4

Systems Engineering Philosophy

All philosophies, if you ride them home, are nonsense, but some are greater nonsenses than others.

First Principles, Samuel Butler, 1912

Why Systems Engineering is Important

Systems engineering has been around for a long time — the Pyramids have been mentioned already, and there are even earlier examples. It may not have been recognized as such, as far as we know, if only because builders at that time did not differentiate between different disciplines; anyone could be an architect, in principle, if he had the will to learn and the talent to practice — and he would not need to be told that he had to do the whole job; that parts alone would not suffice. However, systems engineering really started to be recognized as a discipline during World War II.

There seem to have been several triggers: first, it was recognized that things that were connected could be usefully seen as wholes; even their connecting networks might be seen as entities. Second, it was recognized that systems made up from lots of interconnected parts could sometimes be made to work better, faster, more economically, etc; there was potential to make things more effective by treating them as a whole. Third, it became evident that engineers' methods for designing and controlling technological systems did not work when there were either people in the systems, or the systems were comprised of people. Fourth, systems were starting to become larger, more complicated and complex, with greater variety of parts, increased connectivity, etc., and there were mounting concerns that people knew neither how to design such complicated systems, nor how they might behave in practice. Fifth, it was becoming evident that, correctly synthesized, whole systems could exhibit emergent properties, capabilities and behaviors, affording more than the sum of the parts in terms of capability, performance and effectiveness.

A real stimulus for understanding systems and systems behavior is to be found, at least in part, in disasters. Arthur D Hall III (Hall, 1989) cites the chemical plant leakage in Bhopal in 1986; the explosion of the NASA Challenger space shuttle (1986) and the Apollo fire (1987); the sinking of the Titanic (1912); the nuclear explosion at Chernobyl (1986); and the disaster at the Three Mile Island power plant (1979). He could also have cited the problems facing humanity where some are over-farming land, creating unwanted mountains of food and drink, while others around the world

Systems Engineering: A 21st Century Systems Methodology Derek K. Hitchins
 ISBN: 978-0-470-0585-5

starve; where nations continue to contribute to greenhouse gases in the atmosphere, unwilling or unable to prevent them from building to damage or destroy their economies and environment. He could have cited the emergence and spread of new diseases such as HIV/AIDS, CJD, MRSA, etc., and the reemergence of tuberculosis — not forgetting man's age-old nemesis, malaria.

Systems engineering is important, vital, because it offers the prospect of creating whole systems where man can live, prosper and flourish in harmony and balance with the rest of the environment, where man sees himself as part of, not separate from, that environment. It is that important: but it may not be that simple.

Early Examples Set the Style

Systems engineering is continually evolving around a basic set of tenets and principles. We have met some off the tenets already: synthesis, holism and organicism; principles will emerge later. Some of these ideas appear self-evident — or are they? If a system is large, diverse and complex, is it obvious that changing one part will also change the others, and not necessarily for the better; perhaps only in retrospect?

A retrospective look at the Battle of Britain, and in particular at the command and control system of No. 11 Group in southeast England in 1940 might illustrate the issue; this was before the term 'systems engineering' had been coined, supposedly in 1941.

Battle of Britain

The Luftwaffe on the other side of the English Channel outnumbered No. 11 Group of the Royal Air Force more than three to one. The RAF had an ace in the hole: a new ground radar, Chain Home, which was sited along the south coast of England, and which could detect German aircraft formating over France, preparatory to attack. This new device, about which the Luftwaffe was unaware, could provide essential early warning.

Time was still insufficient, however, to intercept intruders before they reached English shores, so the commander, Air Chief Marshal H.C.T. Dowding, set about 'tuning' the command and control (C^2) system, cutting out delays and speeding up processes. He provided direct telephone lines; he introduced codebooks so that raids could be described in one or two short phrases; he developed map tables with markers being moved in real time to show where friend and foe were located, and their direction of movement; he even redesigned the clock in the control centers with colored sectors such that the markers on the plotting tables could be seen as new, or older and less reliable. He had the pilots sitting beside their Hurricanes and Spitfires at forward dispersal airfields, with the electrical supplies plugged in, so that the fighters could take off at the drop of a hat. While they waited, they were able to listen to relayed intelligence broadcasts, so that they knew what to expect when they did get airborne. In short, he went around the whole command and control loop, tightening up procedures, cutting delays, until the whole system 'came up to speed:' which it eventually did; just.

