Co-evolving Body and Brain in Autonomous Agents using a Developmental Model

Frank Dellaert [1] and Randall D. Beer [1,2]

Department of Computer Engineering and Science [1]
Department of Biology [2]
Case Western Reserve University
Cleveland, OH 44106

e-mail: dellaert@alpha.ces.cwru.edu, beer@alpha.ces.cwru.edu

August 1994

Abstract

We present an extension to the developmental model discussed in (Dellaert and Beer 1994) to be able to co-evolve nervous systems with the bodies of autonomous agents. To this aim we re-introduce this basic model, and then examine the issues that have to be resolved when extending this model towards neural development. To test the extended model, we have handcoded a genome for a functional agent that is able to display a simple behavior related to chemotaxis. We also show that the model can be used in conjunction with genetic algorithms to yield better performing agents.

Technical Report

CES-94-16

Department of Computer Engineering and Science
Case Western Reserve University
Cleveland, OH 44106

Technical Report

Co-evolving Body and Brain in Autonomous Agents using a Developmental Model

Frank Dellaert [1] and Randall D. Beer [1,2]

Department of Computer Engineering and Science [1]
Department of Biology [2]
Case Western Reserve University
Cleveland, OH 44106

e-mail: dellaert@alpha.ces.cwru.edu, beer@alpha.ces.cwru.edu

Abstract

We present an extension to the developmental model discussed in (Dellaert and Beer 1994) to be able to co-evolve nervous systems with the bodies of autonomous agents. To this aim we re-introduce this basic model, and then examine the issues that have to be resolved when extending this model towards neural development. To test the extended model, we have handcoded a genome for a functional agent that is able to display a simple behavior related to chemotaxis. We also show that the model can be used in conjunction with genetic algorithms to yield better performing agents.

1. Introduction

We are interested in using a developmental model to synthesize autonomous agents. In (Dellaert and Beer 1994) a simplified, evolvable and yet biologically defensible model of the developmental process is presented, explicitly designed with this goal in mind. However, that model only addresses the development of emerging patterns of different 'cell types', represented by square elements of different color.

In this paper we present how we extended this model to be able to co-evolve an autonomous agent 'body' with it's controlling 'nervous system'. Using this extended model, we can synthesize fully functional agents, complete with sensors, actuators and a nervous system to control them.

To accomplish this, we had to extend the model in two ways: (i) the abstract cells used before should be able to take on a functional role in a simulated autonomous agent, and (ii) the model should be able to account for the development of functional 'nerve cells' that interact with the other differentiated cells in the organism and bestow the whole with a behavior.

Please bear in mind that the neuronal development model we constructed draws on biology for inspiration, but it is not our aim to make a realistic model of biological development. It would be nice were it to share some fundamental characteristics with the actual process, but simplicity and the ability to be integrated with the basic developmental model of the earlier paper were our primary goals.

In section 2 we re-introduce the basic developmental model, since it is to be the foundation for the extended model. In section 3 we then examine the issues that have to be resolved when building a simplified model of neural development. After having done so, we present the extended model in section 4. Then, in section 5, we show how we tested it by handcoding an agent that can successfully perform a simple task related to chemotaxis, i.e. a Braitenberg hate-vehicle. This implements a task that is both simple and yet requires a fully functional network. Finally, in section 6, we examine whether the extended model can also be evolved, and we round off with conclusions in the final section.

2. The Basic Developmental Model

In this section a brief overview is given of the principal components found in the basic developmental model proposed in (Dellaert and Beer 1994). In this basic model, the developmental process unfolds simultaneously at three different levels: at the level of the organism, of the cell and at the molecular/genetic level.

2.1. The Genetic Regulatory Network

The genetic regulatory network is the first principal component of the model. The unfolding pattern of differential gene expression at the genome-level is modeled by a network of interacting genetic elements, in this case a Boolean network, as first pioneered in this context by (Kauffman 1969). Fig.1 illustrates one possible instance of the wiring of a Boolean network with three nodes each having two inputs. Not shown but crucial to the understanding of such networks are the Boolean functions that are implemented within each node. Each node can have a different function, corresponding to a different way in which it will react to its inputs. This function models the way genes and gene products influence each other's expression.

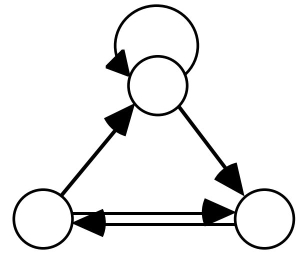

Fig.1. Example of a wiring configuration in a Boolean network with N=3 and K=2.

2.2. The Cellular Simulator

The second component consists of a very simple cellular simulator to model development at the cellular level. We have simulated the physical appearance of a cell by a simple, two-dimensional, square element that can divide in any of two directions, vertical or horizontal. See Fig.2 for an illustration.

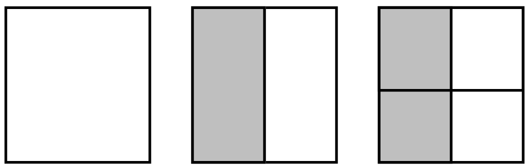

Fig.2. Zygote square dividing two times to yield 4 'cells', or squares

A simulated cell cycle (Fig. 3) consisting of two phases, interphase and mitosis, alternates cell division with the updating of the genetic regulatory network (of which each cell has a copy). The state of the network may be different in each cell and corresponds to the pattern of gene expression in that cell. This pattern in turn determines the cell type, represented in the simple model by a color. It will also determine whether a cell actually goes through cell division (mitosis) or through an alternative rest state.

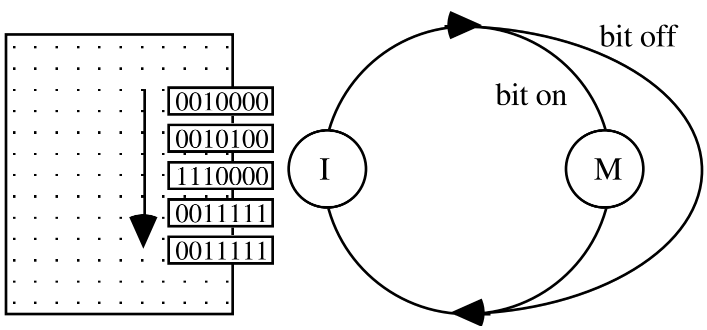

Fig.3. The cell cycles between interphase and mitosis. Mitosis is skipped if the 'dividing bit' is not set.

2.3. Organismal Level

Finally, the last aspect of the model covers all phenomena at the organismal level, i.e. intercellular communication and symmetry breaking.

In the simple model we break the symmetry at the time of the first 'cleavage' event by assuming the existence of a 'maternally' imposed asymmetry in the 'zygote' square, that can lead to different patterns of gene expression in the first two daughter-cells. A second spatial clue is introduced later in development by supplying the developing organism with the notion of a midline.

