

WoxBOT: a Wide Open Extensible Robot for Virtual World Simulations

Marcio Lobo Netto, João Eduardo Kögler Jr., Emílio Del Moral Hernandez and Fábio Roberto Miranda

Escola Politécnica - Universidade de São Paulo

São Paulo, Brazil

Abstract

This paper reports some activities towards artificial life research. We propose the implementation of a society of virtual mobile robots, able to perform requested tasks while behaving according to their own motivation and reasoning. Each robot is an intelligent agent that perceives the virtual environment via a simulated vision system and reacts moving its own body by driving its own wheels. The robot also communicates with other robots placed near to itself, through a simulated talking and listening (prototypical hearing) system. This feature allows them to communicate with each other, giving support for the evolution of their social skills. The conception and specification of the robots and the environment are being conducted very carefully to create an open distributed object architecture that could serve as a testbed freely available and ready to use for testing theories in some computational areas. Our intention is to test concepts in evolutionary computation, artificial life, pattern recognition, artificial intelligence, cognitive neurosciences and distributed objects architectures.

Keywords: *Virtual Life, Genetic Algorithms, Neural Networks, Intelligent Multi-Agents, and Distributed Systems.*

1. INTRODUCTION

Artificial life is a term originally intended to mean the simulation of macroscopic aspects of living beings behavior using microscopically simple components [1]. However, the term spanned to designate a wide variety of simulated living creatures, including virtual characters whose behavior emerges from hierarchical and functionally specialized complex structures like an animal body [2]. Both kinds of simulations are very interesting from the computational point of view: they offer very attractive means to computationally model complex behavior, a subject that had gained a special relevance under both the theoretical and the applied standpoints.

Artificial life worlds are computational simulations like virtual places where animated characters interact with the environment and with other virtual beings of the same or distinct categories. Different degrees of sophistication can be found in these virtual creatures, from unicellular life with minimalist models to complex animals with detailed biomechanical models. However, all of them display behavior emergent from the dynamics of a complex system (for instance, systems with learning, reasoning, ontologies, cognitive processes, etc).

Our goal is the exploitation of diverse strategies used by the creatures to perform tasks while adapting themselves to the environment. This scenario can be useful for many purposes: for behavior modeling, as a laboratory of learning algorithms, for research on societies of virtual characters, in the study of

population collective dynamics, etc. Therefore we have conceived an ARENA, the environment of the virtual world, and a robot - WOXBOT (Wide Open Extensible Robot), as virtual being that inhabit the ARENA. It was conceived to allow more than one robot to be easily introduced to co-exist in the same virtual world where they may cooperate or compete with each other. As an example of application, a given research project can be targeting visual recognition methods, so it will be using WOXBOT as a subject employing the methods under test, and will take ARENA as a laboratory for evaluation experiments. Another research project instead, can be targeting autonomous robot mobility algorithms and traveling strategies.

Sensors (including information processing), actuators and an automaton composed by a deterministic finite state machine define the WOXBOT architecture. The information captured by the sensor is classified through a dedicated neural network. As a result codes are generated and given as input to the automaton, which controls the actuators.

WOXBot has been considered to employ multiple sensors and among the several possibilities we chose to start with the following:

- a vision system consisting on a simulated camera followed by a neural network that classifies the visual patterns, generating symbols (codes);
- a hearing system able to identify time-varying signals produced in the surrounding neighborhood mimicking sounds, with a dedicated neural network that classifies the audio patterns, generating symbols to be interpreted as components of the coded message;

The information got by the vision system is directionally constrained by the visual field, while that one received by the listener is extended to an area covering an angle around the receiver wider than the visual field. However, although having more coverage, this hearing does not renounce the vision because the latter provides a more precise description of the environment than the former. Actually both senses complement each other. After the classification by their correspondent neural networks, the produced symbolic descriptions are combined and give the automaton a better scenario about the external world, allowing it to change to a new state according to a more complete information. In response, the automaton motor control system, responsible for the movement of the robot decides which motion action to take.

