Emergence, Hierarchy, Complexity, Architecture

How do they all fit together? A guide for seekers after enlightenment…

Derek Hitchins

Introduction

What is *emergence,* and what has it to do with systems and systems engineering? It certainly seems to stir emotions: on the one hand, we have engineers, calling themselves 'systems engineers,' saying that it is something to avoid like the plague, equating it with bad engineering. On the other hand, system designers, systems architects, social scientists, etc., suggest that systems engineering is *fundamentally about* creating emergence. They can't both be right: or, perhaps, engineers and 'systems engineers' are really quite different in how they think and what they do...

The word complexity similarly stirs emotions. Some say it is to be avoided, almost at all costs, others say that it is inevitable, and yet others suggest we really need complex systems to cope with complex problems – without defining what *complex* means…

Hierarchy presents a similar issue: is it something real, or imaginary – a mental perception? Is hierarchy the same as architecture, just another way of looking at architecture, or something completely different?

(W)holes

To make some sense of these terms, we need to come to grips with a little systems theory. First, what is a system? Basically, a system is a whole something, anything. Examples:

- A gambling system is only a system if you know *all* of the things you have to do, in the right order, at the right time, to make it work; knowing only part is not enough, you have to know the whole – *process,* or *procedure*.
- A manned trip to the Moon will not work without men, rocket, space vehicle, support, and so on. Leave any part out and you wont get there and back: you need the whole - *shebang*.
- A fighter aircraft with all of its equipment, engines, weapons, sensors, etc., but without a crew is not a whole – and it cannot 'do' anything. It cannot start up, taxi, takeoff, climb, look for a target, intercept, return to base, land. All it can do is sit in a hangar, and leak fuel and oil into strategically placed drip trays. Add the crew i.e., complete the whole *interceptor*, and all sorts of things become possible.

If there is a chunk obviously missing from something, or the 'whole is incomplete,' then you don't have a system: and that something can be a set of ideas, a piece of technology, an organization, a set of government departments, a team, and so on.

Gestalt, et al…

Back in the early 20th Century, researchers noticed that some wholes did not appear to behave quite as expected. One scientist experimented with still photographs, taken at short intervals, of a galloping horse. He found, on flicking rapidly through the stills, that he could see a 'moving picture' of the horse galloping[1]. And he asked the question: 'where does this apparent motion come from, when all we have is a set of still pictures?' Answering that question led to those 'what the butler saw' machines at the end of the pier, then to the cinema and eventually to TV.

But, where *does* the apparent motion come from? Watching TV, you see successions of stills presented at about 25 frames (50 fields) per second. If you had the relatively short visual persistence of, say, a pigeon, you would just see the succession of stills. With we humans, it is different…

---

[1] He could have learned a lot from school kids who draw pictures on the corner of their school notebooks and flick through them to make moving pictures. The whole idea may have come from being bored with Latin declensions!

Meantime, back with photographs of the galloping horse, researchers identified something they called *gestalt*: confusingly, this does not translate directly from the German, but it means something like 'shape,' or 'pattern.' The idea was that, somehow, there was more coming out, emerging, from the whole than could reasonably be expected by looking at the separate parts. And they called this phenomenon 'emergence.' In the vernacular, this became 'the whole is greater than the sum of the parts.'

And that turned out to be far from new. Back in the days of those Ancient Greeks, Aristotle (384-322BCE) gave us:

> *The whole is greater than the sum of its parts;*
> *The part is more than a fraction of the whole.*
>
> Composition Laws

Note Aristotle's use of 'whole.' What might have happened if 'wholes/systems' had continued to develop from Aristotle's time, instead of waiting to be rediscovered in the 20th Century? And, before we leave Aristotle, what do you think he meant by the second line of that quote?

