mapsto} s'$ instead of $(s, a, s') \in \mapsto$, to indicate that action a may move the system from state s to state s'.

It is important to note at this stage that labelled transition systems, whilst based on discrete mathematical structures, are capable of modelling continuous-time phenomena. One way of doing so is to incorporate continuous time into the state space. States would then be pairs (t, s), where t is a real value denoting time, and s is the state of the system at time t. Transitions would then be of the form $(t_1, s_1) \overset{a}{\mapsto} (t_2, s_2)$, provided that $t_1 < t_2$ and $s_1 \mapsto s_2$ is a valid evolution of the system over the time-interval $[t_1, t_2]$ (as a simple example, if s_1 and s_2 are intended to represent velocities, then requiring $s_2 = s_1 + k(t_2 - t_1)s_1$ would represent a constant acceleration).

As an example of a labelled transition system, consider an abstract view of 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:

$$s_1 \overset{\mathbf{a}}{\mapsto} s_2 \overset{\mathbf{i}}{\mapsto} s_3 \overset{\mathbf{s}}{\mapsto} s_4 \overset{\mathbf{m}}{\mapsto} s_5 \overset{\mathbf{r}}{\mapsto} s_6 \tag{1}$$

2.2. Affordances

Using labelled transition systems we can formalise an act of reproduction in terms of a sequence of events taking place over time. Using Gibson's theory of affordances we can say that one or more of the entities present in a reproduction model assist in a particular action.

Note that our models are ecological, in the sense that we specify an act of reproduction that involves some number of entities within the environment of the reproducer. We do not place restrictions on what these entities can be; they can be animate or inanimate, logical or physical. For this reason we use Gibson's theory of *affordances* (Gibson, 1977, 1979) to describe the way in which one or more entities assist in particular actions in the reproduction model. Gibson presented affordance theory as an "ecological theory of perception", in which animals perceive objects in their environment as being able to assist in some action. For instance, for a mouse a cave might afford shelter, or running water may afford nourishment. We can

Please cite this article in press as: Webster, M., Malcolm, G., *R*-Models: A mathematical framework for capturing notions of abstraction and assistance in reproductive systems. BioSystems (2012), http://dx.doi.org/10.1016/j.biosystems.2012.07.003

stretch this notion even further; the mouse might perceive another mouse of the opposite sex which affords it reproduction. The mouse might perceive three stepping stones, which afford it safe passage across a stream. The mouse may even perceive a predator, which affords death. In each case, the notion of affordance binds some actor (e.g., the mouse) with some entities in its environment. The actor requires the entities to perform an action, e.g., a mouse needs a cave to perform the act of shelter.

In our models we can consider the reproduction of any reproducer. The reader should note that we therefore only use affordances as a metaphor; we do not suggest that a reproducer like a computer virus perceives its environment in any non-trivial way, for example. We could have just as easily used words like “assistance”, “collaboration” or “cooperation” in place of “affordance”; however, we chose “affordance” for its ecological connotations.

2.3. *R-Models: affordance-based models of reproduction*

We can formalise affordances within reproduction models by using a function *Aff* that assigns to any action a in the reproductive act a set of entities, $\mathit{Aff}(a)$, that together *afford* the action a, i.e., those entities without whose presence a could not be performed. We make this formal as follows.

Definition 1. An *affordance-based reproduction model*, or *R-model*, is a tuple

$(S, A, \mapsto, \mathit{Ent}, r, \varepsilon, p, \mathit{Aff})$,

where

- $(S, A, \mapsto)$ is a labelled transition system;
- Ent is a set of ‘entities’ with the reproducer $r \in \mathit{Ent}$ the particular entity that reproduces in the model;
- $\varepsilon \subseteq \mathit{Ent} \times S$ 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 $s_0 \overset{a_1}{\mapsto} s_1 \overset{a_2}{\mapsto} \dots \overset{a_n}{\mapsto} s_n$ with $s_{i-1} \overset{a_i}{\mapsto} s_i$ for $0 < i \le n$, and with $r\ \varepsilon\ s_0$ and $r\ \varepsilon\ s_n$; and
- $\mathit{Aff} : A \to \mathcal{P}(\mathit{Ent})$ is a function from actions to sets of entities, such that, for all states s, if a is possible in s (i.e., $s \overset{a}{\mapsto} s'$ for some state s'), then $e\ \varepsilon\ s$ for all e in $\mathit{Aff}(a)$.

As an example, we consider bacteriophage reproduction as described above.

Example 1. An *R*-model for bacteriophage reproduction might look as follows:

- The labelled transition system, $(S, A, \mapsto)$, is

  $s_1 \overset{\mathbf{a}}{\mapsto} s_2 \overset{\mathbf{i}}{\mapsto} s_3 \overset{\mathbf{s}}{\mapsto} s_4 \overset{\mathbf{m}}{\mapsto} s_5 \overset{\mathbf{r}}{\mapsto} s_6$,

  as in (1), with the actions $\{\mathbf{a}, \mathbf{i}, \dots\}$ corresponding to attachment, insertion, etc.;
- $\mathit{Ent} = \{\mathit{virus}, \mathit{host}\}$ where *virus* is the bacteriophage virus and *host* is the host cell which gets infected;
- ε is defined as follows: $\mathit{host}\ \varepsilon\ s_i$ for $1 \le i \le 5$, meaning that the host cell is present in all states of the reproductive process except the final one (as it has been destroyed in the virus release stage). We also specify $\mathit{virus}\ \varepsilon\ s_1$ and $\mathit{virus}\ \varepsilon\ s_6$ as the virus is present in the first state, i.e., before it injects itself into the host cell, and in the final state, after “copies” of the virus are released into the environment;
- $p = (S, A, \mapsto)$ in this case as the transition system contains only the states, actions and transitions of the bacteriophage’s reproductive process;
- $\mathit{Aff}(\mathbf{a}) = \{\mathit{virus}, \mathit{host}\}$ as the virus and host must both be present in order for this action to take place, and $\mathit{Aff}(\alpha) = \{\mathit{host}\}$ for all $\alpha \in \{\mathbf{i}, \mathbf{s}, \mathbf{m}, \mathbf{r}\}$ as the host cell must be present for all of these actions to take place.

