

The Worm has Turned: An Adaptation of a Mind Simulation

Martin Wurzinger
<http://www.otoom.net>
martinw@otoom.net

Abstract

Enabling machines to think like humans has been a significant issue ever since the capacity of computer programs achieved a critical complexity for such considerations to be applied. The present article begins with an overview of the problems seen under the traditional perspective. It continues with a different framework, this time concentrating on functionality rather than content. A program that simulates cognitive dynamics is introduced and how such a view answers to the dimensions of scale, from neurons to thought structures in an individual to group behavior and society at large. The main part focuses on an adaptation of this system, driving an artificial ‘worm’ towards its ‘food’ under various conditions. The evaluation describes the methodology used, the results obtained, and the significance of those results as they pertain to a humanoid robot. Finally the future development is touched upon and the issues are summarized including questions about the role of an artificial mind from the perception of humans as its creator and society in general.

Keywords

artificial mind, humanoid robot, chaos, attractor-type behavior, artificial life.

1. Introduction

How a problem is conceived becomes highly significant once the steps leading to a solution are being implemented. If the task is to model the human mind the conceptualization is doubly critical because ultimately there exists no objective reference to how we see ourselves. We literally have to start at the beginning.

The tendency of humans to transfer their perceptions into the material can be observed from the earliest times onwards. So much so that the creation of tools has been used as a marker for our evolutionary development down the ages. The ability to fashion an object out of what one’s environment supplies is significant enough, but in the present context it doesn’t stop there. The nature of the tool itself has always been representative of not only what we wanted to do but what we perceived as potentially being able to do, however idealistic such musings may have been under the circumstances. When the picture did not extend further than the need to break, pierce, or bend something we transferred our wishes into objects doing just that. As the time came to see, or rather think, further we extended our horizons to more and more power, to greater speed, even to flight. And in due course we abstracted higher still and turned our attention to replicating mental processes, from the abacus to the computer.

As short as this summary may be, it highlights two important aspects. One, the construction of artifacts follows the image we perceive of our selves; and Two, the sophistication of that image reflects the capacity of our minds to abstract.

Somewhat hidden in that scenario is the methodology employed in building the desired object. Since we started with simple things, representative of our relatively lower abstractions of the world and ourselves, the focus was mainly on content, not functionality. The latter naturally accompanied the former, but the artisan was more of a builder, less a designer. A Pythagoras came long after levers were a familiar sight in everyday life.

The preference for content over functionality (meant here in its original sense, that is the property of a function) held pace with the increase in complexity. Although modern aircraft design for example can largely dispense with the trial and error of old, the results are firmly placed within the conceptual landscape of an object-oriented world. Indeed, it would not make sense otherwise: what we need is something material that can address the demands of stress and load in the tangible.

As soon as we cross into the pure abstract however the reference changes. An algorithm is not an object, although we may apply it to one. The process of triangulation does not need us running around with sticks but can be done on paper. Suddenly the functionality comes first, and the corner of a building, the peak of a hill, or one's position at sea is secondary. The problem becomes even more acute once we reach the limits of perception in terms of the conceptual path from abstracted object to process to application.

Provided the building or the peak can be identified in the real for what it is (even if we can't place our hands on it directly) their role in the design process is equally accessible because their definition holds pace with the transparency of intent.

Now take the human mind. Here our powers of definition have been found wanting, as our reflections throughout history demonstrate. The content-imbedded template of imagery falls short of its goal since the system (*ie*, the mind) is required to define itself to the last degree, something it naturally can't do unless it could avail itself of yet another system—a logical impossibility in terms of a comprehensive definition when relying on object-by-object instantiation.

In the field of artificial intelligence the issue has been subsumed under what has been termed the 'grain problem' [22], and now evolutionary psychology even is forced to address the recursive deconstruction of observed events when trying to describe a particular behavior form suggesting an inherent modularity that throws open the question of adaptation against the background of genetic hardwiring [2].

Significantly, attempts at modeling neuronal processes via computer software concentrated on simulating specific cell dynamics, for instance the firing patterns surrounding Purkinje cells [18], or the molecular level model of a single synapse and the cellular level model of the Neocortical column [23]. For the authors the importance lay in the material processes because in the context of content we need to know about their characteristics for an adequate redescription from the molecular level upwards.

As essential as such information is when it comes to understanding the structural elements of the brain, whether it opens the dynamics of the mind to scrutiny remains in question. An analogy would be the modeling of a reciprocal engine. One could simulate the entire process in terms of space, pressure, fluid dynamics, etc, with the output a similar representation of momentum and torque, but all this won't make the computer move across the desk.

On another occasion when writing about those very issues Coiffet comes up against the conceptual boundary established by the finite level of self-awareness compounded by the sheer complexity and dynamic interdependence of the brain [12]. If machines are meant to reflect the cognitive capacity of humans they need to possess a system that displays a set of functionalities representative of what can be identified in familiar mental processes.

In the context of the above the last point is particularly important. Modeling an engine is only the first step; sooner or later the representation has to be made manifest in the material if we want to get some actual work done. Cognitive processes on the other hand do not require such a transposition—their result is yet another abstraction waiting to be implemented. The nature of the medium takes second place to its functionality, provided the outcome meets the expectations.

Attempts at defining the ongoing events at whatever scale has led to the dynamical approach by recognizing the associated timelines as well as satisfying the need for mathematical precision under the umbrella of useful redescriptions [24]. Such models however have been found "inscrutable" and apart from "carefully constrained simple cases" largely failed in their stated intent.

Indeed, the insistence to transpose the observable states of a biological system into the formality of a replicating framework produced a number of concepts summarised to varying degree by what have become known as functionalism, computationalism, and/or the dynamical hypothesis.

It needs to be pointed out therefore that ‘functionality’ in the current context refers to the conceptualized abstraction of a particular behavior pattern, regardless of what label a functionalist may have selected to conveniently categorize a certain module within a larger system. The instantiating elements may be sufficiently specific to suggest a module, but they also may be nothing more than a however fleeting pattern becoming visible to the observer. In the latter case they would not only evade the formalism of mathematical equations, but dealing with such events also highlights the problematic nature of semantic markers as pointed out by van Gelder [15].

The debate under computationalism for instance derives its sustenance from the desire to transcribe the processes of computer models into the hypothetical perspectives of mental states. The associated time frames should be of interest to a dynamicist, not least because in linear as well as distributive algorithms the system clock is highly critical. If an algorithm can be represented through the symbolic framework of a flowchart and two models are similar if their end result is comparable, functionalists should be satisfied even with differential equations as a substitute for a program. The distinctions—or otherwise—have been amply described by Giunti [16]. Although I have not entered into their essential debates, the respective overlaps just touched upon between the three main concepts listed above—processes occurring over time, the representative nature of mathematical equations vs algorithms, and implementation issues in relation to symbols and/or integers—fuelled Harnad’s polemic with Searle’s Chinese Room argument as its main focus [17]. Rather than increasing the potential of the conceptual spaces delineated by such discussions, I would argue that all they managed in the end was to appropriate the English language to conduct their abstractive edifices. I would like to reclaim some of that language.