Besides of serving as an element to detect sounds connected to simulated physical events occurring in the robot neighborhood, its hearing system can be used as the receiver of a communication system among robots co-existing in the ARENA, provided they have also a speech system to generate audible messages. Following this idea, the robot sensorial neural

network outputs are also directed to a speech system, responsible for the emission of symbolic messages to the surrounding environment. These serve as behavioral answers that mimic sounds of a prototypical speech, consisting of signals emitted by the robot over a wide angle around its *voice generator*. In a first phase some kind of primitive sounds are used as a form of communication between distinct robots.

The automata implementing the motion and speech control state machines are obtained from optimization procedures conducted with genetic algorithms. Through evolutionary mechanisms we intend to be able to see an improvement of these automata, leading to virtual creatures better adapted to their environment and to each other. For instance, primitive sounds may lead to basic languages if the robots become able to associate information with sounds.

The ARENA is implemented as a distributed object environment and can run on single processor platform as well as can take advantage of parallelism or multithreading in high performance computer architectures. In this later case it could be used to handle highly sophisticated and complex applications, such as population behavior. It could also be used to evaluate multi-agent distributed computational models using the WOXBOTS as the prototypical agents.

The organization of the paper is as follows: section 2 gives an overview of the project, presenting the chosen requirements and proposed solutions. Section 3 review several conceptual aspects involved in the various components of the project and give guidelines to be used in the implementation. Section 4 details the modeling of the character and the environment and presents the results obtained. Section 5 foresees the further steps to be taken in the project.

2. PROJECT OVERVIEW

Our long-term research line in artificial life is aimed to the development of complex virtual worlds with realistic creatures able to exhibit sophisticated behaviors supported by reasoning, learning and cognition. To achieve it, one is required to consider the following aspects:

- account for the wide variety of mathematical and computational models usually required to represent the complex aspects of living behavior;
- provide an efficient environment and platform with enough computational power to appreciate the details of the simulated creatures behavior in real time, including the ability to support live interactivity with persons;
- specify and build a system that embodies constant evolution and improvement in its constructs at the same time that enables or, indeed, promotes the reuse of well succeeded results and implementations.

Our choice was to fulfill these requirements building an application development environment as an architecture of distributed communicating objects mapped on SPADE, a microcomputer cluster [3]. SPADE is a non-expensive scalable parallel computing architecture with high performance capability that can be used with LINUX and WINDOWS NT. The use of the object-oriented paradigm promotes code reuse and

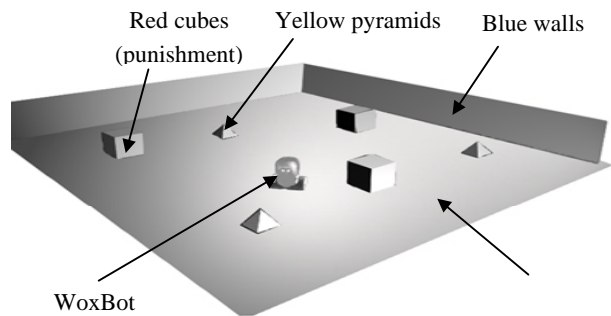


Figure 1: ARENA (Artificial Environment for Animats) – a virtual world for the development of artificial life projects. The character at the center is a robot that avoids the cubes and seeks the pyramids. A contact with a cube reduces the robot life, while the pyramids extend it. The robot goal is to prolong life as long as it can. An evolutionary computing scheme selects the better performing robots. The environment can support several robots.

modularity while giving great simplicity for maintenance and evolution. This is a capital aspect to cope with the necessity of testing many computational and mathematical models to account to the wide variety of behavioral theories. The ARENA implementation exploits parallelism via multithreading, so the program can run both on multi and single processor platforms. While single robot implementations can run perfectly on single processor computers, the multiple robot implementation should run on multi-processor architectures to cope with the application performance requirements.