In this schematic model of the bacteriophage, we posit just two entities: the host 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.

It is important to stress that this is not the only possible *R*-model for bacteriophage reproduction. We could use a different labelled transition system consisting of any number of states in the reproductive process, including branches, cycles and disjoint graphs. We could specify any number of entities, for example, we could include the substrate on which the host cell exists, or the food on which it feeds, or another virus in the system, etc. (Note that we require at least one entity, the reproducer.) Likewise we can specify the presence relation ε, the path p and the affordances in the model however we want (subject to the restrictions given in Definition 1). Indeed, we give a more detailed version of bacteriophage reproduction in Example 5 below.

This flexibility makes *R*-models a convenient mathematical framework for describing reproductive systems, and in particular for capturing notions of abstraction and assistance those systems, as we shall see in the following sections.

2.4. *Assistance and refinements for R-models*

We will use *R*-models as a tool for reasoning about abstraction and the ecology of assistance in reproductive systems. Before we do so, we need to formalise notions of assistance and refinement.

2.4.1. *Assistance in R-models*

In order to formalise assistance with respect to reproduction, it is useful to identify the set of all entities which assist in the overall act of reproduction. We call this set the *ecology* of a model.

Definition 2. The *ecology* of an *R*-model M, $E(M)$, is the union of the sets $\mathit{Aff}(a_i)$ for all actions a_i in the path p.

In other words, the ecology of an *R*-model is the set of all the entities that assist in the act of reproduction. We can then classify an *R*-model M as assisted or unassisted, depending on the ecology of M:

Definition 3. An *R*-model M is *unassisted* iff there is no entity e in $E(M)$ that is different from the reproducer r. Conversely, an *R*-model M is *assisted* iff there is some entity e in $E(M)$ that is different from r.

Note that M is unassisted iff $E(M) \subseteq \{r\}$; in general, this need not be an equality, and we can also classify models according to whether or not the reproducer participates in its own reproduction. 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 (considered by some to be a reproducer; Hofstadter, 2000) apparently 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 (Gardner, 1970). We summarise this distinction in

Please cite this article in press as: Webster, M., Malcolm, G., *R*-Models: A mathematical framework for capturing notions of abstraction and assistance in reproductive systems. BioSystems (2012), http://dx.doi.org/10.1016/j.biosystems.2012.07.003

Definition 4. An R-model model M is *trivial* iff the reproducer $r \notin E(M)$. Conversely, M is *non-trivial* iff $r \in E(M)$.

Note that the trivial/non-trivial classification is orthogonal to the assisted/unassisted classification.

We now demonstrate how an R-model for a copier computer virus can be defined and classified as assisted or unassisted, or as trivial or non-trivial.

Example 2. The following copier computer virus reproduces when it is executed by the 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 command which invoked the shell script. As shell scripts are commonly invoked using their own filename, this computer virus is able to make a copy of the file containing itself.

In general, for a computer virus, we would like to base an R-model on an operational semantics (Plotkin, 2004) 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 $s \overset{p}{\mapsto} s'$ 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 `FS` | `B` | `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 `[FH: 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: 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,

$$[\mathtt{virus}:\mathtt{cp}\,\$0\,\$0.\mathtt{copy}]\|[\$0:\mathtt{null}]\|\mathtt{virus}$$

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`.

Now a simplified operational semantics for the Bash interpreter states that there are three actions *get*, *subst*, and *cp*, and permits the following path:

$$\begin{array}{l}
\quad[\mathtt{virus}:\mathtt{cp}\,\$0\,\$0.\mathtt{copy}]\|[\$0:\mathtt{null}]\|\mathtt{virus} \\
\overset{get}{\mapsto} \\
\quad[\mathtt{virus}:\mathtt{cp}\,\$0\,\$0.\mathtt{copy}]\|[\$0:\mathtt{virus}]\|\mathtt{cp}\,\$0\,\$0.\mathtt{copy} \\
\overset{subst}{\mapsto} \\
\quad[\mathtt{virus}:\mathtt{cp}\,\$0\,\$0.\mathtt{copy}]\|[\$0:\mathtt{virus}]\|\mathtt{cp\,virus\,virus.copy} \\
\overset{cp}{\mapsto} \\
\quad[\mathtt{virus.copy}:\mathtt{cp}\,\$0\,\$0.\mathtt{copy}]\,[\mathtt{virus}:\mathtt{cp}\,\$0\,\$0.\mathtt{copy} \\
]\|[\$0:\mathtt{virus}]\,|
\end{array}$$

This path shows that executing the virus causes its code to be reproduced in the file *`virus.copy`*.

We now construct an R-model M_{cv} in which we identify three entities: the copier computer virus (cv), which is the reproducer in this R-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. 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 R-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 1, and that the result is a non-trivial assisted model.

Several variations on this R-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 an R-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 R-model.

2.4.2. Refinements of R-models

It should be clear from the preceding examples that we are not classifying reproducers *per se*; rather, we are classifying *R-model s* of reproducers, and we allow for the possibility that a reproductive process may be modelled in many different ways using as many different R-models. 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; in other words, 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 R-models that captures the idea that one R-model provides a more concrete view of the same process modelled by another R-model.

