Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

# First Experiments Relating Behavior Selection Architectures to Environmental Complexity

Lola Cañamero Orlando Avila-García Elena Hafner

*Adaptive Systems Research Group, Department of Computer Science, University of Hertfordshire, College Lane, Hatfield, Herts AL10 9AB, UK.*
*L.Canamero@herts.ac.uk, O.Avila-Garcia@herts.ac.uk*

## Abstract

Assessing the performance of behavior selection architectures for autonomous robots is a complex task that depends on many factors. This paper reports a study comparing four motivated behavior-based architectures in different worlds with varying degrees and types of complexity, and analyzes performance results (in terms of viability, life span, and global life quality) relating architectural features to environmental complexity.

## 1 Introduction

The behavior (or action) selection problem for an autonomous robot consists in making a decision as to what behavior to execute in a particular situation in order to satisfy its current goals in the best possible way, while working towards guaranteeing survival in the long term. Assessing the performance of behavior selection architectures is a complex task that depends on many factors. Comparative studies such as [6, 10, 3] have tended to focus on the respective merits and drawbacks of hierarchical (structured) versus flat (parallel) architectures, measured in terms of single global parameters such as reproductive capability (fitness). However, the gap between hierarchical and flat architectures seems too big, and we think that a comparison of architectures that are more similar to one another can give rise to a more systematic analysis. In addition, environmental complexity can be of different types and does not increase in a linear way with the number and types of elements introduced. Therefore, finer-grained analyses of architectures and environments, and of how these relate to each other, are needed to obtain a deeper understanding of the adequacy of different architectures for different types of tasks and contexts [5]. Taking a step in that direction, this paper reports an initial study comparing four motivated behavior-based architectures, all of them variants of the architecture proposed in [4], performing in five different worlds with varying degrees and types of complexity, and analyzes performance results relating architectural elements to environmental complexity. The criteria used to measure and compare performance are based on Ashby's viability theory [1], also used within the animat approach by e.g. [9, 8, 2]. In this framework, a set of survival-related variables that determine the robot's needs or goals must be kept within a viable range of values so that environmental changes do not put the robot's life in danger. Instead of a single measure of performance, we take into account different indicators in order to have a wider range of criteria for measurement and comparison. For this initial study, we have taken three indicators: the degree to which viability (the stability of the internal milieu) is preserved, life span, and the 'life quality' resulting from combining both. Regarding environmental complexity, we have considered the effects of availability of resources, number of objects and their influence on perception and navigation, and dynamism introduced by an enemy species.

## 2 The Valimar Environment

To test our architectures, we have created a typical behavior selection environment (Valimar, Figure 1) comparable to others used in this domain, in which our robot must select among and perform different activities in order to survive. We have used Webots 3.0 (www.cyberbotics.com), a very realistic 3D mobile robot simulator allowing users to create different environments with continuous space and complex physics. For these experiments we have used Kheperas fitted with a camera on their top to create two species of robots—Nessas (green Kheperas) and Enemies (red Kheperas). *Nessas* are more complex creatures used to implement the four behavior selection architectures described in Section 3. *Enemies* have the same sensors and actuators as Nessas, but they have a much simpler architecture and behavior (their main goal is to attack Nessas), since their only role is to introduce dynamism in the environment. Neither Nessas nor Enemies can remember the location of objects, and they also lack any planning capabilities. Valimar is surrounded by a wall and contains cylindrically shaped objects of different colors: food (yellow) and water (blue) sources, nests (purple), obstacles (gray), dull blocks (red), and our two species of robots. Since Enemies and dull blocks have the same color, Nessas can mistake one for another.

