

GENESIS

A Virtual Environment for Genetic Optimization of Adaptive Agents

White Paper #1

GENESIS Project

Project Admin Team

mailto:genesis_proj@yahoo.com

http://www.geocities.com/genesis_proj/index.html

Keywords: Artificial life, genetic algorithms, neural networks, distributed systems, adaptive agents, evolving behavior.

Preface

The GENESIS project is an idea realized by a small group of researchers and postgraduate students from Greece, with common interests in artificial intelligence, adaptive agents and distributed systems. Based in the concepts of A-life, our goal is to create a useful, modular, robust, distributed virtual environment which will be used for the deployment of various kinds of "intelligent" agents. Using a common framework for both environment and agents' basic architecture, we should be able to deploy different kinds of agents with the one and common goal of survival through genetic optimization and behavioral adaptation to the environment. Using various control schemes for each agent class, from genetic algorithms, neural networks, fuzzy rule set, even linear optimization, we should be able to observe the effectiveness of each method in a competitive environment.

The first section describes the basic concept and the motivation for the project. The second section presents the detailed framework of the work to be done, including the goals and basic development stages. The third section presents some details regarding mainly the goals and directions of the design of the virtual environment and a generic framework for future agent implementations. The fourth section lists some of our current thoughts about the way that the basic framework of the environment can be realized. The essay closes with references to our current bibliography and links, along with the naming of the basic components of the system.

1. Introduction

When the first Tierra system was implemented, everyone familiar with the concepts of A-life was thrilled with the idea of a complete virtual world for digital creatures trying to compete with each other. The basic idea was to create a virtual platform (virtual CPU) which would simulate the way that digital computers work. In order to allow the digital creatures to evolve from generation to generation, arbitrary changes in their running code were allowed. Some led to improvement, most led to crash, all inside the virtual machine so that the real computer system could run unaffected. There was only one problem: the digital creatures did not seem to behave in a "smart" way. That is their basic function was to run, mutate a little and then try to run again, hoping that the change improved the way they interact with their environment.

Of course only few were willing to let hundreds or maybe thousands of digital creatures consume the resources of any computer, just to see how good they will do after hours of execution time. After all, CPUs was not so powerful and so cheap as they are today. However, recent developments both in hardware and software engineering gives us the opportunity to try and develop "smart" agents. A smart agent need not be able to play chess like a grand master, or solve a complex resource allocation problem in a few microseconds. It just has to learn to behave in a way that allows it to live longer, consume less, avoid danger and spread it's genes wider through many offsprings. Of course none of this behavioral adaptation can be passed to it's offsprings, but the chances of his survival, and with it the survival of it's genotype, are dramatically improved. The ultimate goal is always to survive.

Starting off with these basic principles in mind, our team decided to undertake a purely research project of developing a common platform for a whole new genre of virtual worlds, exploiting the latest advances both in implementation (programming languages, distributed computing) and in the broad area of "intelligent" systems (genetic algorithms, neural computing, evolving behavior, etc.).

2. Framework

2.1 Character

The GENESIS project is a purely academic work with research character and goals. The designs, code or results from this work cannot be used in any other way by any of the participants other than for the advance of science in a non-commercial way. All the people that work on this project will be more than happy to welcome any individual that wishes to take part and help any way he/she can. The project is just beginning and it would be a wonder if the current (small) group is gradually joined by users all over the world to create something wonderful. All work and ideas of the people working on the project is shared freely among them under the [Mozilla Public License \(MPL\)](#). The Mozilla Public License was adopted to cover issues of source code availability and modifications more efficiently than the [GNU General Public License \(GPL\)](#) and the [Artistic License](#) which were the original choice.

2.2 Goals

The motivation of a project like this is always the joy of creating and experimenting. However, a working system can be extremely useful in numerous areas, especially regarding the easy deployment of smart agents with modular control strategy.

The original goals set by the team are the following:

- Develop a universal platform for virtual environments, distributed and scalable over physical networks.
- Observe agent behavior in a competitive environment and compare with real life biological systems.
- Compare various control strategy methods in a competitive way, creating a profile and benchmark for known algorithmic techniques (GA, ANN, etc.) for game-like environments.
- Compare the significance of structural attributes (genetically inherited) in contrast with behavioral attributes (adaptation).
- Research the relative effect of the application of semi guided genetic evolution (Balwin effect) in contrast with random genetic evolution (Darwin).
- Develop new methods for the design and implementation of "universally adaptable" agents with pure evolving behavior, based on self organized strategy modules.

