CONCEPTS OF COOPERATION IN ARTIFICIAL LIFE

Harold W. Thimbleby Ian H. Witten David J. Pullinger

October 1, 1998

Abstract

We have built some simple, but useful, cooperative Artificial Life agents. Based on this experience and by contrasting our work with computer viruses, we argue that Artificial Life (the simulation of life including evolution) can only remain *reliably* and indefinitely cooperative if it adheres to explicitly-specified social conventions. Breaking or neglecting these conventions results in systems that are worse than useless; in fact, malicious with respect to human social values.

Running title: Cooperation in Artificial Life

Keywords: Artificial life, agents, computer viruses, liveware, cooperative behaviour

Manuscript received ... **SMC 094–04–0461**

Concepts of cooperation in artificial life

Harold W. Thimbleby: *Address for correspondence*
School of Computing Science,
Middlesex University,
Bounds Green Road,
LONDON,
N11 2NQ. UK

Ian H. Witten:
Department of Computer Science,
Waikato University,
Hamilton,
Private Bag 3105,
New Zealand

David J. Pullinger:
Institute of Physics,
Techno House,
Redcliffe Way,
BRISTOL,
BS1 6NX. UK

Acknowledgements. This work arose from a Visiting Fellowship for Ian H. Witten under UK SERC Grant No. GR/F/60601, 1989. The authors also wish to acknowledge the contribution of Stephen Marsh, a SERC/CASE student supported by SERC and Canon Research Europe. Brian Gaines, Calgary University, Canada, and several anonymous referees have made very helpful comments for which the authors are grateful.

1 Introduction

Altruism and cooperation, even when feasible, are achieved only with effort and foresight. To support this claim, we contrast two closely related systems: computer viruses and liveware. Both, we argue, are being developed to satisfy the basic requirements for life. Both have achieved autonomy in that they are no longer under the control of their human originators. Computer viruses are designed to be malicious to their computer hosts and so to computer users, whereas liveware is intentionally cooperative. They differ in the social laws that they embody, explicitly and implicitly.

Artificial Life is a recent area of computer application [19]. We shall criticize its stance of assuming cooperation by emergence. Our experience with viruses and liveware suggests that Artificial Life requires careful constraint — laws for cooperation — to operate appropriately. Such laws are absent from viruses.

1.1 Paradigms and perspectives on life

Life is viewed differently from different disciplines, and within those disciplines from various points of view. Schrödinger [31] published an interdisciplinary examination of life; more recently computers, and the possibilities of simulation, have encouraged new perspectives [28]. Attempts to discover extraterrestrial life also urge on us wider views of what life is [15].

The within-discipline views of Biology, Computer Science and Artificial Life, may be briefly summarised as follows:

1. *Biology* views life as the distinguishing property of (living) systems that reproduce, adapt, evolve and die. Furthermore, such systems are characterised by organisation (local reduction of entropy) and robustness. Boundary cases such as anhydrobiosis indicate that life is relative to an environment; it is a disposition to reproduce and evolve when these activities are possible or appropriate to the goals or conditions of the system.

   Life may be studied abstractly ([14, 29]: to give just two examples) but, more broadly, many biologists would agree that life is a diverse phenomenon and does not allow a meaningful and general definition [10]. Many biological dictionaries do not define 'life' as such. The assumption is made that the world in which we live only provides one (recognisable) instance of life, and so discussion tends to be more concerned with classification of systems based on DNA than with axiomatic definitions.

2. *Computing Science* adopts the perspective of formal systems to analyse and simulate life processes. From this viewpoint any *effective* definition of life can be implemented and characteristic phenomena can be derived mechanically [1, 4, 32]. This formal view claims that there is no formal difference between the organic generation of life and its simulation (e.g., as numbers or symbol systems). We note, then, that the observation of characteristic phenomena (often only when suitably interpreted) does not logically imply life.

   The main drive in Computing Science concerning biological simulation is towards robotics and artificial intelligence, including neural networks. The relation of programmability and evolvability are discussed in Conrad [7]. Hofstadter [17] presents a substantial discussion of the feasibility of the computability of life.