This command and control system for the Battle of Britain set the pattern for subsequent UK Air Defence Systems, of which there have been several since. Each may have employed the latest, up-to-date technology, but the system design changed little: the original set the style and the standard of what was to become operational system engineering.

NASA's Apollo

If Dowding set the standard for command and control systems, then NASA set the gold standard for space travel in the 1960s. In the general case, every 'new' system is really a revision of some prior system. For instance, we have had air defense since we had hawks, bows and arrows, rocks and spears, shields and caves. . . . Rarely, very rarely, there is a genuine 'green-field site,' and the Apollo missions to the Moon were the first time that man stepped on to solid ground that was not Earth. It was also one of the first instances of the conscious application of systems engineering, and arguably still stands as its greatest achievement: so great that some today are unwilling to believe it ever really happened. . . .

The problem facing the NASA system designers was daunting. At the start, they had no clear strategy for how to get to the Moon, how to land, if there was any firm base on which to land, how to move about on the surface, how to get back and land safely back on Earth, etc. They had no initial idea of how many men should go — should it be one on his own, a pair, three, four. . . ? They did know that there was a limit on the payload that the Saturn V rocket could lift, and they knew that what must be lifted would have to include everything for the round trip.

They developed systems engineering principles and practices that have stood the test of time:

- Systems engineering requires a clear singular objective or goal.
- There should be a clear concept of operations (CONOPS) from start to finish of the mission.
- There should be an overall system design that addresses the whole mission from start to finish. The full CONOPS should be demonstrably realized in the design, step by step. . . .
- The overall system design may be partitioned into complementary, interacting subsystems. Each subsystem is a system in its own right, and has its own clear mission and concept of operations.
- The overall system design addresses more than the spacecraft, its contents and the rocket; in addition, the whole design includes the ground control system, the telemetry and wireless systems, the ground maintenance systems, the crew training systems, etc., etc. The whole mission system is all of those, even though only part actively goes on the mission. The parts that do not go on the mission may take a very active part in the mission especially when things go wrong (e.g., Apollo 13)
- Each subsystem should then have its own system designers and systems engineers who look at the whole subsystem in its containing system (the whole system) and its interactions with, and adaptations to, all the other subsystems (i.e., the systems approach). Subsystem design teams work hand-in-glove with the containing/whole system designers, and with their own subsystem engineers.
- Each subsystem may be developed independently and in parallel with the others, provided that fit, form, function and interfaces are maintained throughout. Where any emerging deviations are unavoidable, whole system design may be revisited
- Upon integration of the subsystems, the whole system should be subject to tests and trials, real and simulated, that expose it to extremes of environment and to hazards such as might be experienced during the mission. These would include full mission trials where recovery from defect was possible.

Working in such a clear environment made it easy for system designers to see that any change to any part might affect the overall size, volume, mass, shape, moment of inertia, etc. If such change was inevitable, then complete redesign may be needed, or at the very least a complete rebalance of the affected design parameters. And it was clear, too, that the whole depended on each and every part: a holistic view was essential, but without losing sight of the parts and their interactions.

Their approach was organismic, too, although the term was probably not in use then. The mission system had to operate as a unified whole, with each of the parts having high levels of interaction with the other parts, electrically, electronically, resource-wise, through the operators in mission control, and via the astronauts. The many parts had to adapt to each other, too, if only to keep within tight weight, volume and shape budgets, The end design, with each phase of the mission based around its own craft, and the mission phase craft fitting together like so many Russian dolls, is even reminiscent of an organism.

Is it 'Systems'?

There is an ongoing philosophical discussion about whether systems engineering is engineering or systems. The above two examples show, perhaps, the futility of such arguments. The example from the Battle of Britain was concerned, not with engineering, manufacture, or whatever — that had already been completed. Dowding was working at whole system level, trying to enhance the emergent properties of the whole: particularly, of timeliness. So, he was reducing delays that occurred due to organization, lack of direct communications, use of verbose speech, unclear mapping, fighters taking too long to take off, climb to height, etc., etc. This was really, nothing directly to do with technology — it was to do with process, and particularly speeding up the process. Technology was not the system of interest: the overall process was the system of interest. Some of the process-as-a-system was supported by technology: some was conducted by people (operators, controllers, support staff, aircrew), in the form of verbal communication, map creation, choice of tactics, etc.

