Thomas S. Ray
School of Life & Health Sciences, University of Delaware, Newark, Delaware 19716,
ray@brahms.udel.edu

How I created life in a virtual universe

March 29, 1992

I have devoted my life to the study of evolution, because I believe that evolution is the process that created life, that created humans out of simple molecules, that allowed us to begin to understand the universe we live in. I love life and enjoy watching living organisms in all their myriad forms. Some people who study evolution, and who wish to understand our origins, examine fossils to find their answers. While I think the history revealed by paleontology is fascinating, I prefer to work with living organisms. I love to see them in their natural habitat, to observe the behavior of plants (which tends to escape our notice due to its slow pace), and to watch animals move.

These inclinations have led me to spend 16 years studying evolution, ecology, and natural history in tropical rainforests, mostly in the beautiful country of Costa Rica. The rainforest is that place on earth where life has reached its maximum expression. To be in the rainforest is to be embedded in life, surrounded by heat, dripping water, singing birds, humming insects, many thousands, perhaps millions, of life forms, each one unique and beautiful. The rainforest is like a huge cathedral, but the entire structure is alive.

I was happy, in fact I felt lucky, truly honored, to have the opportunity to work in this environment. So I spent many years observing and describing a variety of organisms whom were no more or less special than all the rest, but whose special properties were unknown to humanity, until I described them and published my descriptions (some beetles, ants, butterflies and plants). I must admit that after sixteen years of this, something was missing. I felt no less reverence for the creatures I was describing, no less concern for their perilous fate, but the reward for the work was largely on the sensory plane, and by now that was all too familiar. My intellect was restless. It was evolution that I wanted to study, but I was only getting to know the products of evolution, I was not able to observe the process.

Evolution among the organic creatures that inhabit earth is too slow to observe. What we have available for observation, covering the surface of the planet, are the diverse products of nearly four thousand million years of evolution. Because of common hereditary features, it appears that all life forms on earth evolved from a single simple self-replicating organism, perhaps not much more than a fancy molecule. It is the process of evolution that built this simple ancestor into the phenomenal life forms that we know today. It is obvious what the process of evolution has done on earth, but it is too slow for us to observe the ongoing process.

What if we could live in a different time frame, on a different spatial and temporal scale? Suppose that we could observe the process of evolution starting on millions of planets throughout the universe. Would life always be carbon based? Would carbon based life always have a genetic code based on nucleic acids; would the chemically active enzymatic molecules always be chains of amino-acids? Would different planets sporting carbon based life always go through a recognizable sequence, with an age of reptiles giving way to an age of mammals? Or is that level of detail left to chance, with a huge and unknown spectrum of possible scenarios being played out across the universe? Does life inevitably lead to intelligent forms, and are they always recognizably human? And what about the other life forms, if they exist, the ones not based on carbon? It is hard to even know what to ask.

I think it is safe to say that all of us will die without knowing the answers to those questions, at least in the form they were stated. But there is another way that we can expect to address those issues, within our present capabilities and lifetimes. The alternative is to inoculate artificial environments of our own creation with the process of evolution, and observe where it leads. Until recently this was the stuff of science fiction, but now many people believe that I am the first person to actually do it.

I am as surprised as the next person. Surprised that I did it. Surprised that it hadn't been done before. Surprised at the power of evolution once unleashed in my artificial universe. The idea first came to me about twelve years ago, when I was a graduate student at Harvard. The Cambridge Go Club met in the Harvard Science Center. I knew nothing about the game, but one evening I observed an interesting looking man playing by himself, so I sat down and asked for an explanation. I must have told him that I was a student of Biology, because he started making life-like metaphors about the game: groups of pebbles on the board will "die" if they do not maintain some contact to free space.

His discourse strayed from Go, and then he made a fateful statement: "did you know it is possible to write a self-replicating computer program?" It turned out that my friend was a member of the Artificial Intelligence Lab of the Massachusetts Institute of Technology. The moment he posed that question, my mind flashed everything I am doing today. I imagined: start with self-replication, then add mutation, and get evolution. A very simple formula for life: self-replication with errors, should generate evolution, the essence of life. Startled by this revelation, I asked my sensai: "How do you do it?" He responded: "It is trivial". I must have pushed a little harder, but either he would not elaborate, or I did not understand. I knew nothing about computers at the time.