In order to model the mind we need not necessarily insist on a direct replication of all the participating elements’ dynamics, as long as the process portrays a similarity in terms of its functionalities as understood in this paper. Furthermore, since the program also employs computation as well as integer groups which can be interpreted as symbolic representations (as we shall see), and chaotic type behavior along time lines, the described processes must be allowed to exist in their own right without misleading conceptualizations borrowed from the traditional hypotheses.

Since we model the behavior of the mind such a framework should offer itself to varying scales, from an individual’s thought structures to group behavior and indeed wider society. As such we can avail ourselves to a far wider resource space to look for confirmation. About 150 major events from society, politics, and science have been collected since 2003 supporting the validity of the model (see Appendix).

In the sections to come I will first introduce the basic computer program simulating cognitive dynamics (including what they are in principle), followed by an extension of the program to show how it can move an artificial ‘worm’ towards some target as it utilizes such dynamics. Next I will summarize the main points derived from the wider reference of society to illustrate the applicability of the model since they are relevant to intelligent machines.

The comprehensive detail of the basic computer model has already been outlined in a separate paper [27]. For reasons of continuity I have included the main points; for a more comprehensive outline of the essentials the reader should consult Ref. 27. It is also available from www.otoom.net, as are the full results of the worm program plus the program itself including source code. This paper deals with a subset of the entire model which is described at length in a text [30] whose acronym makes up the site’s name and uses major philosophical treatises across the centuries to add a historical/evolutionary context, as well as building from neuronal dynamics upwards. The sheer conceptual as well as operational extent of the system of mind makes such an expansive treatment necessary. Therefore the present text restricts itself to the details relevant to a computer simulation that demonstrates the principle behavioral characteristics we would expect from an artificial mind.

To understand the technical aspects of the worm program a reading of this article and possibly the paper is sufficient. Refer to the Appendix for the location of the supplementary material.

2. Basic Computer Model

Under the auspices of functionality (as defined above) rather than object-related content we can view the mind as a system that has emerged as a result of the neuronal dynamics within the physical structure of the brain. As such the definition of mind entails the dynamics that offer themselves for consideration once we leave the neuronal materiality aside and concentrate on the mind's behavior as another layer positioned above the former.

The system can be seen as featuring input, processing space, and output. Delineating further its input covers the nerves from the organs, the processing space is made up of the various cell types within the central nervous system, and the output is carried via nerve paths to the respective destinations of the body.

The physical structure of the processing space is certainly daunting. Approximately 100G neurons, each one connected to between 1000 and 250,000 other neurons, these synapses containing around 5000 neurotransmitters in the form of molecular combinations, and each neuron itself manufacturing several hundred thousand proteins [4].

The resultant overall dynamic is the system of mind, and given its obvious behavior must feature the following characteristics:

- its processes are a function of input;
- similar input produces similar results;
- the results are more or less localized;
- dissimilar input induces changes in the result as a function of the degree of dissimilarity;
- localization allows individual regions to be regarded as sources of further input to other regions within the overall system, and output can be viewed under a similar framework.

Mindful of the neurons' complex characteristics but disregarding their actual nature, the result of some input can be termed a phase state across the relevant cluster of neurons, in other words the dynamic entities.

Therefore we have a system receiving data as input, processing such data resulting in phase states within certain clusters defined by the degree of similarity between such states, and output as a function of relevant phase states.

The output is projected into the system's environment from which data are once again received as input. Such a feedback loop makes for an iterative system, the dynamic nature of its entities being responsive to constant update and so continuing the process.

The underlying complexity of the processes can be seen in a positive light, namely as a sufficiently high level of granularity giving rise to a commensurate high degree of variance in terms of emerging rule sets, representative as they are of the localized phase states. See section 3.3.4. Further comments for some points on granularity.

An iterative system containing dynamic phase states as a result of incoming data and being responsive to their varying nature, is a chaotic one.

We now can move on to the construction of such a system around the identified functional characteristics.

2.1 Layout

The neurons are represented as nodes arranged in a matrix, with each node being another matrix composed of elements. The elements are integers initially arranged in sequence seeding all the matrix nodes at startup. The main or inner matrix is surrounded by an input and an output section, nodes similar to the rest and divided into several regions each depending on the number of in- and output types required.

In- and output nodes are connected to a number of inner matrix nodes according to the connectivity level set by the user. Inner matrix nodes are connected to each other in similar fashion except that each represents the parent node of a tree, its branches within the inner matrix placed in a random manner. The levels of branchings are also set by the user. Since every node is a parent node, it becomes a member of many overlapping trees. Fig. 1 illustrates the layout.

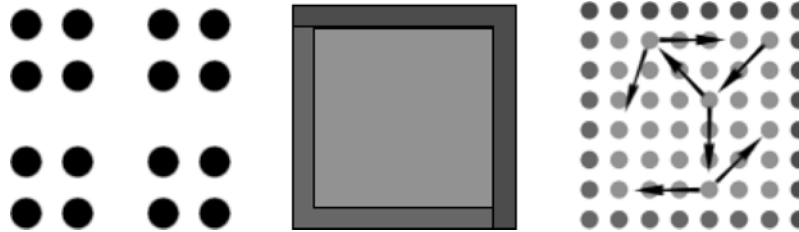


Fig. 1. General layout of the processing space. A main matrix composed of four nodes, each node an element matrix with four elements (left), input and output section surrounding the inner matrix (center), and an example of a connection tree with two nodes branching from every node and depth level 3 (right); only one branch of the parent node is shown.

Input distributes itself across a certain region in terms of block size per node, gets transmitted to some parent nodes within the inner matrix from where it spreads down the respective trees. Output emerges from certain nodes of the matrix that are connected to their output nodes, across which the data are again spread by block size per node. The processing of data moves through the input section, along the parent nodes of the inner matrix and down their trees, and finally to the output nodes (more of this process later).

The inner matrix nodes represent the biological neurons, the elements per node their protein formations, connectivity levels and trees simulate synaptic links, and the results of the processing algorithm stand for the neurotransmitters inside the synapses. The distribution of data across their respective in- and output regions can be compared to the specific nerves coming from the senses or going to organs (such as the visual field-specific receptors on the eye's retina, or connections to individual muscle fibres).

2.2 Processing

In line with the dynamics identified above the algorithm processing the data needs to induce chaotic behavior. That is to say the element values must be changed in such a way that at every iteration the result turns each element into an attractor with respect to the corresponding element in the next node along the tree.

An integer in the former element is termed the reference value, in the latter the resident value. The result of the process, `new_val`, becomes the current resident value. If the reference value is the smaller of the two, the following algorithm applies:

```
diff = res_val - ref_val;
new_val = res_val - ((100 - diff) * diff / 100);
```

If the resident value is smaller the algorithm becomes,

```
diff = ref_val - res_val;
new_val = res_val + ((100 - diff) * diff / 100);
```

When implemented in the programming language C/C++ with two bytes reserved for integer storage (allowing a range from -32,768 to +32,767) this usually has the effect of moving the resident value towards its reference, but not always. Alternative outcomes are oscillations by the resident value or fluctuations outside any recognizable pattern. Fig. 2 shows the three types of possible outcomes, all dependent upon the reference value. Note that the initial stages of integer movements are not necessarily indicative of their eventual type.