Explaining emergence…

Not content with rediscovering emergence, 20th Century researchers tried to *explain* it. Some wholes, some systems, had properties, capabilities and behaviours that could not be exclusively attributed to any of their rationally separable parts. The more they looked, the more they found these 'irrational' phenomena. Self-awareness emerged from the human brain. Chemical compounds behaved nothing like their constituent elements…

As a very simple example of emergence, consider sodium, an alkali metal; in its natural state it is rather unpleasant, fizzing and burning in water – great fun, though, in the chemistry lab at school. Consider, too, chlorine: pale green gas with a choking effect on human lungs – no fun at all. Separately, these two elements are inimical to human life: together, they are essential to human life. Provided, that is, the elements are brought together in the right way to form the compound sodium chloride, or common salt – that translucent crystalline substance with which we are so familiar, and which occurs naturally.

How can translucent cubic crystals 'emerge' from a compound of an alkali metal with a greenish gas? We learned at school that a common salt molecule forms from an atom each of sodium and chlorine: when these two atoms interact in the right way, the resultant molecule has a particular pattern of electrons orbiting in its outer shell; that gives the molecule, and hence the crystal of salt, its characteristic appearance and behaviour.

So, emergence in this instance derives from the bringing together of different parts in such a way that their dynamic interactions result in properties, capabilities and behaviours of the whole that are not exclusively attributable to either of the rationally separable parts. And it seems that is a fairly general explanation of emergence, i.e., that emergence 'arises' from the dynamic interactions of parts within the whole.

Emergence and Not-Really Emergence

Let's look at that idea in simple everyday terms. Think about a ship, with its hull, engines, screws, cargo, crew, etc: it will have an all-up weight (AUW). Is AUW an emergent property? Not really. Although it is a measure of the whole, you could still identify all the constituent parts, most of which have no mutual interactions anyway, and add them up without calculating anything more or less than the sum of the weights of the parts.

So, what about a radar on that ship, comprised of a transmitter, a receiver, a timer, an antenna on a turntable, etc., etc., all interconnected and interacting as they should? Oh, and don't forget an operator and someone to look at the screen and interpret the picture… Is there any emergence there?

Well, sort of... The whole (radar, that is) is able to detect, locate, plot, track, identify and report ships and planes. These might be called emergent capabilities, capabilities of the whole, since none of them is exclusively attributable to any of the rationally separable parts. The transmitter transmits, but nothing else. The receiver receives, but that is all. And so on. Only when all the parts interact dynamically, cooperating and coordinating their actions, do capabilities emerge. But, as any self-respecting radar engineer will tell you, there is nothing magical or mysterious about these 'emerging' capabilities; the actions, interactions, coordination, timing, etc., all have to be carefully worked out constructed and controlled. So, more hard graft than magical-mysterious emergence.

Note that the radar's 'emergent' capabilities are described in a language (track, identify, etc.,) that has no meaning at the level of the parts (transmit, pulse, compress, synchronize, modulate/demodulate, etc.) This 'different language' turns out to be characteristic of emergence…

However, genuine emergence, the real deal, is also strongly associated with nonlinearity: the radar's behaviour is generally thought of as being linear: it is certainly calculable using linear mathematics. So while we might like to think of a radar's capability (track, identify, etc.,) as being emergent, it may be stretching the point somewhat. On the other hand, it is not unreasonable, and is in line with systems theory, to consider that the whole radar, operator(s) included, exhibits the emergent property of *unity*—provided it is a unified whole, of course.

But—and this is important—there may be other emergent properties, too, of a more significant nature. A radar may (or may not) exhibit 'stealth,' 'integrity,' and/or 'coverage,' for example which are very much properties of the whole. And it is often these rather more 'organic' emergent properties, capabilities and behaviours that provide that extra 'something' that system designers seek to 'design-in.' And, at times, they *can* seem a bit magical mysterious, or tricky… How do you design-in 'integrity,' for instance; or stealth?

Complexity and Hierarchy

Open wholes

Most, if not all, systems are also *open*. Which means that 'things' can enter into the whole, pass through the whole, be processed within the whole, and exit from the whole – like food passing through your alimentary canal. Or, in *systems theoretic* terms, an open system exchanges energy, material and information with its environment – and especially with other wholes/systems in the environment.