3. *Artificial Life* concerns itself with systems beyond the normal environments of biology — in particular, 'living' systems that are not based on DNA or organic chemistry. The 'genetic algorithm' style of problem solving [13] implements an abstraction of the mechanism of biological life (based in genetics: cross over, mutation, genotypes), which, under

appropriate environmental pressures, results in what are described as evolutionary phenomena. By formulating problems as selection pressures on organisms, performance can be optimised over a series of generations. Artificial Life systems are frequently composed of many similar, simple organisms that collectively exhibit life-like behaviours, such as Conway's cellular automaton 'Life' [3].

Reviews of Artificial Life may be found in [19, 20]; a brief review raising some pertinent issues is [18].

These three different perspectives on life are not inconsistent. However the fields diverge, concentrating on different properties of life: organisation, specification, and emergence, respectively.

1.2 Definition of Artificial Life

If life can be defined abstractly, independent of its biological roots, it follows from the Church-Turing Thesis [12, 24] that life may be simulated. 'Artificial Life' is that simulation, or an approximation thereto, typically implemented in digital computers rather than in organic chemistry. Genetic algorithms may be used to solve problems of survival: the Artificial Life is then purely a representation of the behaviour of the algorithm — the 'life' of many simulated organisms.

Artificial Life need not 'look alive,' since the form has been abstracted away. If rendered in (say) animated high resolution colour it might look impressively alive, but no current definition of life requires this. Much work is, however, devoted to simulating biologically felicitous representations [30].

Langton argues that there should be no rules in an Artificial Life that dictate global behaviour [19]. Instead he requires any global rules that are exhibited to be emergent, that is, not defined explicitly. However, whenever the same law is represented in each component of an Artificial Life agent — whether explicitly or implicitly by way of its interaction with the environment — it becomes observationally indistinguishable from a global law, and its emergence (or otherwise) is immaterial to its purpose. From a formal viewpoint, the disadvantage of emergence is that conformance of an Artificial Life agent to any particular laws cannot be subject to proof or scientific assessment: one can in general only establish conformance *after* it is operating.

1.3 Philosophical issues

This paper discusses concrete issues using terms such as 'malice' and 'cooperation.' There are serious philosophical issues over the meanings of such terms, particularly when they are applied to different categories of agent. For example, an animal might be useful to a human without intending to be useful; in what sense might one say that the animal cooperates with the human? For this paper, we take a functional stance: terms are used to describe what agents do, rather than the 'reasons' that may be invoked to interpret their actions. (What reasons we do give in this paper are to be interpreted as clarifying metaphors rather than as claims of profound epistemological status.)

Furthermore we gloss over that 'intention' derives from human agency and may inherently be a biological phenomenon that man-made artefacts inherit; it may also be just a convenient way of talking about complex behaviour [9]. Do computer programs intend to bring about circumstances that their designers (being humans) straight-forwardly planned them to cause? Do programs with unintended features have intentions, then, of their own? We chose to avoid such interesting and diverting questions. Nor will we pursue moral values, though the present paper may be viewed as a contribution within the proper scope of 'artificial morality' [8].

Cooperation is a mutual process. Our emphasis on cooperation with humans for their benefit does not imply that we are ignoring the non-humans' point of view: it happens to be a simpler and a more direct English style to take an anthropocentric view. Taking a biological example, we humans cooperate with green plants. We eat them and spread their seeds. From the plants' point of view, in contrast, the benefit of cooperation is more the distribution rather than the consumption. We will see when we discuss Artificial Life that humans also spread it as a side-effect of their normal activities. One difference between cooperative and uncooperative Artificial Life at this level is whether the spread is desirable from the point of view of its human vectors. (In the case of computer viruses, humans spread viruses because they want to spread *other* information.)