The ARENA has a floor plan and walls encompassing the limits of the virtual world. Inside this scenario there are objects of several kinds and functionalities like obstacles, barriers, traps, shelters, energy or food sources etc, serving as pitfalls or resorts for the creatures (figure 1). One or many characters will be introduced in this environment to perform certain tasks. In the first stage of the project the sole character will be the WOXBOT (figure 2) whose task is to keep itself alive as long as it can. Our desire is to provide a very general environment to serve as a workbench for many applications focused in artificial life.

Certain conceptual ingredients play specific roles on composing the artificial life scenario and thus should be present in this implementation: *sensing and perception, knowledge use, and evolution.*

Sensing and perception is the way by which the creature can grab information from the environment essential to update its internal state when performing behavior regulation. This last concept refers to the tasks and purposes of the creature. The invariants detected by perceptual system can be used to construct or update a knowledge base or ontology. A knowledge base (or, more specifically, the ontology) constitutes stable and structured information that can be re-used by the creature to reason about the environment, to predict the effects of its own behavior and to perform task planning. Finally, evolution is the ingredient that enables a creature to change its adaptation to a variable environment. In the following sections we will examine these aspects.

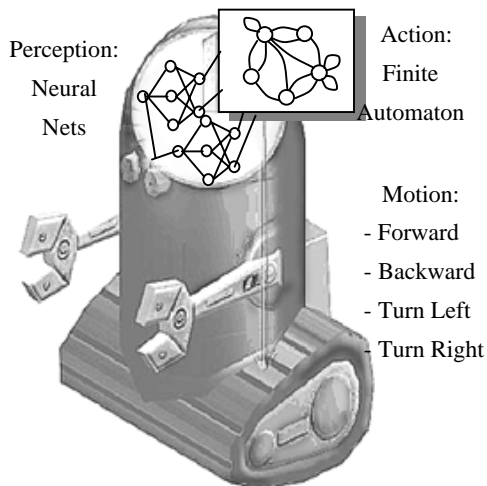


Figure 2: WoxBOT (Wide Open Extensible Robot) – a virtual character that recognizes visual and audio inputs and reacts according to the decision taken by the finite state automaton. Trained neural nets classify the acquired inputs. The WoxBOT has a distributed object open architecture that enables changing the perception and action algorithms very easily.

3. FOUNDATIONS

3.1 Artificial Life

Artificial life has been associated to computer science by many ways, ranging from cellular automata theory [4][5], to computer graphics animation [3][6][7]. Some researchers have been seeking models for the beginning and evolution of life in nature [4]. Graphics animators and scientists have been looking for physical models to give natural appearance and behavior to their simulated characters [7]. Although very different in nature, these works have been related to evolution and natural selection concepts.

There are two computational issues that emerge in the subject of modeling life, a local and global one. The local aspect refers to the ability of an individual creature to regulate its own behavior. The global one concerns with the adaptation of a certain species. The global aspect emerges from the local in the sense that a species is well adapted to its environment if the corresponding creatures are also successful individually. These issues are manifest in the genotype and in the phenotype. The genotype carries the transmitted characteristics of a creature and as an equivalence class it represents the species. The phenotype refers to a specific individual. To design an artificial creature is to find an organism that can reach a specified regulation of behavior when performing some task. The organism can be modeled with some representational language, given by some automaton. Structured design methodologies are very difficult to use in this field, even when a very good specification and description are provided. The main drawback is to cope with an evolving complex environment. Evolutionary computation is a modeling and design paradigm well suited to these cases and its main tool is the genetic algorithm.