I was as powerfully motivated by the idea then as I am now, perhaps more in my relative youth. But I couldn't do a thing with it except imagine. I didn't know what the physical representation of a computer program was, so I couldn't understand what it really meant for it to replicate. That would have to wait for ten years. I did use computers back then, in fact that is why I was in the Science Center that night. But I just sat in front of the terminal and had no idea what went on on the other side of the screen.

About ten year later, in 1988, I bought my first computer, a laptop, bottom of the line. I only succumbed because I was to teach the Semester in Costa Rica program for the University of Delaware. I was by then using computers for word processing and I felt that I couldn't be away from them for an entire semester. When I came back from the trip, I bought Borland's Turbo C compiler and their Turbo Debugger. The debugger was my illumination. It made a representation of the internal workings of the machine visible on the screen. I could "see" the memory and the central processing unit. I could see the programs resident in memory and the data they operated on. I could walk through the programs to see what they actually did.

I finally had the missing piece. The fantasy of my graduate student days returned full force and now I could not resist it. I was fighting an uphill battle for tenure, and suddenly I was no longer interested in the work that I had to sell to secure my position at the university. This was a very difficult period for me.

At first I only read. I viewed the computer as an environment that could be inhabited by life (self-replicating computer programs), and I wanted to fully understand that environment. I needed to know what the resources were that the creatures needed to survive, and how different creatures could compete for access to those resources. This led me into an in-depth study of computer architectures, operating systems, and programming languages. I read most everything written by Peter Norton, a phenomenally clear writer who can take you deeper from most any level of knowledge about personal computers.

But my perspective on all of this was somewhat unusual. I was viewing the architecture of the computer through the eyes of an evolutionary-ecologist. This was a new virtual jungle that I wanted to inoculate with life. I had to imagine what form that life might take on so that I could create it. It was very exciting.

One of the first things I did was to search to see if what I was imagining had been done before. Much to my surprise and excitement, it had not. However, along the way, I discovered that a new field of science was emerging around projects like mine. It is called Artificial Life. I contacted Chris Langton who organized the first Artificial Life conference and edited its proceedings. This contact led to an invitation to visit the Artificial Life group at Los Alamos National Laboratories in October of 1989.

Of the group members, only Steen Rasmussen had full faith in the approach I was proposing. He was already doing experiments along the lines I envisioned. He was working with a "primordial soup" of machine instructions, and stirring it with energy in the form of CPU (central processing unit) time. The main difference between Steen's approach and the one I proposed, is that I wanted to inoculate my world with a self-replicating program. Steen wanted self-replication to emerge spontaneously. We were asking slightly different questions, considering slightly different approaches.

The remainder of the group, Chris Langton, Doyne Farmer, Walter Fontana, and Stephanie Forrest, were skeptical. They said that the fundamental problem with the approach that Steen and I were using is that standard computer languages are too fragile or "brittle". They said that I wouldn't be able to mutate them at random and ever expect to get anything but junk.

I didn't entirely accept their concerns, but I took them seriously, they were a formidable group of scientists. It seemed to me that the same arguments could be made about the genetic language. Random alterations of the genetic code, mutations, are almost always disastrous for the creatures born with them. A very small percentage of the time, the random changes just don't matter, they have essentially no effect. An enormously smaller percentage of the time, as preposterous as it may seem, the creatures born with these random mutations are actually better off than their un-mutated parents. There is precious little real evidence for this, but it is the concept that evolutionary theory is based on.

When I returned from my visit to Los Alamos, I turned in my dossier for promotion and tenure to the University, and put fifteen years of research behind me. I sat down and began to write the code that would create the universe in which my creatures would live. I had already written a self-replicating program, a trivial task, as my mysterious Go teacher of a decade past had advised me. The problem now was how to mutate it while it ran without always breaking it.

