The "Worm" Programs—Early Experience with a Distributed Intelligence

Ian H. Witten
Computer Science, Calgary University, Calgary T2N 1N4, Canada
Telephone (403) 220–6780; email ian@cpsc.UCalgary.CA

AND

Harold W. Thimbleby
Computer Science, Stirling University, Stirling FK9 4LA, Scotland
Telephone 44-786-833730; email hwt@compsci.Stirling.ac.UK

> Electronic calculators can solve problems which the man who made them cannot solve; but no government-subsidized commission of engineers and physicists could create a worm.
>
> Joseph Wood Krutch, 1949

1. Introduction

Following a mention in a recent article of our preliminary experiments on worm programs (Witten & Thimbleby, 1989), we have been deluged with enquiries about the progress of this research project. Although we had planned to wait until further concrete results were available before going to press, we feel obliged to satisfy curiosity by publishing some details of the research program now. Moreover, our lawyers have advised us that this step is advisable in order to forestall others who may plan to patent this new computational technique.[1]

The structure of this paper is as follows. First, we review contributions made by worms to three different areas of advanced technology: a vehicle for autonomous, distributed programming; a novel method of robot locomotion with a rich repertoire of different gaits; and an entirely new paradigm for machine learning. With this background in place, we present the design of a novel distributed device that uses worms as a mechanism—not just a metaphor—for computation. Next we proceed to describe the application of such a mechanism to distributed AI programming. Until very recently this research has been beset with a number of practical problems, none of which appears to have reported before, at least in the open literature. The main result of this paper is a new programming environment which solves all problems together. Finally we draw some tentative lessons from the research.

[1]This probably explains why early publication, in advance of convincing experimental results, has been a longstanding custom in AI research generally.

2. Worm technologies: a survey

2.1 A VEHICLE FOR AUTONOMOUS SYSTEMS

The first use of worms as a metaphor for computation can be attributed to Brunner's powerful novel *The shockwave rider* (Brunner, 1975). This piece of science fiction was apparently taken seriously by researchers at Xerox's Palo Alto Research Center who developed "worm programs" consisting of several segments, each comprising a process running in a separate workstation in a computer network (Shoch & Hupp, 1982). The segments keep in touch through the network. Each segment is at risk because a user may reboot the workstation it currently occupies at any time—indeed, one of the attractions of the idea is that segments only occupy machines that would otherwise be idle. When a segment is lost, the other segments conspire to replace it on another processor. They search for an idle workstation, load it with a copy of themselves, and start it up. The worm thus repairs itself.

Worm can be greedy, trying to create as many segments as possible; or they may be content with a certain target number of live segments. In either case they are very robust. Stamping one out is not easy, for all workstations must be rebooted simultaneously. Otherwise, any segments that are left will discover idle machines in which to replicate themselves.

The Xerox experiments emulated the legendary ability of worms to regrow when cut in half. However, no attention was paid to the intellectual strengths of worms, and the worm community probably considered the use of their name in the context of such simple programs as grossly libelous. Interestingly, Brunner's novel, which describes a program that grows to take over an entire global computer network upon which society has come to depend, seems to invite an analogy with tapeworms—which are not actually worms but are so termed for possessing elongated bodies—rather than the species of *annelida* or segmented worms.

2.2 A MECHANISM FOR LOCOMOTION

Although we were unaware of this when the original article on the gait of spiders and robots (Thimbleby & Witten, 1989) was penned, worms have stimulated research into artificial locomotion. Miller (1988) analyzed the motion dynamics of snakes and worms, which have surprisingly complex internal structure, and produced excellent graphical results from a dynamic model. Not content with mere pictures, he demonstrated a radio-controlled 8-segment worm during the presentation of his paper at SIGGRAPH '88.

Worms progress by capitalizing upon differential friction effects well known to cross-country skiers: their scales slide forward over the ground relatively easily, but when the body segment slides backwards the scales dig in and the frictional force suddenly becomes very great. Miller modeled each segment of the creatures using a cube of masses with springs along each edge and across the diagonal of each face. He was then able to simulate the familiar worm-like motion, which results from the worm sending waves of compression from its head to its tail to avoid the undue stresses at either end that would occur if all the spring lengths were varied in phase.

