Applied Digital Evolution

Attila Nagy
University of Debrecen
Institute of Mathematics and Informatics
nagyat@dragon.klte.hu

February 25, 2001

Abstract

As a relatively new discipline artificial life tries to create complex systems that are adaptive, stable, evolvable and so on. In one kind of experiments the evolving entities are assembly-like computer programs, so called digital organisms. The learning process of these self-replicating strings can be called digital evolution. Although this research is in its beginning phase and the results can be applied several ways. This paper tries to summarize the application possibilities which are currently known.

1 Introduction

This paper is a short summary of a bigger project. I started the development in September of 1999 as a continuation of two former projects (namely Tierra [3] and Avida [1]). My aim was to integrate and enhance the functionalities of the previous software systems. Now my research platform [2] – called Physis – can be used for both scientific and practical purposes.

As the research area is a bit new the terminology sounds quite strange. It seems to be a wild mixture of informatics and biology, so first I'll clarify to some extent the fundemantal concepts. However this clarification should be done without getting into the technical details. After that I'll enumerate the possibilities of applications in informatics, in biological researches and education. The research of digital evolution raises several deep questions which force us to revise our basic notions. This venture is far more than interdisciplinary. Finally I'll discuss one of these questions.

2 Digital organism

Although this paper is about the possible applications I should give at least an informal definition of the digital organisms.

DEF: A special program written in an assembly-like computationally universal language and the corresponding processor architecture. The specialty means that the program is able to self-replicate (copying its instructions to an other memory area).

We can call them digital organisms because they take on the role in the artificial world that natural organisms do in nature: they proliferate, metabolize, fight and die.

3 A very simple example

Although I really wouldn't like to get into the technical details but at this point it may be useful to examine one particular example in order to understand the concept of digital organism.

This organism is a nude-replicator: it doesn't do anything else except for self-replication.

```
count-f
zero
allocate
label-b
copy
inc
if-n-equ
jump-b
divide
label-f
```

Let's debug this short code-fragment!

The first `count-f` instruction simply counts the number of instructions to the next `label-f`.The result is stored in an offset-register. Actually the organism determines its size.

With the second instruction we put 0 into the stack then we allocate memory cells for the child. After that the copy-loop begins. In the loop an instruction is copied and the top of stack is incremented by 1. This counts the copied instructions and adjusts the address of the source cells. If the number of copied instructions equals to the value of the offset register the copy-loop will end.

At this moment we have the duplicated code. The `divide` instruction starts the "cell-division".

After successfull replication the parent program is simply restarted and the child can start working in an other memory location.

4 Digital evolution

DEF: A process in which the principles of evolution (natural selection with mutation as the main engine of development, struggle for life) are applied among the digital organisms in order to increase their complexity.

The main concept was originally described in the *Origin of Species* written by Charles Darwin in 1859. A correspondence can be made between the natural and digital process.

The limited resources of the environment correspond to processor time and the amount of memory. Mutation can be interpreted in the digital process as instruction-change during the copy-process and insertion and deletion during the proliferation. The organisms have to compete for the resources via faster execution and inventing new algorithms.

5 Applications in informatics

And now let's see how can we use digital evolution for practical purposes!

The most important task in informatics is to create efficient, adaptive and fault-tolerant software especially on the lowest levels.

Several methodologies (OO paradigm, UML and so on) try to tackle the problem that how can we handle complexity? Unfortunately most of them work only with the highest levels of a system. The code of an operating system consists of million lines. They had become fragile monsters. In human made systems complexity seems to be against efficiency and safety.

Evolution is a proven route: it was capable of creating the life on Earth. Animals and plants have all the desired properties and some of them are intelligent. They're complex but still stable and adaptive. Why don't we use the evolution for our purposes?

Basically there are two types of application: code optimization and code generation (creation).

6 Code optimization

The starting point is some existing code. This can be generated by a compiler from a higher-level programming language. Of course the compiler can optimize but that happens according to some general rules. For example device drivers would need very low-level optimization.

The existing code completed by a self-replicating part makes a digital organism. This organism then can be inject into a digital ecosystem and after a size-dependent time-period the optimized descendants can be extracted.

7 Code creation

We can start the experiment with a population containing only nude replicators but the environment of the organisms can be arbitrary complex: the environment consists of computable tasks. These tasks should be learned by the organisms. The organisms can pick up pieces of data (integer numbers in the current state of the implementation) and after processing them they simply write the result back to the environment. This process can be considered as some kind of metabolism. Data transformation is just like eating and digesting.

8 Application in biology

A digital ecosystem may be the ultimate research tool for an evolutionary biologist since there are serious problems in studying evolution in biology. The time scale simply doesn't allow us to observe the process. Meanwhile in computers the evolutionary process is many times faster.

The experiment can be repeated exactly the same way but with different settings of the statistical subsystem if we miss something in the first run by adjusting the random seed.