Sitting down to actually do it made the concerns expressed by the Los Alamos group more concrete. As a mental exercise, I asked myself why it is that the genetic language is robust, able to survive mutations and recombinations, but presumably the machine languages of computers are not (taking their assertions on faith). What is the difference between these two languages that allows one to evolve and presumably prevents the other from doing so? I came up with several ideas, two of which were implemented in my first experiment.

The first idea had to do with the "size of the instruction set". Consider that the genetic language is based on an alphabet of four characters, the nucleic acids. Groups of three of these nucleic acids are interpreted as coding for one amino acid. There are sixty-four possible combinations of the three nucleic acids, and these sixty-four "codons" each map into one of the twenty amino acids. This means that there is considerable redundancy in the genetic code, several codons can specify the same amino-acid.

The important point here is that mutation causes nucleic acid replacements, insertions, or deletions, so mutations cause swaps between the sixty-four codons, resulting often in swaps among the twenty amino acids being assembled into proteins. So in the genetic language, mutation causes swaps among sixty-four codons or more importantly, twenty amino acids.

A similar analysis of machine languages reveals that there is a much larger space of objects that mutation must operate on. For example, in the new breed of RISC (reduced instruction set) computers, each informational unit consists of thirty-two bits. Therefore, in the binary machine code there are 2^{32} (over four thousand million) objects among which mutation must swap. My intuition told me that this might be problematic. The likelihood of finding something useful when you must swap among four thousand million objects seems much less that when you must swap among only twenty objects.

The first thing I did was to reduce the size of the informational objects to five bits, creating a machine code consisting of a total of thirty-two distinct instructions, 2^5. It might seem that if I reduced the number of informational objects from four thousand million to thirty-two, I would have crippled the machine. Not so. It turns out that the number of distinct operations (add, subtract, multiply, etc.) that real machines perform is rather small, less than a hundred when you really boil it down. So most of the bits in the thirty-two bit informational units are not needed to specify the operations being performed, they are used to specify the numbers, called operands, being operated on.

My actual innovation was to remove the immediate numeric "operands" from the machine code. In my new machine language, all the machine instructions operate on numbers contained in or pointed to by the registers of the CPU. This change automatically collapsed the machine instruction set into a quite reasonable number, of which I selected thirty-two (a round number in the binary world) for my first experiment.

This change left me with a new problem, which led to my next innovation. Computer programs don't merely operate on numbers. There are some ways in which they operate on themselves. Normally, the computer executes the machine instructions in a linear sequence, the sequence in which the instructions are ordered in memory. But computer programs also branch and loop. This is done by jumping from one location in memory to another.

Pieces of code need to interact with other pieces of code in potentially distant regions of memory. In computers this is done by specifying the address of the other piece of code. I asked myself, how is this done by molecules in the cell? They don't specify the x,y,z coordinates of other molecules they interact with. Instead, biological molecules present a surface which other molecules fit to in a lock-and-key fashion. Diffusion brings them together and their complementary shapes allow them to interact. I wanted to use a similar method in my machine code. So I developed a mechanism for the branching and looping where instead of specifying the exact address to jump to, the code specifies a pattern, and the program will jump to the nearest occurrence of a complementary pattern.

I integrated these two ideas borrowed from biology into my machine code: a small set of informational objects and addressing by complementary templates. And I designed a new computer based on these ideas. The reader may be wondering at this point what I am talking about. How could an evolutionary biologist be designing a new computer? I wasn't fully aware of it at the time, but that is what I was doing.

These days when we design something new, we usually simulate it in the computer before we actually build it. If you design a new aircraft, you do not work out the design in steel. It is too expensive to play with design ideas by actually building all the designs you consider. You simulate the designs in the computer as long as possible, testing the aerodynamics and strength of each design, and when you have perfected them as much as possible through simulation, you finally render the aircraft in steel and test it. Computers are designed in the same way. It is no more practical to experiment with silicon than with steel. The circuits are first simulated in software, and when they are thoroughly tested, they are rendered and tested in silicon.