Interestingly, while worms are only capable of this one kind of motion, snakes, which enjoy a very different anatomy, have four distinct gaits. Worm-like motion, called "rectilinear progression," is achieved by the snake sliding its skin over its ribs (Klauber, 1982). "Horizontal undulatory motion" is the more familiar sinusoidal "snaking" movement, "sidewinding" involves looping the body sideways and is usually performed over shifting sand, while "concertina progression" is a rare motion that involves flexing and straightening of the snake. Caterpillars exhibit a fifth legless gait: they flex their backs vertically in a sinusoidal fashion, lifting the underside, thrusting it forward, and placing it down again; their lack of reliance on differential friction equips them admirably for optimization problems such as hill-climbing, steepest descent, etc.

It seems that as well as offering a metaphor for autonomous computation, worms also provide inspiration for mobile robots. It is still unclear whether even spiders can equal the rich gait repertoire of legless creatures.

2.3 A NEW PARADIGM FOR LEARNING

Planaria (or flatworms), though with a very primitive nervous system, possess the ability to learn both by habituation and associative conditioning. Indeed, by virtue of their small size and low cost, they are ideally suited for classroom experiments in learning, particularly with today's large introductory psychology classes (Katz, 1978). A remarkable experiment in the 1950's yielded the astonishing result that when trained worms were chopped into pieces and fed to a group of untrained ones, the training was actually transferred to the cannibals, in contrast to a control group which had been fed untrained worms (these and other results are reported by Best, 1963). Such experiments have continued, increasing both the complexity of behavior (e.g. Carney & Mitchell, 1978) and the number of generations of cannibalism (e.g. Loomis & Napoli, 1975). The carrier of training appears to be RNA, for if it is extracted from the bodies of taught worms and injected into untrained worms of the same species, it causes similar behavior.

These results point to a new paradigm for learning, "digestion-based learning" or DBL, which represents a significant advance over other paradigms such as explanation-based, case-based, similarity-based, and so on (Kodratoff, 1988) and has profound implications for artificial intelligence. We already know that diet is a major factor in behavior generally: its systematic, if unconscious, control may incidentally provide an evolutionary explanation of the apparently irrational tendency of parents to force certain kinds of food into their reluctant offspring regardless of the cost in terms of damage to family relationships. The advantage of DBL is that it can occur at any stage of development and so avoids the normal bottlenecked life cycle of evolution by reproduction (Dawkins, 1989).

Planaria are primitive animals, and the original research on cannibalism used classical conditioning where a strong light, which normally caused the worm to stretch itself, was immediately followed by a mild electric shock which caused it to contract or turn its head. (More recent work has investigated more complex phenomena, such as whether such worms can detect magnetic fields and, if so, with what exactness; Krebs, 1975.) To enable more useful behavior to be learned, we must move up the evolutionary scale.

3. Universal worm machines

The role of RNA as a basis for memory in more advanced animals is controversial, but *annelida* (segmented worms) represent a convenient level to experiment with DBL. *Annelida* have a more advanced nervous system than *planaria*. They readily demonstrate trial-and-error learning and can be taught to master simple mazes. Successful learning is not only easier to detect, but the more developed musculature of these worms permits them to control their environment. Under suitable feedback conditions, as in Skinner boxes, they demonstrate impressive learning capability. As noted above, DBL operates very much faster than evolutionary timescales and consequently we have considered arranging for it to work in concert with natural selection. Briefly, we were able to utilize the positive feedback effects of slicing successful *and* unsuccessful worms automatically in a two-lever Skinner box arrangement and feeding appropriate slices to the surviving front ends. In this arrangement, worms that learn more slowly than the population average are rapidly weeded out. Although these experiments were technically promising, we have abandoned them following advice from our ethics committee.

It is but a small step from teaching elementary geometric abilities such as turning left or right in a T-maze to teaching elementary computational abilities like "nand" and "nor".[1] These logical operations are universal building-blocks for computation from which any digital device can be built (and indeed have been), and the existence of a "nand"-worm which could be reproduced in arbitrary numbers simply by appropriate breeding and feeding policies raises the possibility of autonomous biological computers of arbitrary size and complexity. In effect, every compost heap could become a Turing machine—which, along with easily-implementable subterranean lines of communication, introduces the particularly exciting prospect of the world's first truly international biorenewable multicomputer.