Definition 5. For R-models $M = (S_M, A_M, \mapsto_M, Ent_M, r_M, \varepsilon_M, p_M, Aff_M)$ and $N = (S_N, A_N, \mapsto_N, Ent_N, r_N, \varepsilon_N, p_N, Aff_N)$, a *refinement* $M \Rrightarrow N$ is a triple (R, g, h), where $R \subseteq S_M \times S_N$ is a relation, and $g : A_M \to A_N$, and $h : Ent_M \to Ent_N$ are functions such that

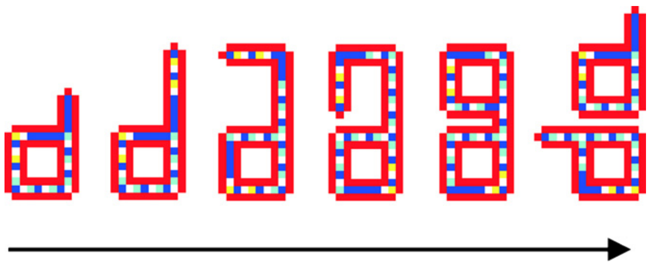


Fig. 1. Reproduction of Langton's loop. The arrow represents time.

1. $s \overset{a}{\mapsto}_M s'$ and $s\,R\,t$ implies there is $t' \in S_N$ with $t \overset{g(a)}{\mapsto}_N t'$ and $s'\,R\,t'$,
2. $s_0\,R\,t_0$ where s_0 and t_0 are the starting states in p_M and p_N respectively,
3. $e\ \varepsilon_M\ s$ implies $h(e)\ \varepsilon_N\ t$ whenever $s\,R\,t$,
4. $h(r_M) = r_N$, and
5. $h(Aff_M(a)) \subseteq Aff_N(g(a))$ 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 $M \Rrightarrow N$ 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 occur in N, although N may provide more states, actions, and entities than figure in M. The following sections give some examples of refinements.

2.4.3. *Comparing abstraction formally using refinements*

Langton's loops reproduce on a two-dimensional cellular automaton grid. A loop consists of an outer "sheath" which contains the self-description: a series of symbols encoded in the states of the sheathed cells. The self-description causes an "arm" to be extended from one corner of the loop, which then turns perpendicularly, before repeating the process a further three times until a child loop is constructed after 151 time steps (Langton, 1984), as in Fig. 1.

One way of modelling the reproductive process of the loop would be to take explicitly the states of the cellular automaton grid, e.g.:

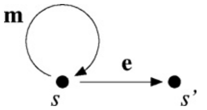
$$s_1 \overset{a_1}{\mapsto} s_2 \overset{a_2}{\mapsto} \ldots \overset{a_{151}}{\mapsto} s_{152}, \tag{2}$$

We will see that this model can be refined in two different ways.

Example 3. Let M be an R-model in which S_M, A_M and $\mapsto_M$ and the path p_N are defined by (2); i.e., this is the "schematic" transition system consisting of just that path. Let the set of entities $Ent_M = \{L, G\}$, where L represents Langton's loop (and is also the reproducer in this model), and G represents the cellular-automata grid. In this model, we consider this grid as affording reproductive actions to L, i.e., $Aff_M(a) = \{L, G\}$ for all actions a. We also specify that $L\ \varepsilon_M\ s$ and $G\ \varepsilon_M\ s$ for all $s \in S_M$, so that this model meets the requirements of Definition 1.

Now let N be the model that contains the entire state-space of the cellular automaton grid, so that S_N consists of all possible configurations of the grid, with a transition from one state to another iff the first state evolves in one step to the second. Similarly to Example 2, this R-model is therefore based on the operational semantics of the cellular-automaton grid. We set $Ent_N = Ent_M$, and for simplicity, we let L and G be present in all states (though it would be possible to restrict L's presence to only those states where a region of the grid contains one of the configurations of a loop). Clearly, N "contains" M, and we have a refinement $M \Rrightarrow N$, where the relation between states is the inclusion $S_M \subseteq S_N$, and with the identity functions on transitions and entities.

A rather more surprising refinement is given by another "schematic" transition system that picks out the **m**iddle and **e**nd of the act of reproduction, e.g.,



(3)

In the following example we show that this is also a refinement of M.

Example 4. Let P be a model in which S_P, A_P and $\mapsto_P$ and the path p_P are defined by (3), i.e., this is the transition system with only two states s and s', and transitions $s \overset{m}{\mapsto}_P s$ and $s \overset{e}{\mapsto}_P s'$. Let the set of entities $Ent_P = \{L\}$, with L present in both states, and let both actions be afforded by the loop, L.

We now define a refinement $(R, g, h) : M \Rrightarrow P$. Let R be defined by: $s_i\,R\,s$ for all i with $1 \le i < 152$ and $s_{152}\,R\,s'$. Let $g : A_M \to A_P$ be defined as follows:

$$g(a_i) = \mathbf{m} \quad \textit{for } 1 \le i \le 150, \textit{ and}$$
$$g(a_{151}) = \mathbf{e}.$$

Finally, we set $h : Ent_M \to Ent_P$ as the function mapping both G and L to L.

It is straightforward to check that the conditions of Definition 5 hold.

It is clear that the refinement in this example is only possible because of the schematic nature of the model M, which is just a path in a larger model (cf. Example 3). Indeed, we could take the existence of a refinement to P as the definition of what it means to be "schematic".

The following example gives a more standard view of refinement, using the bacteriophage model of Example 1.