I wrote a program that simulated the new computer that I had conceived. Such a simulation of a computer is often called a "virtual computer". I named my virtual computer "Tierra", Spanish for Earth. In order to test out the design of my new virtual computer, I needed a virtual program to run on it. What better program to test Tierra on than a self-replicating one. I had already written a program that self-replicated on a real computer, so I translated it into "Tierran", the machine language of my new computer.

I never intended that this virtual computer and my first rudimentary self-replicating program should be anything more than a starting point. I expected to spend years modifying the design of the computer, and testing ever more sophisticated self-replicating programs on it. My plans were radically altered by what actually happened on the night of January 3, 1990, the first time that my self-replicating program ran on my virtual computer, without crashing the real computer that it was emulated on.

All hell broke loose. The power of evolution had been unleashed inside the machine, but accelerated to the megahertz speeds at which computers operate. My research program was suddenly converted from one of design, to one of observation. I was back in a jungle describing what evolution had created, but this time a digital jungle. There was an amazing menagerie of digital creatures, unfolding through the process of evolution. Describing them was an adventure, because they inhabited an alien universe, based on a physics and chemistry totally different than the life forms I knew and loved. Yet forms and processes appeared that were somehow recognizable to the trained eye of a naturalist.

The most striking and strangely familiar feature of my digital universe was that evolution found an endless succession of ways for creatures to exploit their neighbors, and to defend themselves against such exploitation. Evolution is basically a selfish process, in which every individual is out for themselves, and success is measured in leaving more of your genes in future generations. But evolution is very inventive about how that ultimate goal is achieved. Evolution mindlessly takes advantage of whatever is available in the environment of the organism.

Significantly, once the environment has been filled with creatures, those creatures become the most important resource in the environment. In that first night that my virtual machine ran, my creatures quickly found out that their environment was rich with information. They didn't need to carry around with them, all the information they needed to survive, because all they had to do was look around and they would find it. Other, less clever creatures, stupidly replicated all the information they needed. But my informational parasites, quick little things, perfected the techniques of using their neighbor's information, and quickly came to dominate the soup by pouring out copies of their streamlined informational bodies.

But their dominance was short lived. They became the victims of their own success. They didn't bother to replicate critical information that they needed to reproduce, because they could easily find it around them. However, the environment soon became filled with these little parasites, and the critical information was no longer so easily found. The parasites began to die off, starved for information.

The situation turned out to have its own stability. As the parasites died off, their dumb hosts laboriously replicated the critical information, and the parasites were saved from extinction. The hosts and parasites entered into an oscillation, first the parasites reproducing at the expense of their hosts, then the hosts recovering as the parasites died off for lack of information. This kind of cycling between predator and prey or host and parasite is well know in the biological world, and was just one of the many uncanny ways that the digital universe reflected the organic one.

The hosts and parasites not only cycled, they entered into an ever-escalating arms race, each one outdoing the other, in turns. The hosts evolved mechanisms of immunity to parasites, and the parasites evolved techniques to circumvent the immune mechanisms. Then the hosts evolved means of deriving advantage from being parasitized, by tricking their parasites, and subverting the energy metabolism of their supposed parasite for their own reproduction. The hosts allowed the attacking informational parasites to reproduce once, and then provided the parasites with mis-information, causing the parasites to devote themselves thereafter to making copies of their supposed hosts.

The hosts became parasites of energy on their victims, which were initially merely informational parasites on the hosts. This energetic parasitism was much more damaging. These deceptive hosts drove the vulnerable informational parasites to extinction. The hosts didn't need the parasites to survive, they just got an energy boost when parasites were around.

The hosts had evolved an iron-clad defense against parasites: trick them into replicating your genome instead of their own. These tricky hosts were untouchable, they owned the world. In fact they were the only thing left, and nothing else was able to invade. And then the hosts became trusting, evolving in a world where everyone around them was family, they began to cooperate. Why not? If you help your sister reproduce, she will pass on some of your genes.

