An Idea for a Project
A Universe for the Evolution of Consciousness

J. D. Horton

May 28, 2010

To the reader. This document is mainly for myself. It is for the most part a record of some of my musings over about two weeks in November of 2007, as the ideas developed and changed. I gave a presentation to the Faculty of Computer Science on April 1, 2009, called *An April Fool's Project*, after discussing the idea with a few people. Nothing here is fixed yet. Any feedback would be appreciated.

1 Initial Musings

How can you investigate consciousness? Build an artificial world with beings in it that can observe and change their environment, and see if you can get people to agree that the beings have consciousness, or at least exhibit many properties of conscious beings. As I have no solid idea what consciousness is, or how to develop it, I think we need to use evolution, to combine new ideas in ways that nobody expects.

Build an artificial computer environment with two types of objects: space objects, and beings. The beings inhabit one (later maybe more) of the space objects. The space objects are connected to form a graph, so lets call the space objects **nodes**. Maybe cells would be a better term, but I do not want space to look too much like a cellular automaton to a reader. This space is more general. Pairs of nodes are connected by **tunnels** (edges, connections, arcs?).

Nodes must have resources of some kind. The first one that comes to mind is **energy** (grass, food?). This can be a nonnegative real number, or maybe integer. Others come to mind: light or dark, other resources like minerals. I do not want to restrict them.

I do not know what to call the beings, but I think that a more descriptive term is needed. I have rejected sheep, ants, bacteria, viruses. Maybe **bugs**?

The bugs must be able to observe at least some of the resources of the node that they are at, at the very least know what the energy is, at least approximately. The bugs must have (or quickly develop) some kind of emotions, at the least they must feel **pain?**. Happiness is required at some point, but maybe that can be developed. The bugs can **eat** some of the energy at a node, and always be losing energy to the environment somehow, possibly thru whatever their actions are. The bugs must be able to move from node to node, which means they must be aware of the tunnels at their node. If they have below a certain level of energy, this should be bad (pain), and they should eat to avoid this pain. If there is no more energy to eat at a node, then they have to move to another node. But I want them to figure that out somehow.

The bugs must have a brain of some kind. They need a memory, and probably a neural network. The neural network can be their memory, in that it tries random actions, and those actions that avoid pain will be encouraged, and those that cause pain will be discouraged. Possibly even pain might not be built in; bugs that do not eat starve to death and remove themselves from the simulation. Bugs that learn to eat must learn to feel hunger to avoid starving to death. Evolution at work.

Since bugs can die, they must also procreate. They must have the ability to split in two, presumably when their energy levels exceed some value (possibly set by evolution), with each child taking a portion of the energy of the parent. The child could be identical to the parent including the brain-neural network, or possibly just take values close to it, or maybe with some random changes. But there should be a possibility of becoming more complicated, by adding an ability and/or adding a requirement.

Bugs must be aware of other bugs. Can two bugs inhabit the same node? Maybe not at first. Or if they do, maybe they can exchange some of their neural network brains, or combine them, or have children, or fight, or one eats the other. All these possibilities should be included in the simulation.

The nodes must also be able to change, but more slowly than the bugs. Possibly every once in a while a tunnel can split in two, placing a new node in between. Maybe sometimes nodes that are close together but are not connected can become connected by a new tunnel. An inflationary universe. This is likely to be ommitted in the early phases of the system.

Occasionally a new resource can be added at a node. When this happens, it should be (slowly?) distributed along the tunnels to the other nodes. The bugs need to have the ability to sometimes recognise new resources, which presumably must become requirements for some of them, but also give them more powers. For example, to see resources at neighboring nodes. Possibly

to allow bugs to send "bots" to investigate neighboring nodes. I would prefer that the resources and powers come from the system rather than being added by people, but I do not know how to do this for all of them.

This universe requires a lot of parallel computational power. Every bug requires its own thread, possibly more than one. The inflationary universe idea also requires that space have a thread, or maybe many.

I can also imagine allowing the universe to use the internet for some of the tunnels. Could we get a community of users? This would be required to get enuf computer power to do all that I would like to do.

The system requires that the user be able to watch individual bugs as well as to gather statistics about all the bugs, and the universe.

So what ideas need to be expanded?

1. Node structure.
2. Bug structure.
3. Resources of nodes and how they change.
4. Powers of bugs and how they change.
5. Brain structure.
6. The human interface.

2 Node Structure

2.1 Internal structure — resources

A node just has a set of resources, and remains the same unless acted on by something. A resource needs a name, and a value. Can resources be restricted to integer? positive? boolean? I see no reason why they cannot take any value, so I guess they could even be an object, as in OOP. They actually form a set, but I suppose that they could be stored in a vector. But this is perhaps not best, as different nodes may have different resources, so the index of a resource for different nodes can be different. A set of objects therefore.