We do, however, maintain that cooperation with humans is fundamentally more desirable than non-cooperation; also that it is more desirable than (what might be no more than attributed) cooperation between non-human agents. If computers or their program agents cooperate *only* amongst themselves, what impact has that on us? But if they do not cooperate with us, or they behave as if they are not cooperating, that is something that we may wish to be concerned with. In this respect with would disagree with Moravec's view of life (artificial or otherwise) for its own sake [26]: although it may be interesting to create 'children' who surpass us and outlive us, for the present paper we aim to contribute to the improvement of the human condition.

We wish to distinguish carefully between labelling Artificial Life agents as interesting in particular ways (for example that they are cooperative with humans) and claiming that *all* agents are interesting. This would be a risky generalisation and assumes a uniformity that may not be present. As motivation, though, it may be satisfactory for the exploratory stages of a new laboratory discipline. But if agents are released and are autonomous it is methodologically unsatisfactory to decide in what way they are interesting *after* their interesting, perhaps unanticipated, properties have become apparent. Specifically, if Artificial Life is to cooperate with humans, cooperation and the forms it may take has to be *planned*. (Our experience is that this is not at all easy.) Of course, if one decides that Artificial Life has (a right to?) a life of its own, perhaps at the expense of humans, then things become very much easier; contrariwise, this would be cooperation of the simplest kind with an unscientific human research programme!

2 Realisations of Artificial Life

Artificial Life systems are generally simulations within single or tightly-coupled computer systems. Conventional robots may be autonomous for brief periods of time, but the only examples that are truly free indefinitely are computer viruses and liveware. These two classes of systems are beyond the control of their creators and deal directly in information of significance to their hosts.

2.1 Computer viruses

Computer viruses were first described by Cohen [5]. Hoffman [16] is a more recent source.

Briefly, viruses attach copies of themselves to other programs. When an infected program is invoked, it seeks other files that it can change, and infects them by modifying the files to include a copy of the virus code. The virus usually resumes the original program so that the user is unaware of its intervention. A computer virus hijacks the reproductive (e.g., copying) mechanisms of its host, usually by making operating system calls. There is a terminological debate on what constitutes self-replication and how so-called viruses and worms should be distinguished: generally, a worm takes active steps to seek out a host (e.g., by using a computer network), whereas a virus is opportunistic.

Biological terms	Computational terms
Camouflage	Encryption
Mimicry	Trojan Horse
Immunity suppression	Armour

Table 1: Examples of how computer viruses employ biologically familiar mechanisms.

The purpose of self-replication is to propagate the virus and to 'keep it alive' despite its being eradicated from specific sites. Symptoms, triggered by time or conditions (time bombs, logic bombs) may occur suddenly and on a large scale, however slowly the viral infection spreads. Viruses — even so-called 'benign' viruses — consume resources, which may have unfortunate consequences (such as slower computation, consumption of disc space).

The techniques used by viruses are theoretically unlimited, and are reminiscent of biological mechanisms (Table 1) — and often a step ahead of detection techniques. Virus detection software can operate in many ways, for example by combinations of pattern matching and behaviour traps. Some viruses can be immunised against by placing characteristic, but inactive, templates of viruses in appropriate files.

Some computer viruses have cross-infected each other, typically producing sterile or very similar strains (`nVIR A` and `nVIR B` being examples). Some viruses mutate by randomly encrypting their structure in an attempt to evade detection. Genetic algorithms have yet to be explicitly deployed; they could be, though such viruses would likely be bigger and hence easier to detect.

The adversarial nature of virus writers against anti-viral workers results in a directed, goal-seeking, effectively evolutionary progress of viral technology.

2.2 Liveware: 'cooperative viruses'

Computers can share data and can do so autonomously. Liveware is a form of communication in which information self-replicates in a humanly meaningful way [34, 37, 38]. If viruses are viewed as destroying both informational data and programmatic data, liveware is designed to spread data and programmatic elements that are useful to computer owners. In order for it to do this successfully, there are laws that regulate its action.

