Synthesis —The Essence of Systems Engineering

To Knowledge, Analysis may hold the Key
But Synthesis Begets all Creativity
Anon. 1992

Views and Beliefs

Ask the man in the street what 'systems engineering' is about, and chances are he will think it is something to do with 'systems,' therefore with computers, information technology, or similar: but, he will not be sure.

Ask an electronics engineer in industry, and he or she will be quite clear about the *role* of the systems engineer: i.e., what a systems engineer *does*. The systems engineer ensures that all of the various parts of some assembly – modules, backplanes, harnesses, shielding, power supplies, blowers, casing, etc., etc., fit and work properly together to create the whole … i.e., ensures 'fit, form and function' of the parts to come together, both physically and functionally, to form (synthesize) the unitary whole, which then does what is intended. So, the systems engineer is responsible at the level of the *whole*, but not for the design and construction of specific *parts* – that would be engineering: electronic, electrical, mechanical, etc. The systems engineer, it seems, need not be an engineer, or technician, *per se:* instead, he 'puts things together,' using what appears to be a different discipline—at least, it is none of the conventional engineering disciplines—and he/she need not have been trained as an engineer – although, it might help.

Academics charged with teaching systems engineering also have clear views of systems engineering:

> *"... Systems engineers study the whole, integrated system rather than one particular component within it. This involves modelling, design, analysis and implementation, and in particular ensuring that all its components interact together in an efficient way to achieve specific and meaningful objectives."* **Sheffield University Prospectus**

In other words, the systems engineer is concerned with how the parts work together as a *unitary whole* to achieve some *purpose*. Again, this is suggestive of a discipline – one of synthesis and integration rather than of engineering disciplines — which conventionally employ reduction.

Why Synthesis?

Systems engineering emphasises *synthesis*, as opposed to *reduction*. Why? Rene Descartes proposed that the way to understand a complex problem was to divide it into smaller parts, each of which could be understood on its own, and then to bring the understandings of the parts together to form an explanation of the whole: or, Cartesian reduction. It works for many things, but not for all...

It does not work for natural things, animals, people, teams, and life in general. Dissect a human into many parts, and you will not find the source of human intelligence, of purposeful, or goal-seeking behaviour. You might expect to find it in the brain, say, but you will find only grey matter and white matter, neurons and dendrites… enormous complexity, certainly, but where is the intelligence, where the self-awareness? It surely cannot be in a neuron, or in a dendrite.

Of course, on reassembling the parts, you will be unable to ‘restart the motor of life.’ The viability of a human, of any animal, is vested in the continuing actions and interactions between the mutually interdependent parts, or organic subsystems. The human exhibits ‘emergent properties, capabilities and behaviours,’ i.e., properties of the whole that are not exclusively attributable to any of the rationally separable parts, and which may be meaningless in the language appropriate at the level of those separable parts.

Scientists deduced that bringing parts together and causing them to interact, to be mutually interdependent, can result in the whole being different from, and potentially greater than, the sum of the parts; and, that it could work for manmade systems, too. But, what was going on – where did this “mysterious extra” actually come from? Was it, perhaps, something to do with complexity?

Combining sodium and chlorine in the right way forms sodium chloride (NaCl) or common salt, which is essential to human life, whereas the constituents are prejudicial to life. Salt, however, can take a number of physical forms (solute, crystal, etc.), depending upon the environment in which it forms… suggesting that some, at least, of the emergent properties are dependant on the environment in which they may be realized.

Perhaps the most obvious emergent property of any system is that a set of interdependent parts can form *a unitary whole:* an individual person comprises millions of parts, yet is a unitary whole. Nature, it is observed, makes only wholes: atoms, cells, animals, plants, etc. A naval destroyer comprises millions of parts, too, including hundreds of people, yet can be a unitary whole.

For the many and various parts of a naval destroyer function as one, indicates that the parts are complementary, contributory, cooperative, coordinated, etc., creating capability and enabling synergy between the parts, such that the destroyer ‘behaves’ as a unitary whole. It may also, then, exhibit other properties such as timeliness, agility, survivability, battle space superiority, etc; emergent properties of the whole that are not exclusively attributable to any of the parts, and which have meaning only at the level of the conflict environment. In this example, they would also be *relative* properties; e.g., as compared with other combatants in the same environment.