Example 5. 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):

```
C T ↦ C- T
Cell [S]- T4[R] ↦ Cell[S R]
```

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 T4 bacteriophages:

```
Cell[R S] ↦ Cell[R S R]
Cell[b- rna S] ↦ Cell[S T4[brna]]
```

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

```
Cell[S] ↦ 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:

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

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, `Cell`, which is present in a state iff that state has a subterm of the form `Cell[...]`, and bacteriophages, present in a given state iff the RNA, `b-rna` occurs as a subterm. Furthermore, we can take the condition in Definition 1 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).

To see that this model refines the bacteriophage model of Example 1, define the relation *R* so that s_1 is related to all terms that contain a cell and a bacteriophage (so that the attach action is possible); s_2 to all terms where a bacteriophage is attached to a cell (so that the inject action is possible); and so forth. It is straightforward to see that this gives a refinement of the schematic model of Example 1.

2.4.4. Comparing assistance formally using refinements

It is also possible to compare the assistance of two *R*-models of the same reproductive act using refinements. In the following example we show how two different *R*-models of the same reproductive process can be related using a refinement.

Example 6. Let us define N_{cv} as an *R*-model based on the copier computer virus *R*-model M_{cv} from Example 2:

- $S_{N_{cv}}, A_{N_{cv}}, \mapsto_{N_{cv}}, r_{N_{cv}} (= cv)$ and $p_{N_{cv}}$ are identical to those in M_{cv};
- $Ent_{N_{cv}} = \{cv, sra + cp\}$;
- $Aff_{N_{cv}}(subst) = Aff_{N_{cv}}(cp) = \{sra + cp\}$; and
- $sra + cp \;\; \varepsilon_{M'_{cv}} \;\; s$ iff $sra \;\; \varepsilon_{M_{cv}} \;\; s$ or $cp \;\; \varepsilon_{M_{cv}} \;\; s$.

A refinement $(R, g, h) : M_{cv} \Rrightarrow N_{cv}$ can be constructed as follows. Let *R* and *g* be the identities $1_{S_{M_{cv}}} : S_{M_{cv}} \rightarrow S_{M_{cv}}$ and $1_{A_{M_{cv}}} : A_{M_{cv}} \rightarrow A_{M_{cv}}$, respectively, and let *h* be the function mapping *cv* to *cv* and both *sra* and *cp* to *sra* + *cp*. Note that in this case, both M_{cv} and N_{cv} are assisted *R*-models.

Even though N_{cv} and M_{cv} are both assisted *R*-models, the way in which assistance is defined differs in the two models. In M_{cv}, there are two assisting entities, *sra* and *cp*, whereas in N_{cv} there is only one assisting entity *sra* + *cp*. (The entity *sra* + *cp* is notionally the result of "amalgamating" the entities *sra* and *cp*.) In this way *R*-models with differing kinds of assistance can be related using refinements. We shall see in the next section how unassisted *R*-models can be related to assisted *R*-models using refinements.

3. Formal results: exploring the *R*-model "space"

In this section we give some formal results on *R*-models. In particular we show that for every assisted *R*-model, there is a corresponding *R*-model describing the same reproductive act that is classified as unassisted, and vice versa:

1. **The unassisted reproduction theorem**: every *R*-model can be refined by an unassisted *R*-model; and
2 **The assisted reproduction theorem**: every non-trivial *R*-model refines an assisted *R*-model.

These theorems apply to every *R*-model regardless of whether it is unassisted or assisted.

If we consider *R*-models to be a reasonable model of assistance in reproduction, then we can take these theorems as proof that every view (assisted or unassisted) of a reproductive act can be related to another view with the opposite classification. The significance of this will be discussed in more detail in Section 4.

3.1. The unassisted reproduction theorem

The idea of the proof is that, given an *R*-model *M*, we amalgamate the ecology of the reproducer into one entity, giving an *R*-model that refines *M*. In Definition 6 we construct this *R*-model $M^{\#}$, and in Proposition 1 we show that $M^{\#}$ is, indeed, an *R*-model. We then prove in Proposition 2 that $M^{\#}$ is always classified as unassisted. Finally, we show that there is always a refinement $M \Rrightarrow M^{\#}$, which completes the proof.

Definition 6. Given an *R*-model $M = (S, A, \mapsto, Ent, r, \varepsilon, p, Aff)$, we define

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

where

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

Proposition 1. *For any R-model M, $M^{\#}$ is also an R-model.*

Proof. By Definition 1, we require that for all $e \in Aff^{\#}(a)$, and for all states *s*, if action *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 6(3), $e = h(e_0)$ for

$$\begin{tikzcd}
M \arrow[r, Rightarrow, "\gamma"] \arrow[dr, Rightarrow, "\alpha"'] & M^{\#} \arrow[d, Rightarrow, "\beta"] \\
& N
\end{tikzcd}$$

Fig. 2. Refinement arrows between M, $M^{\#}$ and N.

some $e_0 \in Ent$, and because M is an R-model, it follows that $e_0\ \varepsilon\ s$. If $e_0 \in \mathrm{E}(M)$, then $e = h(e_0) = r$ and $e\ \varepsilon^{\#}\ s$ by Definition 6(2); if $e_0 \notin \mathrm{E}(M)$, then $e = h(e_0) = e_0\ \varepsilon\ s$ and so $e\ \varepsilon^{\#}\ s$ as desired. □

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

Proposition 2. *For any R-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 $\mathrm{E}(M)$. Therefore, for any a_i in p, if $e \in Aff^{\#}(a_i)$, then $e = h(c)$ for some $c \in \mathrm{E}(M)$ and so $e = r$. □

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

Proposition 3. *For all R-models M, there is a refinement $M \Rrightarrow M^{\#}$.*

Proof. The refinement $M \Rrightarrow M^{\#}$ consists of the triple $(1_S, 1_A, h)$, where 1_S and 1_A are identities on states and actions, and $h : Ent \to Ent^{\#}$ is as defined in Definition 6. Clearly, transitions are preserved (Definition 5(1)), and preservation of occurrences of entities (Definition 5(3)) follows immediately from Definition 6(2), and we need show only $h(Aff(a)) \subseteq Aff^{\#}(a)$ for all actions a, but this is immediate from Definition 6(3). □

This gives us our main result for this section:

Theorem 1 (Unassisted reproduction theorem). *For every R-model, M, there is an unassisted R-model, $M^{\#}$, such that $M \Rrightarrow M^{\#}$.*