To implement intercellular communication a modified version of the Boolean network is used that allows for nodes in the regulatory network to connect to nodes in the networks of neighboring cells. This can then be interpreted as the existence of a cell-surface receptor, sensing the presence of specific chemical agents introduced by cells the environment. In a similar fashion a cell can find out whether it is exposed to the external environment (the space around the perimeter of the organism).

3. Modeling Neural Development

In this section we focus our attention in turn to all the major phenomena that play a role in neural development. No attempt is made to explain these aspects of development at length, but instead we would like to refer the reader to (Purves and Lichtman 1985), whose order of presenting these topics we will loosely follow here. We intend to examine what aspects are important to incorporate in the model and how we could do this.

3.1. Neuronal Differentiation

In addition to the primarily induction-like mechanisms of cell identity control of the earlier model, we will have to model cell-lineage events like asymmetric cell division. Any inspiration we draw from biology on neural development is mostly from simple invertebrates like the leech or the nematode C. Elegans. In these simpler organisms, cell lineage seems to play a much more important role than in more complex animals, thus the need to have a counterpart for it in our model.

3.2. Pattern Formation

Pattern formation mechanisms will be partially supported in the model. Partially, because with the simple mechanisms that we have at our disposal it is possible to have some amount of pattern generation, but generally the term covers mostly the more intricate patterns that come about by the workings of smooth gradients of diffusable morphogens, which are not included in the basic developmental model. The artificial organisms that we are capable of modeling right now, however, have a small number of cells (64 to 256) and it seems reasonable to start off with simply modeling a small number of neurons and their axon growth. Pattern formation of the more complex kind certainly plays a role in systems where a large number of elements is involved, e.g. the formation of the composite eye in Drosophila or cortex-patterning, but we think we can safely leave it out in a first approximation. One thing we might miss out on, however, are segmental structures, like ganglia in repeated body segments (although it is possible to have a segmented pattern in the current model, see below).

3.3. Migration of Neurons

Although neuronal precursor migration plays a key part in the neural development of many complex animals, we will not include it in our model. One reason is simply that our earlier developmental model does not allow for cell movement, because of the computational complexity that would introduce. Another is the fact that cell migration might not have such a large role in the neural development of many simple organisms, on which we would like to base the extension of our model toward nervous systems.

3.4. Axon Outgrowth

In contrast, the processes that are sent out by the neuron are crucial to the functioning of the nervous system even in the simplest animals, and we will have to make sure that outgrowth and subsequent navigation of neurites is adequately modeled. Indeed, what makes neurons so special is their ability to communicate with each other, in many cases via axonal processes that they send out during development, and that somehow manage to innervate targets in a remarkably robust way.

For axon-path finding we propose to model the displaying of cell adhesion molecules on the surface of the cells upon which the growth cone of the axon travels. The growing axon would then sense these CAM molecules and only grow where they are laid out, in the direction where it can find more. In (Purves and Lichtman 1985) a number of other possible (co-existing?) mechanisms are discussed: we chose the one that appeals most to us and decided to model only that in the first cut. Maybe this will limit the range of developmental patterns we will be able to develop, but on the other hand we will have a fairly good idea of why these other mechanisms could be important, rather than ending up with a melting pot of diverse mechanisms that complicates understanding.

Taking inspiration form lower invertebrate nervous systems, we will only consider axons that directly synapse on another cell's soma, without modeling dentritic trees.

3.5. Neuronal Death and Trophic Dependencies

We will include the notion of a trophic factor in the model, as this seems to be a major mechanism whereby the fan-in, fan-out properties of neurons (and other cells innervated by them) are regulated. Neuronal death can be modeled by simply letting the cell lose it's neural properties if it does not secure enough trophic factor within a certain time-frame.

3.6. Formation of Synapses

It seems reasonable then to use the modeled trophic factor simultaneously as a signal for growing axons to stop wandering and form synapses at the location where this trophic factor is available. It may not be the case in biology that this is mediated by the same molecule, but we do not want to make the model needlessly complicated. The amount of trophic factor available could also be used to specify the strength of the synaptic connection: one could draw the analogy with biology by imagining many synaptic boutons where there is a lot of trophic factor available, and conversely less where there is a lower concentration of trophic factor.

3.7. Nature of the Synaptic Interactions

We propose to make the nature of the synaptic interactions dependent on two variables: the type of neurotransmitter expressed in the pre-synaptic neuron, and the nature of the receptors expressed in the post-synaptic cell. Both these variables can be under genetic control. The receptors should be selective to a certain neurotransmitter only, so that innervated cells can selectively tune in to signals provided by a certain group of neurons only.

4. The Extended Model

To extend the model towards neural development, we first have to provide a framework to enable genetic elements to exert an influence within the cell. After having done so, we will discuss some general cellular functionalities that were also present in the basic developmental model but that we now have to fit into the new framework. Then we will discuss the implementation of the specific neuronal functionalities discussed in the previous section, and provide a more detailed description of one of the crucial extensions to the model, i.e. the behavioral model of a single growth cone.

4.1. Incorporating Functionality of Genetic Elements

Extending the model

The previous model did not allow for the state of a particular genetic element to have any consequences other than specifying the color of the cell. This simple color-model only necessitated interpreting certain bits in the state vector: the only 'active' role played by the genetic elements was (1) to regulate the state of other elements, and (2) to influence neighboring cells via the neighborhood vector (see Dellaert and Beer 1994). Now we will provide a framework to make 'cellular function' possible, e.g. sending out an axon or releasing a chemical agent.

To that end, we have implemented the Boolean networks using a new genome-cytoplasm simulation model. As the name suggests, there are two components to this model: the genome and the cytoplasm. we will discuss each of these here in somewhat more detail:

Cytoplasm

The simulated cytoplasm is a set of gene-products that is produced by the expression of some genes in the genome (see below). In the model every product is simulated by a number between 1 and 1000. The set is not ordered, and only supports the operations ADD, REMOVE and CONTAINS.

Genome

The simulated genome consists of a set of operons, mimicking the operation of a gene. Every operon specifies (1) a gene product that will be produced when the operon is active, (2) the other products that can regulate its expression and (3) a Boolean function that specifies how the expression is regulated. In Fig.4 this is schematically depicted and compared to a biological gene. A product will contribute a one as input to the Boolean function of another operon if it is present in the cytoplasm and zero otherwise.