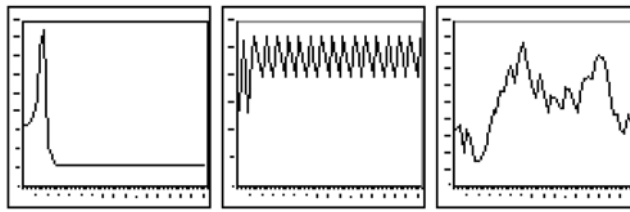


Fig. 2. Changing values of integers in the node elements when subjected to a processing algorithm that produces attractor-type behavior. A stable (left), periodic (center), and strange attractor (right).

If four bytes are allowed for integer storage the probability of a strange attractor occurring is markedly decreased. The choice of algorithm had been as much a matter for its ability to generate chaos-type outcomes as the characteristics available under C/C++.

Since the process moves iteratively through all the nodes of the inner matrix and each node belongs to several trees, the nodes also become attractors with respect to their neighbors (those nodes touched consecutively by the algorithm as it moves through the matrix). As the outcome depends on the integer value, where a particular movement causes the attractor type either to be maintained or changed into another type, the overall result produces domains of similar behavior in terms of the type of attractor generated. These domains achieve their own stability through the input, whether such input comes from the outside or occurs between the domains and their clusters.

Note that the integer values themselves are irrelevant with respect to the representative nature of the domains—what matters are the relationships they are able to generate between the clusters. The latter's potential to remain similar to their neighbors can be termed their affinity. Hence under a traditional perspective the integer values per node could be viewed as a symbol, yet they are indisputably part of a computation through an algorithm. In the end it is the affinity relationships induced by the inputs that determine the representative quality of the domains, and to what extent they are being maintained depends on how influential the subsequent inputs are in relation to previous data.

2.3 Behavior

Examining the characteristics of the domains produced by the algorithm reveals differing degrees of affinity with each other. Relevant domains can be grouped into sets, and as such can feature intersections due to their mutual affinity relationships. Since domains are representative of input (between the environment and the matrix as well as between the domains within the matrix), an intersection can be viewed as yet another domain containing the essential characteristics of two or more sets.

Due to the interdependence between input and the resultant node states an intersection therefore represents what can be termed an abstraction on a higher interpretative level.

As mentioned before, the functional perspective allows the dynamic framework to be applied at varying scales, since it is the process which is first and foremost under focus with the actual content (*ie*, what the dynamics stand for at a particular scale) coming second.

Human articulations such as written or verbal statements can be deconstructed in order to analyze the thought structures behind them. On that higher scale an idea can be viewed as a domain, concepts as clusters of domains, and considering the modifying influence a certain idea has on the latter the underlying thought structures portray their denominative nature with respect to the former. Ideas can be grouped to produce an abstraction, concepts can gain a greater significance by generating their own clusters, or can dissolve an existing one.

Whether these phenomena become already apparent on the level of neuronal activity in the brain is impossible to say at this stage, but they can certainly be identified amongst the matrix nodes of the computer model since the dynamic framework holds in both cases. It is worth mentioning that conscious

and subconscious thought structures are not being addressed here as such, although their inherent dynamics come under the same perspective.

3. Worm Program

While the basic computer program generates an output that allows it to be analyzed in conjunction with the characteristics of individual nodes, it needs to be shown that the internal phase states of the system are capable of influencing an object in a meaningful way; it should actually ‘do something’. The in- and output regions, the resultant phase states within the main processing space, and the representative quality of these states in relation to in- and output are necessary preconditions to what a scaled-up system would require.

3.1 The worm

As object a stylized version of a ‘worm’ has been chosen. It consists of several discs, the first being its head, and connected via links. The links can change their relative direction to each other and are positioned according to three such parameters, direction, direction-change, and additional-direction-change. Hence the discs in their relative position to each other are not limited to single concave or convex curves—the thing can ‘wiggle’.

Movement is affected by going through each link from the head down and making the last segment the anchor point on the surface. At the next cycle the same process is repeated from the tail upwards and now the head becomes the anchor. Thus the action can be compared to the way a mosquito larva propels itself through the water. The worm moves within a square (its ‘cage’) and at the center a larger disc represents the ‘food’. When the head reaches the food the goal has been achieved.

The input to the system comprises the direction parameters as well as the distance from the worm’s head to the food (with one exception: when the worm hits the side of the cage only those parameters are used that keep it inside). The effective output consists of the direction parameters only, which are then fed back into the system. The cycles are therefore iterative.

Fig. 3 shows the worm and a possible path as tracked by the program.

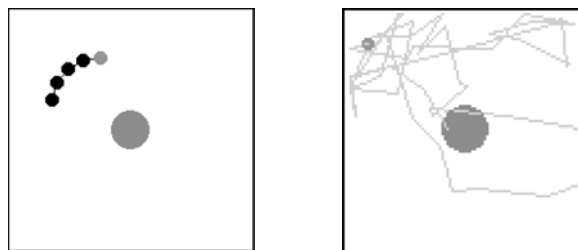


Fig. 3. Sample screen shots of the worm (left) with its head in light-gray and a certain path of its head by the time it has reached the food (right). Note overshooting of the target at some stage and veering away from it, also sliding along the edge of the cage. The starting position in this case is indicated at the top left.

To allow a precise evaluation no other information accessible to the system is provided. In other words, there is no awareness about the position and how to move the segments with respect to the food, and at any given time the system is therefore unable to take a ‘higher view’ in order to enact an additional modification of the direction parameters to close in on the target. Either the worm’s path puts the head on the disc or it doesn’t. As a consequence it is possible for the worm to overshoot the center or veer away from it at the last moment. Nevertheless, it can be shown that the system can generate positive outcomes under certain conditions through its ability to produce meaningful patterns via its phase states.

The ‘higher view’ becomes possible once additional input streams are available which in turn produce their own representative states in conjunction with those already present. In principle those states can be viewed as another set of conditions processed in terms of the affinity relationships discussed earlier. For

them to be implemented a considerably enlarged matrix together with a greater range of input is necessary. Refer to the earlier remarks about rule sets and their respective granularity. Allowing for an appropriately high degree the processing space needs to be available but the essential dynamics are the same.

3.2 Methodology used in the evaluation

Tests of the basic computer program have shown the significance of matrix sizes as well as the size of the element matrices at each node [28]. A larger size permits more diverse domains, more elements make for higher responsiveness to varying input, and a domain's latency (*ie*, its potential to affect affinity relationships) is enhanced through a greater number of nodes and elements per node. This is something to be expected when considering the dynamic equivalent at the higher scale of groups and/or societies, where more people have a greater chance to develop ideas and diversify them into evolving concepts.

The matrix sizes and their numerical dependencies are listed in Table 1.

Size type	Matrix nodes	Matrix elements	Tree nodes	Tree elements
1	25	225	75	675
2	64	576	192	1728
3	64	2304	2688	96768
4	100	3600	4200	151200
5	100	6400	4200	268800
6	100	10000	600	60000
7	210	21000	5250	525000
8	210	47250	2100	472500
9	210	47250	8820	1984500
10	210	47250	10920	2457000

Table 1. Number of matrix nodes and elements, number of nodes per tree and elements per tree. *Note:* Only inner matrix nodes are counted; in- and output nodes are disregarded.

The numbers of nodes and elements are determined by matrix rows and columns, elements per node (again arranged in a matrix with element rows and columns), connections per node, and the depth level of trees. All these are parameters set by the user.