The idea of openness can be difficult for those who like to think of a system as having a fixed boundary inside which things exist, function, perform, etc. A biological cell has a boundary, inside which things exist, function and perform, yet at the same time 'stuff' permeates through the cell wall, energizes the cell functions, gets processed, and the waste from metabolism permeates outwards, again through the cell wall. A human cell, any biological cell, is a factory. An industrial factory, say one for assembling automobiles, behaves remarkably like the cell. Parts, subassemblies and so forth enter through the factory gates, get progressively assembled in a series of linked processes, and the assembled vehicle goes out the factory gates. Cell and factory are both open systems. All systems are open in some degree, else we would not know of their existence. Engineers may consider systems as closed: systems engineers do not.

Open systems, by their nature, interconnect with other open systems, forming networks and networks of networks, such that each is resourced, each acquires throughput, each works on its throughput, while others utilize the outflow from each. Open systems interact with others dynamically, and adapt to such interactions in a never-ending dynamic complex: which, if we were to let it, might overwhelm us in complication and complexity.

Hierarchy

Looking at a real-world network of open systems, it is generally possible to see that some work together in mutual harmony: those that work together form a candidate 'greater' whole, so put a conceptual boundary around them—but, don't sever any connections, since 'stuff' must still pass through that conceptual boundary. Then look for other open systems, similarly engaged in harmonious interactions, and put a conceptual boundary around them too. Soon what you see is not a network of many small wholes, so much as a new network of fewer, conceptually bound, greater wholes. Now, working with these larger, interconnected wholes, you can repeat the exercise to get even fewer, but still larger interconnected wholes… See Figure 1. (Note that these networks, or clusters, of open, interacting systems should, in each instance, form a whole, i.e., have a (local) sense of wholeness or unification.)