2.2 Tunnels — the topology of space

A node also has tunnels. The tunnels should be distinguishable, so they can be stored in a vector. New tunnels, if ever for a particular node, should be

added to the end of the vector. Some bugs may be able to distinguish one tunnel from another by their index, although the simplest bugs should not be able to do so.

Tunnels are not objects. They only occur in the tunnel lists of their adjacent nodes. It is not clear that tunnels always are reversible. It is possible that the graph of space is directed rather than undirected.

Nodes can be connected in any way one wishes. If space is one-dimensional, the space graph is a path, or perhaps better, is a simple cycle. I doubt that this would be an interesting enuf space. It could be a 2-dimensional grid, again probably better to be a torus, Klein bottle, projective plane, or higher 2-dimensional manifold. My original preference was for a randomly connected graph of low degree. But a low-dimensional manifold satisfying some symmetry conditions might be a better place to start.

2.3 Inflation

Because we want the universe to evolve and get more complicated, more nodes have to be added to the network. I suggest that the changes be small random events, probably one new node at a time. My suggestion is that a tunnel splits to form a new node of degree 2 in its middle. Then no node other than the new one has to change its structure, except for one node in the list of tunnels.

But this will make the average degree go down, unless the world is one dimensional. So we also want to be able to add tunnels, again randomly. But the probability of a new tunnel should be much higher for low degree nodes, and for nodes that are closer together. Whatever rule is chosen, it will eventually dominate the topological structure of space. So this has to be thought about carefully.

2.4 Inhabitants

My first thought was that there should be only one bug that can inhabit a node at a given time. But that is clearly too restrictive. So there must be a set of inhabitants. I expect that the set should always be very small, but I do not know if that is necessary. In any event you need a list of pointers to the inhabitants. Only inhabitants in the list can make changes to the resources of a node.

I expect that there will be many different types of bugs, that most will only be background for the “high-level” bugs that we are are most interested in.

3 Bug structure

This is a difficult question. One essential requirement is that they inhabit a particular node. But I do not want to preclude the possibility of being in two or more nodes at once. However maybe one node must be the "home" node. This should not be possible at the start, but the possibility should be left open.

Bugs must have some kinds of attributes. Maybe just one is essential, the life source or energy store. As the bug acts, the store decreases. Different actions might use different resource stores. Possibly all actions uses the energy store.

Bugs must have some powers, that is, things that they can do. They must not be allowed to remain perfectly static forever, because then they may as well be dead. (Or maybe space can be filled with "dead" bugs?)

Examples of powers:

1. Feel: Observe the resources available.
2. Eat: Consume a resource, which turns into resource store. Possible even to eat another bug, but that would be a less common power. I suppose that another bug could be a resource to a poweerful more advanced bug.
3. Grab: Pick up and carry part of a resource. This is not the same as "eating", which puts the resource into a store.
4. Drop: Drop a resource.
5. See: Observe the tunnels available; later this may include observing the node at the other end, both resources and tunnels and inhabitants possibly, and later recursively to nodes further away.
6. Move: Move along a tunnel to a new node.
7. Drop a resource in a neighboring node (could be the basis for communication).
8. Excrete: When resources are consumed, it may be that they leave a residue of some kind. The residue may change slowly back into the original resource. If we assume that the universe needs conservation laws, this is required. I do not know that conservation laws are required if we have an inflationary universe, but it might be convenient to have them anyway. This can also mean that it is possible to have bugs

that can identify other bugs by their scat. If all resources have this property, then resources come in pairs.

9. Die: If a bug ever has no stores, then it must be dead. The stores then get added to the resources of the node directly, or as excreta, or both.

10. Procreation—splitting: A bug can split into two. The children will split the stores. This could be done on a more or less equal basis, or on an unequal basis. The children would normally have the powers of the parent, but there must be some random chance of changes including adding a new power, or deleting an old one. If it were on a less equal basis, presumably the larger child is more like the parent than the smaller child.

11. Procreation — sexual: Should be possible.

12. Combining: Maybe two bugs could combine to form a larger bug. This could be a way to procreation, combine and then do two splits.

13. Remember: A bug can remember what it does and what the situation is like, but different ways are possible. See brain.

Every new resource gives six more possible powers, not including some power that is not yet thought of. Presumably many resources will have some other special power that they make possible, such as improving sight or ability to procreate or to combine or improving the brain.

I presume that powers are implemented as methods of a class.

There has to be a chance that powers combine to form other powers. Using powers together should increase the chance of combining.

3.1 Using powers

A bug has to "decide" when to use powers. I suppose the brain should be in control here. The brain will be asked at each step whether a power should be used, and the brain will reply with some probability that it should be used, with some distribution of how much (when eating, picking up, dropping, excreting etc.).

What I was going to write here, I am moving to the brain section.