Four starting positions of the worm are used for every size type: top left, bottom left, top right, and bottom right. In addition several conditions are applied, either singly or in some combination. Generally speaking they are,

- **block factor:** the number of times per cycle input is processed before an output is generated;
- **approval factor:** the number of times per cycle input is processed if the direct distance between the worm's head and the food has decreased on the previous cycle;
- **position cycling:** rather than stopping the program when the food has been reached and changing to the next starting position with a newly initialized matrix, the worm resets automatically to the next position with the current phase states intact;
- **teach mode:** the worm is moved from the current starting position to the food by an algorithm applied from the outside, such that the direction parameters are calculated in order to get closer to the food; the parameters become an input to the system where they are processed but the resultant output is discarded; this can be done more than once before the worm resets to the starting position and the system proper takes over with the previously generated phase states intact.

Recall the three direction parameters and the distance to the food; they are the only values processed by the system and only the direction values influence the positioning of the worm. There is no other way it can be directed towards the center. Therefore combinations of conditions are selected to determine how effective each one has been with regards to the resultant domains, which after all are ultimately responsible for the outcome.

The test runs are numbered with letters, categorizing the set of conditions they represent. Table 2 is a summary.

Test run	Condition/s
a	Basic configuration, no condition
b	Initial direction settings diametrically opposite to the default values
c	Block factor = 4
d	Position cycling
e	Position cycling, block factor = 4
g	Position cycling, approval factor = 4
h	Position cycling, block factor = 4, approval factor = 4
i	Teach mode (3 cycles)
j	Teach mode (1 cycle)
k	Block factor = 4, teach mode (3 cycles)
l	Position cycling, teach mode (3 cycles)
m	Position cycling, block factor = 4, teach mode (3 cycles)
n	Position cycling, approval factor = 4, teach mode (3 cycles)
o	Position cycling, block factor = 4, approval factor = 4, teach mode (3 cycles)

Table 2. Series of test runs with their respective condition/s. Note: Test run 'f' served as comparison to experimental run 't'; both have been omitted. Test run 'b' has been done as a matter of interest only and has not been used further in the evaluation.

Thus the overall arrangement of the test runs is as follows (Table 3).

size type:	1	2	3	4	5	6	7	8	9	10
a										
b										
c										
d										
e										
g										
h										
i										
j										
k										
l										
m										
n										
o										

Table 3. Relationships of test runs between matrix size plus connectivity and condition/s. Note: 14 test runs, 10 size types, and four starting positions make a total of 560 tests.

Although the relationships between affinity and domain in terms of emergence, stability, and possible deconstruction can be ascertained within the expected patterns across the range of matrix sizes and input formats, the behavior by the program also reveals some outcomes which every now and then fall outside the range due to small differences in the integers.

The sets of test formats seek to address the problem of identifying the significance of the test results within the probabilities expected from the conditions, while at the same time allowing for stray values. Despite the complexity of the processing space and the immense number of calculations the latter are formal and produce the same result provided exactly the same parameters have been used (on the other hand, the slightest difference in the environmental conditions can affect the result, even running the program in another subdirectory on one's hard disk).

Table 4 shows the type of comparisons between test runs.

Test run comparison		Conditions compared
1	a - c	none - block factor
2	a - d	none - position cycling
3	a - e	none - position cycling, block factor = 4
4	a - g	none - position cycling, approval factor = 4
5	a - h	none - position cycling, block factor = 4, approval factor = 4
6	a - i	none - teach mode (3 cycles)
7	a - j	none - teach mode (1 cycle)
8	a - k	none - block factor = 4, teach mode (3 cycles)
9	a - l	none - position cycling, teach mode (3 cycles)
10	a - m	none - position cycling, block factor = 4, teach mode (3 cycles)
11	a - n	none - position cycling, approval factor = 4, teach mode (3 cycles)
12	a - o	none - position cycling, block factor = 4, approval factor = 4, teach mode (3 cycles)
13	c - k	block factor = 4 - + teach mode (3 cycles)
14	d - e	position cycling - + block factor = 4
15	d - g	position cycling - + approval factor = 4
16	d - h	position cycling - + block factor = 4, approval factor = 4
17	d - l	position cycling - + teach mode (3 cycles)
18	d - m	position cycling - + block factor = 4, teach mode (3 cycles)
19	e - h	position cycling, block factor = 4 - + approval factor = 4
20	e - m	position cycling, block factor = 4 - + teach mode (3 cycles)
21	g - n	position cycling, approval factor = 4 - + teach mode (3 cycles)
22	h - o	position cycling, block factor = 4, approval factor = 4 - + teach mode (3 cycles)
23	i - j	teach mode (3 cycles) - teach mode (1 cycle)

Table 4. Juxtapositions of test runs in terms of respective conditions. Note: '+' denotes conditions of first test run are repeated.

When it comes to some results amongst the entire range of possible results therefore the question is not how to project from a small sample space to a population at large, but one of assembling a sufficient number of factors and juxtaposing them in a systematic fashion, all within a categorically defined and finite environment. The types of conditions relate to the essential propensity of chaotic processes within a structure such as the matrix with its node matrices and connection trees: the emergence of attractor-type behavior, the increase as well as decrease of the stability of domains through repeated processing, and the possibility of several inputs being affinitive with each other and so enhancing the significance of a domain (which in turn affects its neighbors under the same auspices).

For the purpose of analysis the kind of results are, in order of importance,

1. total number of cycles needed for the worm to reach its food with each starting position (a **food cycle**);
2. mean distance to the food during a food cycle;
3. the path by the worm during a food cycle recorded by the program, checked for a significant proximity at an earlier stage during this cycle.

3.3 Test results

3.3.1. Cycle numbers

First the total cycle numbers required to find the food were compared in terms of test run type (Fig. 4).

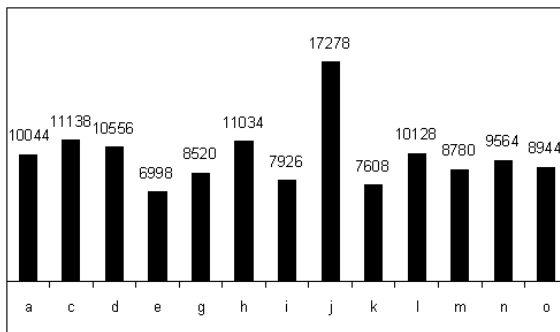


Fig. 4. Test runs with their respective total cycles, grouped by type.

With ‘a’ as baseline, it can be seen that test runs (e, g, i, k, m, n, o) fared better, whereas test runs (c, d, h, j, l) fared worse. In other words, seven conditions resulted in less cycles needed and under five other conditions it took more cycles.

Repeated processing overall (c), continuing to the next starting position (d), repeated processing after a lesser distance but with position cycling and repeated processing overall (h), only one cycle in teach mode (j), and position cycling and three cycles in teach mode (l), do not improve the performance in terms of the total number of cycles during the entire test run.

Repeated processing overall means the results from the previous cycle are disseminated through the matrix nodes regardless how close or how distant to the center the worm has come. Every current phase state is reinforced throughout the relevant domains, for better or worse.

Continuing to the next starting position simply makes the existing phase states available to the next configuration, no matter what the potential of those states may have been.