3.2 Genetic Algorithms and Evolution

Genetic algorithms are computational procedures to find the optimal solution in particularly hard problems [9][10][11]. Each creature is represented in this context by a character string that constitutes its own phenotype. A population of such creatures has a genotype that is the equivalence class of the phenotypes under a similarity relation that can characterize the species. The genetic algorithm used in the creature design consists in assisting some transformations carried in the genotypes of members of a population of such virtual creatures, allowing them to change from one generation to another and providing a selection of the best ones. The first generation starts with an arbitrary set of genotypes, *a priori* sub-optimal. Genetic mutation and combination (by reproduction) provide efficient ways to modify the creature’s characteristics, as in natural life. Selection comes after, choosing those ones that, by some criteria, are the best suited, and therefore are allowed to survive and reproduce.

In our work evolutionary processes are crucial components for our virtual character. A finite state automaton controls each WoxBOT and the challenge is to design one that performs tasks efficiently to cope with complex inputs coming from the perceptual modules. The design procedure is carried with a genetic algorithm in which the automaton evolves through generations, based on genetic mutations and crossing-over. Each robot descendant arises with some innovative features that may result on a better or worse adaptation to the environment than those from previous generations. The individuals are selected with some given criteria (for instance, keeping greater energy, living more time, performing faster its tasks, etc) and the ones that achieve higher grades have higher chances to be selected to compose the next generation. This process tends to an optimum according to the efficiency of the problem specification and domain representation [9][10].

3.3 Intelligent Agents

The use of knowledge in the artificial life environment can be done under two aspects: (i) it can be present in the environment and in the conception of the creatures and, (ii) it can be used by the creatures themselves when performing their actions. In both cases, the effective and rational use of the knowledge (about the world, the actions and the tasks) leads to what is called intelligent behavior. Intelligence can also be considered a property emergent from evolutionary systems [9]. Thus, the evolving characters can be modeled as intelligent agents performing tasks in the virtual environment.

Intelligent agents may be understood as computational entities that behave with autonomy in order to manipulate the information associated to its knowledge [12]. This autonomy must regard the agent design, the environment, goals and motivations [13]. They have the capability of reasoning about what is going on in the surrounding environment and about the consequences of their own actions, or to respond to some query conducted by other agents. As this are evolved many features of agents have been identified, as their capacity to communicate with other agent, their mobility (migrating from one computer to another through networks), their intelligence (how efficiently they can perform or use reasoning), and so forth.

4. IMPLEMENTATION

4.1 Intelligent Agents

The WoxBOT was conceived starting from its specification as an intelligent agent. In this implementation, a set of two interacting WoxBOTS has been tested on a single processing node computer. This paper focuses only on aspects of the WoxBot design and training.

The WoxBOT was specified having vision and hearing sensors through which it receives all input from the exterior. And it regulates its behavior in response to the inputs and its own purposes through a finite state automaton. This architecture supports also a knowledge base (ontology) and reasoning, although they have not yet been incorporated. The actions performed are communication (speech) and mobility.

By now we are interested in the perception and behavior regulation issues. The perceptual module and the finite state machine have been designed for these tasks, and we made use of learning and evolution as the ingredients responsible to provide a somehow complex agent. The robot tasks now seem simple: it should look for energy sources, while avoiding traps. However, with a single agent, the possible situations are still many and the tasks could be performed following a sophisticated plan, considering that the performance criteria could be set to include more constraints, like minimal time, minimum use of energy, etc. As explained ahead, based on visual information, the robot plans its way to get closer to the energy sources, and this planning is in fact conducted by the finite state machine that evolved through generations. The hearing and speech of the robots are currently being designed and in this work we present some of the results obtained with the first version of the vision system.

On next versions of ARENA we will explore multi-agent interaction, and WoxBOT and other derived creatures will be able to display sociability skills, which is an important feature to promote a better adaptability. Furthermore, they will also implement mobility features, since we plan to have the multiple creatures living in a virtual environment, supported by the distributed computer cluster, the SPADE. In this implementation the simulated vocal conversation among robots will be of paramount importance to obtain a realistic scenario of artificial life characters.