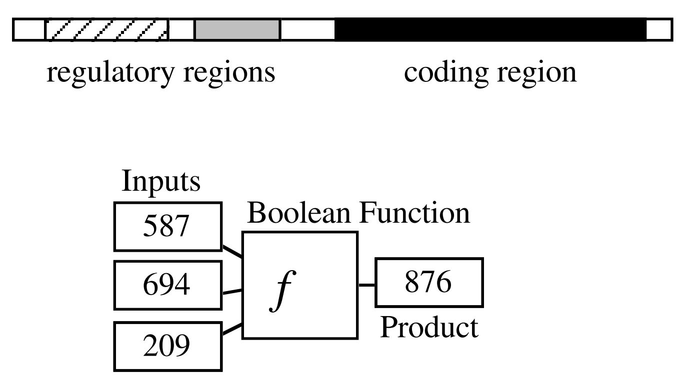

Fig.4. The 'operon' compared with a biological gene.

There are some obvious similarities but also some important differences with the straightforward Boolean network implementation used earlier. A similarity is that in fact an operon could be seen as the description of one node in the Boolean network. In contrast, in the new implementation each operon can be of different fan-in and fan-out. Also, since the genome is a set of operons, the number of operons is not constant nor constrained: by 'deletions' and 'insertions' modeled after those we find in biological evolution, the genome can shrink or grow.

The cell-cycle

A new phase is inserted to allow for the gene products to execute their function. The cell cycle is similar to the one used in the earlier model: an interphase period where the operons evaluate their Boolean functions and express their products is alternated with a mitosis phase in which the cell divides. However, before mitosis is entered, the cytoplasm is examined on gene-products with an active function and if so, these functions are executed. If you will, interphase now consists of a regulatory phase and a functional phase.

In the regulatory phase, every time the operon's Boolean function evaluates to true, its product is added to the cytoplasm, and, conversely, it is removed from it if the function evaluates to false. A difference with the earlier model here is quite crucial: the operons are updated synchronously but in an undetermined order, as they are inserted in the set in random order (although they are always picked from the set in the same order once the set is stable).

4.2. Non-neural functionalities

Here a summary is given of gene-products that implement basic developmental mechanisms (see Table.1), either extending the basic model or implementing mechanisms that were implicitly defined in the basic model but are now made explicit by associating them with certain gene-products.

```
tDividing=20,
tAsymmGene=5,tAsymm,
tMAsymmGene, tMAsymm,
tSymmGene=10, tSymm,
tA0=50,tA1,tA2,tA3,tA4,         // agents
tR0,tR1,tR2,tR3,tR4,            // receptors
tD0,tD1,tD2,tD3,tD4,            // messenger
tEnvSensor=30,tEnv,
tMidSensor=40,tMid,
```

Table.1. The non-neural gene products used in the model.

Simple regulatory genes

Simple regulatory genes code for gene-products that serve to regulate the expression of the other genes. They make up the vast majority of possible gene products.

Cell-Cycle

A special gene-product (tDividing) serves to signal whether a cell should go through mitosis or not. By making use of this element, the genome can specify regions where great detail is needed in contrast with other regions where the cells can be coarser. This may play not such a large role in the square geometry that we work with right now, but might be an important regulator of form in future extensions of the model where more complex geometry's could be used than the present one.

Asymmetric Cell division

Some genes can set up an asymmetric cell division: this is mimicked by letting its gene-product be distributed to only one daughter cell at the time of division. It is this kind of gene that will lie at the basis of much of the cell-lineage events inspired by C.Elegans development. One special gene is reserved for a maternal-effect like event: the initial asymmetric distribution of some factor at the time of the first cleavage.

Receptors

Some genes (tRx) code for receptors: these will look for a specific gene-product in the cytoplasm of neighboring cells (tAx gene products). When that is detected, a 'messenger' product (tDx) is inserted in the cytoplasm, that can then regulate the expression of other genes. Note that this was done in the basic model by letting Boolean networks influence each other across cell boundaries.

Sensing external stimuli

Very similar to receptors are the environment and midline sensor genes (tEnvSensor and tMidSensor): when these are expressed, they will insert messenger-products (tEnv and tMid, resp.) into the cytoplasm to signal that the cell lies at the perimeter of the organism or on the midline, respectively.

4.3. Neural phenomena

The neural developmental events depend on the coordinated expression of a number of neural gene products (see Table.2), which we will discuss now.

```
tC0=100,tC1,tC2,tC3,tC4,        // Cams
tAxon=90,                       // Axon outgrowth
tTrophic=80,                    // Trophic factor
tN0=150,tN1,tN2,tN3,tN4,        // NeuroXmitter
tI0=200,tI1,tI2,tI3,tI4,        // inh post-synaptic receptor
tE0=250,tE1,tE2,tE3,tE4         // exc post-synaptic receptor
```

Table.2. The neural gene products used in the model.

Cell Adhesion Molecules

Specific gene products represent the cellular adhesion molecules (CAM's) that will serve as the guidance for growing axons. They have no active function: they just sit there to be sensed by the growth cone of the axon (see below). You can think of them as being displayed on the surface of the cells.

Axonal Processes

A special gene (tAxon) will be responsible for sending out a simulated axon into the environment of the cell. Whenever this gene is expressed, the cell will be checked whether it expresses a certain CAM. If it does, an axon will be sent out that is looking for CAM's of the same type. The detailed workings of the axonal growth cones (where all the action is) are discussed below.

Trophic Factor

The targets of neuronal processes express the gene coding for a trophic factor: this gene builds up an amount of factor available, that increases as long as the gene is being expressed. When a growth cone reaches a target, it will make synaptic contact and take some of the trophic factor away. The amount of trophic factor available and the number of axons trying to reach this axon can be significant factors in the determining the type of innervation of a certain region. Also, the weight of the synaptic contact can be modulated by the amount of trophic factor available (see below).

Neurotransmitters and Post-Synaptic Receptors

Three other kinds of genes will determine the nature of the synaptic contacts between neuron and innervated target:

- The neurotransmitter used must be expressed in the neuron and determines the selectivity of the neuron: only targets expressing the appropriate receptors will be able to respond to the neuron.

For each neurotransmitter there are corresponding receptors that can be expressed:

- Inhibitory receptors
- Excitatory receptors

It is conceivable to let these different choices that each cell can make determine the dynamical characteristics of the synaptic contact. Although it is not implemented at this stage, one kind of receptor could have a fast response to released neurotransmitter, while others would respond slower.

4.4. The Growth Cone Model

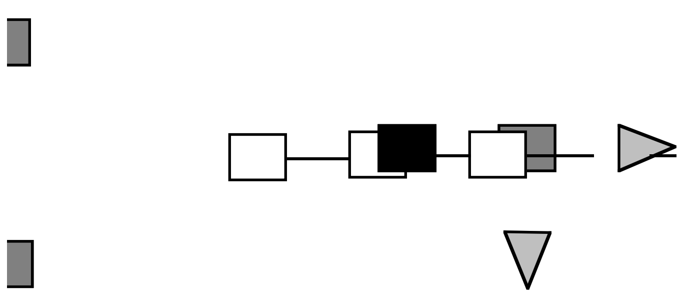

Fig.5. The different axon-element states.