Repeated processing after a lesser distance but including the previous conditions allows the latter to exert their influence and overshadow any benefits a ‘reward’ could entail.

One cycle in teach mode is clearly not enough, and three cycles in teach mode show some improvement but are again neutralized by the continuation to another starting position by which time anything learned from the teach cycles has been ‘forgotten’.

On the other hand, repeated processing during position cycling does reinforce existing phase states even when the starting position has changed (e). Using repeated processing selectively during position cycling improves the outcome (g), so does a teach mode only applied three times (i), and the same goes for a three-cycle teach mode together with repeated processing overall (k). Using repeated processing overall and teach mode three times under position cycling does improve the performance somewhat but to a lesser degree given the potentially negative influence of the block factor (m). An improvement but again to a lesser degree is achieved using a three-cycle teach mode with selective repeated processing under position cycling (n), but adding overall repeated processing to the previous set reduces the cycle numbers further (o).

The figures so far represent the total cycle numbers for each food cycle under the given conditions. The iterative processes entail the possibility of radically different output values as the input is incrementally changed by the feedback loop. Fig. 5 gives some idea how scattered the individual cycle values can be when the results are broken down by test runs and starting positions vs matrix sizes.

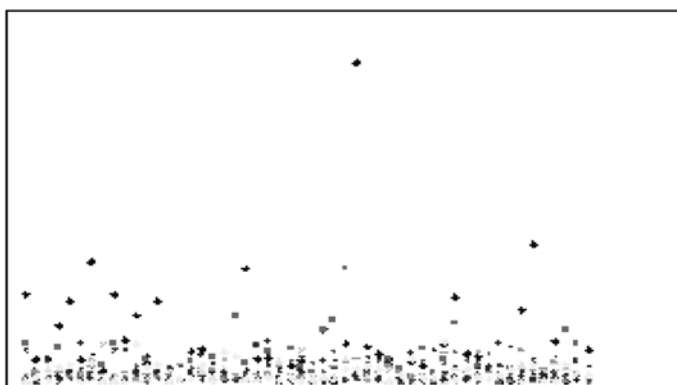


Fig. 5. Distribution across the range of achieved total cycle numbers for each of the four starting positions and matrix sizes in all the test runs.

To gain a more accurate view of the performance under the sets of conditions, firstly the mean of total cycle numbers in one test run was used as reference in relation to the best and worst outcome in another test run and the improvement and/or deterioration was expressed in terms of a downward factor and/or upward factor respectively in relation to the mean. Secondly the same mean was compared with the mean

of the better results and the mean of the worse results in the other test run, and again the downward and upward factor was calculated. The combinations of test runs are as listed in Table 4.

Table 5 contains the these outcomes.

Test run combination	Mean in 1st test run	Best result in 2nd (%)	Downward factor from 1st	Worst result in 2nd (%)	Upward factor from 1st	Mean of better results in 2nd	Downward factor from 1st	Mean of worse results in 2nd	Upward factor from 1st
a - c	251.10	1.59	62.78	919.16	9.19	188.10	1.33	362.99	1.45
a - d	251.10	1.59	62.78	672.24	6.72	89.19	2.82	352.67	1.40
a - e	251.10	1.59	62.78	618.08	6.18	78.08	3.22	133.53	0.53
a - g	251.10	0.80	125.55	497.01	4.97	49.67	5.06	288.88	1.15
a - h	251.10	2.39	41.85	867.38	8.67	68.90	3.64	247.20	0.98
a - i	251.10	0.80	125.55	403.82	4.04	139.84	1.80	346.98	1.38
a - j	251.10	2.39	41.85	2442.05	24.42	93.46	2.69	631.58	2.52
a - k	251.10	0.80	125.55	214.26	2.14	50.27	4.99	217.70	0.87
a - l	251.10	0.80	125.55	652.33	6.52	21.23	11.83	319.95	1.27
a - m	251.10	1.59	62.78	296.30	2.96	99.33	2.53	277.73	1.11
a - n	251.10	0.80	125.55	1050.58	10.51	50.43	4.98	326.50	1.30
a - o	251.10	0.80	125.55	394.27	3.94	91.36	2.75	208.55	0.83
c - k	278.45	0.72	139.23	193.21	1.93	105.79	2.63	277.02	0.99
d - e	263.90	1.52	65.98	588.10	5.88	215.04	1.23	227.96	0.86
d - g	263.90	0.76	131.95	472.91	4.73	184.63	1.43	251.00	0.95
d - h	263.90	2.27	43.98	825.31	8.25	100.60	2.62	242.08	0.92
d - l	263.90	0.76	131.95	620.69	6.21	139.70	1.89	417.90	1.58
d - m	263.90	1.52	65.98	281.92	2.82	91.24	2.89	203.40	0.77
e - h	174.95	3.43	29.16	1244.93	12.45	113.46	1.54	443.10	2.53
e - m	174.95	2.29	43.74	425.26	4.25	115.13	1.52	283.01	1.62
g - n	213.00	0.94	106.50	1238.50	12.38	73.94	2.88	458.08	2.15
h - o	219.50	0.91	109.75	451.03	4.51	118.60	1.85	384.88	1.75
i - j	198.15	3.03	33.03	3094.63	30.95	103.54	1.91	564.12	2.85

Table 5. Test run comparisons of cycle numbers by scale of improvement and deterioration in terms of best and worst result, and by scale of improvement and deterioration in terms of means of better and worse results.

In all the 23 juxtapositions the best results were on a higher scale than the worst results, and only on four occasions the means of the worse results were on a higher scale than the means of the better results (in a-c, e-h, e-m, i-j). Note that these combinations tally with the relativities apparent in Fig. 4.

3.3.2. Mean distance

As Fig. 4 indicates, test runs c, d, h, j, l showed worse overall total cycle numbers than test run a. The second-tier evaluation involves comparing the mean distances reached by the worm during the former with the mean distances under the latter.

In Table 6 these values are listed.

Test run	Sum total of mean distances
a	3383
c	3375
d	3296
h	3413
j	3481
l	3228

Table 6. Comparison of sum totals of mean distances.

In test runs c, d, and l the total mean distances were less than the reference, and in only two cases they were larger (h, j).

Although the cycle numbers themselves had been higher, in three cases out of five the conditions managed to keep the worm closer to the center overall.

This should be understood in conjunction with possible further conditions provided to enable the ‘higher view’ as mentioned before. Shorter mean distances should make it easier to reach the target under additional prompts coming from phase states that have been induced through other input.

3.3.3. Worm paths

In the third-tier evaluation the worm’s paths in test runs h and j were checked to see whether its head came close to the food at some earlier stage in the food cycle, overshot the disc or veered away from it and so continued to move around the cage. Fig. 6 shows the results for test run h with position cycling, block factor = 4, and approval factor = 4.