If the parts of a whole are mutually interdependent, their interactions may be non-linear: non-linear behaviour is associated with emergence. Non-linear behaviour can arise even where the internal parts are linear, as in the case of a man-made product comprised of linear parts/subsystems, where there is *close coupling* between the parts. Figure 1 and Graphs 1 together illustrate the effects of increased coupling on a simple system: even the simplest system can exhibit complex behaviour.

It follows, for both natural and manmade systems, that the *degree* of coupling, and the *nature* of coupling (interactions) between the parts/subsystems can determine the emergent properties and behaviours of the whole. Further, looking to natural systems, complex/chaotic behaviour need not be a ‘bad thing;’ many organic systems behave chaotically, and that may be their strength…

It also follows that the degree and nature of the coupling for manmade systems has to be carefully established during design, and preserved during the physical realization processes so that they ‘generate’ the expected emergent properties in operation: so, no reduction throughout, where reduction treats parts in isolation and so prejudices the integrity of the interactions and mutual interdependence of the parts/subsystems. Which is philosophically incompatible with conventional engineering, where reductionist methods and practices are the order of the day.

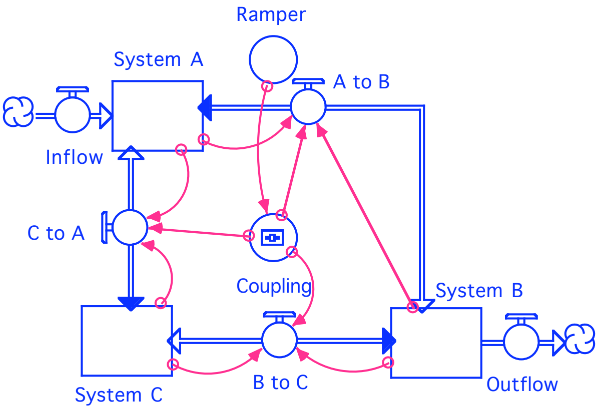
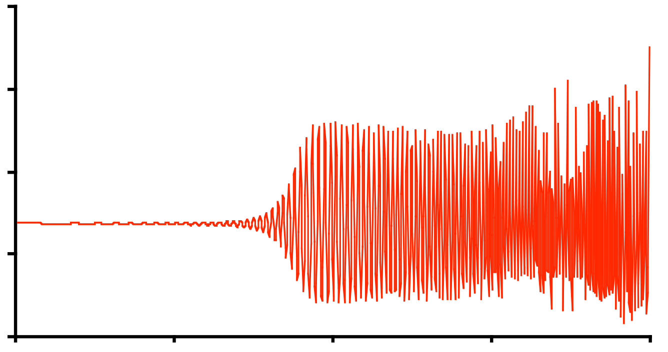
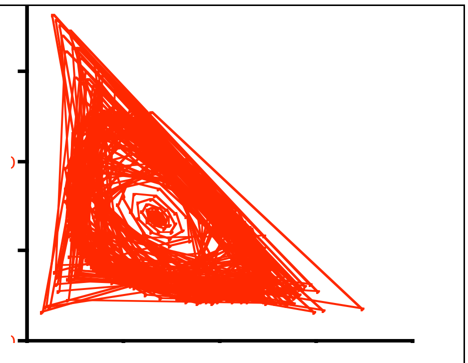
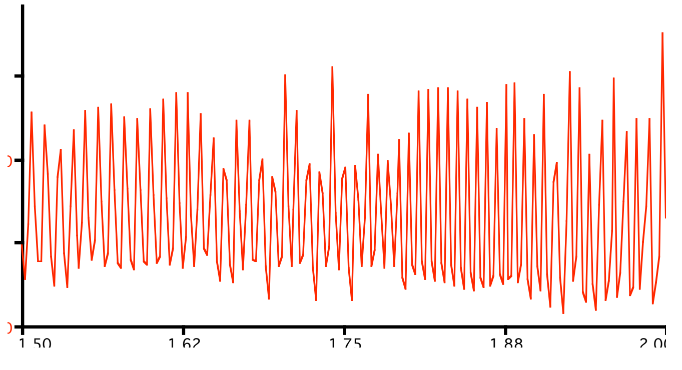