Viruses spread destruction very quickly without necessarily having intervention from human users. If users noticed and could intervene, they would naturally want to stop viruses spreading. Suitably harnessed, computer viruses could in principle, however, spread information and programs that are useful both to the operation of the computer and to human users. One can compare this with the medical use of biological viruses to carry desirable genes into cells. The introduced genes then persist in the body as a side effect of normal cell replication. This is the idea of liveware: a controlled viral form that requires little intervention from the user.

Liveware self-reproduces, adapting to the needs of both its computer hosts and human users. The users need not be identified in advance, and their grouping need not be stable. Indeed, the user group working with a particular liveware system can split, merge and so forth without any administrative overhead. Liveware's self-reproducing properties permit the group of cooperating users to vary flexibly. Further, a liveware system can evolve to fit itself better to the social niche in which it is used, allowing cooperative work to take place in even loosely-coupled socially-organised groups. It does this by explicitly implementing laws of cooperation, which, as is expanded below, draw on concepts from mechanisms of biological reproduction.

The types of liveware system currently in trial use are programs on personal computers that allow data to be entered by users. When users want to obtain data from others and exchange their own, they pass a disc to the other persons. This then updates the data from

Computer virus		**Liveware**	**Biological life**
autonomous	$=$	autonomous	autonomous
uncontrolled	$\neq$	controlled	both
self-replicating	$=$	self-replicating	self- and sexually replicating
potentially evolving	$\approx$	evolving	evolving
'natural' selection	$\approx$	goal-seeking	natural selection
mutation/cross-over	$\approx$	'mutation'/cross-over	mutation/cross-over
un-cooperative agent	$\neq$	cooperative agent	both
malicious	$\neq$	benign	both
anonymous	$\neq$	signed owner	both
spreads destruction	$\neq$	spreads information	both
fragile	$\neq$	robust	very robust
normally ageless	$\approx$	ages	ages and dies

Table 2: Summary comparison of *typical* computer virus, *typical* liveware and *typical* biological life (see text).

each to each. Provided the group of users doing this are not fragmented into small exclusive cliques, each user's data can then be passed automatically to everyone else. The advantage of liveware is that it gives the effect of a large shared database without the administrative and organisational problems of having to maintain a master copy, the conventional solution to a large number of people sharing a pool of data. The same mechanism can be used not only with information data, but also with the way in which the program itself is run — for example, in the interface or functionality presented to the user.

Computationally it is not necessary to make a distinction between program and data. Indeed, the program part of one liveware system implemented to date has been arranged so that software upgrading was performed using exactly the same autonomous mechanism as users' information. This is not evolution in an autonomous sense, but is a very convenient mechanism for software maintenance. When this is extended to the whole group of users exchanging program alterations, some of which may conflict with each other, it indicates that both the behaviour and the appearance of liveware may evolve even though they are not under the control of any one programmer

2.3 Viruses and liveware are Artificial Life

Liveware goes beyond the self-replication of present viruses, for the information it carries splits and merges the data analogously to genetic mechanisms, and with the same teleonomy. It does so in an attempt to improve its ecological fitness to the computer host *and* humanly structured database requirements. In this sense liveware is autopoietic [23].

Table 2 compares the relevant properties of computer viruses, liveware and biological life. Since there are great varieties of viruses and life, the table necessarily compares typical or characteristic properties. For example: although almost all (computer) viruses are anonymous, some actually proclaim who wrote them and why; or, although almost all biological life forms age and die, biological viruses do not age in any meaningful sense.

In contrast to other forms of Artificial Life, computer viruses score by being literally out of control. They work autonomously beyond the knowledge and reach of their originators.

We place 'natural' selection in quotes in the virus column of Table 2 since the selection mechanism (anti-virus products) is intentionally driven by competing programmers and hardware developers.

'Mutation' is quoted in the liveware column since it is intentionally driven by explicit

modification of the genome. Liveware supports cooperation between individuals and therefore survives by *convergence* to the intentions of its users, in a selection strategy that encourages success. In contrast, life and viruses survive by *diversity*, against a selection strategy that eliminates failures.

Table 2 shows viruses as spreading destruction as opposed to information. Of course viruses spread information, namely *but trivially* the description of their own behaviour. This level of spreading information they share with any form of self-replication. Viruses do not vector information.