4.2 Perceptual Modules and Learning

One of the tasks that WoxBOT has to perform in a first project is to be aware of nutrients (yellow pyramids) and hurting entities (red cubes) that are present in ARENA (figure 1). The visual information is gathered from the 3D-surrounding scene by projecting it to a viewport with 3 color channels (RGB). Figure 3 depicts a typical input, which at the present state is taken in low resolution, since there are no textures and the accuracy is not yet affected. To differentiate and spatially locate these entities, two specialized neural networks interpret the visual information received by the robot, ANN-I targeting nutrients and ANN-II targeting hurting entities.

code	scene
0	no object
1	target object at left
2	target object at center
3	target object at right

Table 1: viewer ANN I & II

Both networks ANN-I and II are trained for the identification of their respective objects (yellow pyramids and red boxes) as well as for a general evaluation (classification) of the position occupied by them in the robot's visual field (Table 1). In this way, the networks make the robot aware of the presence and location of nutrients and hurting entities.

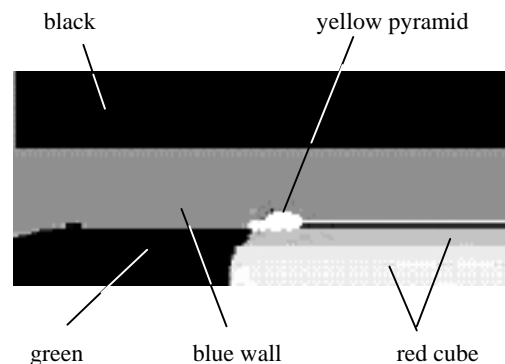


Figure 3: A typical image of the ARENA environment as seen by the WoxBOT vision. The actual image has the depicted colors with shading resulting of the illumination. This image is the input of the neural nets that perform perception. It is also possible to add textures to the 3D scene and make yet more complex patterns.

The architecture chosen for ANN-I / II is the standard multi layer perceptron with one single hidden layer of moderate size and learning through the error back-propagation algorithm. Only eight hidden nodes were enough for this recognition task. Notice that this low number is consistent with the fact that we are dealing with a low number of output classes (only four).

Among the observations that we did during the training of the neural networks we want to mention that the detection and position classification of visual targets was facilitated by the use of samples (pyramids and cubes) of similar sizes. ANN-I and ANN-II do not have good abilities to implement scale invariance. That was in fact expected, since the complexity of the network is relatively limited, and its architecture does not have any specialized organization proper to implement scale invariance. This indicates that some kind of size invariance mechanism could be added in the pre-processing stage. Another possibility is to use a larger and more complex neural architecture so to make possible the interpretation of targets positioned at very discrepant distances. Again, WoxBOT was conceived to allow these improvements and others to be easily implemented, granted by its modularity and object oriented open architecture.

An important issue to consider in the context of the neural networks of WoxBOT is adaptability. In these initial experiments with the robot, the training of ANN-I and ANN-II was done in an isolated phase, which was performed previously to the exploration of ARENA. Only after WoxBOT was able to do the differentiation between nutrients and hurting entities, he started his exploration in ARENA, and then, no further adaptation of the visual recognition system was allowed. In future phases of this project, we want the recognition system to be able to deal with unpredicted changing conditions such as change of illumination, and we will include the change of the features of the nutrients targeted by WoxBOT (shape and color for example). Of course, this depends strongly on the ability of the visual recognition system to adapt, since the target recognition prototypes change with new experiences and new needs of the robot, as its life and exploration of ARENA evolves. At that point of the project, the intrinsic adaptive nature of neural networks will be crucial, and it will be fully explored.