The growth cone is the central component of the axon growth model. To be precise, the whole axon is modeled by a linked chain of ‘axon-elements’, each of which can be in one of four states. The different states are illustrated in Fig.5 and are summarized here:

- Link : this was once a growth cone, but now it serves merely as a link between the neuron and the active growth ones.
- Growth Cone: this is the element that directs the active search through the organism.
- Flank: every growth cone has several flanks, one in every possible direction (except backwards).
- Spike: this is a process sent out by a flank, that can sense the presence of a CAM or trophic factor.

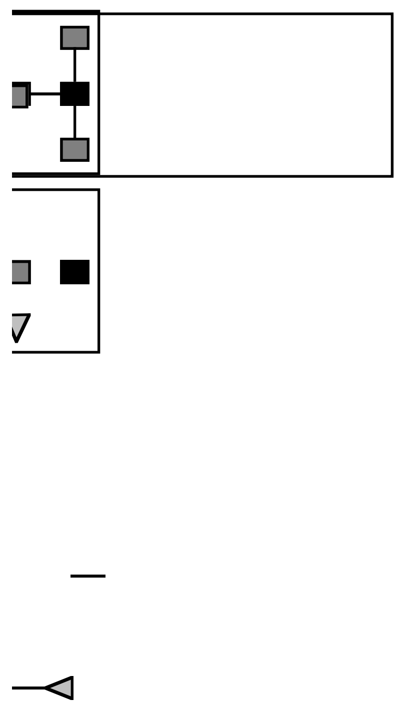

Fig.6. The consecutive steps of axon outgrowth.

The algorithm for axon outgrowth is as follows (see Fig.6):

- The cell sprouts one growth cone, with a flank in every possible direction, except backwards.
- At each step, the flanks send out a number of spikes, again in every possible direction. The spikes that do not encounter a CAM molecule of the right type are pulled back.
- The flanks with the most spikes remaining will become the new growth cones, with axon branching if there is a tie. The original growth cone now becomes a link element.

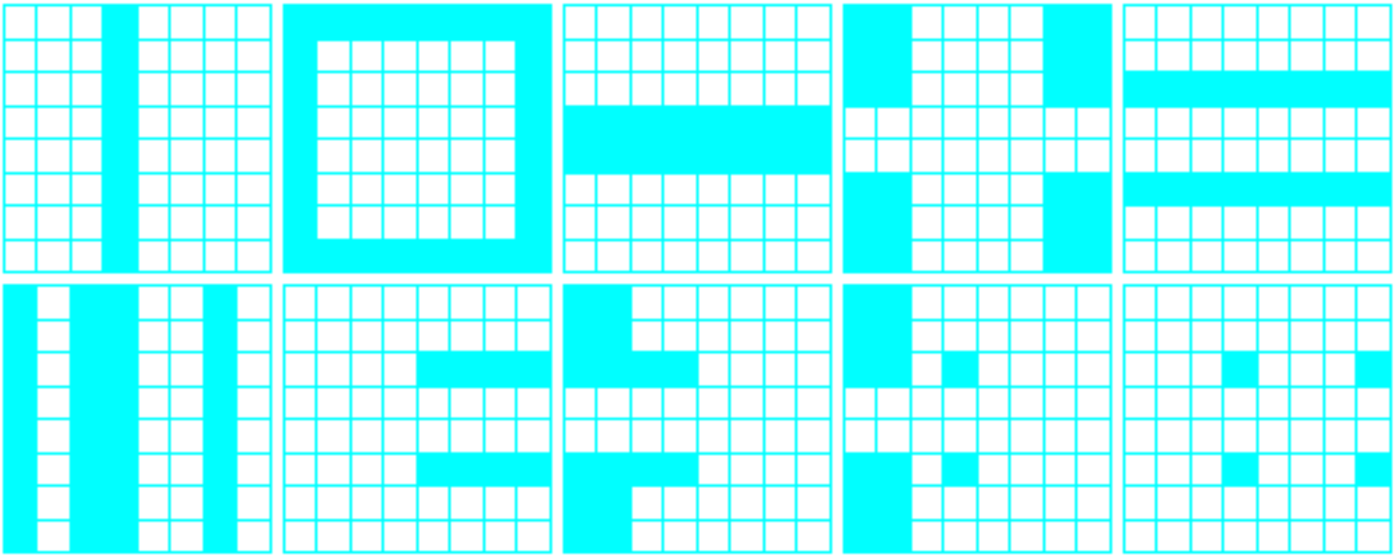

Fig.8. The expression of selected genes/gene products in the adult organism. Top row, from left to right: stripe, e, m, corner, spinal. Bottom row: runt, CAM100, CAM101, trophic, send axon.

When a trophic factor is encountered, the growth cone stops looking for CAM's and instead synaptic contact will be made with the cells expressing this factor. The store of trophic factor in the target-cell is decreased for each increase in synaptic weight, which will occur once each cell-cycle until the trophic factor is exhausted. Note that several neurons can conceivably innervate the same target and thus compete for available factor. A nice touch for future extension would be to implement a 'winner-take-all' mechanism, where only the most successful axon will innervate the target at the end of the day, while other axons pull away. Note also that the switch from CAM-searching mode to trophic-factor-searching mode is a local event and other branches of the same axon might still be happily feeling their way on some CAM pattern (the same CAM!).

5. A Test-case for the Extended Model

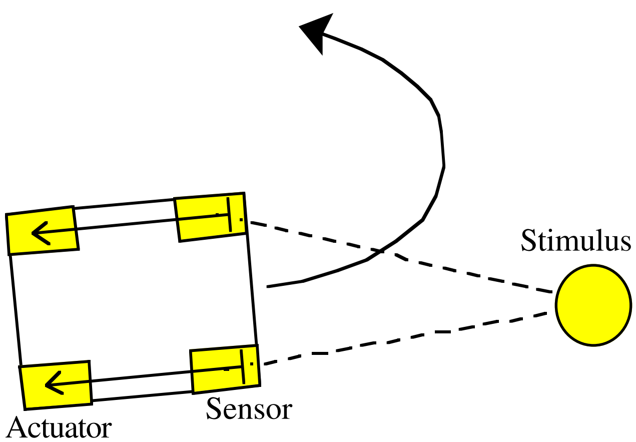

Fig.7. Schematic representation of an organism capable of avoiding a stimulus in a simulated environment.

To test whether this extended model can indeed be used to develop functional agents, we handcoded an organism that executes a simple task. Here we describe that organism, give an idea of what its genome looks like and report on its behavior in a simulated environment.