From the above goals, the first three can be characterized as near-future. The very first is really a platform-specific issue which we quickly decided to solve using the only truly system-independent programming language with support for distributed programming, which is of course Java. The basic programming platform to be used is pure Java, using only the original API from latest Sun's JDK version and excluding any extensions from specific development suits (like JBuilder or J++). This way a maximum modularity can be achieved and developers can share designs and code easily. All supporting libraries should be based on the JDK so that all developers can use and improve the common library framework.

The second and third goal is of course the heart of the project for the near future. It's ok to watch the agents move around, conducting battles or sleeping during night-time, but the true usefulness of such a system is to test various control strategies for the agents, simple or complex, in order to make them behave in a "smart" way. Of course, the goal is not to create a super-intelligent agent which adapts perfectly because it has perfect information about the environment (temperature fluctuations, resources allocation, etc.) or it's enemies locations at any time. A successful control strategy should be efficient for a wide range of environments, using only local perceptual informations for the agent's current state.

The fourth and fifth goals are actually the search for the significance levels for both structural (genetic) and behavioral (adaptive) attributes of the agents. A successful behavior leads to structural propagation, but every offspring should create a behavioral model from scratch. What if some of the behavioral (adaptation) information could be passed through genes to the next generation? A semi-configured behavior for the specific environmental conditions gives a dramatic advantage to a newly born agent, as long as the environment stays the same. But for a highly dynamic environment how much of the ancestors' behavioral information is

beneficial? The famous Baldwin Effect suggests a whole new model of genetic evolution of organisms, but how and when can be (should be) used?

Finally, the ultimate goal is to create methods to design and implement agents that behave like real world organisms. That is no one has programmed the way that their behavior evolves as they try to adapt to the environment. They rather try to learn more and more about the environment that they live in and learn to behave optimally over time, like avoiding dangers or staying close to resource-rich locations. An agent like this should be able to adapt the whole design of its control strategy, that is a self organized strategy, giving not just adaptive but evolving behavior. What a surprise to watch these little virtual creatures suddenly develop co-operative behavior never actually programmed or foreseen but the programmer...

2.3 Development plan

The GENESIS project has started and the initial design phase is currently under way since early June of 2000. The following plan is only an estimate of the distribution of work during the next months:

- Phase-A: [Jul-Aug] Project concept, goals, characteristics, architecture, general framework, platform. Team organization (web site, s/w repository, contact lists).
- Phase-B: [Sept-Dec] Environment & resources model design and implementation. Agent framework.
- Phase-C: [2001+] Agent designs & implementations, testing and evaluation (complete initial goal #3).

We hope that by the end of the year 2000 a fairly robust implementation of the environment should be ready and everyone would then be able to work in the area of agent's control strategy modules and algorithms.

3. Design Goals

3.1 Characteristics of biological organisms

All living beings are characterized by several attributes which distinguishes them from non-living things:

- Motion: find food, hide from danger, battle with enemies, mate.
- Structure and Functionality: organization, processes.
- Metabolism: use resources and energy to grow, move, interact.
- Sensitivity: interact with the environment, passive or active.
- Reproduce: genetically duplicate, mutations
- Stay in the same state, fight the environmental changes.
- Goal is to survive and reproduce, as long as possible.

The only actual difference between an artificially created entity that completes these seven criteria and a real world organism is said to be the reproduction via DNA based mechanisms, along with the last two attributes which usually do not apply to common

use man-made machinery. A refrigerator is functioning according to the 6th attribute, using energy to keep the internal temperature steady, but it does so to serve a purpose other than its own functionality (7th attribute). Although it is fairly easy to design a virtual agent that behaves according to the first five attributes, 6th is a tough one and implies the existence of an efficient control strategy, while the 7th implies that this control strategy should serve the agent's needs and nothing else, something that finally points to a self-organized evolving behavior with no initial designer (then the agent serves the designer's wishes and hopes, not its own need for survival).

3.2 Virtual entities design

The structure and functionality of virtual agents can be formulated in various ways, but the main issues here have to do with gene-based reproduction (reproduce) and self-contained resource management (metabolism). Also, a realizable agent should be able to communicate (sensitivity) and act (motion) in the environment only locally as in any real world biological system. All structural and functional attributes should be coded in the agent's genotype, but any information contained in the control strategy are part of the behavioral model and are volatile in regard with reproduction.