0-7803-7398-7/02/$17.00 ©2002 IEEE

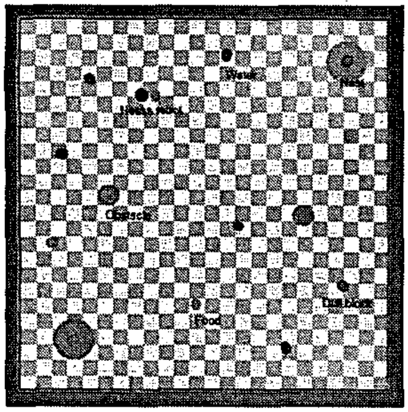

*Figure 1: Top view of the Valimar world used in the first set of experiments (Valimar 1).*

## 3 Architectures for Behavior Selection

The architectures we have studied are neither strictly flat (parallel) nor hierarchical (structured). They consist of two layers—motivational and behavioral—that lead to a two-step computation of intensity. This computation is parallel within each layer, but motivational intensity must be computed before calculating behavioral intensity. All the architectures have the same elements, but they vary in the way in which these are combined (their arbitration mechanisms).

### 3.1 Elements

**Sensing and acting.** The robots are equipped with the following *external sensors*: eight proximity (infrared) sensors, six on the front and two on the back; eight binary collision sensors (located at the same points as proximity sensors); a radio emitter/receiver used to transmit and detect the attack of another robot ($pain_e$), which can have different intensities; and a color camera returning a RGB pattern of $90 \times 90$ pixels. Visual input is used to detect direction and discriminate objects. Object learning and recognition is performed by a combination of three ART1 neural networks, each of them specialized to detect patterns in one of the RGB components. In addition to external sensors, we have programmed *internal sensors* to perceive the values of physiological variables.

The robots have differential wheel steering. Navigation (with obstacle avoidance) is controlled by a neural network that improves over time through Hebbian learning.

**Physiology.** The robots have a synthetic physiology of survival-related essential variables that must be kept within a viable range of values so that environmental changes do not put the robot's life in danger. These variables set the (internal) needs or goals of the robot. Nessa's essential variables are: damage, energy, glucose, moisture, internal pain[1] ($pain_i$), and stress.

*Table 1: Nessas' motivations.*

| Motivation | Drive | Incentive stimulus |
|---|---|---|
| Confusion | $\downarrow stress$ | nest |
| Excitement | $\downarrow energy$ | enemy |
| Fatigue | $\uparrow energy$ | nest |
| Hunger | $\uparrow glucose$ | food |
| Overmoisture | $\downarrow moisture$ | none |
| Overnutrition | $\downarrow glucose$ | none |
| Repair | $\downarrow damage$ | nest |
| Self-protection | $\downarrow pain_i$ | enemy, $pain_e$ |
| Thirst | $\uparrow moisture$ | water |

**External stimuli.** In addition to internal variables, behavior selection is also influenced by the presence of external stimuli that affect the motivational state of the robot or the intensity and execution of behaviors, depending on the architectures. There are six types of external stimuli to which Nessas can react: the different (colors of the) objects in the environment plus external pain ($pain_e$).

**Motivations.** Motivations constitute urges to action based on bodily needs related to self-sufficiency and survival. They implement a homeostatic process to maintain an essential physiological variable within a certain range. Nessa's motivations are characterized by: a controlled (essential) physiological variable, a drive to increase or decrease the level of the controlled variable, an (external) incentive stimulus that can increase the motivation's intensity (only in architectures A2, A3, A4, see Section 3.2), and a behavioral tendency of approach or avoidance towards the stimulus. A feedback detector generates an error signal—the drive—when the value of this variable departs from its setpoint, and this triggers the execution of a behavior to adjust the variable in the adequate direction. The error is a number normalized in the interval $[0, 1]$, where 0 indicates no error and 1 results when the actual value of the variable overflows/underflows the upper/lower limit, in which case the robot dies. Each motivation receiving an error signal from its feedback detector receives an intensity (activation level) proportional to the magnitude of the error. Several motivations can be active at the same time, with varying degrees of intensity. Table 1 shows Nessas' motivations.

**Behaviors.** Our behaviors are coarse-grained subsystems implementing different competencies, as in [6, 4]. Following the usual distinction in ethology [7], Nessas have consummatory (goal-achieving) and appetitive

[1] Pain has a double characterization as internal and external stimulus. Internal pain receives its value from externally felt pain but has more inertia, decreasing more slowly and lasting longer.