5.1. A Simple Braitenberg Hate-Vehicle

The organism that we want to hardwire should be capable of moving in a simulated world, and avoid a patch of 'chemicals' that it is capable of detecting by simulated 'smell'. We defined sensory regions in the front of the organism and actuator regions in the back. The actuators propel the organism, whereas the cells in the sensory regions are capable of detecting the patches of chemicals present in the environment. The simplest way for a nervous system to endow the organism with the correct behavior is to connect left-sensors with left-actuators and right-sensors with right-actuators. This is illustrated in Fig.7.

5.2. The Hardwired Genome

In Table.3 the genome we designed is represented: every number at the beginning of a line corresponds to a gene-product. Next to that number you can find either the intuitive name for the gene and/or its conditions for being expressed. Note that this only serves to give the reader a qualitative idea of how the expression of certain genes depends on other genes. Below the text there is a short explanation of the basic structure of the genome, and Fig.8 shows the expression of selected genes.

300 right
52 right = also agent
57 = receptor for right
62 detect right
307 detect right but not right = stripe
50 m -> acting agent 50
51 e*m
56 receptor for 51
61 next to 51
20 divide, always
30 e sensor
40 m sensor
31 e

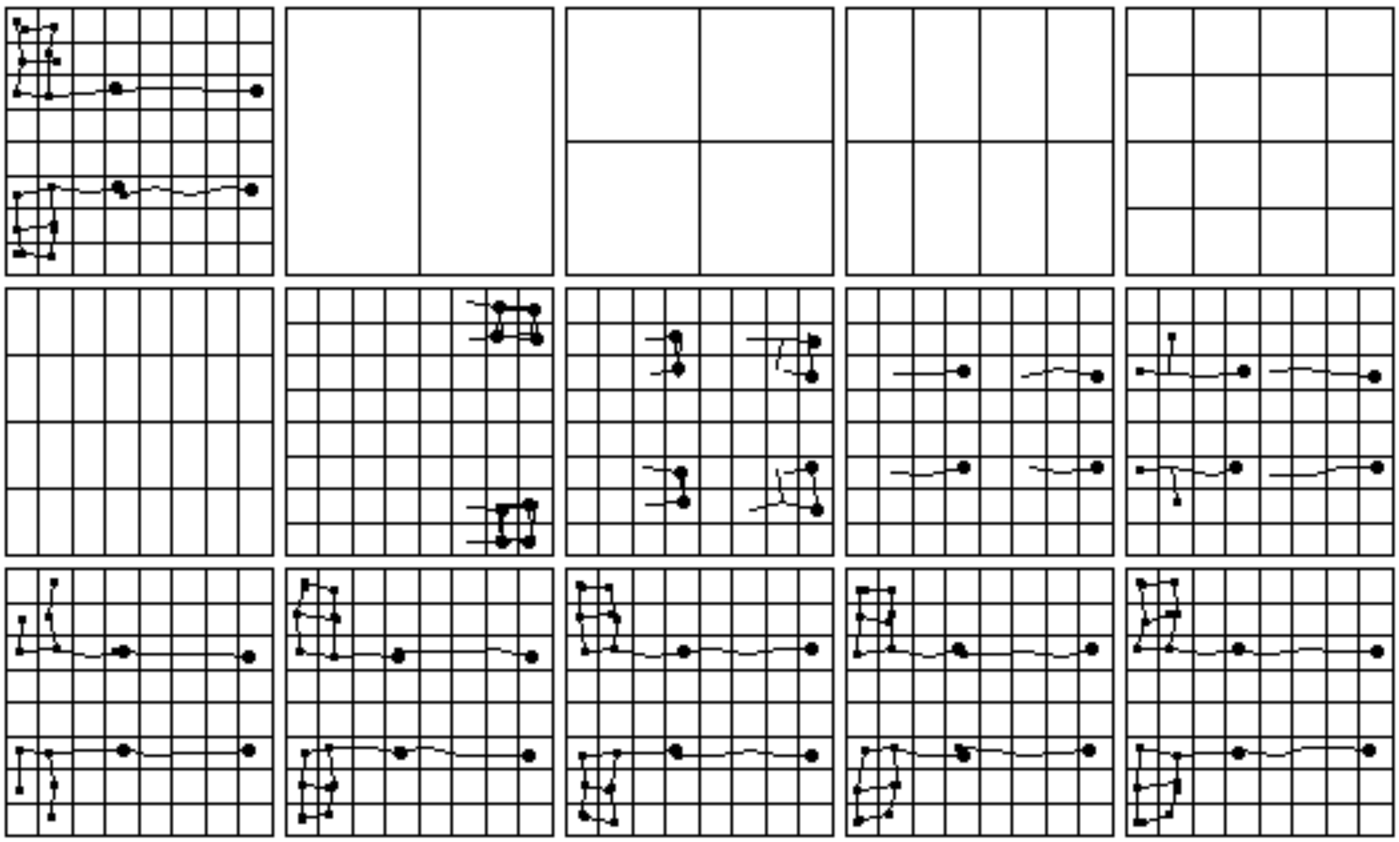

Fig.9. The consecutive steps in the development of the organism with the designed genome. The adult organism is shown in the upper left corner. The larger black blobs represent neurons, the small square blobs are synaptic contacts.

41 m
301 corner = not midline and 61 and self-reinforcement
55 receptor for 50
60 next to 50 = cord
302 spinal = cord but not mid
303 motor = corner and not right
304 eye = corner and right
53 runt = stripe or hairy detect
58 runt receptor all over
63 detect runt
54 hairy = next runt but not runt
59 hairy receptor when not hairy
64 detect hairy
305 wedge : spinal or motor
100 CAM 100 if wedge and right
101 CAM 101 if wedge and left
306 sensor wants axon : CAM 100 and 61 and env
308 runt and stripe and not env = stripey
309 stripey and cam 101 = interneuron
80 trophic if motor or interneuron
310 interneuron wants axon
90 send axon if sensor or interneuron wants to

Table.3. The genome designed to solve avoidance task.

Up to *spinal* all the genes serve topological functions, i.e. to define certain regions on the organism. The genes *spinal*, *motor* and *eye* make use of these topological genes to define the 'spinal cord', sensory region and actuator region, respectively. The genes *hairy* and *runt* are inspired by the pair-rule genes in Drosophila and serve to establish a segmentation pattern used later to help define the interneurons.

The rest of the genome defines the coordinated expression of the neural genes: in Fig.8 you can see that the gene 'send axon' is expressed both in the interneurons (in the middle of the square) and in two cells in the sensory region (to the right). The axons will grow from these sensory neurons to the interneurons over the 'CAM100' pathway and from the interneurons to the actuators by way of 'CAM101' (see below). Both in the interneurons and in the actuator cells the gene 'trophic' is expressed, to signal that they are neural targets.

5.3. Development

This genome then leads to the development of an organism with a nervous system. Two sensory cells at the anterior side (right side in the figures) of the organism project to two interneurons in the middle of the body. These in turn innervate the actuator regions at each flank of the body on the posterior side (left side). The consecutive steps in the development of this designed organism are shown in Fig.9.