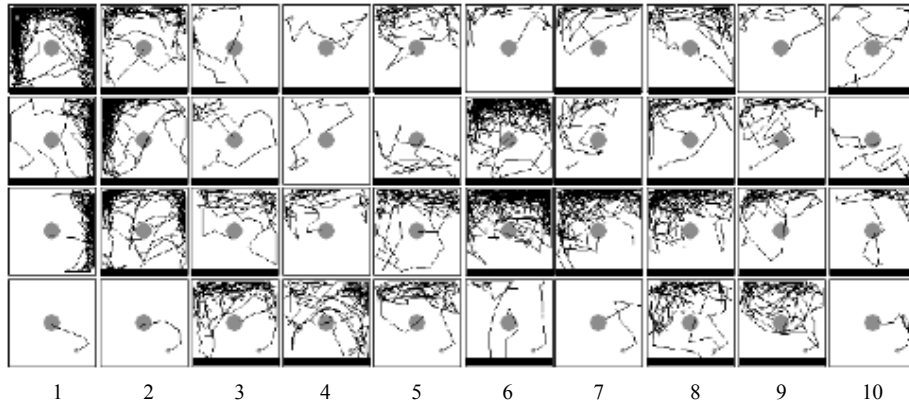


Fig. 6. Worm paths in test run h. Starting positions 1 to 4 arranged in rows and indicated by a small dot, matrix sizes 1 to 10. Positive results are marked by a bar. More or less straight hits are not counted here.

No intermittent proximity occurred at starting position 1 (for matrix sizes 6, 9), at starting position 2 (4), at starting position 3 (1, 4, 5), and at starting position 4 (1, 2, 5, 7, 10). Out of 40 tests 29 featured a close proximity at some time before the food was reached.

Fig. 7 illustrates the paths in test run j under teach mode conducted for one cycle. There was no intermittent proximity at starting position 1 (matrix sizes 1, 3, 4, 6, 10), at starting position 2 (4, 5, 6), at starting position 3 (2, 7, 8), and at starting position 4 (2, 4, 8, 9). Therefore out of 40 tests 25 produced a close proximity during their food cycles.

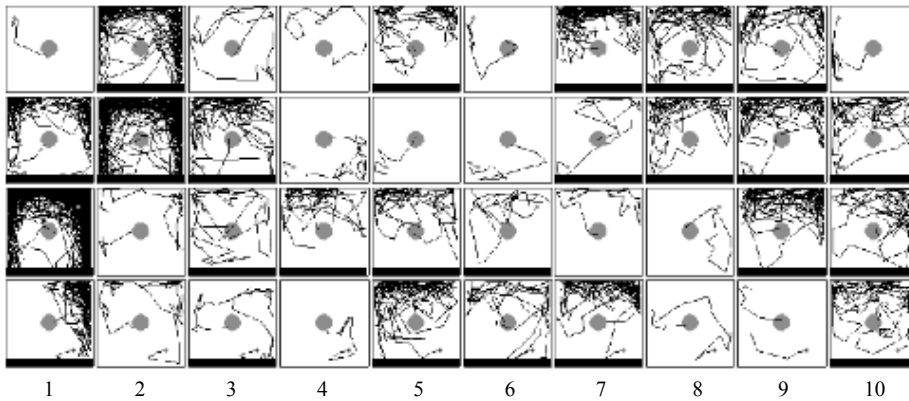


Fig. 7. Worm paths in test run j. Starting positions 1 to 4 arranged in rows and indicated by a small dot, matrix sizes 1 to 10. Positive results are marked by a bar. More or less straight hits are not counted here.

Note that a more or less direct path to the food does not count here, although a low-cycle hit could be regarded as a success. The formality of the test requires the definition of path evaluations be adhered to. Including the hits in test run h after all would add (1, 2, 7, 10) for starting position 4, making the win/lose ratio 33:7. For test run j one would have to add (1, 6) for starting position 1, (5) for position 2, and (4) for position 4. Out of 40 tests 29 would have been successful.

Similar considerations as offered in the context of mean distances apply. Although the overall proximity to the target had been less promising here, additional conditions (circumscribed as the ‘higher view’) may well prompt the head to continue the short distance to the target in these cases.

3.3.4. Further comments

When examining the program and its evaluation questions could arise as to the choice of object and the set of priorities for the test criteria. Both relate to the chaotic nature of such a system, the inherent unpredictability and sensitivity to initial conditions having been amply described [11].

Firstly, from a conceptual point of view it is easier to relate to something which has some semblance to a biological organism, as pared down as a few discs moving on a two-dimensional screen may be. More importantly the object must be simple enough to permit the effects of a single condition becoming apparent for the purpose of testing its functional environment. The degree of interdependence of node behavior with its resultant complexity obscures the respective effects of even a few parameters very quickly. Although the actual outcome at each operation remains unpredictable, a series of events under consistent parameters does allow trends to emerge provided there are any. At any moment the outcome (positive or negative) is an instantiated form of the potential of the previous phase state, in other words its latency. The latency can be identified in terms of the processing elements their functional space contains [29].

The last point touches on the role of patterns. If a system is fed data and produces seemingly random output, the individual results could be described as a set of random outputs. However, if it is possible to rearrange the members of such a set in a certain order so that a pattern becomes now visible (note that each test stands on its own), the previous definition does not hold. Further, if the inputs can be rearranged so that the output resembles an ordered set straight away, the correct interpretation suggests itself immediately. In the case of an iterative system relying on feedback loops the reference can be extended to include its environment, yet another system but at a wider scale. From an array of values the issue has moved to their applicability in a certain context; that is to say, a localized randomness is redefined by a utility from the outside. This relationship between sequences and wider frameworks has been discussed at length by Knuth [20]. Thus the concept of changing conditions applies to a wider processing space and requires an appropriate logic, a significant issue addressed in the context of nonmonotonic reasoning [1]. The size of the overall processing space must be sufficient for patterns to be ascertainable; it cannot be smaller and it does not need to be larger.

Secondly, the possibility of unpredictable values in the result does not necessarily unseat a pattern provided such values are insignificant enough in relation to the trend. Just as a stray value may drive the worm away from the target, a similar outcome may bring it closer. The actual position of the worm’s head in relation to the food at any given time is therefore of lesser importance than the average proximity during an entire run.

At the same time it needs to be shown that the system is capable of achieving a result that is unambiguous under the given conditions. Regardless of any ameliorating factors such as general proximity or an ultimately unsuccessful movement towards the disc, in the end the head has to land where it should. Hence total cycle numbers are given the highest priority.

As much as the phase states are comparable to cognitive events observed in human behavior (alluded to before), the question of conscious vs subconscious dynamics has not been entered into. It is debatable whether such a distinction, teased away from the biological contingencies in humans (and possibly non-humans), serves the ultimate purpose of constructing an artificial cognitive system. Arguments for and against go beyond the scope of this paper. In terms of the program’s scalability however it should be pointed out that a major difficulty in the design of intelligent systems has been the granularity of rule sets underpinning its processes.

Based on the famous ‘halting problem’ it concerns the ability of a system to recognise the limitations of its subsystems [21]. The coarser the rules, the more processing-intensive a specific event will be for the system; the more finely-grained the rules, the less processing is required for a particular event but the set

features far more members and/or greater complexity. In applications where a robot needs to adapt itself to a changing terrain for example, the relevant rules are derived from more general sets that encompass the type of surface to be dealt with [7] and work from the top downwards. The modifications are self-configurable, but their templates are not. Again on the subject of navigation, this time involving a robot to reach a set target, the set of algorithms allows the original fuzzy behavior to be defuzzified in order for the robot to close in [25]. The relevant rule sets come from the system's designers.