*Table 2: Nessas' behaviors. Names in italics indicate appetitive behaviors, the rest are consummatory.*

| Behavior | Stimulus | Effects | Motiv. (A1&A2) |
|---|---|---|---|
| *Avoid* | none | ↓ *energy*, ↓ *glucose*, ↓ *moisture*, ↓ *pain*, ↑ *stress* | avoidance motivations |
| Attack | enemy | ↓ *energy*, ↓ *glucose*, ↓ *moisture*, ↓ *pain*, ↓ *stress* | excitement, confusion, self-protection |
| Drink | water | ↓ *energy*, ↓ *glucose*, ↑ *moisture* | thirst |
| Eat | food | ↓ *energy*, ↑ *glucose*, ↓ *moisture* | hunger |
| Rest | none | ↑ *energy*, ↓ *glucose*, ↓ *moisture* | fatigue |
| RunAway | enemy | ↓ *energy*, ↓ *glucose*, ↓ *moisture*, ↓ *pain*, ↑ *stress* | self-protection |
| *Search* | none | ↓ *energy*, ↓ *glucose*, ↑ *stress*, ↓ *moisture* | approach motivations |
| Sleep | nest | ↓ *damage*, ↑ *energy*, ↓ *glucose*, ↓ *stress*, ↓ *moisture* | confusion, fatigue, repair |
| Wander | none | ↓ *energy*, ↓ *glucose*, ↓ *moisture* | overmoisture, overnutrition |

(goal-directed) behaviors. A consummatory behavior is executed only if it has been selected by the motivational state of the robot and its incentive stimulus is being observed. The execution of a behavior has an impact on (increases or decreases) the level of specific physiological variables. Behaviors can be activated and executed with different intensities that depend on the intensities of the motivations (and in some architectures also of the stimuli) related to them. The intensity with which a behavior is executed affects motor strength (speed of the wheels) and the modification of physiological variables (and hence the duration of the behavior). Table 2 shows Nessas' behaviors[2].

### 3.2 Arbitration mechanisms

We have designed four architectures by varying the way in which these elements are interconnected along three parameters (summarized in Table 3):

**Link between layers.** Motivations and behaviors are connected either through fixed weights indicating the relevance of the link (in architectures A1 and A2) or indirectly through physiological variables (in A3 and A4).

**Locus of influence of external stimuli.** The effect of external stimuli can be computed to influence the intensity of behaviors (A1) or of motivations (A2, A3, A4).

**Point of decision.** The main selection decision can be made at the level of motivations, as in A3, where a single motivation is in charge of selecting the behavior that best satisfies it. It can also be made at the level of behaviors, as in A1, A2, and A4, where behaviors receive activation from all the motivations that they contribute to satisfy, and behavior selection is postponed until behavioral intensity has been computed. In this case, all the behaviors are considered for the final selection, and the robot can satisfy several goals simultaneously.

[2]The table shows the motivations related to particular behaviors in architectures A1 and A2 only, where the connection between these two elements is fixed through weights, as explained below.

*Table 3: Characterization of the four architectures.*

| Arch. | Links | Stimuli computed | Decision point |
|---|---|---|---|
| A1 | fixed weights | on behaviors | behaviors |
| A2 | fixed weights | on motivations | behaviors |
| A3 | physiology | on motivations | motivations |
| A4 | physiology | on motivations | behaviors |

The behavior selection loops are as follows.

*Behavior selection loop in A1.* At every cycle:

1. The intensity of each motivation's drive is calculated as proportional to the error of its controlled variable ($e_{vj}$).
2. The intensity of each behavior is calculated as $b_i = \sum(m_j \times w_{ij}) + \sum(s_k \times v_{ik})$, where $b_i, m_j$ are the intensities of behavior $i$ and motivation $j$, respectively, $w_{ij}$ is the weight between behavior $i$ and motivation $j$, $s_k$ is the intensity of stimulus $k$, and $v_{ik}$ is the weight between behavior $i$ and stimulus $k$.
3. The behavior with highest intensity is selected to be executed.