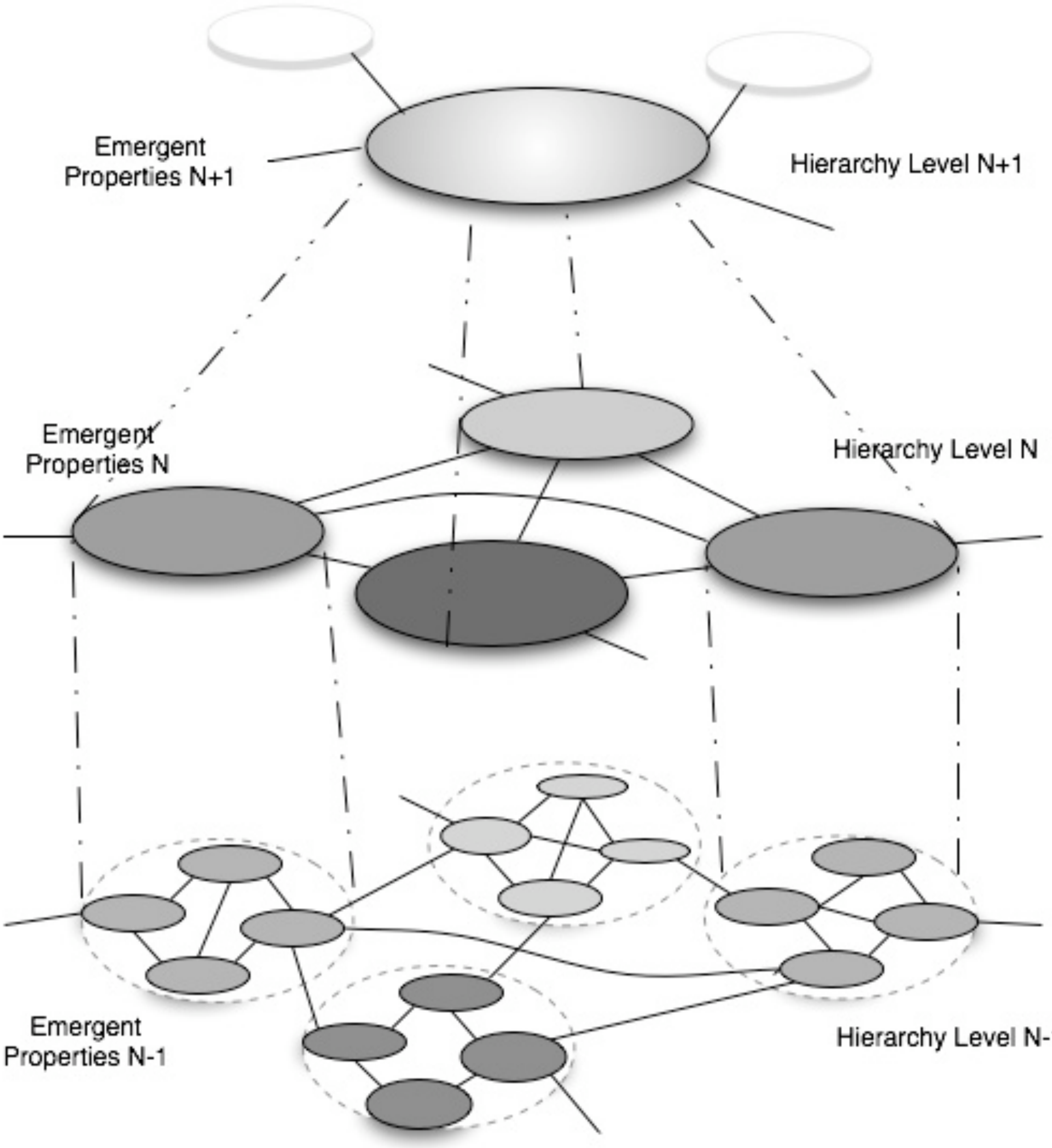

Figure 1. Systems Hierarchy Model. The conceptual diagram shows three levels of systems hierarchy in perspective. At the bottom are shown 16 interacting wholes, open systems. These have been grouped into four sets, each forming a 'greater' whole, such that four wholes, open systems, may be perceived in the centre level. Each of these four 'contains' those systems at the lower level, N-1. Again, the four wholes in the centre are grouped into a single whole, open system, at the top – Hierarchy Level N+1: this whole might be referred to as a Containing System, since it 'contains' the four wholes at Level N, which in turn 'contain' the sixteen wholes at Level N-1. Complexity has been 'managed,' since the single whole at N+1 may be easier to comprehend than the sixteen interacting, networked wholes at level N-1. Alternatively, it can be argued that the whole at Level N+1 'contains' more complexity than each of the four wholes at Level N, which in turn 'contain' more complexity than each of the sixteen wholes at Level N-1.

So, have we achieved anything? Well, yes, we have started on the road to creating hierarchy of whole systems within whole systems within whole systems. For we can 'cap and label' the conceptually-bounded wholes from the previous paragraph, and we find ourselves looking at a perceptually-simpler picture of open, bounded wholes, mutually interacting with other bounded wholes…

Getting complicated? Not really. On the contrary, we have just set out on the path towards managing complexity by 'encapsulating' it: the complexity is still there, but now it is hidden under its cap and label. You can, for example, think about a passenger aircraft as a whole, without having to think separately about the flight crew, the cabin crew, the engines, the airframe, the powered flying controls, the avionics, etc.,

etc. Sure, the whole aircraft may be complex, but for the most part you don't need to engage with that complexity – you can think about the aircraft as a whole, one of many, perhaps, in a fleet.

Or, think about a business organization. It has people, formed into teams, formed into groups, formed into departments, formed into divisions, which together form the company. That is a hierarchy. At each level, there are unified open systems interacting with other unified open systems. And, if you think about it, you will see that at any level in the hierarchy, the interacting open systems will be mutually different, yet complementary. (If they were all the same, they might constitute a collection or association, but not a system.)

Looking 'down' from the 'top,' the CEO may see a handful of divisions, but have no need to see scores of groups, hundreds of teams and thousands of individuals. He/she knows they are there, but they are encapsulated within divisions, which have their distinct properties, capabilities and behaviours. Why try to think about the different activities of thousands, when you can think more easily, and much more effectively, about the activities of the divisions…?

So, in a major aerospace business, you might have several divisions: space, land, sea and air; or, perhaps space-air, land-air, sea-air divisions. (Why choose the second set over the first?) At each level in the hierarchy, you will expect to see a complete set of divisions, departments, groups, teams, or whatever: the set has to be complete at each level, else it will not constitute a unified whole—no whole, no emergence. (At the very least, no emergent unity…) For instance, a team requires people of different skills and capabilities – even different personalities – to be classed as a team: and a good team exhibits emergent properties, capabilities and behaviours that tell the world that it is a good team.