The observation doesn't disturb the process itself. This requirement can't be satisfied in biology: scientific investigations always have impact on the system itself.

The research platform is useful for genetics as well. Hypothesis in the theory of the development of sequence spaces can be quantitatively and instantly tested.

These facts are acknowledged by one the most significant periodicals in natural sciences: by *Nature* as it published an article about digital evolution. [4]

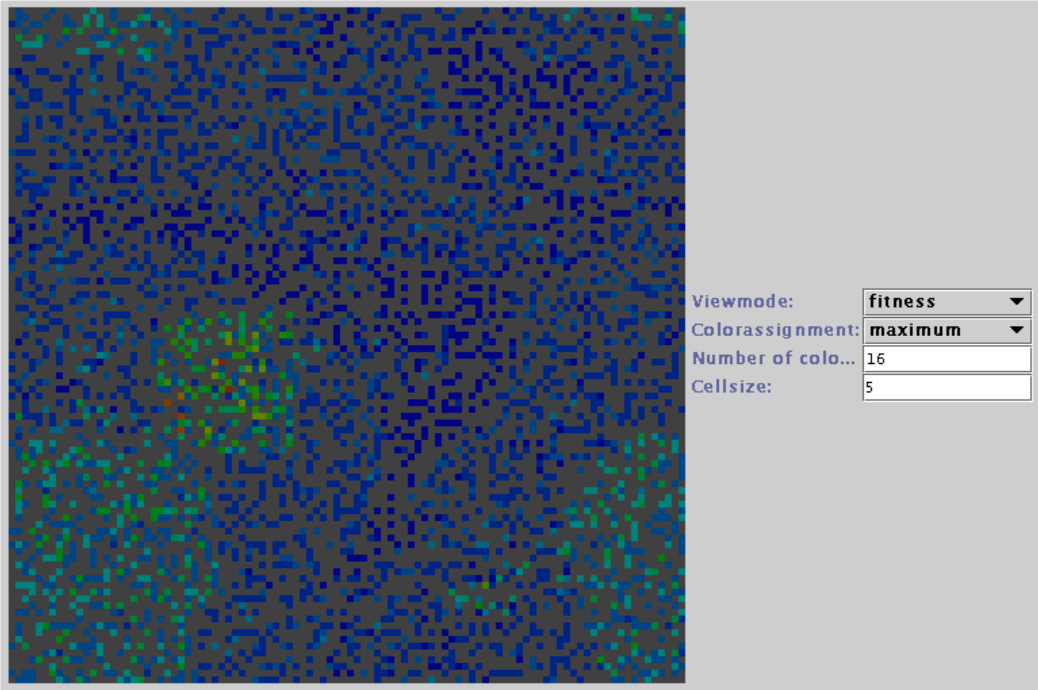

Figure 1: Physis in action

9 Education

Ultimate tool for teaching evolution This is a screenshot of the Physis system (see Figure 1). Here you can see the whole population: each little square represents one digital organism. The colorful ones are fertile. Green and red mean higher fitness.

Somewhat left to the middle an evolutionary innovation has just happened and the new code is propagating over the whole population.

This screenshot shows that how easily the evolutionary process can be observed real-time. The user interface allows the student to change the parameters on the fly.

Instead of reading about the fossils the student can play with the process itself in order to achieve deeper understanding of the phenomenon.

10 Is it simulation?

There are several deep and hard questions in this field. Let's see just one of them!

Is the digital evolution some sort of simulation? Both answers can accepted.

yes It's a much more "realistic" model of life than the traditional genetic algorithms. (There's no predefined fitness-function. Theoretically the evolution is open-ended.)

no What does the digital organism simulate? Viruses? Bacterias? Some of the researchers [6] including me say that they're independent life forms. Not in carbon-based chemistry but in a digital world. We call it artificial life which is also the name of the discipline.

11 The future development

In a scientific research the results should be verifiable. That's why the development is open and the source code is publicly accessible. [2] This development model will remain and the actual work focus on the following issues:

Creating evolvable processors : by creating a universal processor the evolution of the decoding system becomes possible. (The hardware industry has just started to use these idea in Crusoe processors. [5])

High level semantical analyzers The progress evolution renders the organism's code hard to understand for humans. That's why we need tools for examining the activity of the organisms.

References

[1] Avida. `http://dllab.caltech.edu`. Digital Life Laboratory.

[2] Physis. `http://physis.sourceforge.net`. The homepage of the open-source development.

[3] Tierra. `http://www.isd.atr.co.jp/~ray/`. The homepage of Thomas S. Ray.

[4] Richard Lenski Chris Adami, Charles Ofria. Genome complexity, robustness and genetic interactions in digital organisms. *Nature*, 400:661–664, August 1999.

[5] Alexander Klaiber. The techology behind the crusoe processors, 2000. `http://www.transmeta.com`.

[6] Thomas S. Ray. Artificial life. ATR Human Information Processing Research Laboratories, Kyoto, 1995.