*Behavior selection loop in A2.* At every cycle:

1. Calculate the intensity of each motivation $j$:
   (a) Compute the intensity of the motivation's drive as proportional to the error of its controlled variable ($e_{vj}$).
   (b) Compute the effect of the presence of external stimuli on the intensity of the motivation $j$: $a_j = \sum(s_k \times u_{jk})$, where $s_k$ is the intensity of stimulus $k$, and $u_{jk}$ is the weight between $j$ and $k$.
   (c) $m_j = e_{vj} + a_j$ is the final intensity of $j$.
2. The intensity of each behavior is computed as $b_i = \sum(m_j \times w_{ij})$, where $b_i, m_j$ are the intensities of behavior $i$ and motivation $j$, respectively, and $w_{ij}$ is the weight between $i$ and $j$.
3. The behavior with highest intensity is selected to be executed.

*Behavior selection loop in A3.* At every cycle:

1. The winner motivation $j_{winner}$ is calculated.
   (a) For each motivation $j$:
      i. Compute the intensity of the drive as proportional to the error of its controlled variable ($e_{vj}$).
      ii. Compute the effect of the presence of external stimuli on the intensity of the motivation: $a_j = \sum(s_k \times u_{jk})$, where $s_k$ is the intensity of stimulus $k$, and $u_{jk}$ is the weight between $j$ and $k$.
      iii. $m_j = e_{vj} + a_j$ is the final intensity of $j$.
   (b) The motivation with highest intensity is selected.
2. The intensity of each behavior linked (through the physiology) with the winner motivation is computed as $b_i = m_{j_{winner}} \times f_{iv}$, where $b_i, m_{j_{winner}}$ are the intensities of behavior $i$ and the winner motivation, respectively, and $f_{iv}$ is the effect that the execution of behavior $i$ has on $v$, which is the physiological variable controlled by $j_{winner}$.
3. The behavior with highest intensity is selected to be executed.

*Behavior selection loop in A4.* At every cycle:

1. Calculate the intensity of each motivation $j$:
   (a) Compute the intensity of the motivation's drive as proportional to the error of its controlled variable ($e_{vj}$).
   (b) Compute the effect of the presence of external stimuli on the intensity of the motivation $j$: $a_j = \sum(s_k \times u_{jk})$, where $s_k$ is the intensity of stimulus $k$, and $u_{jk}$ is the weight between $j$ and $k$.
   (c) $m_j = e_{vj} + a_j$ is the final intensity of $j$.
2. The intensity of each behavior is computed as $b_i = \sum(m_j \times f_{iv})$, where $b_i$, $m_j$ are the intensities of behavior $i$ and motivation $j$, respectively, and $f_{iv}$ is the effect that the execution of behavior $i$ has on $v$, the physiological variable controlled by $j$.
3. The behavior with highest intensity is selected to be executed.

## 4 Experiments

The purpose of our experiments was to investigate the adequacy of each architectures to deal with different levels and types of environmental complexity. We have used three indicators to measure and compare performance:

**Viability:** The average level of satisfaction of all the essential variables, measured at each step as $v_{step} = 1 - (Err/max_{err})$, where $Err$ is the total sum of errors of the robot's physiological variables normalized between $[0, 1]$ with $max_{err}$, the worst error possible in each step. $Err$ corresponds to the sum of the intensities of the motivations' drives ($Err = \sum(e_{vj})$), and $max_{err}$ is the number of compatible motivations[3], since the worst motivational state corresponds to the situation in which all compatible motivations have their highest intensity (1.0). Average viability for a run is given by $V = \sum_1^{t_{life}}(v_{step})/t_{life}$, where $t_{life}$ is the number of simulation steps that the robot lived.
**Life span:** The time that the robot survived (remained viable) during each run in simulation steps normalized with the total simulation time, $S_{life} = t_{life}/t_{simul}$, where $t_{life}$ is the number of simulation steps that the robot lived and $t_{simul}$ is the total simulation time measured in number of simulation steps.
**Life quality:** $Q_{life} = V \times S_{life}$.