Looking at NASA'a Apollo, there was plenty of technology about, so was systems engineering about the technology there? Not really. Again, there was a process, described effectively in the CONOPS (concept of operations). The mission and the astronauts had to pursue that process, step by step, supported and enable by the technology. Getting the CONOPS right was the first solid step on the road to success. What about the design of the tightly packed space launch vehicle? Surely, there was lots of technology in there? But again, the system design was less about technology, more about physics, balance, structure and flow. More fuel needed here, less mass permitted there. More volume here, need to rearrange there to make way.

So, it was a balancing act, rather than classic engineering. The balancing act could have been undertaken with mocked-up blocks of wood, metal, material — the problem would have been largely the same, and so would the solution. . . .

Is it Engineering?

At some point, the design of the whole system, and of the whole subsystem may give way to the need to actually change something — physically. If the something to be changed is a piece of technology, then engineering a new piece of technology, or re-engineering an existing one, may be involved. That this is done within the overall systems engineering process makes engineering either a part of systems engineering, or an adjunct to systems engineering.

However, in many cases, the 'something to be changed' may not be technology. It may be operator training, or a revised process, a reorganized team, a reconstituted communication channel, a new maintenance scheme, all of the above, etc., etc., none of which can be sensibly classed as engineering.

So, not engineering in any classic sense, then. On the other hand, the bulleted list of principles and practices that were devised and employed in both examples above can be applied to a device that is to

form a largely technological solution to some problem. If this turns out to be engineering, it is different from classic engineering, which is linear and based, generally, on Cartesian Reductionism. Instead, it would observe holism, synthesis and organicism, and it would take note of the bulleted list above. But, would the outturn be the same? Probably not. . . .

Would you, then, call it engineering? Or, perhaps would it be more reasonable, since it would use the same principles and practices as any other system-to-be-changed, to call it systems engineering?

It is entirely possible, and practicable, to design and develop a technological system, part of some sociotechnical whole, using the systems approach, and the same systems methods and practices that would be used for any open system functioning in, and adapting to, its environment. To do this would be systems engineering. It is, on the other hand, possible to construct/build a technological system, that is to form part of some sociotechnical system, from piece-parts; this too would be systems engineering, in the sense that it would form part of the whole systems engineering process. However, this construction process need not apply the systems approach, nor employ systems methods and practices; classic, tried and trusted engineering practices may be appropriate.

So, it seems there may be two ways of designing and constructing a technological system:

- The classic engineering approach, based on reduction of the whole to recognizable parts, their manufacture and subsequent integration and test; there is an emphasis on functions, form, structure and physical architecture. Emergence may be neither recognized nor sought; instead the objective may to be make the whole equal the sum of the parts. . . .
- and the systems engineering approach, in which the technological system is regarded in its dynamic, open, interactive context, potentially adapting to other systems in its environment, and capable of exhibiting emergent properties, capabilities and behaviors. This approach emphasizes dynamic interactions between the parts, and traces threads of coupled interactions from external systems, through the internal parts and back out to the environment; hence the emphasis is on behavior, function, functional architecture, process and dynamics.

In simple systems, the results may be similar: in complex systems, different. In practice, too, there are other characteristic differences; one such is the concept of what constitutes a system. For many engineers (not all), a system is seen as a technological artifact, which a human may use or operate. For systems engineers, a system is more likely to include the human who is interacting with some artifact to extend and enhance his limited human powers and capabilities in some way.

So, a command and control system might be seen as a set of linked workstations with data communications to a central computing system with databases, or, teams of men working on intelligence, operations, logistics, etc., using networked processors in support. Either viewpoint appears to describe the same thing; both might wish to describe their view as that of systems engineering. The latter view, which focuses on human activities, is, however, the richer viewpoint, since it highlights purpose, function, behavior, adaptability, flexibility, diversity, capability, etc., of the whole, much of which is vested in the teams of people, even without their technological support — which serves to enhance, perhaps, but not replace.

Problem Solving, Resolving and Dissolving

Russel Ackoff suggested that there were three ways of addressing problems: they could be dissolved, resolved or solved (see page 17). To dissolve was to move the goalposts, or to change the situation such that the problem no longer existed. To resolve was to satisfice, to find a solution that was ‘good enough.’ To solve was to find a ‘correct’ answer, a best solution.

Systems engineering has been principally associated with the notion of solving complex problems. This is not to say that there is only one solution to a problem; in the real world, there may be many solutions. However, it is to say that some solutions are better than others in the sense that they will fulfill their mission more precisely, more effectively, and that they will satisfy the needs of those with the problem better, too, in terms of affordability, and of restoring harmony and balance where dysfunction and disorder previously existed.