The most generic way to view an agent is like a state-transition, gene-based automaton. Each internal variable represents structural/functional or behavioral information, while processes and interactions alter the agent's state. The system's memory is integrated into the control strategy and it is used to improve the way the agent acts, and it is updated both from external events (sensors) and its own feedback. The whole entity is really a cell that gathers resources and uses them in an optimal way, in order to stay alive and reproduce as long as possible.

A virtual world can be modeled in minimal with some basic components:

- Time: all agents have internal awareness of "global" time so that their actions can be synchronized by the system.
- Space/Motion: all agents have internal awareness of relative movement according to changes in position (3-D)
- Energy: every function requires energy quanta, idle situations have energy (heat) leaks to avoid stationary agents.
- Materials: every agent has specific mass which can be increased selectively with resources to gain bonuses.
- Information: every agent can sense the environment, or can broadcast a signal.
- Integrity: Functional efficiency, degrading due to aging, battle damages, etc.

These basic components are bundled with the internal structure and functionality of the agents so that the five basic requirements for living organisms are met. The design, combination and implementation of these components should ensure a solid framework both for the environment and for the agents that live inside it. For example, an agent might be permitted to absorb energy ("heat") directly from the environment like plants do, but it should be ensured that the energy leaks due to process deficiencies force the agent to look for resources ("food") to avoid a completely stationary world. Aging ensures that deficiencies gradually become too severe and finally the agent is unable to complement its losses and dies out. Integrity is used to control the agent's priorities, that is the agent should be aware of any

damages caused by battle and try to repair them (using up resources) in order to restore its overall efficiency.

4. Implementation Issues

Although the design is currently on an initial stage, some implementation issues are worth noticed because they might affect the way the final design goes. In this section we present some ideas regarding the model design and implementation of the virtual environment & resources, and some parameters about the design of the agent framework.

4.1 Environmental model

- Each site is a virtual world, composed of a torrus-like continuous grid plain.
- All individual worlds make up a distributed virtual "cosmos" in which agents can migrate from world to world.
- Each world is composed of a 3-layer architecture: elevated terrain, resource allocation and agent allocation.
- The environment has ambient "heat" with periodic long-term, short-term and random (noise) fluctuations.
- Easy monitoring and parameter change through a robust GUI.
- Complete state storage and retrieval.

4.2 Resources model

- Each resource is a "food" quantum, used either for reconstruction or energy needs.
- Resources are scattered randomly on the (elevated) plain.
- Fresh resources have value of 10, degrading steadily to 1, destroyed (if not collected) at 0.
- Resource density and digression can easily regulate the degree of competitiveness among the agents.

4.3 Agents model

- Self-sufficient, autonomous.
- Act based on sensor signals and current state.
- Limited aspect and range sensors, bonuses (+/-) due to terrain elevation.
- Move on the elevated terrain, sense on level plain (no line-of-sight obstacles).
- Energy leaks always, severe on damage, growing with age.
- Initial mass can be increased to gain bonuses, but requires more integrity control.
- Battles occur when agents try to simultaneously collect same resource or breed with same target.
- Reproduction with 2 sexes (male/female), genotype combination, variable number of "eggs".

5. References

Here are some papers and links that will be used as guide for the system's design and implementation.

- "The Evolutionary Emerge route to Artificial Intelligence" - Alastair Shannon.
- "Evolving Novel Behaviors via Natural Selection" - A.D.Shannon, R.I.Damper.
- "Perpetuating Evolutionary Emerge" - A.D.Shannon, R.I.Damper.
- The Baldwin Effect (intro) - <http://ai.iit.nrc.ca/baldwin/introduction.html>
- Complex Adaptive Systems and Artificial Life - http://lslwww.epfl.ch/~moshes/alife_links.html

6. Appendix

This is the naming conventions that have been proposed so far for the basic components of the system (subject of change).

Project name:	GENESIS
Global environment:	VirCo
Local environment:	VirEn
Resource unit:	quark
Energy unit:	quantum
Repository unit:	gluon
Agent/Entity	cell
Male agent	m-cell
Female agent	f-cell

Last updated: 24-Jul-2000.

[GENESIS Project](#)
Project Admin Team
mailto:genesis_proj@yahoo.com
http://www.geocities.com/genesis_proj/index.html