Computer viruses are fragile: errors in their program generally lead to failure. Some viruses have a degree of simple redundancy in them, and some may be able to survive as mere replicators without their original malicious activity. This fragility (or very limited robustness) is in contrast to both liveware and biological life. Liveware can sustain substantial corruption that will be repaired by subsequent merges. Living organisms can sustain major trauma; in some cases they can self-repair (e.g., by growing a lost limb), and in most other cases reproduction results in untraumatised offspring.

2.4 The speed and impact of Artificial Life evolution

Since Artificial Life is programmed, evolutionary strategies can be specified and implemented with fast algorithms. Systems can mutate (or be re-programmed) without the delay and overheads of reproduction. Thus, adaptation is possible without reproduction. Once initiated, the pace of artificial reproduction, and evolution, is very rapid in comparison with biological processes.

If such rapid evolution can be autonomous, then the moral direction it takes is of significant interest — Artificial Life may have a major impact on humanity [25]. Although it can be trivially altruistic, there is no reason to suppose that it will be generally, or that it will 'wish' to cooperate with its users. See [6] for a naïve discussion of the 'benefits' of computer viruses.

3 Laws for cooperation

The complexity of human society and the benefits of cooperation have led to the adoption of constraints on human behaviour. Appropriate systems of law (and, increasingly, economic systems) channel agents' options, limiting their courses of action towards cooperative (or, at worst, more predictable) behaviour. Adopting a standard of legitimacy restricts behaviour to more readily computable choices, as individual human rationality is too limited to solve many problems [22]. In short, complex problem solving requires cooperation and precommitment [11, 33], although this is clearly not sufficient! From Axelrod [2], necessary conditions for cooperation include the proper identification of: (a) individuals, that is, the identity of Artificial Life systems themselves; (b) their respective creators, or other responsible agents; and (c) the owners of the specific information carried.

Except for liveware, Artificial Life has not yet been created to communicate non-trivial data — this may explain why there is a prevalent view that cooperation can emerge both within Artificial Life forms and for human use without requiring explicit planning.

We have developed both liveware systems and computer viruses (taking appropriate precautions). A virus is trivial to implement (the program texts of certain computer viruses that are freely available) whereas liveware systems have proved difficult to implement correctly. Whereas a blind strategy of diversity is often sufficient for virus survival (some are programmed to change at random in an attempt to evade detection), liveware has to converge to both the computer hosts being used and the intentions of the social work-group it is supporting.

In our experience we have had to develop a number of laws to achieve and to guarantee convergent cooperation. These are general laws that have guided us; we do not yet know how specific some of them need to be made. Since Artificial Life is artificial, it is futile to argue whether such laws are both necessary and sufficient: the constraints on Artificial Life are just what one wishes to impose. A sufficient law would be to forbid Artificial Life having any impact whatsoever, but this is not necessary if one wants to benefit from it! A designed trade-off has to be made (and we claim that liveware is the first Artificial Life form where this trade-off has been explored in any depth). At least, our laws are a contribution that might encourage the development of more substantial laws (of possibly quite different form), or even their formulation here may encourage and enable an explicit refutation of our claims.

Although our liveware systems have pro-active implementations, this is not the only possibility; for example, laws may be enforced by constraints or programming devices such as guardians [21]. We note that the requirement for laws (properties, principles) is common with other complex system design areas, such as human computer interaction [35] — and they serve a similar purpose of constraining behaviour. Conversely, one's concept of cooperation might be very much vaguer than would be appropriate for a scheme such as liveware; in our opinion whether something might be labelled cooperative retrospectively is not sufficient (though it could be academically interesting): what if it had not turned out that way?

These are the laws we propose:

3.1 Species isolation

To permit different Artificial Life forms to coexist on the same computer hosts, they must be typed (speciated) to inhibit transfer of species-specific information from one life form to another.

Chimeras inherit from parents but are incapable of reproducing with the parent species. Isolation can be achieved either by laws embodied *in* the Artificial Life form, that is under its own discretion or by environmental factors *beyond* its control.