A similar neural network approach will also be used to classify sounds. However, a first implementation did not require any special classification mechanism, because the used sounds were simple tons (different frequencies).

code	sound
0	no sound
1	high frequency
2	low frequency
3	medium frequency

Table 2: listener

4.3 Action and Behavior

The WoxBOT character is an intelligent agent with simulated sensors to get information from surrounding. The neural networks inside the agent analyze this information, classifying it, and generating tokens as outputs, which in turn are used to make a decision about which action to take. Based on this information the robot can decide to go ahead, turn right or left in order to reach a pyramid or avoid a cube (see Table 3). In the same way it can generate a sound (speech).

A finite state machine performs the motion and speech control. This automaton was designed without specifying how the inter state transitions should occur. It is set initially with a random structure that is improved based on evolutionary computation concepts (see figure 4 for an example).

code	action
0	turn left
1	go ahead
2	turn right
3	go backward

Table 3: Movement encoding

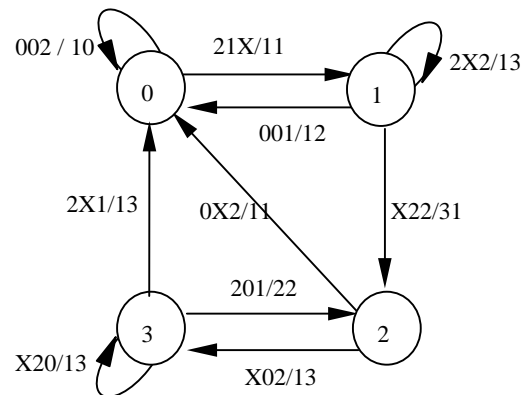
code	speech
0	no sound
1	low frequency
2	medium frequency
3	high frequency

Table 4: Sound encoding

The state machine is coded into a string of bits representing its states, inputs and actions. This string is named the robot's chromosome. The 64 possible inputs are the combinations of the outputs of the two viewers and one listener (4 possibilities each) as shown in Tables 1 and 2, coded in octal in the example of figure 4. They apply to every state leading to a total of 256 chromosomes in this example. Figure 5 depicts some sample chromosomes, corresponding to the inputs 002 and 210 occurring at state 0 and input 122 at state 1 of this automaton.

State	...	0	0	1	...
Inputs	...	002	210	122	...
Transition	...	0	1	2	...
Actions	...	10	11	31	...

Table 5: Sample chromosome corresponding to the automaton of figure 4. States, inputs and actions are coded in octal.



Sensor Inputs Actions
Viewer1/2-Listner Movement Controller - Speaker

Figure 4: A sample finite state automaton used by WoxBot to control its reactions to the visual stimuli.

The binary string is composed in the following sequence: current state / inputs / new state / actions. At the start of the robot life the contents of this string are set randomly, resulting in an arbitrary state machine. The optimization procedure via genetic programming begins with a set of robots initially created with arbitrary state machines (figure 6). The optimization criterion is set to accomplish the goal of maximizing the duration of the robot life. Then each of these initial robots is put to live individually in the ARENA, and their lives duration is recorded.

The best-suited, i.e. those living longer are then preserved. They can generate descendants by reproduction, through the crossover of the chromosomes producing a new breed. The newer generation is then tested in the same manner, until no expressive changes in the robot life duration occur. Then, a random mutation is produced in some of the chromosomes, to induce variety and to proportionate a new strategy. Based on this concept we expect to be able to see after many generations the survival of the most adapted creatures.