### 4.1 Method

We have explored the effects of three sources of environmental complexity: (a) number of objects, (b) availability of resources, and (c) dynamism—in this case the presence of Enemies that can attack and kill Nessas, and also hamper foraging activities. To vary these sources of complexity, we created five Valimar settings (Table 4) with different elements sparsely distributed across the world. Valimar 1 (V1, see Figure 1) should allow Nessas to have a "comfortable" life—a fair amount of resources and nests should permit them to satisfy their needs, while some obstacles hinder navigation and perception of resources; it only has complexity of types (a) and (b), since Enemies are absent, although Nessas mistake dull blocks for Enemies (they have the same color), and therefore they can use them as "resources" to satisfy their aggression needs. V2 and V3 progressively decremented (a) and (b), and we added (c) in V4 (one Enemy) and V5 (two Enemies).

[3]Two motivations are compatible if they do not control the same variable in opposite directions.

*Table 4: Elements of the five Valimar worlds.*

| Elements | V1 | V2 | V3 | V4 | V5 |
|---|---|---|---|---|---|
| Food sources | 4 | 2 | 1 | 2 | 2 |
| Water sources | 4 | 2 | 1 | 2 | 2 |
| Nests | 2 | 1 | 1 | 1 | 1 |
| Obstacles | 2 | 2 | 0 | 2 | 2 |
| Dull blocks | 2 | 1 | 1 | 0 | 0 |
| Enemies | 0 | 0 | 0 | 1 | 2 |
| Nessas | 1 | 1 | 1 | 1 | 1 |

We tested the four architectures in 10 sets of runs for each Valimar world, each set being comprised of a run for each architecture. This made a total of 200 runs[4] (about 40 hours simulation time), each run lasting 10,000 steps of 64 ms of simulated time. Physiological variables were randomly initialized (to values within their viability range) for each set of runs.

### 4.2 Results

Figure 2 shows the average performance of the four architectures in the five Valimar worlds in terms of viability (left), life span (center), and global quality of life (right).

Results for A1, A2 and A4 were very similar in the first three (static) worlds, where A3 obtained the worst results in terms of viability, as its winner-takes-all policy deals worse with extreme situations in which at least one variable is near its limit. In terms of life span, survival was very good in the static worlds. The introduction of Enemies in V4 and V5 leads to significant changes with respect to the three indicators. Viability and life span become considerably worse for all the architectures, due to the negative impact of Enemies' attacks on the physiology of Nessas. In these dynamic worlds, A3 outperforms the other architectures because its winner-takes-all policy makes it more reactive to external changes, dealing better with situations of self-protection (see below), while the opposite is the case for A4.

Let us now discuss some phenomena commonly studied in animal decision making (see e.g. [7]) that we have observed in our simulations, and that allow us to understand

[4]Although more runs would yield more reliable results, analysis of the standard deviation of each set revealed high uniformity in the static worlds, but more randomness in the worlds with enemies.

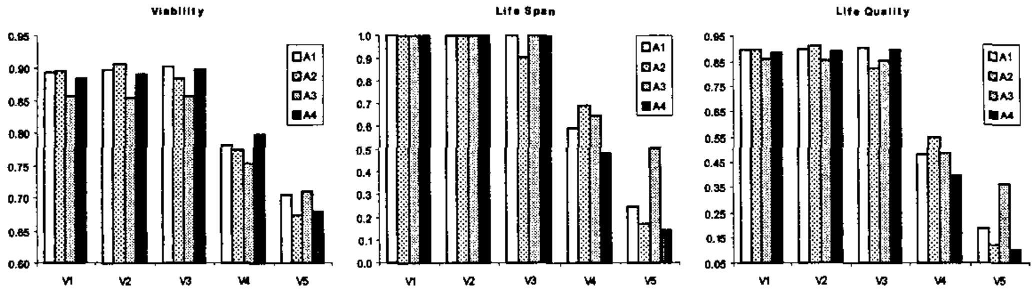


*Figure 2: Average performance of the four architectures in the five Valimar worlds.*

better the differences among the architectures and how they deal with different properties of the environment. Table 5 ranks our architectures regarding some of these phenomena that [6] proposes as desirable features of architectures to achieve flexible behavior selection.