Thimbleby [36] shows that computer viruses can be satisfactorily limited by using encryption to mutually isolate communities of (possibly) infected agents: provided the encryption is implemented in hardware or by other means beyond any virus.

3.2 Environmental awareness

An Artificial Life may be benign in one environment but destructive in another (this applies to many current PC viruses). It should therefore be aware of both its computer host and changes to it so that it does not embark on inappropriate or undefined actions in a different or under-specified context. An extreme example, which should also be avoided, is a system that runs correctly on one brand of computer and operating system but fails catastrophically on another model.

3.3 Controllability

The system must have limitations that allow it to be controllable at several levels, while still allowing it to have autonomy. In liveware it has been helpful to distinguish between the global control imposed by the original creator of the system and the local control imposed by the owner of an instance of it. Global control delineates the range of possibilities open to the liveware in different environments and uses.

The extent of local control depends on the tasks the system is designed to support. In the case of liveware, local control may be trivial or non-trivial; in the former case it may simply

be a matter of discretionary use or not, or extended in the latter case to a new personalised delineation. (Some users may be programmers capable of making arbitrary changes.)

3.4 Sterilisability, a special case of controllability

The sterilisation of Artificial Life refers to the termination of its self-reproducibility, not to the destruction of the information it carries. A system must provide means for sterilisation, at the level of reproduction of specific information carried by it; its aggregate use of a single user's resources: its aggregate use of a group's resources; and its replication anywhere.

The first point means that a user is able to eliminate unwanted or undesirable replication of components of an Artificial Life: components such as private data. The second and third imply that a user, or group of users, should be able to detach themselves from the attentions of the Artificial Life. The final point requires the (distributed) system itself, regardless of the above, to be capable of being sterilised centrally — though this raises issues of authority in the general case.

It is assumed that if an Artificial Life is sterilised, any information it carries is eliminated by deleting it. Conversely, without sterilisation, deletion is often impossible as the system can regenerate (as readily happens with computer virus infections).

3.5 Hosting

There is a distinction between Artificial Life being *used* on a system and being *hosted* by it. In the latter case, we mean that the system responds to its host environment appropriately. It may be used under temporary conditions on the same machine, but this need not imply that it has established itself there as benign (to the machine) and useful (to the human).

A user must know whether he hosts an Artificial Life system. Unless he is aware, he cannot take initiative to control the system, particularly to eliminate it. Conversely, he may wish to restrict the information the system disseminates on his behalf. The system's policy about the private information must also be visible to the user. Privacy is a serious issue since self-reproducing facts (e.g., based on non-intrusive statistical inference) can travel beyond the original group of users. Legal notions of agency are crucial; Nelson [27] discusses royalty and copyright issues.

If a system is *permanently* invisible the user is unaware of its presence, and therefore, by definition, it is of no concern to him. (An apparent paradox is that viruses are only temporarily invisible; their delayed destructive action is only too visible.) An invisible Artificial Life would be useful if the user's resources are acting as hosts or vectors for other users.

Early versions of liveware attempted to be transparent, in direct imitation of the virus idea, to spread information without burdening users. This was emotionally unacceptable to users, and now all are discretionary.

3.6 Ownership

Following Axelrod [2], using cooperative Artificial Life with human beings requires the imposition of a human social etiquette, specifically:

- Users are responsible for their own data ("what's mine is mine");
- Users are not to change other users' data ("what's yours is yours");
- Users are known publicly by their data ("we all know who's who").

The rules apply even when data are not personal: they ensure that precisely one person is responsible for each unit of information, and that there is no possibility of a unit being updated at different times unless the last update is required by the actual owner. Of course, the resulting exchanges may lead to something well beyond the imaginings of any. The second rule ensures that users can work asynchronously — their personal activities are unaffected by changes in the total pool of users and their activities. The final rule is required because data owners may invite the same liveware to operate on data at different places and times: there must be a naming convention to avoid any later conflict with ownership.