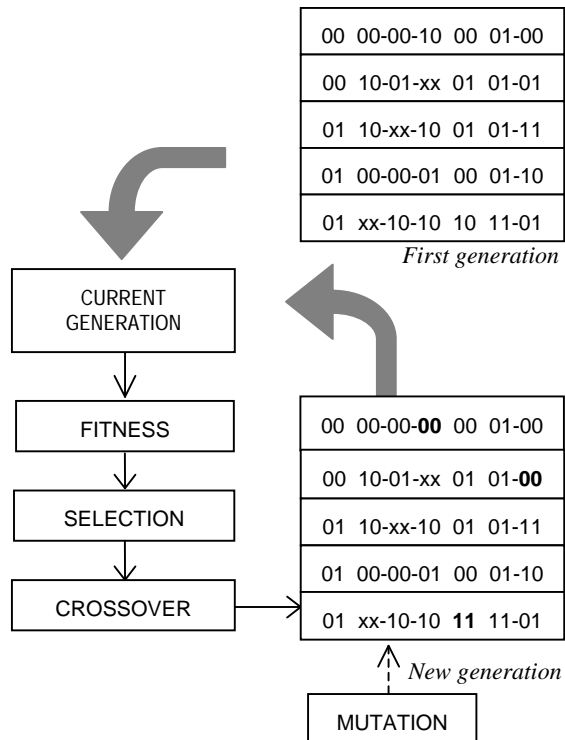


Figure 6: The evolutionary process starts with a generation of WOXBOTS with arbitrary automaton. These undergo successive genetic optimization steps to produce the best adapted automaton that gives WOXBOTS an optimal performance. The presented chromosomes are a subset of the total, corresponding only to those associated with the states 0 and 1 in figure 4 (in boldface are the changed bits).

4.4 Implementation Issues

Before building an application intended for virtual world simulation, one key decision to be made is whether it will be a real time application or not. Many 3D animation packages today, like Maya™ or Softimage™, have integrated programming interfaces, allowing algorithmical animation of characters. This approach, yet useful for some kinds of simulations, has the drawback of few or none capabilities for user interaction.

Having decided for real-time capability, the next issue is the choice of the API. Some requirements we decide to fulfill with ARENA included: portability through many platforms,

shareability and open architecture, and we wanted to make it easy to be used. Given these issues, our choice was made for the Java Programming Language for development and Java3D as 3D graphics package. Java is a worldwide spread and freely available language tightly object-oriented. Its three-dimensional extension Java3D is available on a variety of platforms (Windows™, Linux, Irix™ and others), is freely available and can be run through web browsers with the proper plug-in. The most powerful feature of Java3D is its scene-graph oriented approach, that shifts the focus of 3D programming from vertexes and the rendering process to hierarchies of scene objects and events, shortening development time.

Additional advantages include: loaders for content authored in 3D modeling and animation packages; use of underlying OpenGL or DirectX accelerated hardware; automatic use of threads for executions; taking advantage of parallel hardware; and the free Java Integrated Development Environments (IDEs), like Java Forté, from Sun Microsystems and JBuilder, from Inprise.

The scene graph structure of this application is composed of one locale, with three branch-groups directly attached to it. There is one branch-group for the user view of the scene (view branch-group), one branch-group holding the 3D objects in the scene that can not move (floor and walls) and one branch-group holding the animated scene objects. The reason for this division is that Java3D performs a series of optimizations based on what a given branch-group can or cannot do. By holding all the 3D objects that cannot move in one branch-group we can increase the efficiency of the optimizations. The animated branch-group include the cubes, pyramids and robots. This branch-group also holds a control object that hold the dynamic aspect of the WOXBOT. WOXBOT control models the behavior of the robot, keeping track of visual data (accessing the WOXBOT's Canvas3D), presents the data to the neural networks for pattern recognition and feeds the patterns into the state machine that decides robot's actions.

5. CONCLUSION AND FUTURE WORK

The first stage of the project was to build the ARENA with a single WOXBOT to test the main idea and to orient the specification of the further steps. There are two main directions to follow simultaneously from now. One is to proceed with the ARENA introducing more complexity in the environment, in the tasks and using two or more WOXBOTS simultaneously. The other is to improve the WOXBOT to build new and more sophisticated characters.