This prospect has philosophical ramifications that are even more dramatic than its potential economic benefits. Imagine, after Searle (1980), a human interrogator outside a room which contains such a "worm machine" painstakingly programmed to respond to a story, written in Chinese, by answering questions about it. (This *gedanken* experiment presupposes some means of input and output, but—although we have yet to work out the details—this seems to involve merely technical problems.) Of course we assume that the worms cannot understand Chinese. Given that the programming has been sufficiently cleverly done that intelligible, even appropriate, answers are given to the input questions, can we say that the worms—which after all are only following digestively-programmed rules—actually *understand* the stories? This appears to provide a specific

[1] Indeed, many respected computer scientists have difficulty with left and right even though they can manage the logical connectives and can be left to write left to right the right Boolean operations.

counterexample to Voltaire's (1770) remark that "those who have the best stomachs are not the best thinkers."

4. The chess worms

Motivated by the key role that worms are playing in the development of computer science, as very briefly surveyed above, we have begun to investigate their computational capabilities more systematically. In contrast to the "worm programs" of Shoch & Hupp (1982), which involved simulated worms, our own research explores the possibility of using worms not as a computational *metaphor* but as a computational *engine*. One of the basic problems faced is that *group* computation clearly has no survival value for any individual worm. To provide the necessary stimulus one must organize worms into complex, coordinated, systems, and arrange for competition between different systems, so that evolutionary forces can work to motivate computation. We have found that two-player zero-sum games provide an excellent environment within which teams of worms can compete.

Following the pioneers of AI (Newell *et al.*, 1958) we have begun with the game of chess, and have achieved success in teaching a multi-worm system some standard gambits.[1] Figure 1 shows a photograph of an early experiment. We began work with a standard-sized tabletop chessboard. While chess is often conceptualized as an abstract game that operates according to a set of rules and does not depend on any physical artifacts, this idealized view does not hold when we step outside the computer into a real worm environment. We encountered three principal problems, none of which seems to have been reported previously in the AI literature.

First, the only worms we could find were dwarfed by the chess pieces and had trouble moving them. This led to a preference for minor pawn moves—not for strategic reasons but simply to save energy. Indeed, one team which chose to castle early in the game were completely exhausted by the move and had to retire. Needless to say our policy of natural selection mitigates heavily against such moves being chosen in the future.

Second, garbage collection proved to be a problem, as can be discerned in Figure 1. Worms leave behind a gelatinous residue which congeals in the form of a cast. The problem is not so much in clearing up the garbage, which the worms can be programmed to do using techniques similar to those developed for conventional LISP systems, as in the fact that these worm-casts are occasionally taken for pieces on the board by the opposing team. The notion of a program *deliberately* creating garbage to confuse its opponent seems to be quite new.

Third, the chessboard environment is worm-hostile and we have experienced a disturbingly high mortality rate. As noted in our previous article (Witten & Thimbleby, 1989), current research at the Open University, England, is tackling the problems of being able to produce large quantities of earthworms

---

[1]Successful gambits require an appreciation that present piece sacrifice may obtain future gain and thus provides an ideal testbed for cannibalistic learning techniques.

adapted to living in different environments and we hope to be able to benefit from this (Knight, 1989).

Despite these limitations, and the severe strain-inflated morbidity placed on DBL, we have had some impressive breakthroughs. Figure 2 shows just one example: here, worms *from both sides* are fiercely cooperating to resign white—indicating an appreciation for the outcome of the Rook and King against a lone King.

We have recently hit upon a way of solving all of the above three problems at once, by using different equipment. Instead of the ordinary tabletop chessboard we have begun to employ a pocket chess set in which all pieces are equipped with a small peg that fits into a hole in the center of each square. In the space under the board we construct a worm-friendly environment from compost. This constitutes a "virtual machine layer" with ordinary housekeeping facilities, not unlike that in a conventional operating system. The holes in unoccupied squares provide an ideal interface between the virtual machine and the application layer. Worms can cross this boundary freely (although a security system is contemplated for the future that will restrict access to those with special tokens). Garbage is confined to the operating system layer where it can decompose naturally, and does not clutter up the user interface.

5. Conclusions

Worms have already provided powerful and productive metaphors for computer science researchers, ranging from distributed computation through legless locomotion to digestion-based learning. However, our proposal is, as far as we are aware, the first to take them seriously as a computational vehicle. Early experiments with chess-playing worms were plagued with practical difficulties stemming from the real-world nature of the work, which contrasts sharply with the idealised chess abstraction to which previous researchers have confined their vision. Such problems will have to be addressed in other domains too when we bring our AI efforts out of the epistemological laboratory and into the field (so to speak).