A recent attempt to get closer to the biological version is an interpretation of minimal conscious functions affected through several modules representative of certain dynamics, all linked to an overall control [9]. The model has not been realized yet, but the dynamical distinctions may well lead to some implementation. Note however that those modules have been derived from a summary conceptualization of cognitive processes and once again represent a macro view that is implemented from the top down. We are led back to the traditional hypotheses along the functionalist road.

Another approach, this time focusing on emotion, led to the modeling of problem-solving processes via the ACT-R architecture and its sub-symbolic capabilities by Belavkin [5]. The symbols, representing declarative knowledge units and production rules, function under set parameters some of which seek to model the influence of emotion during decision making. Although the results in this case corresponded to generally acknowledged optimization methods, the parameters themselves are more representative of the model's framework than a wholly integrated system where emotive and cognitive factors exist in tandem and operate from the bottom up.

To preserve the obvious variance in cognitive phenomena and hence their flexibility a way had to be found to deconstruct the rule sets down to their fundamental dynamical entities. Chaos-type attractors allowing affinity relationships to emerge answer that demand. In any case, the problematic transposition of macroscopic behavior—however implemented—into a meaningful dynamical detail has already been pointed out by Eliasmith [13].

4. Future Development

Clearly those 210 inner matrix nodes are too few by far for anything much to happen. Even the lowly *Aplysia* slug has up to 250,000 neurons in its nervous system, and the brain of an octopus contains 300,000 [10]. One of the first tasks is therefore to provide the hardware for many millions of nodes, keeping in mind the considerably much higher number of overall node elements as a consequence.

Conventional applications are already pushing the design and space envelopes of semiconductors. The balance between scaling and functionality within a chip and the synchronization of clock speeds are being addressed [3].

The greater volume would allow partitioning of the entire space so that particular functions associated with their relevant inputs reserve their own processing space and hence their localized domains. Whether such compartmentalization is achieved through physical barriers (*eg.* routers, buses) or through a functional one (*eg.* unique process cycles, integer range of the element values) would be subject to experimentation. Regarding the latter, just as it is possible to tune a conventional artificial neural network, here the degree of affinity amongst clusters could be used in a similar fashion (if nothing else it would produce interesting results!).

Another parameter to be explored is the degree of connectedness amongst the nodes. Connectivity has proven to be a deciding factor in the cognitive ability of organisms, including humans [14]. The same relationship can be observed in the current system. Tests on the basic computer program have shown that higher connectivity achieves a more detailed output [29].

The code would have to be transposed into a distributive one. Linear processing not only wastes time, the sequence of operations does not necessarily follow the functional requirements of affinitive input streams. Having clusters of domains processing their own data in tandem allows the phase states to become more specialized and therefore more meaningful.

One promising avenue is the development of the Cell chip [8], an architecture incorporating parallel processing on single units which can be scaled up in number.

The range of in- and outputs should be widened to accommodate various sensors providing information to the system (recall the 'higher view' touched upon earlier), and output mechanisms in the form of visual and auditory re-representations as well as mechanical devices would have their obvious uses. Naturally, the processing space of the system would need to be adequate.

In the real world a system of this kind should be able to avail itself of a much larger series of inputs as well as outputs. Particularly in the case of inputs such a wider range would provide the conceptual convergence we expect from a biological organism. That is to say, visual processing for example can confirm if it is getting closer to some target, olfactory or tactile sensations support the interpretation of the scenario, and especially in more complex brains the contribution of memory in the form of learned experience enhances the degree of expectation as their host goes through the movements.

In the biological version emotion plays a crucial part in terms of such confirmations. As has been shown by Belavkin and others, the conventional approach has been to incorporate that factor by having it represented as a parameter under the computational framework. Considering the bottom-up emergence of phase states in a chaotic, attractor-driven system however, the chemical propensity of emotion in the organic context can also be seen as a feature which operates interdependently with cognitive processes. In other words, emotion can be viewed as a primitive from of thinking. Its chemical nature in organisms should be transposed into its electrical equivalent suitable for machines.

In the case of humans the sheer plethora of participating detail in cognitive processes usually requires a considerable deconstruction for it to be identified; the same goes for the artificial version. In the end it becomes a matter of scale and utility, the implementations of which are certainly less problematic than dealing with living beings.

5. Artificial minds and Society

The introductory remarks referred to the conceptualization behind the making of an object. However, since the initial idea is an abstract still awaiting its implementation, the eventual artifact is now removed from the possibly idealizing nature of the mental image and is situated in the real. That is to say it functions according to the sum total of its characteristics embedded within the wider functional space of its environment, however comprehensively or otherwise these may have been understood.

Even relatively simple inventions caused unexpected modifications in their surrounds because the contemporary mindset was unable to exactly mirror reality at large; just consider the monumental changes brought about by the printing press, unforeseen at the time.

A machine that simulates the mind not only entails a significant utility for the user, its very system suffers from the ambiguity of incomplete understanding in common environments. Considering society as a system, there may be a point beyond which its resources are insufficient to administer a self-created entity because the latter's complexity is out of reach for the conceptual framework. Under the present auspices a system is successful if its constituents are able to interact with each other in a congruent manner. For instance, the existence of a few knowledgeable individuals becomes irrelevant unless they can communicate their expertise to the rest.

Science fiction prefers dramatic events but sometimes a more subtle scenario is presented, such as the story of some existentialist robot sentinel which kills anyone who is blind to the hidden meanings of its questions [26]. Assumed rationality meant death. Although this is just a story, its interpretation of the human mindset vs the artificial one is interesting in itself. No matter how sophisticated future machines and/or robots may become, their inherent dynamics can only then be interpreted adequately if there is a conceptual tool set capable of addressing their processes. In turn this capacity must be an element of a superset which is society. To put this another way, the functional processes existent in society need to be congruent with respect to their subsets—a feedback loop in itself that is a reflection of its counterpart on the smaller scale as presented in these pages.

In an article on transhumanist values Bostrom ascribes an ever increasing functional space available to animals, humans, transhumans, and posthumans [6]. There the extent spans all activity, from daily life to health, to psychological and moral projections forming the interactive medium. The view presented entails a similarly growing potential in those areas for the participants, but the potential for diversity is also directly proportional to volume. Yet the emergence of domains answers to the relative degrees of affinity within their subsets rather than any judgmental preference decided upon by human observers.

The discrepancy between more complex domains and those of a lesser nature are equally visible on the still larger scale of cultures. The contextual detail of present-day politics already suggested a profound conflict involving civilizations [19], but any new “world order” would still entail the familiar dynamics seen much closer to home. The only thing that would have changed is the content, the type of people carrying on the processes.

The above would usually be found in more philosophical texts rather than a scientific journal. However, over the recent decades science has advanced into realms whose importance to the wider society makes it imperative for researchers to dwell on the implications to a degree hitherto overlooked. The beneficial usage of artificial minds depends on a synchronicity of systems—ours and theirs.

6. Conclusion

The worm program demonstrates the functional aspects of cognitive dynamics as they can be identified at the higher scale of individuals, groups, and society. Although its configuration is still small various conditions can be enacted that elicit a response in line with those dynamics. No outside rule set apart from the attractor algorithm governs the internal processes. Any result is a function of emergent phase states.