In other words, for any R-model, be it assisted or not, there is another R-model which captures the same reproductive process but with modified entities, and which is classified as unassisted.

3.1.1. Further results relating to the unassisted reproduction theorem

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

Proposition 4. *If M is unassisted, then $M = M^{\#}$.*

This is proved by inspection of Definition 6, noting that if M is unassisted, then $E(M) \subseteq \{r\}$. 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. This is a common idea in mathematics and is known in category theory as a *universal property* (Lane, 1971). Consider the situation shown in Fig. 2, where γ is the refinement of Proposition 3. If there is some other unassisted R-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—to be precise, there is a unique refinement β such that the composition of γ and β is equal to α. The existence of a refinement β can be thought of as saying that the changes made to M to construct $M^{\#}$ are minimal changes, and that N must also incorporate those changes. The fact that β is unique intuitively says that there is effectively only one way of making those changes to get an unassisted refinement of M. This property is stated formally in the following

Proposition 5. *For all R-models M, $M^{\#}$ is the least unassisted refinement of M; i.e., if $\alpha = (\alpha_R, \alpha_g, \alpha_h)$ is a refinement of M by an unassisted reproduction model $N = (S_N, A_N, \mapsto_N, Ent_N, r_N, \varepsilon_N, p_N, Aff_N)$, then there is a unique refinement $\beta = (\beta_R, \beta_g, \beta_h) : M^{\#} \Rrightarrow N$ such that α is the composition of γ and β, where $\gamma = (1_S, 1_A, h)$ is the refinement $M \Rrightarrow M^{\#}$ of Proposition 3.*

Proof. Since γ is the identity on states and actions, we clearly require $\beta_R = \alpha_R$ 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., $e \in Ent \setminus E(M)$), we set $\beta_h(e) = \alpha_h(e)$. This clearly satisfies the requirement that β be the unique refinement whose composition with γ is equal to α, and we need only show that the refinement β does indeed exist; i.e., we need to show that $\alpha_h(e) = r_N$ for all $e \in E(M)$. If $e \in E(M)$, 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. □

3.2. The assisted reproduction theorem

From Theorem 1 we know that all R-models can be refined by an unassisted R-model, and therefore all R-models can be viewed as unassisted. In this subsection we show that the converse is also possible for non-trivial R-models: i.e., every non-trivial R-model is a refinement of an assisted R-model.

The idea of the proof is, given an R-model M, to construct an R-model $M_{\#}$ with a new entity that affords all the actions that are afforded by the reproducer in M. We define the construction of $M_{\#}$ for an arbitrary R-model M in Definition 7. In Proposition 6 we show that $M_{\#}$ is an R-model and that it is refined by M. Finally, we demonstrate that $M_{\#}$ is an assisted R-model whenever M is non-trivial, which completes the proof of Theorem 2.

Definition 7. Given an R-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 $r \in Aff(a)$;
- for all states s, $G\ \varepsilon_{\#}\ s$ iff $r\ \varepsilon\ s$;
- $e\ \varepsilon_{\#}\ s$ iff $e\ \varepsilon\ s$ for all entities $e \neq G$.

Proposition 6. *For all R-models M, $M_{\#}$ is an R-model and there is a refinement $M_{\#} \Rrightarrow M$.*

Proof. The entity G has adopted all of the actions afforded by r in M, and is present in all of the states in which r was present in M, and so it is clear from the construction that $M_{\#}$ is an R-model. The refinement $M_{\#} \Rrightarrow M$ 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 5 hold. Conditions (1) and (2) hold trivially, because S, A and $\mapsto$ are identical in M and $M_{\#}$. Condition (3) holds by construction of $M_{\#}$, as does condition (4). The final condition, that $h(Aff_{\#}(a)) \subseteq Aff(a)$, holds because $Aff_{\#}$ replaces r by G and h maps G to r. □

The assisted reproduction theorem follows from this, with one provision: that the original R-model M is *non-trivial* (cf. Definition 1), as this entails that $M_{\#}$ is an assisted R-model.

Theorem 2 (Assisted reproduction theorem). *For every non-trivial R-model M, there exists an assisted R-model $M_{\#}$ such that $M_{\#} \Rrightarrow M$.*

Proof. This follows directly from Proposition 6, 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 R-model. □

Please cite this article in press as: Webster, M., Malcolm, G., *R*-Models: A mathematical framework for capturing notions of abstraction and assistance in reproductive systems. BioSystems (2012), http://dx.doi.org/10.1016/j.biosystems.2012.07.003

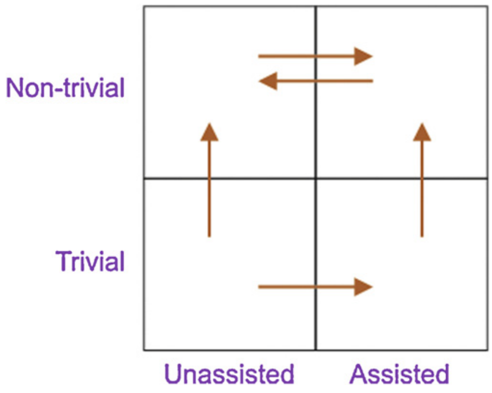


Fig. 3. Allowed refinements between classes of R-models. Arrows indicate that refinements exist between R-models of those classes (where the refinement arrow $\Rrightarrow$ is in the same direction as the arrow shown), and the absence of an arrow indicates that no such refinements exist.

3.3. Allowed refinements between classes of R-models

The definitions of assistance and triviality are independent dichotomies, which means that we can divide the space of R-models into four disjoint parts, depending on the assistance and triviality of an R-model, cf. Fig. 3. It is interesting to note that refinement between these four parts is limited in certain directions, which implies a structured space of R-models. Firstly, there are no refinements from non-trivial R-models to trivial R-models.

Proposition 7. *If M and N are R-models, and there is a refinement $M \Rrightarrow N$, then M being non-trivial implies that N is non-trivial.*

Proof. If M is non-trivial then we know that $r_M \in E(M)$. By Definition 5, we know that $h(r_M) = r_N$ and $h(Aff_M(a)) \subseteq Aff_N(g(a))$. Therefore $r_N \in E(N)$ and N is non-trivial, as desired. □

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

Proposition 8. *For all trivial, unassisted R-models, N, there is no refinement $M \Rrightarrow N$, where M is trivial and assisted.*

Proof. Proof is by contradiction. Suppose that a refinement $M \Rrightarrow N$ exists. Since M is assisted, then by Definition 3 there must be some entity $x \in Ent_M$, different from the reproducer r, such that $x \in Aff_M(b)$ for some action b in the path. By Definition 5 we know that $h(Aff_M(b)) \subseteq Aff_N(g(b))$ and therefore $h(x) \in Aff_N(g(a))$. However, $Aff_N(b') = \emptyset$ 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. □