We overcame these difficulties by developing a new programming environment more suited to the task at hand. Now that the appropriate tools are available and the most difficult conceptual problems have been solved, rapid progress in implementation is anticipated up to the point where hand-held chess systems based on worm technology can compete with, and ultimately overthrow, their electronic counterparts.

We are dismayed that some conventional AI researchers with no sensitivity to the difficulties involved in our research have suggested that we teach the worms the Hedgehog Defence (against the Spanish Opening) as this has a more obvious survival value. We in fact expect very shortly to be in a position to report actual quantitative results to the AI community, and then *we* will be laughing.

References

Best, J.B. (1963) "Protopsychology," *Scientific American 208*(2): 54–62; February.

Brunner, J. (1975) *The shockwave rider*. Ballantine, New York.

Carney, R.E. & Mitchell, G.L. (1978) "Reactions of planarians after cannibalization of planarians exposed to four stimulus combinations," *J Biological Psychology 20*(2):44–49; December.

Dawkins, R. (1989) *The selfish gene*. Oxford University Press, Oxford .

Katz, A.N. (1978) "Inexpensive animal learning exercises for huge introductory laboratory classes," *Teaching of Psychology 5*(2): 91–93; April.

Klauber, L.M. (1982) *Rattlesnakes*. University of California Press.

Knight, D. "Nice work for a worm," *New Scientist 1682*: 55–59; July.

Kodratoff, Y. (1988) *Introduction to machine learning*. Pitman, London.

Krebs, E.K. (1975) "Factors in conditioning and orientation of planarians: learning in a polar field," *J Biological Psychology 17*(2): 21–23; December.

Loomis, L.L. & Napoli, A.M. (1975) "Transfer of training through two cannibalisms of planarians," *J Biological Psychology 17*(1): 37–40; July.

Miller, G.S.P. (1988) "The motion dynamics of snakes and worms," *Computer Graphics 22*(4): 169–178; August (Proc SIGGRAPH '88).

Newell, A., Shaw, J.C. & Simon, H. (1958) "Chess playing programs and the problem of complexity," *IBM J Research and Development 2*: 320–335; October.

Searle, J.R. (1980) "Minds, brains, and programs," *Behavioral and Brain Sciences 3*: 417–457.

Shoch, J.F. & Hupp, J.A. (1982) "The 'worm' programs—early experience with a distributed computation," *Communications of the ACM 25*(3): 172–180; March.

Thimbleby, H.W. & Witten, I.H. (1989) "Bugs in the Church-Turing thesis," *Canadian AI*, 17–18; April.

Voltaire (1770) Letter to d'Alembert. August 20.

Witten, I.H. & Thimbleby, H.W. (1989) "Worms and bugs in the Church-Turing thesis," *Canadian AI 21*, 3–4; October.