To construct a hierarchy, think about a large electronic device, say, the radar mentioned above. Start at the level of discrete components and create a hierarchy with several levels, the whole radar being at the top of the tree. To start you off, components might be grouped into 'boards,' boards into 'modules,' modules into… uh, your turn. Now go up one level more, above the radar… and now up another level? Now, looking at what you have done, ask yourself: have you actually *changed* anything; has your perception of the radar changed in any way; what have you achieved?

Systems Science

At this point, it might be worth reminding ourselves about Systems Science, variously billed as 'the science of wholes,' and as 'the science of complexity.' Systems science seeks to understand complex systems, and their behaviour, as a whole, i.e., without explaining the behaviour of the whole by explaining the behaviour of the parts, which is reductionist, of course.

A classic example might be the sand pile – a classic experiment in which grains of sand are dropped on to a flat plate mounted on a sensitive chemical balance. As successive grains of sand land, they start to form a pile, or cone, which gradually rises until it reaches a point at which 'avalanches' of sand occur, with some grains falling down the pile and off the balance. The number of grains in each avalanche may be calculated by observing the loss of weight associated with each avalanche.

As grains of sand continue to be added, the pile may rise above some critical height, only for a fresh avalanche to bring the pile down again, perhaps below the critical height. Over time, the height of the pile varies above and below, but always averages out at the critical height. This is self-organized criticality.

As might be expected, avalanches come in various sizes, as measured by the number of grains per avalanche: few are large; many more are small. No surprise there. However, in plotting the size of avalanches against the frequency of avalanches on a log-log scale, a straight line is observed—and that *is* a surprise. It seems there is some rule, or law, underlying this behaviour. Not only are there more small avalanches than large ones, but the pattern can be described by a simple power curve:

$$Y = aX^b$$

The first thing to note is that this rule or law describes the behaviour of the whole (sand pile, in this case) without resorting to any detail about individual sand grains and their interactions. Second, if you were to build a simulation model of the sand pile, grain by grain, the resulting model would be unlikely to behave as the real sand pile behaves. Third, the rule or law does not really help you to predict the next event: for instance, knowing the recent pattern of avalanches does not really help you predict the size or timing of the next avalanche. Indeed, the behaviour of the sand pile may be described as 'weak chaos.'

Systems science, then, looks at wholes, and provides explanations of the behaviour of wholes without resorting to internal actions and interactions within that whole – which may be exceedingly complex. And systems science is interested in 'behavioural isomorphs,' too, i.e., systems that may be quite different, but which may be expected to behave in analogous ways. For instance, self organized criticality, weak chaos and the associated power curve may be observed in:

- frequency of earthquakes of particular magnitudes,
- frequency with which comets of a certain size enter the earth's atmosphere,
- pattern of share price movements on the stock exchange,
- distance between cars on a crowded motorway,
- 1/f noise occurring in an electrical conductor,
- population in the Old Kingdom of ancient Egypt
- …and many, many more.

All of which should suggest that there is a quite different way to understand complex behaviour – one that does not require us to undertake prodigiously complex analytical and modelling tasks, which would, in all probability, give us the wrong answer anyway.

And, once you have recognized this concept of self-organized criticality, you may find yourself looking around you at work, at play, out shopping, etc., etc., and you will observe systems of all kinds either in a state of self-organized criticality, or moving towards it. In business, success often leads to self-organized criticality as demand starts to outstrip supply, and as the supplier 'ups his game' to match the demand…

Similarly, in the Battle of Britain, where the Royal Air Force was outnumbered by the Luftwaffe, the RAF employed radar, networks of observers on the ground, and control and reporting centres to anticipate Luftwaffe raids, and generally to neutralize them—just: it was self-organized criticality, again.