3.7 Authentication

An Artificial Life form must be distinct to itself and to its users; this is a problem of visible identity and authentication. Thus systems should be identifiable within the environments for which they are planned and for the users. This is particularly important for a system like liveware, which might have generic programs but different data elements — in the same way that a species has a characteristic genetic makeup but distinct individuals. (Public key cryptosystems permit appropriate authentication and licensing.)

There is a distinction between mimicry and deception. A life form may mimic another to gain some of its advantages, such as safety from predators, relying in this case on deception — the inability of the predator to recognise individuals. For Artificial Life, situated in human society, mimicry is an efficient way to realise certain properties. This is not necessarily deceptive. In contrast, mimicry of data or of ownership is generally fraudulent. Authentication prohibits deception but permits mimicry.

4 Conclusions

This paper has contrasted computer viruses and liveware, as examples of Artifical Life forms that are autonomous and beyond their implementors' control. Both forms run on widely distributed computer systems. Yet they differ crucially in what they achieve for humans. Viruses are essentially trivial, and are destructive; liveware is complex, and is cooperative.

Apart from computer viruses, ideas in Artificial Life adopted from biological life have had trivial impact on humans. If only for this reason our attempt to build an alternative to destructive computer viral-forms bears closer examination. This paper suggests that without deliberate laws to the contrary, Artificial Life will be — indeed, already is — un-cooperative to both computer hosts and to the human users of those computers. Quite minor perturbations to the laws we have developed to guide our own system development result in virus-like behaviour — diversity, unconstrained growth, and destruction of data. It would seem that appropriate laws for cooperation must be planned and built in, and for a purposive, cooperative Artificial Life (such as liveware) we see no alternative.

Biological evolution is opportunistic and probabilistic, but constrained within a range of possibilities. When using the ideas of biological life to develop systems that might be of more benefit to human life, one task is to establish the delimiters and range of possibilities. Without such study, those that design Artificial Life systems hoping that cooperation will emerge have no guarantee that: (a) it will emerge; (b) that it does not go through an excessively un-cooperative phase; (c) that cooperation will be other than temporary or bluff; and (d) cooperation between artificial life-forms will be cooperation *with* human users.

References

[1] Arbib, M. A., 1967, "Self-Reproducing Automata — Some Implications for Theoretical Biology," in *Towards a Theoretical Biology*, Waddington, C. H. (ed.), pp. 204–226, Edinburgh: Edinburgh University Press.

[2] Axelrod, R., 1984, *The Evolution of Cooperation*, New York: Basic Books.

[3] Berlekamp, E. R., Conway, J. H. & Guy, R. K., 1985, *Winning Ways*, **2**, Academic Press.

[4] Chaitin, G. J., 1979, "Toward a Mathematical Definition of Life," in *The Maximum Entropy Formalism*, Levine, R. D. & Tribus, M. (eds.), Massachusetts: MIT Press, pp. 477–498.

[5] Cohen, F., 1987, "Computer Viruses: Theory and Experiments," *Computers and Security*, **6**, pp. 22–35.

[6] _____, 1991, "Friendly Contagion: Harnessing the Subtle Power of Computer Viruses," *The Sciences*, Sept/Oct, pp. 22–28.

[7] Conrad, M., 1988, "The Price of Programmability," in Herken, R. (ed.), *The Universal Turing Machine*, A Half-Century Survey, Oxford: Oxford University Press, pp. 285–307.

[8] Danielson, P., 1992, *Artificial Morality (Virtuous Robots for Virtual Games)*, London: Routledge.

[9] Dennett, D. C., 1989, *The Intentional Stance*, Cambridge, Massachusetts: The MIT Press,

[10] Eigen, M. with Winkler-Oswatitsch, R., 1992, *Steps Towards Life*, trans. P. Woolley, Oxford: Oxford University Press.

[11] Elster, J., 1979, *Ulysses and the Sirens*, Cambridge: Cambridge University Press.

[12] Galton, A., forthcoming, "The Church-Turing Thesis: Its Nature and Status," *Proc. 1990 Turing Conf.*, Oxford University Press.