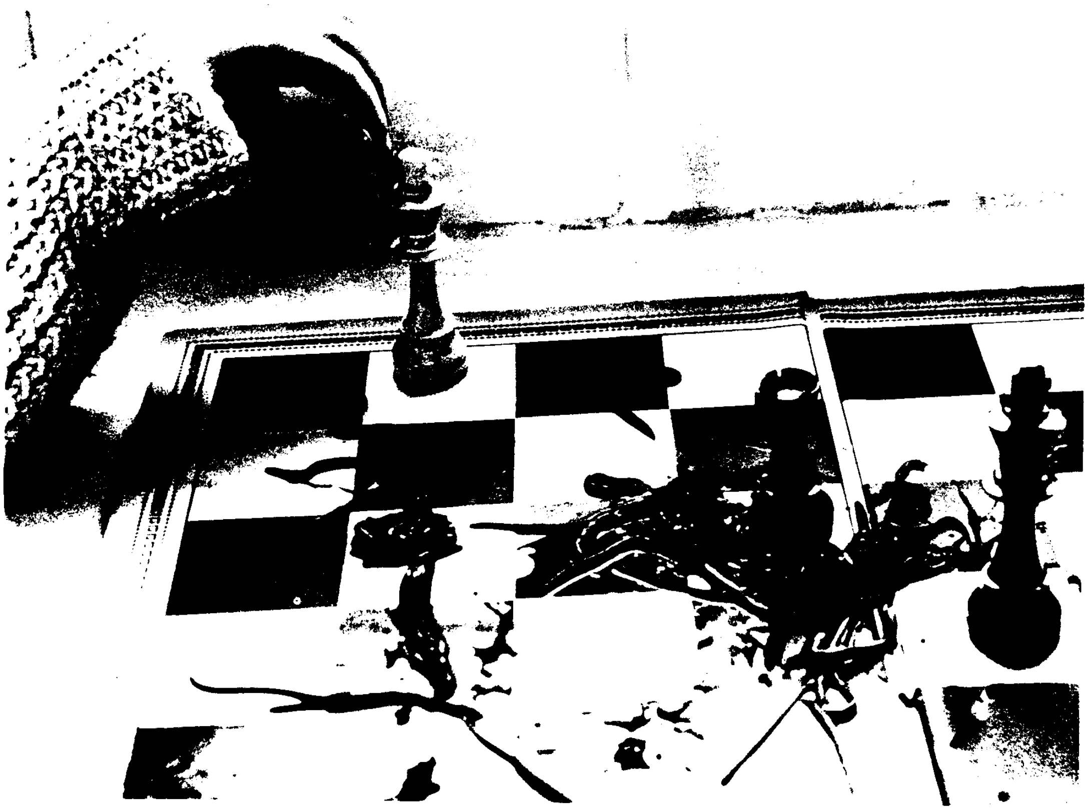

Figure 1. K+R *v*. K, a known loss for white. Experimenter's hand indicates scale.

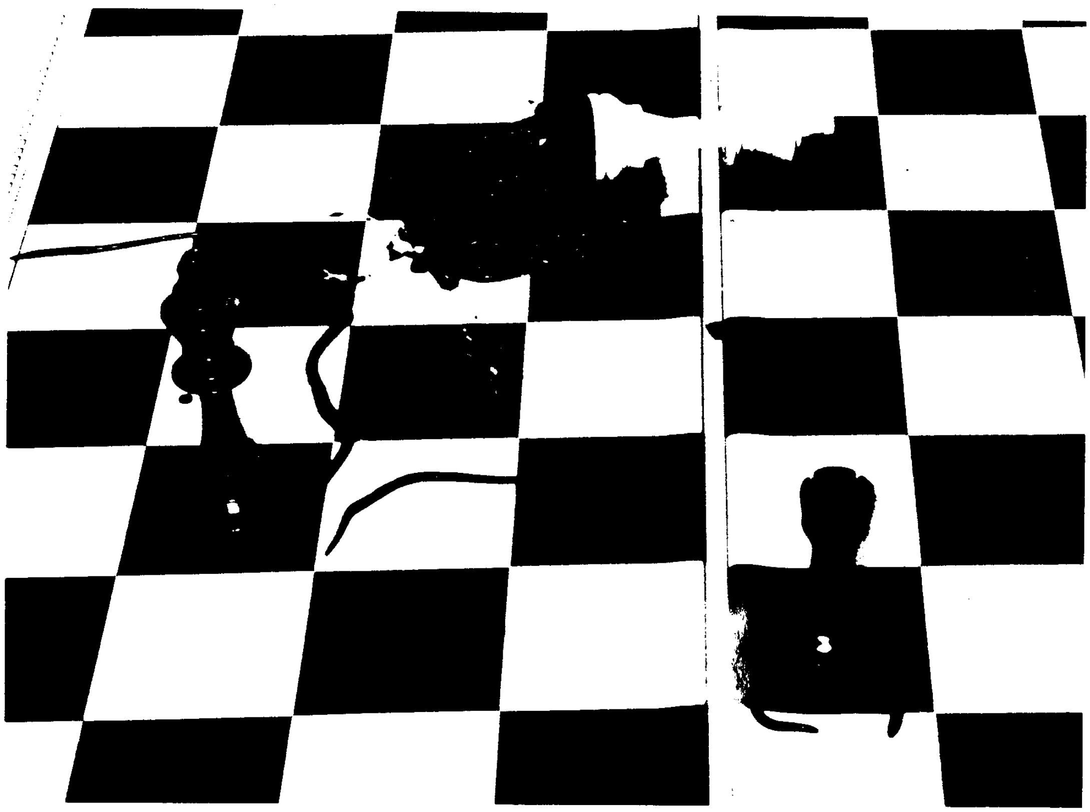

Figure 2. Worms from both sides conspire to resign white. Note the massive resources required to remove a piece from the board.