So, the notion exists of a 'best' solution to a problem, where 'best' takes account of the environment, interaction with other elements, adaptation, stability/homeostasis, synergy, and both the 'value' and the 'cost' of the solution. It may be that this 'best' solution to a problem is unattainable, but it nonetheless serves as a benchmark against which to judge solution options.

There is no good reason why systems engineering should not be associated with dissolving and resolving problems, too. Often, dissolving a problem by changing the rules, or the situation, or even the viewpoint, requires an understanding of the whole, an overview if you like. Systems engineering is adept at developing such overviews, and, in searching for potential solutions to problems, should almost inevitably find ways of dissolving problems. Indeed, classically that has been one of the principle tasks for systems engineering: to find fundamentally different, innovative ways of addressing problems. In recent years, the practice has grown of some customers of deciding upon the solution that they want and presenting this *fait accompli* to systems engineering organizations. It is to be hoped that the customers have already examined ways in which they might dissolve or resolve their problem, and discarded them, before presenting their solution to be created. . . .

Be that as it may, systems engineering has traditionally been concerned with dissolving problems, too. Resolving problems may be different. If to resolve is to find a solution that is 'good enough,' then there is an implication that the solution found is less than 'best.' Satisficing in this way is indicative of working with lack of time, or lack of information, such that judgments have to be made. Systems engineering has been traditionally employed in this arena, too, although it may not be so obvious.

The Apollo missions were progressive, in the sense that each of the early missions went a step further than its predecessor, with none of them able to complete the full mission of getting a man to the Moon and back again. There were earth orbit missions. There was a figure-of-eight mission where a craft flew out to the moon and back in a figure-of-eight pattern, making a loop behind the Moon, but without landing. Then there was the first mission that landed, and subsequent missions that further explored the surface of the Moon using a vehicle; and so on. So, each of the successive missions 'took another bite out of the cherry,' until the problem was eventually resolved.

Looking at Apollo in this way suggests that satisficing (i.e., resolving the problem by producing a series of answers, each closer to the final answer than its predecessor), is systems engineering, too. And, after all, this is little different from prototyping or trialing of solutions to find out what works and what does not, what the environmental reactions are, how the solution will adapt, etc.

Systems Engineering: Definitions and Descriptions

So, systems engineering adopts the systems approach to solving, resolving and dissolving problems. Does that define systems engineering? Apparently not.

INCOSE, the International Council on Systems Engineering, defines as follows:

> *INCOSE A*. Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with

> design synthesis and system validation while considering the complete problem. . . . Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.

This definition is, perhaps, more about the 'how' than the 'what.' Stripping out from the statement INCOSE's view of their particular methods, INCOSE defines systems engineering as '. . . (a way of) realizing successful systems. . . .'

Walden University does not define: instead, it offers the following description of one of its courses:

> The M.S. in Systems Engineering program prepares students to solve complex multidisciplinary problems in the real world, With an up-to-date understanding of the methods of analysis, synthesis and design, graduates of this program are equipped to be leaders in industries as diverse as aerospace, agriculture, manufacturing and medical device technology. . . .

Evidently, Walden University takes a broad view of systems engineering as complex problem solving in the real world.

Sheffield University offers the following description of their systems engineering course, within the Automatic Control and Systems Engineering Department:

> Many systems in daily use are relatively complex in nature and include mechanical and electronic components, computer hardware and software and possibly additional sub-systems. Systems Engineers study the whole integrated system rather than one particular component within it. This involves modeling, design, analysis and implementation and in particular ensuring that such a system is designed so that all its components interact together in an efficient way to achieve specific and meaningful objectives.

Sheffield University looks upon systems engineering as the synthesis of a complex whole system. It concerns itself particularly with synergy.

Virginia University, Department of Systems and Information Engineering, offers the following introduction to their graduate systems engineering course:

> An integrated introduction to systems methodology, design, and management. An overview of systems engineering as a professional and intellectual discipline, and its relation to other disciplines, such as operations research, management science, and economics. An introduction to selected techniques in systems and decision sciences, including mathematical modeling, decision analysis, risk analysis, and simulation modeling. Overview of contemporary topics relevant to systems engineering, such as reengineering and total quality management. . . .

Virginia is concerned with systems methodology, highlighting design and management, with professional and intellectual discipline, so with systems of all kinds.

INCOSE offers a second definition as follows:

> *INCOSE B*. Systems Engineering is an engineering discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner

throughout a system's entire life cycle. This process is usually comprised of the following seven tasks: State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate.