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 R-models is summarised in Fig. 3.

3.4. Examples of applying the theoretical results

In order to illustrate the Unassisted Reproduction Theorem, we give an example of the R-model $M^{\#}_{cv}$ that is a refinement of the copier computer virus R-model M_{cv} (cf. Examples 2 and 6). We recall that M_{cv} was an assisted reproduction model, but applying the unassisted reproduction theorem gives us an unassisted R-model of the same copier computer virus.

Example 7. Let $M^{\#}_{cv}$ be constructed from the R-model M_{cv} of Example 2, as in Definition 6. This gives us:

- $S_{M^{\#}_{cv}}$,$A_{M^{\#}_{cv}}$, $\mapsto_{M^{\#}_{cv}}$, $r_{M^{\#}_{cv}}(= cv)$ and $p_{M^{\#}_{cv}}$ are identical to those in M_{cv};
- $Ent_{M^{\#}_{cv}} = \{cv\}$
- $cv \ \varepsilon_{M^{\#}_{cv}} \ s$ for all $s \in S$
- $Aff_{M^{\#}_{cv}}(a) = \{cv\}$ for all actions a.

The refinement $M_c v \Rrightarrow M^{\#}_{cv}$ consists of the identity functions on states and actions, and the function that maps all entities to cv.

We can now use the R-model $M^{\#}_{cv}$, classified as unassisted, to demonstrate the assisted reproduction theorem in action. The result is another assisted R-model of the copier computer virus, $(M^{\#}_{cv})_{\#}$:

Example 8. Let the R-model, $(M^{\#}_{cv})_{\#}$, be constructed from the unassisted R-model $M^{\#}_{cv}$ following Definition 7. 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 alone affords all the actions of the copier virus's reproductive cycle, making $(M^{\#}_{cv})_{\#}$ an assisted R-model.

These examples show that our approach does not say that assisted and unassisted models are the same thing: the constructions of Definitions 6 and 7 are not bijections, as M_{cv} and $M_{\#}$ are different. What our approach *does* say is that a reproductive process may be viewed in qualitatively different ways. Moreover, the notion of refinement serves to rank these qualitatively different approaches to modelling the same reproductive process.

4. Conclusion

In this paper we have introduced R-models as a method for modelling assistance and abstraction in reproductive systems, with examples including bacteriophages, computer viruses and Langton's loops. R-Models are based on transition systems, and while this does allow the modelling of continuous processes, the real strength of our approach lies in individual-based models (Giavitto et al., 2004; Fisher et al., 2000). As our examples of computer viruses and artificial-life systems suggest, R-models can be built on top of operational semantics for programming languages; indeed, Examples 2 and 5 use the notation of Maude, an executable language which has been used to describe operational semantics for a large number of programming systems (Meseguer and Roşu, 2007).

This formal approach also allows us to address the issue of modelling systems at different levels of abstraction. The notion of refinement exactly captures the situation where one model concretely describes the same process described more abstractly in another model. This in turn allows us to address the issue of assistance in reproductive systems. Our approach to assistance is based on the notion of affordance of actions in transition systems, in analogy with Gibson's theory of affordances (Gibson, 1977). Our main results construct two refinements $M_{\#} \Rrightarrow M \Rrightarrow M^{\#}$ for any R-model M, where $M_{\#}$ is assisted (if M is non-trivial), and $M^{\#}$ is unassisted. Because of the refinement relationship, all three R-models are models of the same reproductive system. The assisted model, $M_{\#}$, is obtained by adding an entity which affords all the actions afforded by the reproducer in M. Notionally, this new entity could be thought of as 'the laws of physics', or of the cellular-automata grid in the case of Langton's loops: some very basic entity that enables the actions performed by the reproducer in the model M. On the other hand, the unassisted R-model $M^{\#}$ is obtained by amalgamating all the entities in the ecology of the model M, i.e., all the entities that afford at least one of the actions in the reproductive path in M. Notionally, this takes the entire ecology of the model M as one super-organism that affords all the actions necessary for reproductions, and allows one

Please cite this article in press as: Webster, M., Malcolm, G., R-Models: A mathematical framework for capturing notions of abstraction and assistance in reproductive systems. BioSystems (2012), http://dx.doi.org/10.1016/j.biosystems.2012.07.003

way out of the philosophical quandary as to whether the genome uses the phenotype, or vice versa. These notional interpretations of the constructions may be debatable, but the neatness of the formal results shows that affordance and assistance make the space of reproductive systems quite highly structured.

We emphasise that our approach does not classify reproductive processes; rather, we classify *models* of such processes. Our approach is therefore only formalistic insofar as we assume that models of such processes already exist; given these models, our results apply, regardless of the quality of the model. It may be that modelling a process in one particular way throws "half of the problem out of the window, and it may be the more important half" (von Neumann, 1966, p. 77), but our notion of refinement allows for modelling processes in different ways, and at different levels of abstraction. It may seem that this 'pluralistic' approach to modelling makes the classification as assisted or unassisted somewhat arbitrary, but the important point is that the refinements $M_{\#} \Rrightarrow M \Rrightarrow M^{\#}$ have a *direction*, from unassisted to assisted as 'most refined'. Moreover, the notion of ecology used in the construction of the unassisted model $M^{\#}$ seems to have useful counterparts in many models. For example, in Kauffman's autocatalytic sets (Kauffman, 1986), the ecology would consist of the entire autocatalytic set. An analogue of this in the area of computer viruses is given by Filiol's *k-ary viruses* (Filiol, 2007), which consist of k programs $p_1, \ldots, p_k$, where p_i makes a copy of p_{i+1} (mod k). Here again, the ecology of each reproducer p_i consists of all k programs. In both cases, our notion of refinement allows the construction of an unassisted R-model that captures the mutual dependencies of these entities. However, it seems likely that our approach is best suited to scenarios where formal models really do capture the intended behaviour of the reproductive process being modelled; we briefly sketch some cases in the sections below on related and future work.