Figure 1. STELLA™ Model of an Open System, with inflow and outflow. The overall system comprises three interconnected 'reservoirs,' Systems A, B and C, arranged so that the degree of coupling between them can be progressively increased, initially promoting oscillation, then chaos and finally instability…







Graphs 1. From the Model of Figure 1. Top left simulates the 'contents' of System A with time, as the coupling between Systems A, B and C is progressively increased. Bottom left shows the final section of the simulation expanded, to reveal the chaotic nature of the variations. Above, a phase-plane graph shows the growth of the single-point attractor, starting as a point in the centre, and ending in the triangular pattern shown. If the ramping increases further, the whole goes unstable…

So, the essence of systems engineering appears to be founded in synthesis, but not just any synthesis. Specifically:

> *The essence of systems engineering is in selecting the right parts, bringing them together in the right way, causing them to interact in the right way, and orchestrating such interactions to create requisite emergent properties.*

All of which may describe synthesis, yet leaves much unsaid. Which are the right parts, which are the right ways, and what are requisite emergent properties? The answers to those questions are going to be problem- and solution-specific. However, it may be possible to go some way to answering by showing how problems may be addressed and solutions conceived, designed and realized…

Systems Design – Conceptual Methodology & Ontology

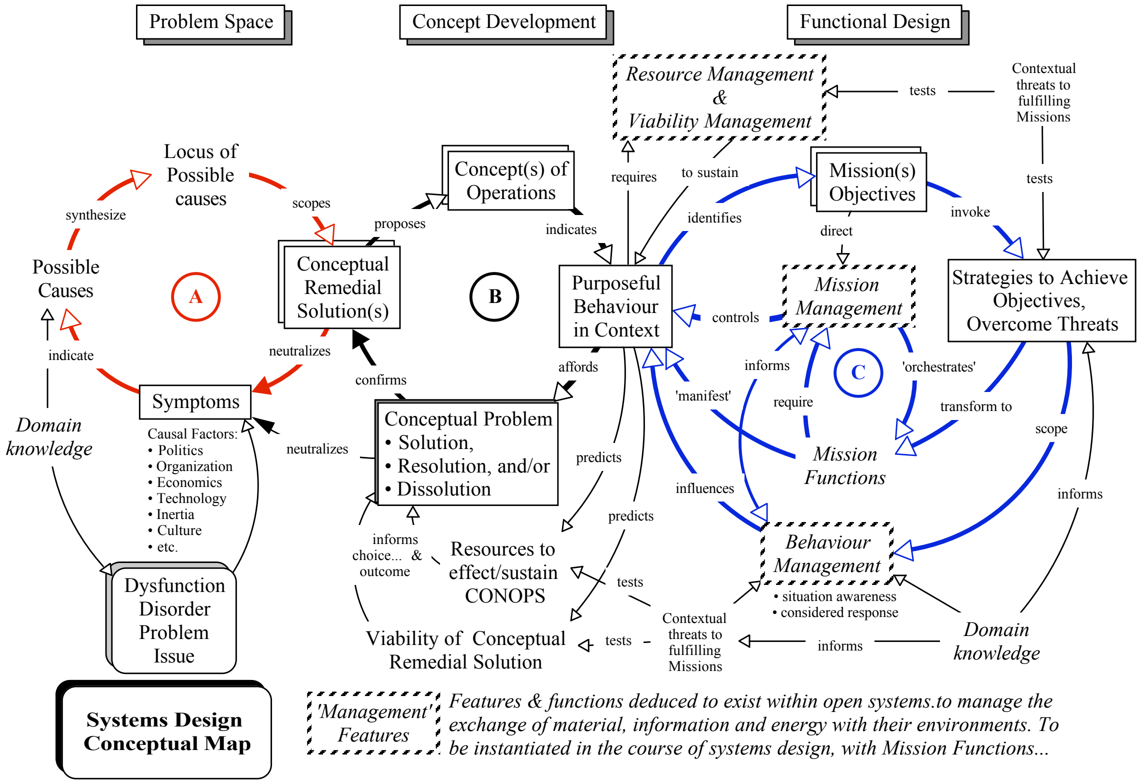


Figure 2. Systems Design Conceptual Map.