The contribution of this work was to provide a rich environment for artificial life experiments that is shareable, portable, scalable and implemented in a widespread language, Java and an extensible 3D graphics package, Java3D. The tests already performed have employed sophisticated ingredients in a combination that is somehow innovative, that is to use a neural network based vision system to provide input to a finite state automaton designed with a genetic algorithm. These two modules are components of an intelligent agent that can be cloned over distributed object architecture to perform several actions, with cooperative or competitive behavior. The whole system of ARENA and WOXBOT is very flexible and has a very

great potential to evolve to extend its use to many appealing applications.

The immediate steps being taken are the implementation of the voice communication between WoxBots and the improvement of the vision system to take into account scale invariance and distance estimation.

We thank Prof. Dr. Sérgio Takeo Kofuji for his support to this project as well as for providing the computational resources of the SPADE cluster for the implementation.

6. REFERENCES

- [1] E.W. Bonabeau and G. Therulaz, "Why do we need artificial life?" in *Artificial Life an Overview*, edited by C.G. Langton, MIT Press (1995).
- [2] D.Terzopoulos (org.), *Artificial Life for Graphics, Animation, Multimedia and Virtual Reality*, SIGGRAPH 98 course notes 22 (1998).
- [3] Volnys Bernal *et al*, "PAD Cluster: An Open Modular, and Low Cost High Performance Computing System" in *Proceedings of 11th Symposium on Computer Architecture and High Performance Computing*, Natal, Brazil (1999).
- [4] C. Adami, *Introduction to Artificial Life*, Springer Verlag (1998).
- [5] A. Dorin, "Physically Based Self-Organizing Cellular Automata", in *Multi-Agent Systems – Theories, Languages and Applications*, C. Zhang and D. Lukose (eds), Lecture Notes in Artificial Intelligence v. 1544, Springer Verlag (1998).
- [6] C.Phillips and N.I. Badler, "Jack: A toolkit for manipulating articulated figures", in *ACM/SIGGRAPH Symposium on User Interface Software*, Banff, Canada (1988).
- [7] N.I. Badler, C.B. Philips and B.L. Webber, *Simulating Humans: Computer Graphics, Animation and Control*, Oxford University Press (1992).
- [8] P. J. Bentley (ed.), *Evolutionary Design by Computers*, Morgan Kaufmann (1999).
- [9] D.B. Fogel, *Evolutionary Computation - Toward a new philosophy of machine intelligence*, IEEE Press, (1995).
- [10] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press (1992).
- [11] J. Koza, *Genetic Programming*, MIT press (1992).
- [12] S. Russell and P. Norvig, *Artificial Intelligence – A Modern Approach*, Prentice-Hall (1995).
- [13] J.S. Sichman and Y. Demazeau, "A model for the decision phase of autonomous belief revision in open multi-agent systems", *Journal of the Brazilian Computer Society*, N 1, V 3 pp 40-50 (july 1996).
- [14] R.D. Beer, H.J. Chiel and L.S. Sterling, "A biological perspective on autonomous agent design", in *Designing Autonomous Agents*, P. Maes (ed.), MIT Press (1990).
- [15] Henry Sowitzal (org), *Inroduction to Programming in Java3D*, SIGGRAPH 98 course notes 37 (1998).

About the authors

Márcio Lobo Netto is Assistant Professor at Polytechnic School from the University of São Paulo
E-mail: **lobonett@lsi.usp.br**

João Eduardo Kögler Jr. is Associated Researcher at Polytechnic School from the University of São Paulo
E-mail: **kogler@lsi.usp.br**

Emílio Del Moral Hernandez is Assistant Professor at Polytechnic School from the University of São Paulo
E-mail: **emilio@lsi.usp.br**

Fábio Roberto Miranda is Master's Student at Polytechnic School from the University of São Paulo
E-mail: **fmiranda@lsi.usp.br**

Address:

Department of Electronic Systems Engineering
Polytechnic School from the University of São Paulo

Av. Prof. Luciano Gualberto, trav. 3, 158
05508-900 - Cidade Universitária
São Paulo - BRAZIL