*Table 5: Ranking of architectures (1 is best, 4 poorest).*

| Phenomenon | A1 | A2 | A3 | A4 |
|---|---|---|---|---|
| Openness (reactivity) | 3 | 2 | 1 | 4 |
| Stability | 2 | 2 | 4 | 1 |
| Opportunism | 3 | 1 | 3 | 2 |
| Displacement behaviors | – | – | – | 1 |
| Varying attention | 3 | 2 | 1 | 4 |
| Efficiency maximization | 3 | 2 | 4 | 1 |

**Stability** of a sequence of behaviors occurs when behavioral intensities are (nearly) similar for all the sequence. A non-stable sequence results in sudden changes[5] in the robot's velocity and modification of its variables. A3, being more reactive as it uses a single motivation to drive behavior selection, was the least stable architecture, while A4, with its "maximum profit" approach was the most stable.

**Opportunism management** varied considerably in the four architectures as a consequence of the way in which the influence of external stimuli is taken into account. Architectures A1 and A3 are less opportunistic than A2 and A4, since the influence of external stimuli is computed only once to calculate the intensity of the winner behavior. Taking the example of the sleep behavior, the influence of its incentive stimulus (the nest) is computed in A1 only at the level of the sleep behavior, and in A3 only at the level of the motivation that selects sleep to satisfy its drive—either fatigue, confusion or repair. However, in A2 and A4 the influence of the presence of the nest, computed at the motivational level, can be taken into account several times—as many as active motivations the sleep behavior can help satisfy, i.e. a combination of fatigue, sleep and repair. Although opportunism is in general a desirable feature that provides flexibility, too much opportunism can present disadvantages in environments with few resources located far from each other.

**Displacement behaviors** were only observed in A4[6], as its "maximum profit" policy, together with the fact that links between motivations and behaviors can be positive (excitatory) or negative (inhibitory), can lead to *mutual inhibition* of two motivations with high intensity. These features are also responsible for the considerably lower activation levels that behaviors receive in A4, compared to the other architectures, making it less reactive.

**Situations of self-protection** are more difficult to deal with when Nessas are executing a consummatory behavior next to a resource, as they are more exposed to Enemies, which can attack them on the back, where they are not detected, and block them against resources, as shown in Figure 3. This roughly corresponds to what [6] denotes as *varying attention*—the fact that animals pay lower attention to danger when they are in an extreme motivational state, e.g. very hungry. A4 is more often trapped in these situations than the other architectures due to its lower reactivity and to the fact that its lower intensity levels make Nessas spend more time next to resources. A3, being more reactive (like a simple emotional system), is the best in these situations.

**Maximizing efficiency** of behavioral choice is an important desiderata for behavior selection mechanisms. Executing a behavior that satisfies several motivations simultaneously is usually more efficient (in terms of viability) than a behavior that only satisfies one goal. In this respect, an interesting difference was observed in situa-

[5] Recall that the intensity of the winner behavior has an impact on motor strength and on how physiological variables are modified (and therefore on the duration of the behavior).

[6] Displacement behaviors would also have been possible in A1 and A2 if we had not considered only positive weights between motivations and behaviors, but are not possible in A3, where only one motivation drives behavior selection.

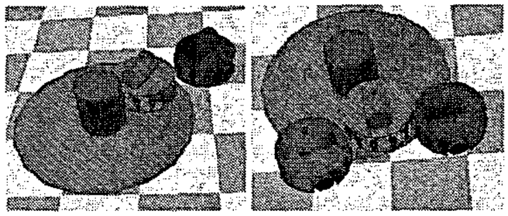

*Figure 3: Nessas (lighter Kheperas) attacked and blocked by Enemies next to a nest.*

tions of fatigue between A3 and the other three architectures. While A1, A2 and A4 tend to resolve these situations with the sleep behavior—which is more costly because it can only be executed on a nest, but it allows to satisfy the fatigue, confusion, and repair motivations simultaneously—A3 usually deals with them by executing the rest behavior—which can be executed anywhere but only contributes to correct fatigue. The explanation lies in the fact that, for sleep to be executed in A3, the confusion motivation has to win the competition against fatigue, and this does not happen frequently as confusion is produced by excess of stress, which increases very slowly. In addition, if fatigue wins, the rest and sleep behaviors will have the same intensity, and since rest does not need the presence of a stimulus to be executed, it is always executed first. This difference is important because the execution of sleep in A1, A2 and A4 improves the robot's viability considerably more than the execution of rest in A3, and this can partly explain the poorer performance of A3, in particular in static environments.