Interestingly, if we let all the cells divide twice more to yield four times as many cells, the developmental process adapts and produces a qualitatively similar pattern. This indicates that using a developmental model for neurite growth does provide some robustness, even in the face of rather important changes of the underlying morphologies. Fig.9 and 10 show the

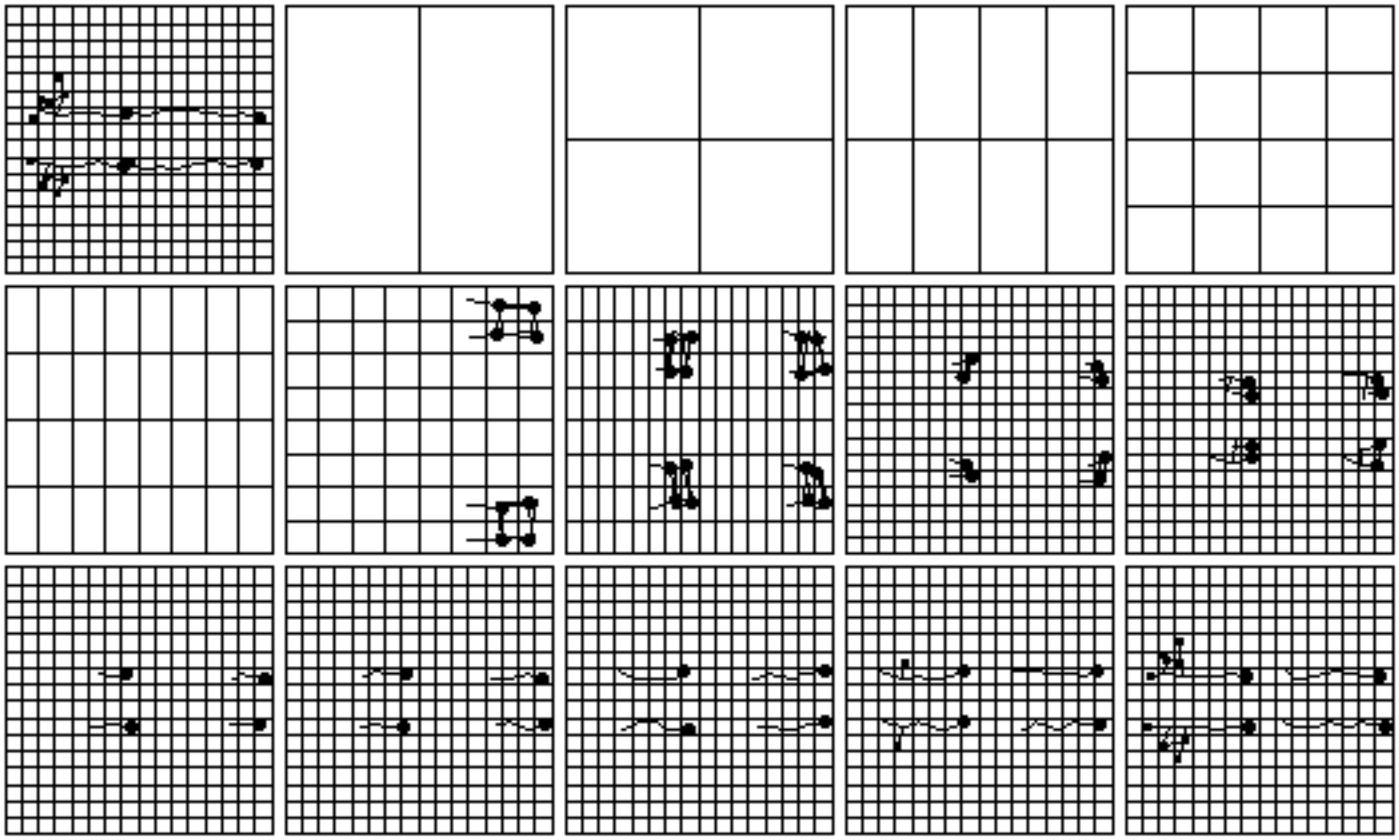

Fig.10. The developmental sequence directed by the same genome, but now with 8 divisions instead of 6.

results for 6 resp. 8 divisions, yielding 64 resp. 256 cells in the final organism.

5.4. Behavior in a simulated world

Finally we extracted a functional agent from the developed organism. A neural network is constructed by checking what cells are innervated by the neurons that send out axons. The neurons are connected appropriately to the input from the sensors, and the output of selected neurons is directed towards the actuators of the simulated agent. At this moment there are fixed regions on the square agent where sensory cells and actuators are located, although in future work we intend to let that be specified by the genome as well. The weights, thresholds and time constants in the extracted neural network are the same for all neurons, and their values were optimized by trial and error.

We then put this agent in a simulated environment and looked at its behavior when we set it on a course towards a patch of chemicals. Fig.11 shows the results of an experiment, wherein the organism is put down to the left side of a large patch (that it will want to avoid) and set on a course due East. The patches' smell falls of inversely proportional to the square of the distance. As you can see the organism does indeed steer away from the patch when it's sensors pick up the smell.

We then used 'software intracellular recordings' to show the activities of the neurons while it was executing its behavior. In Fig.12 the activities of the neurons of the agent are displayed. The square elements represent the neurons, and next to each neuron it's activity over time is displayed.

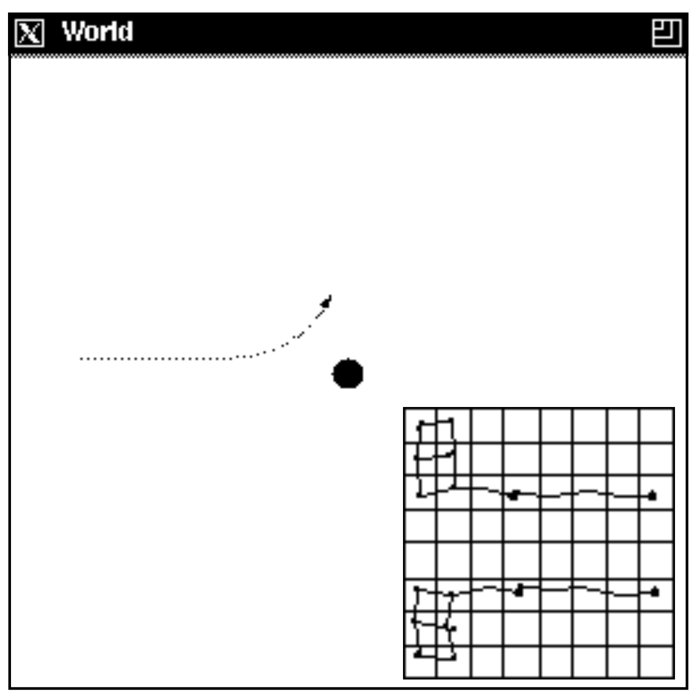

Fig.11. The behavior of the handcoded organism in response to a patch of chemicals.