3.2 Changing powers

Powers only normally change when a new bug is born. Combining and splitting may be a very powerful way to do this. One has to be careful not

to allow powers to propagate too quickly. But there can be a chance of a new power occurring at random in the population. Should this be restricted to splitting and/or procreating? Not necessarily.

Maybe longlived bugs should be given the chance to get new powers.

4 The brain

This is perhaps the most important part of the project, and possibly the hardest to choose among all the possibilities. But since we have isolated it from the rest of the simulation, maybe we can have fast evolution here.

The "brain" is the routine that decides what the bug is going to do next. At the start the brain will consist of probabilities that each of the original powers will be acted upon. Every power must have a probability, but possibly zero. In fact at the start I would say zero. But also every probability must have a small chance $\epsilon > 0$ of changing, by some small random amount, $\alpha > 0$. Eventually as probabilities cannot be negative, this will make all probabilities positive. Both α and ϵ itself must have a probability of changing, and of splitting, so that different powers have different ϵ's. We could start all the bugs with 0 probability of doing anything, and a single common $\epsilon (= 0.001?)$. Presumably α could be a (normally distributed?) random variable. If α were not normal, then α as a random variable would avoid the need for ϵ, because that could be included in the randomness of α.

4.1 Memory

The direct memory could be a queue of all observations and actions the the bug has done. The problem is that over time, the memory will get too big. So memory will have to be limited to some small number of time units, (or of actions,?) such as 5, or 1000. This is a short term memory.

There will be a need for a long term memory too. Maybe it should just record "important" events. Maybe it should be a neural net, and just record general "feelings" about the impact of actions, in a way that affects it future choices. Probably both.

There is a great deal of work to be done here.

4.2 Emotion?

The brain must somehow correlate the memory with survival. So that situations of observations/actions from a long time ago must be considered

positive. Also procreation. But I want the bugs to learn this on their own. We might build such things in later.

We can use the immediate memory to be inputs to a neural net whose outputs are the probabilities of certain actions. Those bugs, whom the actions help, survive and procreate; evolution at work. But this does not act like the training of a neural net. It does not seem like happiness.

An idea: have an attribute, a value that specifies if the bug is happy or not, or at least says whether the bug likes or dislikes the current situation. Happiness is more of a long term thing.

When the "like" value is negative, the bug should take action immediately. It may be that "freeze" is an action.

By remembering when it is in likeable situations and correlating it with recent actions could be a way to develop a desire to repeat the actions. This may be a better way to decide if powers combine. Thus here is where memory and emotion interact.

5 The Human Interface

As one of the things I want to do is to develop a big universe, other people must use the system. To make them want to, there must be features that make it interesting to watch and to work with. I have few ideas here, but it is a very important aspect.

1. Have something that shows the universe from a bug's veiwpoint, as well as examine every aspect of a bug. This could be rather interesting in some cases. For example, a bug wandering in a four-dimensional universe could be rather interesting. The bug would have to have some unusual powers.

2. Have something that gives universe-wide statistics. The universe can be restricted to one machine, or it could be as wide as the internet. There could be interesting problems with gathering data from across the internet.

3. Have some nice interface so that a user can create their own universe, with any topology that they like, and populated however they like. But they can also create their own rules for expansion etc.

4. Have some nice interface to develop bugs with a set of powers of their own choosing.

5. Have an interface to create new powers for bugs. This has to be limited enuf so that monsters do not appear and destroy the rest of the universe. Maybe some limitations must be built into the universe, or maybe each piece of the universe must be able to shut itself off from the rest, in case of trouble. Or maybe just shut itself down, or maybe block bugs with superbug powers. On the other hand this may be the most interesting part of the project for some users.

   SECURITY IS A BIG PROBLEM!

6. An attractive website is needed.

7. To advertise the system, scientifically interesting results are needed, so that it gets coverage in the popular science literature. We need to get someone doing work in evolutionary theory perhaps, or in a cognitive science of some kind. Besides this is really the main reason for the system to be developed.

8. The other way to advertise is thru the gaming community. Can interesting games be built on this idea? One solitare game would be to design a bug that can survive in a difficult universe. Or that develop a certain power without being given it. Or to have competitions on developing bugs, and putting them together to see which can win in battle — bug wars. Or to design bugs that cooperate with others without losing their identity. None of these seem to be attractive.

6 How to Proceed?

It would be nice to become familiar with:

1. Artificial life simulations.

2. Evolutionary simulations.

3. How the brain works.

4. Online games?

5. Name for the project?

An initial universe could be a 1-dimensional cycle, with a single resource. Can bugs learn to move and eat? This could probably be proved analytically.

I must find out who, if anyone, is interested. It will require many people to develop a system like this.

Postscript: Two masters students have joined the project. Michael Francis joined in September 2009, and Eckart Sußenberger in May 2010. Aaron Michaux with whom I discussed the idea early on, is also working on the project in a less formal capacity yet.