These processes, being part of the model’s framework, can be confirmed through their counterparts in human affairs, whether in general society, politics, or scientific representations (see Appendix, Parallels). They allow a formal analysis of events which influence human civilization, such as the Iraq war, climate change, or peak oil. The entirely natural and replicable phenomenon of emergence into higher complexity would also be one of the most powerful arguments against movements such as Intelligent Design for example. Important questions not only about the possibility of artificial minds but our own situatedness in a complex and ever-changing world can be raised. How the machine fares in the end will become a measure of our understanding of the Self.

Appendix

Locations of supplementary material:

- i. How the Mind Works: the Principle Dynamics in the Bio- and Non-bio Version
<http://www.otoom.net/Downloads.htm> > IPSI-2005 Venice paper
- ii. Basic computer program
<http://www.otoom.net/Downloads.htm> > OtoomCM program
- iii. Worm program including source code
<http://www.otoom.net/Downloads.htm> > OWorm program
- iv. Complete test results of OWorm
<http://www.otoom.net/Downloads.htm> > OWorm tests
- v. Sample animation of worm movement
<http://www.otoom.net/Downloads.htm> > OWorm animation
- vi. Sample application of the mind model on a large scale
<http://www.otoom.net/Downloads.htm> > The social Europe: a formal view
- vii. Events since 2003 confirming the mind model
<http://www.otoom.net/Parallels.htm>

References

- [1] G. Antoniou, *Nonmonotonic Reasoning*, The MIT Press, Cambridge, 1997, p. 271.
- [2] A. P. Atkinson, M. Wheeler, *Evolutionary Psychology's Grain Problem and the Cognitive Neuroscience of Reasoning*, in: *Evolution and the psychology of thinking: The debate*, Hove: Psychology Press, 2003.
- [3] A. E. Barun, S. Hillenius, *Semiconductor International* 8/1 (2006).
- [4] M. F. Bear, B. W. Connors, M. A. Paradiso, *Neuroscience - Exploring the Brain*, Lippincott Williams & Wilkins, Baltimore, USA, 2001, pp. 25, 515, 800.
- [5] R. V. Belavkin, *The Role of Emotion in Problem Solving*, in: *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing*, 2001, pp. 49-57.
- [6] N. Bostrom, *Transhumanist Values*, <http://www.nickbostrom.com/ethics/values.html>, 2006.
- [7] Z. Butler and D. Rus, "Distributed Locomotion Algorithms for Self-Reconfigurable Robots Operating on Rough Terrain", in: *Int'l Conf. on Computational Intelligence in Robotics and Automation (CIRA) '03*, July 2003.
- [8] The Cell project at IBM Research, <http://www.research.ibm.com/cell/>, 2006.
- [9] N. Charkaoui, *A Computational Model of Minimal Consciousness Functions*, *Transactions on Engineering, Computing and Technology* V9 November (2005).
- [10] E. H. Chudler, *Brain Facts and Figures*, <http://faculty.washington.edu/chudler/facts.html>, 2006.
- [11] K. Clayton, *Basic Concepts in Nonlinear Dynamics and Chaos*, in: *A Workshop presented at the Society for Chaos Theory in Psychology and the Life Sciences meeting*, July 31, 1997 at Marquette University, Milwaukee, Wisconsin, 1997 .
- [12] P. Coiffet, *An Introduction to Bio-Inspired Robot Design*, *International Journal of Humanoid Robotics* Vol. 2 No. 3 (2005) 229-276.
- [13] C. Eliasmith, *The third contender: A critical examination of the dynamicist theory of cognition*, *Philosophical Psychology* Vol. 9 No. 4 (1996) 441-463.
- [14] G. N. Elston, *Cortex, Cognition and the Cell: New Insights into the Pyramidal Neurons and Prefrontal Function*, *Cerebral Cortex*, Oxford University Press V 13 N 11 (2003) 1124-1138.
- [15] T. J. van Gelder, *The dynamical hypothesis in cognitive science*, *Behavioral and Brain Sciences* 21 (1998) 615-628.
- [16] M. Giunti, *Dynamical Models of Cognition*, in: I. van Gelder, T. and R. Port, (Eds.), *Mind as motion: Explorations in the dynamics of cognition*, Cambridge MA: The MIT Press, 1995, pp. 549-571.
- [17] S. Harnad, *Minds, Machines and Searle 2: What's Right and Wrong About the Chinese Room Argument*, in: M. Bishop & J. Preston, (Eds.), *Essays on Searle's Chinese Room Argument*, Oxford University Press, 2001.
- [18] F. W. Howell, J. Dyhrfeld-Johnsen, R. Maex, N. Goddard, E. de Schutter, *A large scale model of the cerebellar cortex using PGENESIS*, <http://citeseer.ist.psu.edu/howell99large.html>, 1999.
- [19] S. P. Huntington, *The Clash of Civilizations and the Remaking of World Order*, Touchstone Books, London, 1998.

- [20] D. E. Knuth, *The Art of Computer Programming*, Addison-Wesley, Sydney, 1998, p. 149.
- [21] Epp, S.S., *Discrete Mathematics with Applications*, Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 1995, p. 270.
- [22] M. Lockwood, "The Grain Problem", in: *Objections To Physicalism*, Clarendon Press, Oxford, 1993.
- [23] H. Markram, *The Blue Brain Project: Simulating Mammalian Brains*, <http://bluebrainproject.epfl.ch>, 2004.
- [24] R. F. Port, *Dynamical Systems Hypothesis in Cognitive Science*, draft entry for *Encyclopedia of Cognitive Science*, MacMillan Reference Ltd, London, 2000.
- [25] A. Saffiotti, *Fuzzy Logic in Autonomous Robot Navigation*, <http://www.aass.oru.se/Agora/FLAR/HFC/home.html>.
- [26] R. Silverberg, *The Sixth Palace*, Deep Space, Corgi, Great Britain, 1977, p. 105.
- [27] M. Wurzinger, *How the Mind Works: the Principle Dynamics in the Bio- and Non-bio Version*, IPSI-2005 Venice Conference, 2005.
- [28] M. Wurzinger, *How the Mind Works: the Principle Dynamics in the Bio- and Non-bio Version*, IPSI-2005 Venice Conference, 2005, p. 13.
- [29] M. Wurzinger, *How the Mind Works: the Principle Dynamics in the Bio- and Non-bio Version*, IPSI-2005 Venice Conference, 2005, p. 14.
- [30] M. Wurzinger, *On the Origin of Mind*, Brisbane, 2003.

Vitae

Martin Wurzinger received his Bachelor of Information Technology degree in 1999 and his Bachelor of Information Technology Honours degree in 2000 from Griffith University, Australia, both with a focus on Artificial Intelligence. From 2000 until 2003 he worked on the text "On the origin of Mind" as well as developing a prototype of an artificial mind. In 2005 he completed an extension of the computer program, driving a point in 3D space. In 2006 he completed a further adaptation, enabling a flexible object to reach a target under various conditions.

Martin Wurzinger was elected to become a member of the Golden Key National Honours Society during his university studies. In 2005 he was invited to give a seminar on the computer simulation at the Vision, Touch & Hearing Research Centre, School of Biomedical Sciences, University of Queensland, Australia. He was invited to attend the IPSI-2005 Venice conference to which he submitted a paper. In 2006 he was invited to participate in upcoming Dialogue Workshops held by the European Commission Directorate-General, Directorate K - Social sciences and humanities.