The map in the figure contributes to a 'systems ontology,' indicating a rational basis for the design of purposeful system solutions to complex problems. The map comprises three linked loops:

- **Loop A** explores the problem space, identifies symptoms of dysfunction and disorder indicating an issue of problem, surveys the problem and scopes conceptual remedial solutions, the test of which is that they would, if realized, neutralize all of the symptoms, so 'solving, resolving or dissolving' problem or issue.

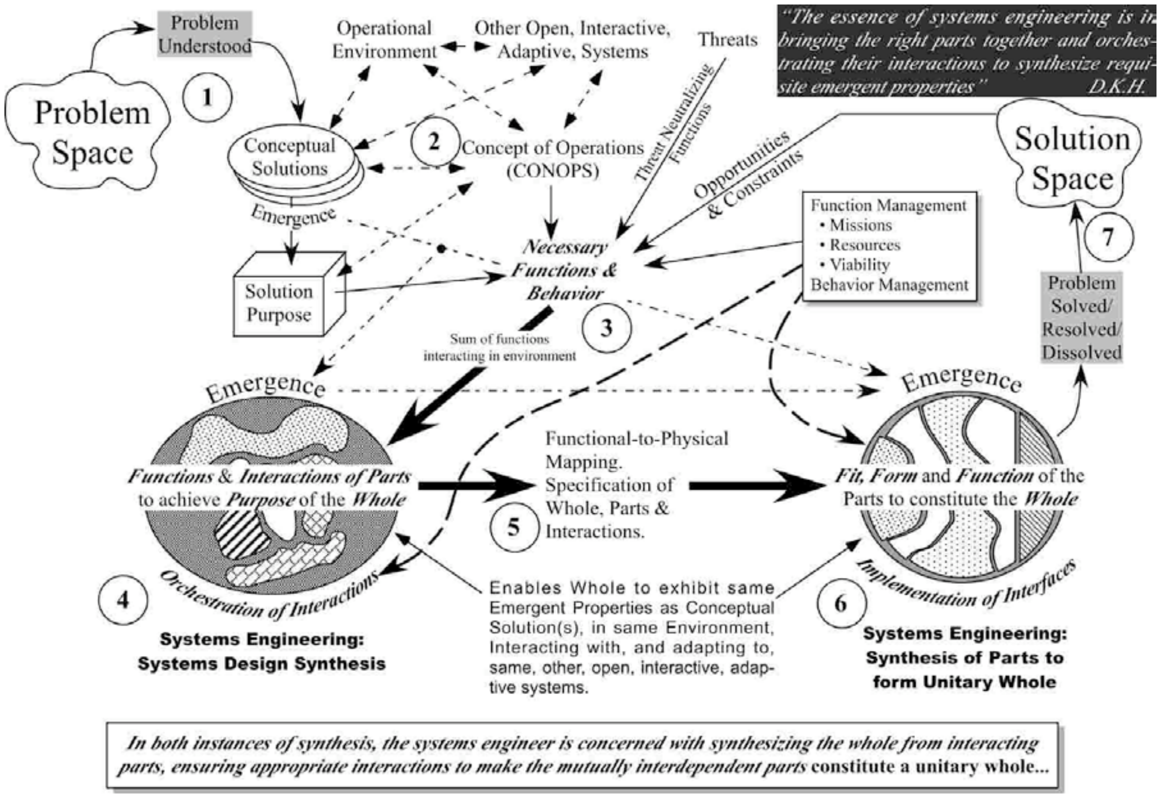
- **Loop B** proposes and develops concepts of operations for candidate remedial solutions, identifying and elaborating the way in which the candidate would operate to pursue its mission, achieve its purpose, taking account of the environment, other interacting systems, threats and risks, energy, information and resource requirements. Again, the crucial test for a potential CONOPS will be neutralization of all problem symptoms by solving, resolving or dissolving… Competing CONOPS may be compared by the risks and threats they incur, the energy and resources they expend, and by their potential impacts – positive and negative – on their future environment.
- **Loop C** develops functional design of the solution system, essentially by bringing functions together and coordinating their actions and interactions to substantiate the Purposeful Behaviour in Context, generated in Loop B and proved in Loop A.