4.1. Related work

Classification of reproduction based on the reliance of a reproducer on the environment can also be found in the works of Freitas and Merkle (2004, chapter 5), Taylor (1999, 1999) and Luksha (2003). Freitas and Merkle give categories for the location of replication information (i.e., self-description) and replicator parasiticity (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 is 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 classification to a purely formal exercise. Of course, the process of encoding a reproduction system within an R-model remains subjective; however, with the unassisted and assisted reproduction theorems we have shown that unassisted and assisted R-models are related in refinements, and therefore any bias is mitigated by the fact that any unassisted R-model has a corresponding R-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 (2000) and Rosen (1991, 1999), who given formalisation of the act of reproduction based on mathematical functions. Chaitin (1979) works towards a formal definition of life based on algorithmic information theory. The works of von Neumann, Codd, Langton and others (Sipper, 1998) 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 and Misra (2007) have given an interesting approach to recognising evolving artificial life forms based on the recurrence of similar patterns due to evolutionary reproductive behaviour.

Our related work (Webster and Malcolm, 2008) gives an account how entities can be combined formally to produce more complex entities; this is an example of how the "amalgamation" of entities, as in the Unassisted Reproduction Theorem might work in practice. In particular we focus on Langton's loop, in which the reproducing loop is modelled as a composite entity built from its constituent "data-paths".

4.1.1. Löfgren's assisted reproduction

A version of this "amalgamation" of entities was given by Löfgren (1968), who states that there are two different forms of reproduction: symbiotic reproduction, in which a set of entities reproduce each other; and atomic reproduction, in which a single entity reproduces itself. We can see symbiotic reproduction as a kind of assisted reproduction, as each of the entities is assisted by one or more entities in the set. Furthermore, atomic reproduction is unassisted reproduction, as the reproducer is not assisted by any other entities. Interestingly, Löfgren notes that "atomic self-reproduction shall result from the symbiotic self-reproduction when all the distinct entities of the symbiotic case coalesce." This makes intuitive sense. If we take the assisted view of the reproduction of von Neumann's reproducing automaton, in which there is another entity (the cellular automaton) which assists in the act of reproduction, then (by following Löfgren's method) we can form an unassisted view of von Neumann's reproducing automaton if we coalesce the automaton with the cellular automaton, resulting in a single entity which reproduces in an unassisted way.

4.1.2. Rosen's paradox

As we have seen, it is possible to classify a reproductive act as assisted or unassisted, depending on the way in which the reproductive act is specified as an R-model. However, there is a logical paradox in the idea of assisted reproduction that was first described by Rosen (1959) (see also Cárdenas et al., 2010).

Rosen's paradox is based on a functional view of reproduction. We assume we can characterise a reproducer r as a mathematical function, $r : A \rightarrow B$. The reproducer, r, takes input from its domain, A, and produces outputs in its range, B. If r is capable of reproducing itself entirely, that is, it contains all that is necessary to make a copy of itself, then it is logical that $r(x) = r$ (for some input x). Therefore r is in the range B. However, in order to define a mathematical function we must first know the contents of its domain and range, A and B respectively. However, as B is a well-founded set, we cannot define it until we have first defined r. Therefore we have a paradox: to define r we must first define B, but to define B we must first define r. Therefore r cannot be defined.

Therefore, if we allow a functional view of reproduction, which would seem reasonable, then we have to conclude that unassisted reproduction is paradoxical, and therefore unacceptable. In contrast, there is no paradox associated with assisted reproduction, i.e., when there is some "assisting entity" in the environment of the reproducer that assists in the act of reproduction. Rosen showed that his paradox was absent in this situation by defining an assisting entity, a function E, which assists another entity ab which reproduces. We can characterise this assisted reproduction as follows. Let

$$E : A \times B \times (A \times B) \to (A \times B) \times (A \times B)$$

which takes a construct ab and components a and b, and produces a copy of ab, i.e., $E(a, b, ab) = (ab, ab)$. In other words, ab is the reproducer, which is assisted in reproduction by the function E.

Rosen's paradox gives us a new insight into unassisted and assisted versions of a reproductive act, and lets us distinguish between them. Perhaps, if unassisted reproduction is paradoxical, but assisted reproduction is not, should never consider unassisted reproduction again. We might think of all reproduction as assisted, instead. However, there are many cases in which unassisted reproduction appears to be the natural way of framing things, e.g., von Neumann's self-reproducing automaton or the case of a bacterium which takes in materials from its environment in order to produce a copy of itself. In both of these cases we can avoid Rosen's paradox by including in our conceptual model the assisting environments we had not previously considered (the cellular automaton and the environmental materials, for example); by doing this reproduction is no longer unassisted, and therefore there is no logical contradiction.