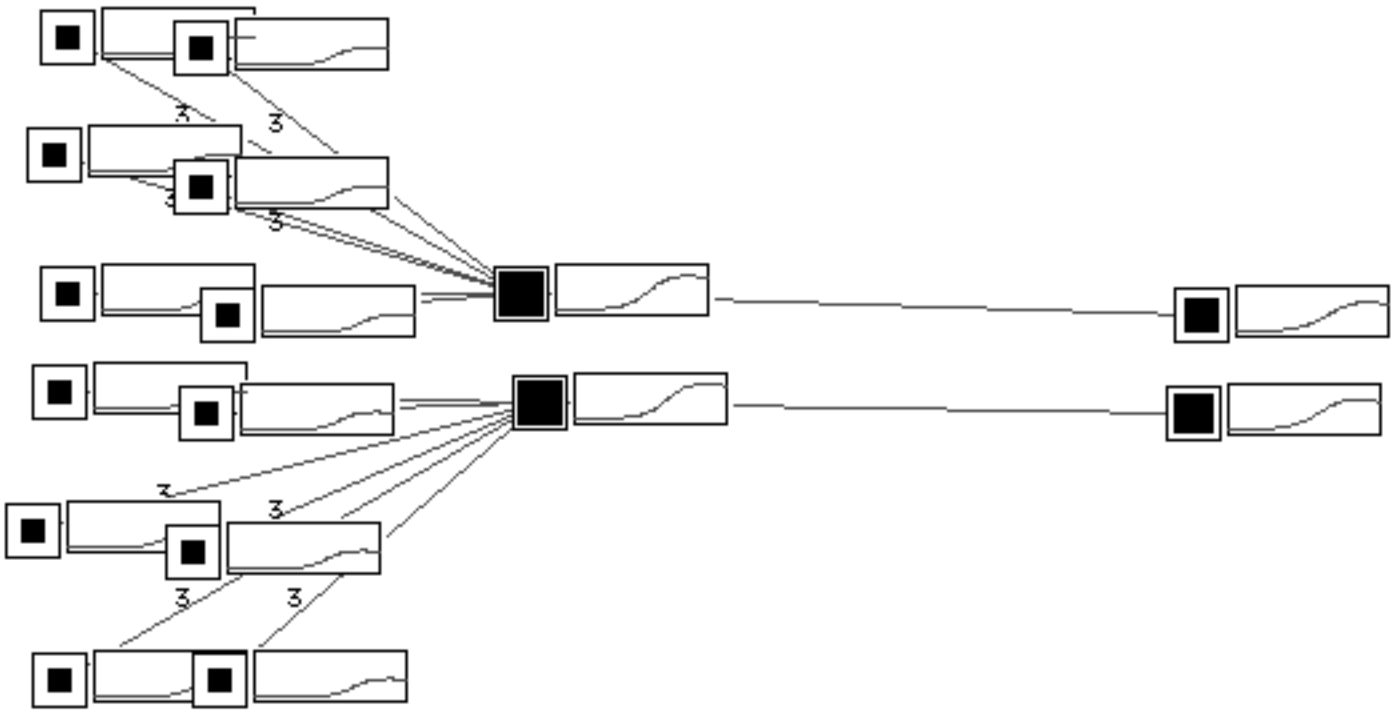

Fig.12. The ‘intra-cellular recordings’ of all the neurons of the handcoded organism during the experiment of Fig.11.

6. Evolvability of the Extended Model

The next interesting question is whether we can now use the genetic algorithm to improve on the behavior of the handcoded organism. As already stressed in (Dellaert and Beer 1994), note that mutations and cross-over will take place on the level of the genome, but that we are looking to optimize a performance function in the fully developed organism, which is not a trivial problem.

We used the handcoded organism to seed the starting population for the GA and we found that indeed, a better performing organism emerged, where the actuator regions were innervated in a different and stronger way. In this particular run of the GA we used a population size of 100 and the organism shown emerged after only 22 generations.

The developmental sequence of this newly evolved organism is shown in Fig.13, the intracellular recordings of it’s nervous system during the same experiment as before are shown in Fig.14, and finally it’s behavior in the simulated environment is shown in Fig.15.