Synthesizing Solutions to Problems

The Systems Design Conceptual Map can inform a systems design process or methodology: see Figure 3. Creating solutions to problems, lends itself to a number of steps or stages:

- Explore the problem space; survey the problem; understand the problem and the problem domain. Number 1 in Figure 3
- Generate conceptual remedial solutions to the problem and judge, test, evaluate, compare… their potential validity and completeness in solving the problem.
- Generate concepts of operations (CONOPS) for conceptual remedial solutions, test the various CONOPS step-by-step, identify resource implications, potential threats in the operational environment, etc. and their problem-solving integrity. Number 2 in Figure 3
- Bring together 'mission functions' identified from the problem, the conceptual remedial solution, the CONOPS, threats neutralization, problems and opportunities identified in the solution space, etc. Number 3 in Figure 3. Instantiate internal features required within any open system to manage the throughput of information, material and energy:
    - Function management (missions, resources, viability); viability management addresses: synergy, maintenance, evolution, survivability and homeostasis.
    - Behaviour management (cognition, interpretation, selection, excitation).
- Coordinate cooperative mission functions using function management to achieve missions in the (simulated) environment in accord with the CONOPS, creating interaction matrices, and formulating functional routines. Number 4 in Figure 3
- Develop functional subsystems and functional architecture using conventional functional binding and coupling. Also Number 4 in Figure 3
- Map the functional architecture on to a physical configuration within the constraints of the solution space. Number 5 in Figure 3

- Specify the requirements of function, fit and form for the whole and for each of the parts and their interconnections/interactions.
- Realize the physical parts: synthesize the whole by bringing the parts together and causing them to interact. Number 6 in Figure 3
- Test (prove) that the whole/system functions/performs effectively within the wider environment while interacting with, and adapting to, other systems.
- Hence solve, resolve or dissolve the original problem. Number 7 in Figure 3.



"The essence of systems engineering is i bringing the right parts together and orchestrating their interactions to synthesize requisite emergent properties" *D.K.H.*

Enables Whole to exhibit same Emergent Properties as Conceptual Solution(s), in same Environment, Interacting with, and adapting to, same, other, open, interactive, adaptive systems.

In both instances of synthesis, the systems engineer is concerned with synthesizing the whole from interacting parts, ensuring appropriate interactions to make the mutually interdependent parts **constitute a unitary whole...**

Figure 3. Schema for Synthesizing Tangible Solutions to Conceptual Problems

So, the figure outlines how to establish the right parts, how to bring them together in the right way, and which are the requisite emergent properties, i.e., those exhibited by the chosen conceptual remedial solution in its conceptual operational environment.

The figure emphasizes the role of synthesis, particularly, at Numbers 4 and 6. **Systems Design Synthesis**, Number 4, concerns itself particularly with the *management of mission functions*: arranging their activation, coordination, cooperation, contribution, etc., such that the whole functions optimally in pursuing its mission(s). At the same time, Design Synthesis concerns itself with the *management of systems viability,* with the continuing capability of the solution system to mount its missions.

Synthesis of Parts, Number 6 replicates the synthesis of Number 4, realizing the same functional activation, interactions, coordination, cooperation, contribution, etc., but in this second case the functions and interactions are tangible, not disembodied. (This synthesis is, of course, that being performed by the systems engineer, described in the opening paragraphs above…) The various physical parts will exhibit fit and form as well as function: in principle the whole should perform and behave in the real world environment as the functional design performs and behaves in the simulated operational environment. Provided, of course, that the problem has remain unchanged throughout, that the exploration of the problem space was thorough, that the conceptual remedial solutions were generated and tested effectively, that the competing CONOPS were established, evaluated and compared appropriately, etc., etc.

Essentially, then, systems design has to be comprehensive, thorough, purposeful, insightful, creative, innovative, etc., and may still be inadequate. No wonder systems design has been likened to "knitting fog." On the plus side, it is potentially creative and innovative, and while systems designs can never be *proved* 'correct,' some designs can be shown to be incomplete, deficient, misguided, or plain incorrect. Which is, of itself, a powerful tool, preventing, as it may, the enormous waste of time, money and resources that could otherwise be spent realizing the wrong solution to the problem.

Derek Hitchins **August 2008**