

The Organisation of Complex Systems

H.A. Simon

```
@incollection{compref303,  
  author = {Simon, HA},  
  year = {1973},  
  title = {The Organization of Complex Systems},  
  booktitle = {Hierarchy Theory - The Challenge of Complex Systems},  
  editor = {Pattee, HH},  
  publisher = {George Braziller},  
  address = {New York},  
  pages = {1-27}  
}
```

In this paper he will attempt to be the “pion” that binds the elementary particles “physicists” and “biologists”. Ho ho ho!

This paper looks as though it'll draw heavily on Simon's other paper “Architectures of Complexity” since he'll be looking at and describing complexity's hierarchical structure.

Hierarchy

In this paper by “hierarchy” he means a recursive kind of structure: a box within a box within a box. There might be more than one box within a box and each of the smaller boxes contain yet more boxes. “*nature loves hierarchies*” since they're everywhere in nature.

He lists 4 major hierarchies that can be observed:

1. chemicals → molecules → atoms → protons neutrons electrons → elementary particles
2. organism → tissues → cells → macromolecules (I guess this means proteins?) → organic compounds
3. inheritance → genes → chromosomes → DNA → etc.
4. human societies → organisations → small groups → a human → cognitive programs in CNS → elementary information processes.

The last one has been referred to as “the hierarchy of final causes called the mind” by Walter Pitts. It used to be only present in people but now we have computers that process information and run programs, this makes them useful for study!

“since programs are much more readily accessible to study in their artificial than in their natural environments, we have learned enormously more about them in our generation than in all previous history.”

“For this reason the digital computer is taking its place ... as an experimental system of the greatest importance”.

Some Theory of Hierarchy

He can (apparently) show that “*the time required for a complex system containing k elementary components, say, to evolve by processes of natural selection from those components is very much shorter if the system is itself comprised of one or more layers of stable component subsystems than if its elementary parts are its only stable component*”.

He illustrates this point with a story about watchmakers. This is the same story he uses in “architectures of complexity”. He estimates that a system with k elementary components, with s components on each level (on average) then the expected evolution time is proportional to $\log k s$, so

to evolve systems of 10^{25} atoms from systems of 10^{23} takes as long as evolving systems of 10^3 from systems of 10 atoms.

This theory of extra evolution speed being caused by complexity is supported by the fact that in nature there are large complex systems everywhere (all plants and animals for instance).

Near Decomposability

interactions decrease in strength with distance, so a “particle” will have the strongest interactions between its neighbours, so *“a system is likely to behave either as made up of a collection of localised subsystems or to form a more or less uniform “tissue” of equally strong interactions”*.

He then goes on to talk about how if we make observations on a system for time T, the interactions that are slow (i.e. take much longer than T) are effectively constant and can be ignored, similarly if we can't observe anything quicker than τ seconds then interactions that are faster than τ are invisible and ignorable as system noise. He calls systems that do this “nearly decomposable”.

Hierarchies in Computing Systems

He uses some program called EPAM as an example of a large complex system. It's written in some obscure old language called “IPL-V”. In this language he can write a routine with primitive commands and then give that routine a name to make it a function. That function can then be used as a primitive command(!) there can even be as many functions as you like(!!!), EPAM is up to 10 levels deep. The basic IPL-V commands themselves aren't very primitive because they are translated into some machine code specific to that computer. EPAM isn't concerned with how IPL-V is translated it just knows that its commands will run on that computer *“EPAM has a 'meaning' that is independent of the particular machine language in which it is expressed”*. The machine code itself is still complex because it then interacts with the physical logic devices which implement the behaviour that EPAM wants. EPAM is independent of the computers it runs on, the computer itself is indifferent to what program is run on it, it just knows how to turn machine code into noughts and ones. They communicate with machine code, which is the minimum amount of information needed for them to interact. *“the art of subroutines, in writing complex computer systems, consists in discovering the points of cleavage at which the least information needs to be passed from one subroutine to another”*.

Loose Horizontal Coupling

There is “vertical” separation in that there are high frequency and low frequency subsystems and they are distinct from each other, but is there separation between subsystems that are on the same hierarchical level?

Loose horizontal coupling is different to the loose vertical coupling. The vertical coupling allows stable sub-assemblies to be “treated as simple givens” i.e. irrelevant to the overall larger structure *“only their equilibrium properties affecting system behaviour at the higher levels”*. The loose horizontal coupling allows the sub-assemblies to operate independently of the others, its inputs and outputs are relevant to the larger system behaviour. These couplings can be observed on all hierarchic levels, an example is the systems in the body. The nervous system is unrelated from the circulatory system but there may be minor interactions between the two somehow, neither system cares how the other does something, just so long as it gets done. Internal modifications / improvements to one of these systems will not disturb the other. This is yet more support for the idea that complexity quickens evolution.

Production Systems

This just repeats the idea about stable subsystems with loose horizontal coupling being good. In a production system if products A and B are independent but A uses B as an input then you can make lots of B and use it as a buffer so that A is independent of fluctuations in B.

A thought: the reason computers have evolved so quickly since the 50s is perhaps because they're made of stable, interchangeable subsystems? Eg harddrive, motherboard, processor, graphics cards etc etc. Each of those is developed independently but interact between each other in a standardised manner. Consequently people can work on them independently and they each will get better over time without breaking the computer (ie all these things stuck together) as a whole.

Alphabets

Here Simon presents the idea that having a limited number of items for a subsystem to choose from (an alphabet) is somehow significant for “evolutionary success”. For instance a milk protein has 9541 atoms, all of which are C, H, O, N or S, five of the 92 natural elements. From these 92 elements are billions of different proteins, but then at the level up from proteins they have an alphabet of billions, so only certain hierarchic levels are alphabetic. The higher level stuff might be referred to as a language. A language should have the following characteristics:

1. *sufficient variety in its primitive processes so that no meaning is absolutely excluded from expression*
2. *sufficient flexibility in its rules of combination so that any nuance can be expressed by building up composite structures.*

Which I suppose means that there needs to be as little variety in element and interactions as possible, but there should be enough there to represent everything you'll ever need.

Summary: Loose Coupling

“Everything is connected, but some things are more connected than others”

The hierarchic structure allows higher level stuff stability and “functional efficacy” (ability to do its job) to be independent of their components.

Reduction

One idea is that reducing something to its fundamental parts allows you to understand it. The atoms and whatnot can be described by a really big equation which would be far too hard to solve. There is a second idea that constructing laws that describe the relationship between hierarchic levels is the way to go. This school of thought develops bodies of scientific law eg particle physics, molecular chemistry, biochemistry etc. This fits in with the idea of “near decomposability”. This is a bit of an inexact science though.

He illustrates the point that the second idea is better by saying that he could supply the IPL-V code of EPAM or the machine code, which one would be more useful to you? Not even the machine code though but the electrons in the computer and a complicated equation about how they move on average.

“Scientific knowledge is organised in levels, not because reduction in principle is impossible, but because nature is organised in levels, and the pattern at each level is most clearly discerned by abstracting from the detail of the levels far below”.