4.2. Future work

We conclude by sketching out some areas for future research.

4.2.1. Metrics for reliance on external agency

Above, we described how R-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 R-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.

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 depend less on host functions than those with small genomes' (Granoff and Webster, 1999). 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.

4.2.2. 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 (Symantec Press Centre, 2000). 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 (Baeten, 2005). In real-life ecologies, reproducers are capable of interesting 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.

4.2.3. 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 (Filiol, 2005). As discussed in Section 2.4, we can classify reproduction models as assisted or unassisted, depending on how we model the entities and affordances in those models. In (Webster, 2008) 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 R-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,

Please cite this article in press as: Webster, M., Malcolm, G., *R*-Models: A mathematical framework for capturing notions of abstraction and assistance in reproductive systems. BioSystems (2012), http://dx.doi.org/10.1016/j.biosystems.2012.07.003

because this cannot be detected at run-time, and should therefore be high priority.

References

Baeten, J., 2005. A brief history of process algebra. Theor. Comput. Sci. 335, 131–146.

Chaitin, G.J., 1979. Toward a mathematical definition of "life". In: Levine, R.D., Tribus, M. (Eds.), The Maximum Entropy Formalism. MIT Press, Cambridge, Mass., pp. 477–498.

Cohen, F., 1987. Computer viruses—theory and experiments. Comput. Secur. 6 (1), 22–35.

Cárdenas, M.L., Letelier, J.-C., Gutierrez, C., Cornish-Bowden, A., Soto-Andrade, J., 2010. Closure to efficient causation, computability and artificial life. J. Theor. Biol. 263 (1), 79–92.

Dawkins, R., 1990. The Selfish Gene. Oxford University Press, USA, first published 1976, ISBN:0192860925.

Filiol, E., 2005. Computer Viruses: From Theory to Applications. Springer, Paris, France, ISBN:2287239391.

Filiol, E., 2007. Formalisation and implementation aspects of *k*-ary (malicious) codes. J. Comput. Virol. 3 (2), 75–86.

Fisher, M.J., Malcolm, G., Paton, R.C., 2000. Spatio-logical processes in intracellular signalling. BioSystems 55, 83–92.

Freitas Jr., R.A., Merkle, R.C., 2004. Kinematic Self-Replicating Machines. Landes Bioscience, Georgetown, TX, ISBN:1570596905, http://www.molecularassembler.com/KSRM.htm (accessed 26.03.08).

Gardner, M., 1970. Mathematical games: the fantastic combinations of John Conway's new solitaire game 'life'. Sci. Am. 223, 120–123.

Giavitto, J.-L., Malcolm, G., Michel, O., 2004. Rewriting systems and the modelling of biological systems. Comp. Funct. Genomics 5, 95–99.

Gibson, J.J., 1977. The theory of affordances. In: Perceiving, Acting and Knowing: Toward an Ecological Psychology, pp. 67–82.

Gibson, J.J., 1979. The Ecological Approach to Visual Perception. Houghton Mifflin, Boston, ISBN:0395270499.

Granoff, A., Webster, R.G. (Eds.), 1999. Encyclopedia of Virology, vol. 3. Academic Press, pp. 1414–1415 (Entry on "Virus Multiplication Cycle").

Henz, M., Misra, J.,2007. Towards a framework for observing artificial life forms. In: Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife 2007). IEEE Press, pp. 23–30.

Hofstadter, D.R., 2000. Gödel, Escher, Bach: An Eternal Golden Braid. Penguin, p. 499 (Chapter 16).

Kauffman, S., 1986. Autocatalytic sets of proteins. J. Theor. Biol. 119, 1–24.

Koonin, E.V., 2012. Defining life: an exercise in semantics or a route to biological insights? J. Biomol. Struct. Dynam. 9, 603–605.

Lane, S.M., 1971. Categories for the Working Mathematician. Vol. 5 of Graduate Texts in Mathematics. Springer Verlag, New York.

Langton, C.G., 1984. Self-reproduction in cellular automata. Phys. D: Nonlinear Phenomenon 10, 135–144.

Löfgren, L., 1968. An axiomatic explanation of complete self-reproduction. Bull. Math. Biol. 30 (3), 415–425.

Luksha, P.O., 2003. The firm as a self-reproducing system. In: Proceedings of 47th International System Science Society Conference.

McMullin, B., 2000. John von Neumann and the evolutionary growth of complexity: looking backwards, looking forwards. . .. Artif. Life 6, 347–361.

Meseguer, J., Roşu, G., 2007. The rewriting logic semantics project. Theor. Comput. Sci. 373 (3), 213–237.

Plotkin, G.D., 2004. A structural approach to operational semantics. J. Logic Algebr. Program. 60–61, 17–139, http://dx.doi.org/10.1016/j.jlap.2004.03.002.

Rosen, R., 1959. On a logical paradox implicit in the notion of a self-reproducing automaton. Bull. Math. Biophys. 21, 387–394.

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

Rosen, R., 1999. Essays on Life Itself. Columbia University Press, New York, ISBN: 978-0231105118.

Schrödinger, E., 1944. What is Life? Cambridge University Press, Cambridge, UK.

Sipper, M., 1998. Fifty years of research on self-replication: an overview. Artif. Life 4, 237–257.

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 26.03.08).

Taylor, T., 1999. On self-reproduction and evolvability. In: Floreano, D., Nicoud, J.-D., Mondada, F. (Eds.), Proceedings of the Fifth European Conference on Artificial Life (ECAL99). Springer-Verlag.

Taylor, T.J., 1999. From artificial evolution to artificial life. Ph.D. Thesis, University of Edinburgh, http://www.tim-taylor.com/papers/thesis/ (accessed 26.03.08).

von Neumann, J., 1966. In: Burks, W. (Ed.), Theory of Self-Reproducing Automata. University of Illinois Press, Urbana, Illinois.

Weber, B., 2006. Life. In: Zalta, E.N. (Ed.), The Stanford Encyclopedia of Philosophy. The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University, CA, USA, http://plato.stanford.edu/archives/spr2006/entries/life/ (accessed 26.03.08).

Webster, M., Malcolm, G., 2008. Hierarchical components and entity-based modelling in artificial life. In: Bullock, S., Noble, J., Watson, R., Bedau, M.A. (Eds.), Artificial Life. XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems. MIT Press, Cambridge, MA, pp. 678–685.

Webster, M.P., 2008. Formal models of reproduction: from computer viruses to artificial life. Ph.D. Thesis, University of Liverpool.