The Systems Model

We now have the basis of a model that brings together open systems, emergence, hierarchy and complexity. Consider:

> *"... the general model of organized complexity is that there exists a hierarchy of levels of organizations, each more complex than the one below, a level being characterized by emergent properties that do not exist at the lower level. Indeed, more than the fact that they 'do not exist' at the lower level, emergent properties are meaningless in the language appropriate at the lower level. "The shape of an apple," although the result of processes which operate at the level of cells, organelles and organic molecules which comprise apple trees, and although we hope eventually explicable in terms of those processes, has no meaning at lower levels of description. The processes at those levels result in an outcome which signals the existence of a new, stable state of complexity—that of the whole apple itself—which has emergent properties, one of them being the apple's shape."*
>
> P. Checkland, Systems Thinking, Systems Practice, John Wiley, 1981

Complexity

Systems thinkers tend to agree that complexity is something to do with systems, but they cannot agree what complexity is, let alone whether it's good, bad or indifferent. Some say that it is about a complex of varied parts set in intricate arrangements. For my part, I suspect that it stems from three indivisible aspects: variety, connectivity and tangling/folding: something with lots of identical parts, like a brick wall, is not complex.

Unlike the bricks, the parts of a system vary one from another, however, e.g., to perform different functions or processes. The more functions/processes a whole performs, then the more complex it is likely to be –

but, on the other hand, it will probably be more capable and more adaptable as a result, and it may be able to exhibit more behaviours. There must be many different interaction pathways, too, so increasing both connectivity and the *variety* of that connectivity.

Last, but not least, tangling and folding. If a whole consists largely of a set of linked functions/processes connecting inflow to outflow, then you might expect some sort of linear arrangement. For most systems, that would be impractical, and the various processes/functions end up being fitted into a sphere, a box or a building: something to do with maximizing the volume-to-surface area ratio? Processes and functions are arranged to fit into these often-arbitrary containers, resulting in the interconnections being folded, bent back on themselves, and tangled.

Hence:

$$\text{Variety} \cup \text{Connectivity} \cup \text{Tangling} \Rightarrow \text{Complexity}.$$

… yet, as we all suspect, complexity is largely in the mind of the beholder: once you get used to something, it starts to seem progressively less complex; which may seem to make nonsense of this simple equation.

So, is complexity bad, good, or indifferent? From the above, it seems likely that the more complex a system, the more varied and adaptable is its capability likely to be, the greater will be the degree of internal energy expended in sustaining its throughput, the greater will be its ability to accommodate change in its environment, and so on.

On the other hand, people who seek to understand, or create, complex systems, face a more daunting task… So, perhaps it's not complexity, *per se*, that is the issue: perhaps it's the ability of designers, builders, managers, organizers, etc. to manage the complexity that they perceive in the first place.

Systems engineering is supposed to be about managing complexity: as opposed to some of the so-called standards for systems engineering, which seem to proliferate complexity and complication, rather than manage it. But then, somewhere along the line some folks confused systems engineering with *engineering management:* we are talking here about *systems* engineering – i.e., the conception, design, creation and operation of wholes.

The 5 Cs of Systems Design

The 5 Cs form a useful set of systems design principles:

- **Complementary**. The parts/subassemblies of a system complement each other to create a whole, i.e., the set of parts is complete, comprises a full complement.
- **Co-operative**. The parts of a system act and interact cooperatively and harmoniously within the whole.
- **Co-ordinated**. The functions and processes within a system coordinate and synchronize their actions and activities to create requisite capabilities, behaviours and synergies
- **Contributory**. The parts/subsystems contribute, separately and together, to the objectives of the whole – this defines their value in the context of the whole.
- **Concinnity**: The parts/subsystems are constructed, configured and conformed to synthesize a dynamic, balanced whole.

… and then there is homeostasis. Homeostasis is the tendency toward a relatively stable dynamic equilibrium between interdependent elements... Design for homeostasis is essential for the viability of the created whole in its future operational environment.

The Essence of Systems Engineering

Which really leaves the question of what systems engineering is really about, and how do you 'do it?'

Fundamentally, systems engineering is about solving problems: whole[2] problems, usually fairly complex ones. So:

Systems engineering is the art and science of creating
whole solutions to complex problems.

If that seems a bit vague, remember that systems engineering is about so very much more than technology. Systems engineering can potentially take on just about any problem facing the team, the company, the organization, the nation… even the globe?

