Escaping Addresses

Too much of a good thing can be bad.

An electrical signal can be presented to many elements by connecting all of the elements to a single circuit node. This often requires elongating the conductor that realizes the node so that connections in a physical circuit are possible. One advantage of signaling with electrons is that they can carry a signal useful distances in less time than elements normally react. A collection of similarly connected nodes can serve as an *address buss* through which numerical addresses are presented to many elements so that one can come to attention. The technique works well and is at the heart of both processor and memory design on a variety of scales. Address busses make computers a logical machine for when they are properly clocked we can reason knowing all elements have been considered.

Biological systems signal through *codes* such as the genetic code or the anti-idiotype network. They signal along *transmission lines* such as the myelin sheathed axon and branching *trees* like the dendrite. They *switch* signal expression with *flags* and *transport* messengers along *routes*. They organize along *fields* such as the concentration gradients that cue differentiation. But they don't (or don't often) buss addresses. That is because the signal carriers (atoms, not electrons) are too sluggish to make an address buss useful.

As we increase the number of elements on a chip, in a cabinet or on a network above a threshold even the speedy electron slows to a level that makes busses impractical. Then the system designer is challenged to maintain the fiction of addressability. Registers, arrays, caches, servers, and singletons are all artifacts we maintain as if addressing were always as easy as it once was. The co-evolution of our hardware and software designs have enshrined the address buss so firmly in our computational world that only a restart can displace it.

Imagine a world where the very first computers were manufactured from twelve inch wafers, deployed into an environment where every person had hundereds, and that they were all connected, world wide. Would we choose to make addresses presented on a buss a fundamental idiom? I think not.

When we abandon address busses we abandon the logical determinism that comes with them. We can no longer be sure that all elements have been considered. Such is already the life of the network engineer. But in programming methods and language design and even processor architecture, we hang on to logic as if it were all that were possible. This is not because no one has pointed to another way. Rather, it is because no community of circuit, system, network, language and methodology designers have looked the same way at the same time. -- WardCunningham

---

Notes

We will know that we have escaped the confining logic of addressability when we consider proliferation to be a sign of success rather than failure. See OnceAndOnlyOnce.

Similar arguments have been made regarding the Von Neumann bottleneck. I find both the bottleneck and the usual alternatives non-biological and seek here to find something more fundamental to replace.

?TomRay chose complementary matching as an alternative to offset addressing in Tierra's instruction set. His motivation was to make small changes have small effects so as to not break things hopelessly with every mutation.

The following comments were inspired by Ward's remarks although they do not follow from them directly. In simulating 'biological' agents, the choice of space (and hence 'addressing') seems critical. Many simulations use a 2D grid: this is nice, because you can easily display the simulation on a 2D pixel grid aka 'monitor'. But it's quite limited.

I've moved to using 3D grids (and 3D graphics software for display). A cubic cell has common faces with 6 others rather than 4, but the increase in complexity is higher than the numbers suggest. Biology makes good use of 3D, but it's not clear how much could be achieved in 2D.

If our agents are computer code/data, we can use memory addresses as our 'space'. There are many ways of doing this. Tierra uses a linear space. But a 32-bit word can also be interpreted as 1, 2, 4, ..., 32 dimensions and, as Kanerva has shown ("Sparse Distributed Memory", MIT Press), binary hypercubes of high dimension have interesting properties. Distances are easy to compute with the Hamming metric and 'spheres of influence' (Hamming distance < r) have nice properties.

On a slightly different note, Ward mentions the difficulty of self-replication in 2D. I suspect it's even hard in 3D! But biological organisms don't self-replicate, they lay eggs instead. The only self-replication problem they solve is copying the DNA, which is approximately 1D. -- PeterGrogono

---

I strongly concur with Peter. I think life is able to self-replicate but only at the very lowest level. And this is the level that I think is possible to mimic through computer programs. Lisp, for exmaple, has the very unique feature that code is data, and this property allows it to "encode" in metadata, the structures that are to emerge in a morphogenesis process.

More on this at BiologicalRealm

MikeBeedle

Edit

Last edited April 8, 2002
Return to WelcomeVisitors