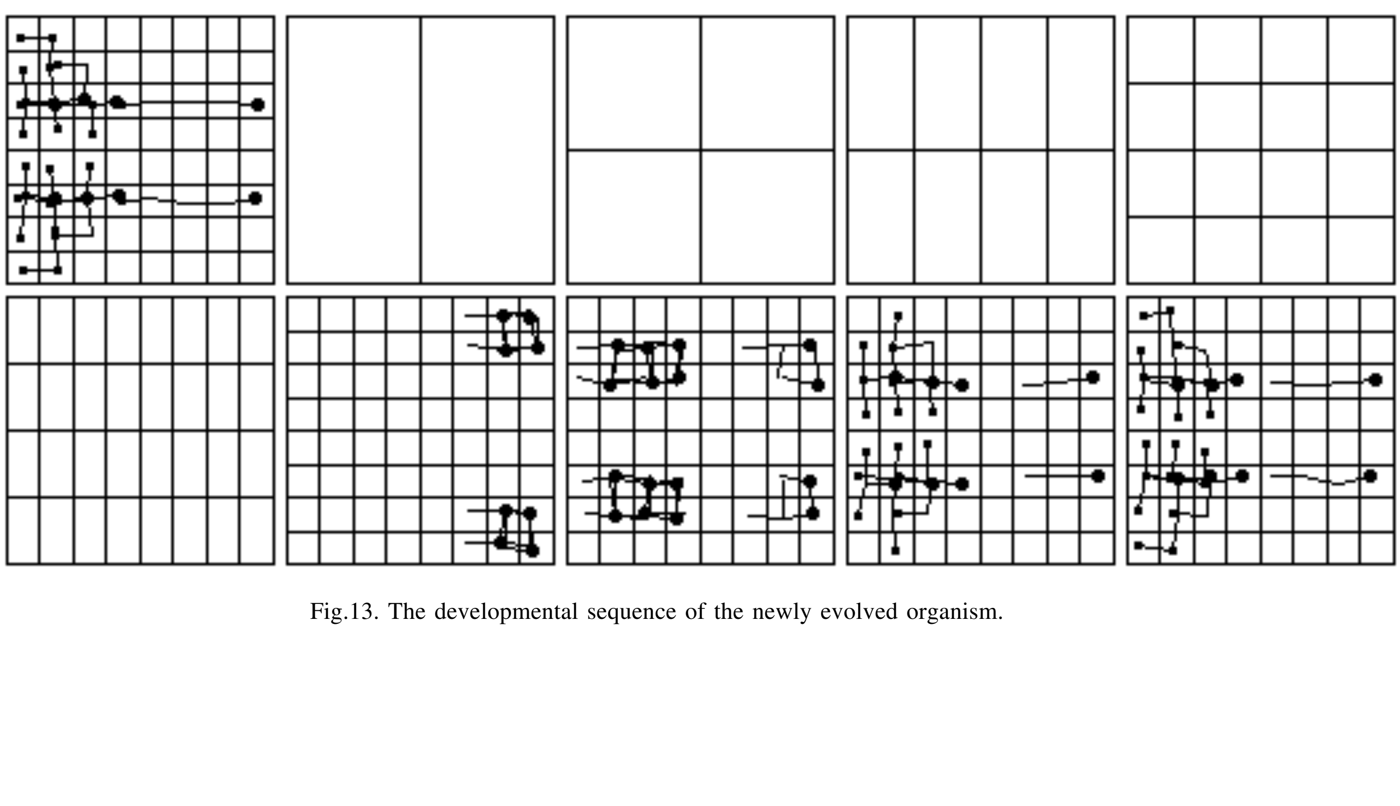

Fig.13. The developmental sequence of the newly evolved organism.

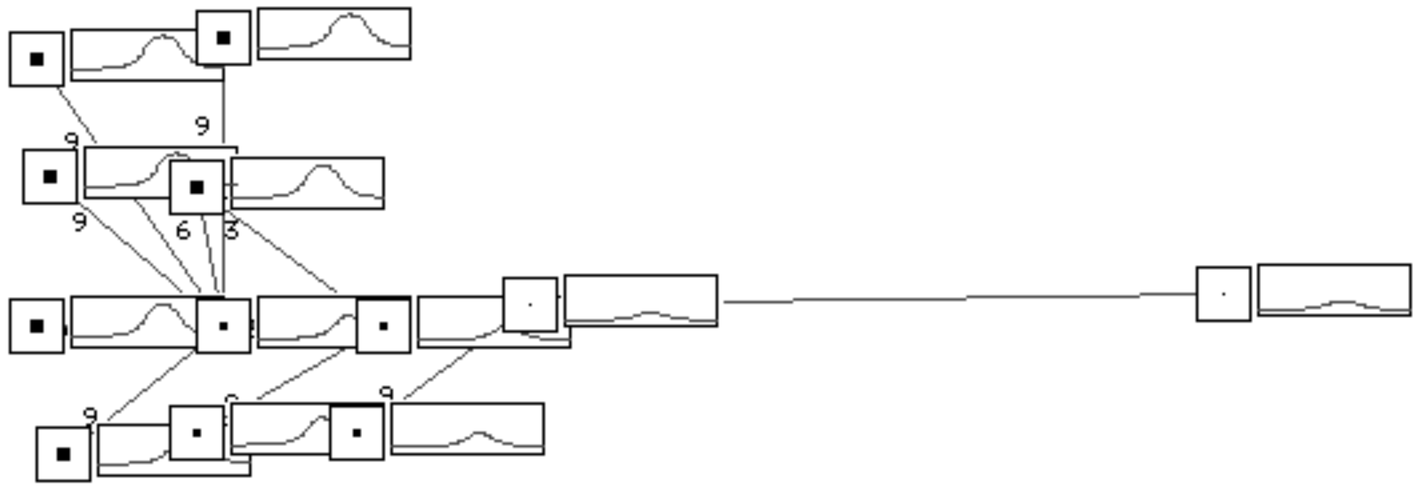

Fig.14. The ‘intra-cellular recordings’ of the topmost neurons of the newly evolved organism during the experiment.

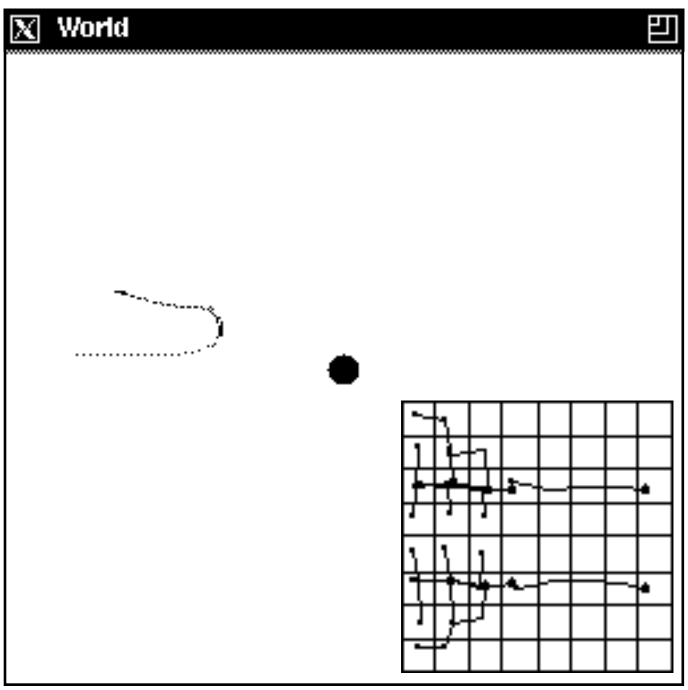

Fig.15. The markedly better behavior of the organism that was evolved from the original handcoded organism.

7. Conclusions

We have presented an extension to the basic developmental model discussed in (Dellaert and Beer 1994) to account for neural development, tested it by handcoding a functional agent, and we have shown that the model can be used in conjunction with genetic algorithms to yield better performing agents.

In addition, the developmental process used seems to be robust in the face of changes in the morphology of the agent. For example, we have seen this when we quadrupled the number of cells: the resulting phenotype was qualitatively the same, and we get functional agents without any change in the genome, which is remarkable.

Some issues have been touched upon but have not yet been completely implemented. For example, although the genome can code for different kinds of neurotransmitters and receptors, at this point there is not yet any functional significance attached to them in our implementation. The same argument holds for trophic deficiencies and resulting neuronal death.

Moreover, although we have shown here that we can incrementally improve on a working design using the genetic algorithm, we have not yet been able to evolve a functional agent from scratch. We would like to explore that issue more closely in future work.

On a positive note, the tedious work involved in hardcoding the genome has given us a good insight in the inner workings of the model and what it’s limitations and strengths are. This knowledge will enable us to improve on the model and hopefully help us to evolve working agents from scratch.

References

Dellaert, F. and R.D. Beer. 1994. Toward an Evolvable Model of Development for Autonomous Agent Synthesis. In *Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, edited by Brooks, R. and P. Maes. MIT Press: Cambridge, MA.

Kauffman, S. 1969. Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets. *J. theor. Biol.* 437-467.

Purves, D. and J.W. Lichtman. 1985. *Principles of Neural Development.* Sinauer Associates: Sunderland, MA.