## 5 Conclusion and Future Work

In this paper, we have presented an initial study of the performance of different motivated behavior selection architectures in environments with varying degrees and types of complexity. Instead of a single measure of performance, we have used different indicators, drawn from viability theory, in order to have a wider range of criteria for measurement and comparison. We have analyzed performance results in terms of how architectural elements relate to various sources of environmental complexity. Our results show that small variations in the way in which the same architectural components are combined can greatly influence the way in which behavior selection is performed, and therefore the adaptivity of the robot to different environmental conditions. This supports our idea that a finer-grained analysis and comparison of architectures and environments favors a more through understanding of the adequacy of different architectures for different types of tasks and contexts.

To continue this study we envisage several directions for future work. First, we would like to complement our viability and life quality indicators with a measure of the internal equilibrium achieved in terms of the standard deviation of the motivations' error. This would provide another indicator of quality of performance that cannot be simply integrated with the previous ones, as it is not easy to compare which is best, measuring internal stability in terms of a good global viability level or in terms of how uniformly the different motivations are satisfied. Second, we want to vary the complexity of the world in terms of dynamism to take into account not only the presence of enemies but also extinction and mobility of resources. Finally, we plan to add basic emotions to our behavior selection architectures, following [4], and compare the performance of the different architectures with and without emotions in static and dynamic worlds.

## Acknowledgments

Orlando Avila-García is supported by a scholarship of the University of Hertfordshire. Elena Hafner was supported by a Socrates program with the University of La Laguna, Spain.

## References


[1] Ashby, W.R. (1952). *Design for a Brain: The Origin of Adaptive Behavior.* London: Chapman and Hall.

[2] Aubin, J.P (2000). Elements of Viability Theory for Animat Design. In J.A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S.W. Wilson, eds., *Proc. Sixth Intl. Conf. on Simulation of Adaptive Behavior,* 13–22. Cambridge, MA: MIT Press.

[3] Bryson, J. (2000). Hierarchy and Sequence vs. Full Parallelism in Action Selection. In J.A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S.W. Wilson, eds., *Proc. Sixth Intl. Conf. on Simulation of Adaptive Behavior,* 147–156. Cambridge, MA: MIT Press.

[4] Cañamero, L.D. (1997). Modeling Motivations and Emotions as a Basis for Intelligent Behavior. In W.L. Johnson, ed., *Proc. First Intl. Conf. on Autonomous Agents,* 148–155. New York: ACM Press.

[5] Cañamero, L.D. (2002). Designing Emotions for Activity Selection in Autonomous Agents. In R. Trappl, P. Petta, S. Payr, eds., *Emotions in Humans and Artifacts.* Cambridge, MA: MIT Press (in press).

[6] Maes, P. (1991). A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature. In J.A. Meyer and S.W. Wilson, eds. *Proc. First Intl. Conf. on Simulation of Adaptive Behavior,* 238–246. Cambridge, MA: MIT Press.

[7] McFarland, D. (1995). *Animal Behaviour,* 3rd. edition. Harlow, England: Longman.

[8] Meyer, J.A. (1995) Artificial Life and the Animat Approach to Artificial Intelligence. In M. Boden, ed., *Artificial Intelligence.* Academic Press.

[9] Steels, L. (1996). Discovering the Competitors. *Adaptive Behavior,* 4(2).

[10] Tyrrell, T. (1993). The Use of Hierarchies for Action Selection. *Adaptive Behavior,* 1(4): 387–419.