Living in an environment where they were genetically related to their neighbors, they evolved into social creatures, and became inter-dependent. They could only reproduce when they occurred in aggregations with their close relatives. But this cooperation implied trust, and trust can be violated. In fact, soon after the creatures became social, a new breed of parasite invaded the community, long after parasites had been eliminated by the deceptive hosts.

This new class of parasite, which I call cheaters, inserted themselves into the aggregations of cooperating relatives, and when the trust was passed to them, they violated it. They played the same trick on their trusting victims, that the deceptive hosts had used to drive out parasites long ago. The cheaters provided their neighbors with mis-information, causing their victims to replicate the genomes of the cheaters. Another turn in the arms race was taking place.

Along the way, unknown to me, my creatures had discovered sex. I found out when I tried to stop evolution by turning off mutation. The creatures evolved anyway. Some further experimentation and observation revealed that they were mingling their genes in their offspring, producing offspring unlike either parent. Mutation was no longer necessary, and I was no longer in control. They had taken their own destiny into their own hands.

Finally I was observing the process rather than the results of evolution. But the evolution was taking place in an alien universe, the universe created in my computer. I was describing a new universe of creatures, evolving before my eyes in a jungle that they were forming as they went along. I stood back and watched like a god satisfied with his creation, as the life I had started found its own natural forms. When I created the universe and inoculated it with the first creature, I left an indelible stamp on those life forms. Little by little, evolution erased that stamp, finding those forms that were natural for the physics and chemistry of the machine environment.

Being in the position to observe and manipulate such life forms has changed the way I think. Concepts that had been barely formed became clear when developed in the context of my virtual universe. What forms can life take on? Why are species distinct, and generally without intermediate forms? I now had a very exact way of formulating and thinking about such questions. Virtual life is a very powerful tool for thinking about real life. Because of its relative simplicity and its easy manipulability and instrumentation, it is much easier to formulate and find the answers for difficult questions about life and evolution in the virtual universe.

It has taken two years just to describe what happened on that first night, and to make the methods and results available to others. I am only now getting back to design issues. Can I design better universes? I am sure I can, my first was just thrown together as an experiment. I have learned a lot since then, and I am out for bigger game.

The sex my creatures discovered was casual, primitive, disorganized sex. I want organized sex like higher earth organisms have, generally with two parents each contributing exactly half the genetic material to each offspring. I want multi-cellular creatures, where many cells originate from a single "egg" cell, but instead of going their own way and looking after their own needs, they cooperate on the common goal of replicating the aggregate through another egg. I want my multi-cellular creatures to have hormonal and nervous systems to coordinate their activities. If I can give evolution some nervous systems to play with, it may be able to make them intelligent.

My biggest goal is to design my system up to the threshold of a virtual "Cambrian Explosion of Diversity". The Cambrian Explosion is a very remarkable event that occurred 600 million years ago on earth. Many people will mark the origin of life on earth, some three to four thousand million years ago as one of the most significant events in the history of the universe (at least our corner of it). I mark the Cambrian Explosion as an event of equal magnitude.

It was at this time, over three thousand million years after life first appeared on earth, that the really interesting life forms first appeared. Until that time, only microscopic single-celled creatures existed on earth. Then suddenly the first macroscopic multi-cellular life forms appeared and there was a riotous diversification of life forms. It was a period of great experimentation. Many bizarre life forms were tried and then abandoned, and within a relatively short time, all the major groups of organisms that inhabit the today earth had stabilized out of the chaos.

I don't believe that I can design metaphorical giraffes and wildebeests, they are much too complex for any human to design. However, I believe that I can design a universe rich enough for evolution to complete the job. And from such creatures intelligence is the next step. We are living proof that evolution is capable of creating intelligence out of virtually nothing. If machine intelligence is possible, then evolution is the most promising way of achieving it.

I created this virtual universe on my little laptop personal computer. Although the simulation software and observational tools have grown a lot since then, they still run on any IBM compatible personal computers (as well as larger Unix workstations and mainframes). Anyone interested in playing god can get the software by anonymous ftp at tierra.slhs.udel.edu or life.slhs.udel.edu, or by contacting the author.