[13] Goldberg, D. E., 1989, *Genetic Algorithms*, Reading, Massachusetts: Addison-Wesley.

[14] Goodwin, B. & Saunders, P., 1989, *Theoretical Biology: Epigenetic and Evolutionary Order*, Edinburgh University Press.

[15] Heidmann, J., 1992, *Life in The Universe*, trans. Leonard, I. A., McGraw-Hill.

[16] Hoffman, L. J. (ed.), 1990, *Rogue Programs: Viruses, Worms, and Trojan Horses*, New York: Van Nostrand Reinhold.

[17] Hofstadter, D. R., 1979, *Gödel, Escher, Bach*, Hassocks: Harvester Press.

[18] Hurst, L. D. & Dawkins, R., 1992, "Life in A Test Tube," *Nature*, **357**, pp. 198–199.

[19] Langton, C., 1988, "Artificial Life," In *Artificial Life*, Santa Fe Inst. Studies in the Sciences of Complexity, Langton, C. (ed.), Reading, Massachusetts: Addison-Wesley, pp. 1–47.

[20] Langton, C., Taylor C., Farmer, J. D. & Rasmussen, S. (eds.), 1991, *Artificial Life II*, Santa Fe Inst. Studies in the Sciences of Complexity, Reading, Massachusetts: Addison-Wesley.

[21] Liskov, B. & Scheifler, R., 1988, "Guardians and Actions: Linguistic Support for Robust, Distributed Programs," *ACM Trans. Programming Languages and Systems*, **5**(3), pp. 381–404.

[22] Luhmann, N., 1979, *Trust and Power*, Chichester: Wiley.

[23] Maturana, H. R. & Varela, F. J., 1972, *Autopoiesis and Cognition*, Dordrecht, Holland: D. Reidel Pub..

[24] Minsky, M., 1967, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice-Hall.

[25] Moravec, H., 1988a, "Human Culture: A Genetic Takeover Underway," In *Artificial Life*, Santa Fe Inst. Studies in the Sciences of Complexity, Langton, C. (ed.), Reading, Massachusetts: Addison-Wesley, pp. 167–199.

[26] _____, 1988b, *Mind Children*, Cambridge, Mass.: Harvard University Press.

[27] Nelson, T. H., 1990, *Literary Machines*, Sausalito, Calif.: Mindful Press.

[28] von Neumann, J., 1966, *Theory of Self-Reproducing Automata*, Urbana, Illinois: University of Illinois Press.

[29] Polyani, M., 1968, "Life's Irreducible Structure," *Science*, **160**, pp. 1308–1312.

[30] Prusinkiewicz, P. & Lindenmayer, A., 1990, *The Algorithmic Beauty of Plants*, New York: Springer-Verlag.

[31] Schrödinger, E., 1944, *What is Life?* Cambridge: Cambridge University Press.

[32] Simon, H. A., 1982, *The Sciences of the Artificial*, Massachusetts: MIT Press (2nd ed).

[33] Thimbleby, H. W., 1988, "Delaying Commitment," *IEEE Software*, **5**(3), pp. 78–86.

[34] _____, 1990a, "Liveware: A Personal Distributed CSCW," Institution of Electrical Engineers Colloquium, CSCW: Computer Supported Cooperative Work, Digest No. 1990/133: pp. 6/1–6/4.

[35] _____, 1990b, *User Interface Design*, Reading: Addison-Wesley.

[36] _____, 1994, "An Organisational Solution to Piracy and Viruses," *Journal of Systems and Software*, **25**(2), pp. 207–215.

[37] Witten, I. H. & Thimbleby, H. W., July 1990, "The Worm that Turned: A Social Use of Computer Viruses," *Personal Computer World*, pp. 202–206.

[38] Witten, I. H., Thimbleby, H. W., Coulouris, G. F. & Greenberg, S., 1991, "A New Approach to Sharing Data in Social Networks," *Int. J. Man-Machine Studies*, **34**(3), pp. 337–348.