The essence of systems engineering is in selecting the right parts, bringing those parts together in the right way, and in orchestrating their actions and interactions to create requisite emergent properties, capabilities and behaviours.

… where the requisite emergent properties, capabilities and behaviours are those previously determined as needed to solve some problem or resolve some issue.

Which leaves the definition of a 'system' outstanding:

A system is an open set of complementary, interacting parts with properties,
capabilities and behaviours emerging both from the parts and from their interactions
to synthesize a unified whole.

The Systems Approach

To see why there is a need for an approach of any sort, consider an open system, interacting with, and adapting to, other systems within its environment. The open system is comprised of open, interacting subsystems, which similarly interact with, and adapt to, each other, and to external inflows and outflows.

Consider now making a significant change to any of the subsystems. The change will affect, at least, its outflow, which will change the inflow to other interacting subsystems, directly and indirectly, which will change the behaviour of those other subsystems, which will change their outflows, some of which will impact on first subsystem that was changed.

So changing any one subsystem is likely to impact every subsystem directly or indirectly, and hence will change the whole. But, the whole is interacting with other wholes in the environment, which will mutually change and adapt, so that the change to the original subsystem may reverberate through the local environment… or, even beyond. Because of the many and various interactions, it is likely that this response to the initial change will be *nonlinear…*

At first it may all seem so complicated that there is no way of understanding what will happen whenever changes, and adaptations to changes, occur. That is where the systems approach comes in. The systems approach is used widely in psychology, economics, sociology, biology (of course), jurisprudence, and systems thinking/systems design/systems engineering.

The systems approach always considers a system in context. To be more precise, any system of interest is considered to be:

a) open,

b) part of some greater, or containing whole, and

c) in dynamic interaction with, and adapting to, other open systems in its environment.

… or, looked at with a minor hierarchy shift, any system of interest is considered as a subsystem within some greater whole/system, interacting with, and adapting to, other subsystems within that environment.

With all the potential for action, interaction, adaptation, etc., it may prove difficult to conceive of all the activities and adaptations that may occur. Think, for instance about human personality seen, not in isolation, but in the context of the mind, interacting with and adapting to other parts of the mind, while the

---

[2] It can be shown that addressing and solving only part of some problem is unlikely to solve the whole problem, and may, indeed, make matters worse.

mind and the developing personality are exposed to continually-changing family and societal stimuli and influences: media, culture, paedophilia, violence, etc. Not straightforward. Without the systems approach, however, understanding the nature and development of personality would be difficult-to-impossible.

For manmade systems, one sensible way forward is simulation and modelling, leading on to so-called model-based systems engineering (MBSE). It is possible and practicable to create dynamic, generally nonlinear, simulations based on the systems approach within which it is possible to conduct experiments, trying out different system designs, for example, with different orchestrations of the interactions between the various subsystems, and to observe emergent behaviours in the dynamic simulation, choosing the configuration which solves the (simulated) problem in the (simulated) environment. This is the essence of systems engineering, above, taking the systems approach in simulation…

But, whether MBSE is the chosen way forward or not, the systems approach is *de rigeur* for systems thinking, systems design and systems engineering – including test, integration and proving of the system-to-be-delivered to the customer. No systems approach? No sensible solution! And, frankly, no valid or credible systems engineering…

Systems Architecture

A word on systems architecture, which seems to be grossly misunderstood. Systems architecture is about configuration and pattern. In Figure 1, the basic architecture is the arrangement of systems and links at Level N-1. Or, according to situation, you might see the architecture as being the clusters and links at Level N: and so on. It tends to be the horizontal slice through the hierarchy at a level of interest in a particular context.

There are all kinds of systems architecture: there is a number of archetypal configurations that recur throughout both the natural and manmade worlds. As system designers, we may find some architectures emerging naturally in finding solutions to problems – just as the internal shape of the lock defines the shape of the key that will unlock it. Functional architectures fall into this category.

On the other hand we may incorporate some architectures into our systems designs to achieve some purpose: to connect, to secure, to isolate, to enable self-healing, for closed user groups, for damage tolerance, to improve effectiveness, etc. And some architecture may even be dictated by context, by opposition and competition. Fans of football, for instance, will be aware that the 11 players in a team may be arranged in different 'formations' (e.g., 4:4:2) to improve their effectiveness, and that different formations may be appropriate for different competitors. Much the same can be said of warfare on land, in the air and at sea: the latter is particularly interesting since a naval action group may be engaged on three fronts at once; surface, subsurface and in the air. And the three fronts are not independent; so, complex… and the three architectures are continually shifting, morphing, interacting and adapting.

Layered architectures are common for defence and security alike. A layered defence might be set up around a sensitive area, with fighter aircraft intercepting would-be intruders as the outer layer; surface to air missiles might form a middle layer; and, a close-in weapon system might operate as a back-stop layer. For an intruder to get though all layers should be unlikely. With more layers it becomes even less likely, until at some point adding even more layers affords diminishing returns. There may, or may not, be communications between the layers, which can make a significant difference to the overall performance, which can be measured as 'leakage probability;' i.e., the proportion of intruders that penetrate all layers.

Other archetypal architectures include nodal, non-nodal, pipeline, circular pipeline, and many more. Each has its pros and cons, according to situation. But, those who impose particular architectures on complex systems better know what they are doing: the wrong architecture can prove disastrous. And, in the real world, systems architectures tend to adapt and evolve in operation, rather than stay rigidly in one form…

Summary

Emergence really is what systems engineering seeks to create. Systems engineering is also about managing complexity, and about creating whole solutions to whole problems, so about very much more than emergence – but emergence is fundamental, since it signifies the creation of a unified whole - *solution*. No emergence: no whole. Emergence can be defined as:

Properties of the whole that cannot be exclusively attributed to any of the parts, which do not exist at the lower level of the parts, and which are meaningless in the language appropriate at the lower level.

So, self awareness, a property of the whole mind, does not exist as a property at the level of neurons and axons, and self awareness is meaningless in terms of groups of neurons (grey matter) interacting with other such groups through bundles of axons (white matter.)

Similarly, battle-space dominance, a property of a network-enable squadron of interceptor aircraft, cannot be exclusively attributed to any of the interceptors: battle-space dominance as a property does not exist at the level of the individual, networked interceptors, and is meaningless in the language appropriate at this lower level.

Systems engineering is not the same as engineering. Classic systems engineering, as its name should indicate, is fundamentally about systems, and so engages with 'systems' as a discipline: systems science, systems theory, systems thinking, the systems approach, systems methods, systems design, systems architecting, etc., etc. 'Systems' can be anything that meets the definition of a system given above. Systems engineering can determine if technological products need to be specified and engineered, as part of a whole solution to some problem. But systems engineering is not conventional engineering, and – despite vociferous claims to the contrary – never has been.

Complexity is an aspect of systems: it seems to be a union of variety, connectivity and tangling – at least. It is neither good, nor bad: for the most part, it just is. Complexity in systems may imply varied capability, adaptability, and resilience; it may also imply homeostasis and internal energy expenditure. On the other hand, it need not indicate any of these… instead; it may signify disorder and confusion…

Hierarchy is a view of complex systems in layers, with each layer smaller and simpler in perception than that beneath, yet containing and concealing the complexity of the layer beneath. A hierarchy layer is defined by emergence of properties-of-the-layer, which are not exclusively attributable to any of its lower level parts. Hierarchy is also a means of managing perceived complexity.

Systems architecture is pattern and configuration. There are functional architectures, physical architectures, body-form architectures, communications architectures, and many, many more. There are several archetypal architectures, or classic patterns, each with its pros and cons when applied in the real world. In that real world, architectures are likely to interact and adapt, as the systems they configure interact and adapt. Always, the 'systems approach' rules!

Derek Hitchins. **March 2008**

Further reading from this author on these topics:

Systems Engineering: A 21st Century Systems Methodology, John Wiley & Sons, Chichester, England (2007/2008)

Advanced Systems Thinking, Engineering and Management, Artech House, Norwood MA (2003)

Putting Systems to Work, John Wiley & Sons, Chichester, England (1992)