INCOSE's first definition (A, above) identified realization of a successful system, while this second definition identifies the creation and execution of a process as its goal: one identifies with the end, the other identifies with the means to that end.

Is it possible, reading through the various descriptions and definitions and elaborations, to detect a common thread running through them? Is there one systems engineering, or many? Or, is there, perhaps, a common systems engineering philosophy that guides each of the definitions and descriptions presented above?

Looking through the various offerings, certain features recur:

- Wholes. Systems engineering is about looking at wholes, understanding wholes and creating wholes.
- Synthesis of the whole from complementary parts. Repeated mention of 'interdisciplinary' or 'multidisciplinary,' indicates that parts differ from each other, but are encouraged to operate harmoniously — hence complementary. Also implicit in 'integrated.'
- Finding answers/solutions to 'whole problems:' this feature recurs explicitly or implicitly.
  - Addressing the whole problem is entirely rational, since it can be shown that addressing only part of a problem can result in counterintuitive responses, such that the whole problem becomes worse, rather than better.
  - Politicians, amongst others, often appear to misunderstand this, and address symptoms rather than the root problem: or, is this expediency?
- Analysis, in the sense of detailing constituent parts and how they relate to, and interact with, each other.
- Design, whence innovation.
- Complexity and its accommodation.
- Discipline and science.
- Integrity: quality, risk, etc.
- Planning, decision-making.

So, it seems that there may be one systems engineering philosophy underlying a seeming plethora of definitions and descriptions. The seeming differences emerge from situating the basic philosophy in different disciplines: situating systems engineering philosophy in an aerospace industry results in an apparently different definition of the discipline from situating it in, say, control engineering, agriculture, commerce, economics, social services, healthcare, etc. Despite these surface differences, then, there does appear to be a common notion of systems engineering, that it is a discipline, that it is about addressing whole complex problems and about finding complete, whole, solutions to those problems.

Is it, then, possible to provide a general definition of systems engineering, i.e., one that is not situated in some particular industry or sphere of activity? Lexicographers are good at addressing this kind of problem. Chambers dictionary defines architecture, a particularly complex subject with obvious parallels to systems engineering, succinctly as 'the art or science of creating buildings.' This avoids the trap of describing the 'how' of architecture, which would take many books and on which few pundits would agree.

The definition of architecture suggests a useful template for that of systems engineering: not how it does it, which would also take many books upon which pundits would not agree, but what systems engineering *is*.

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

A glance back at the definitions and descriptions above will indicate that this simple definition does indeed encompass all of them, although pundits within particular industries or disciplines may find it lacking specificity relevant to their application. But then, like architecture and the systems approach, systems engineering is a broad, philosophically based, universally applicable discipline, with its dedicated theory, science, tenets and principles. None of these need be changed by the environment in which systems engineering is applied, although the words used, the descriptions, the methods, etc., will properly and inevitably adapt to that environment.

Looking at this definition, and recalling the ongoing discussions about 'systems' or 'engineering,' there is a possibility that systems engineering might have something in common with, say, doctoring, or police detective work. Doctoring seeks to solve the problem of what is wrong within a highly complex system (the human body, or animal body for veterinarians) and to identify and apply a cure. Police detection seeks to find out what really happened, and to piece together a trail of evidence leading to the identification, prosecution and conviction of a perpetrator. Systems engineering similarly explores and unravels the real problem within some complex whole and seeks to conceive and manifest some way of 'curing' the dysfunctions: and like doctoring and detection, the methods and processes, can be applied to a wide variety of different problems, situations, types, categories, etc. The analogy suggests, too, that domain knowledge and experience will be important.

This catch-all definition of systems engineering combines art and science; art is very much in evidence during the innovative, creative stages of conceptualizing potential solutions to problems, while science, logic and rationale are in evidence in the approach taken to conceptualizing, to problem solving, to planning, to detailing, validating, etc. It has to be said, however, that many practicing systems engineers care little for art or science, recognizing neither in their everyday work. When painting the Sistine Chapel ceiling, Michelangelo might have been more concerned with adjacent colors running and brush strokes showing, than with the overall picture — which, after all, he could not see from close up.

The Real Objectives of Systems Engineering

Given this catch-all definition, it follows that the objective of systems engineering is to create a whole solution to some complex problem. This supposes that the problem can be identified, and that it is solvable (resolvable or dissolvable) — not all problems are. So, it seems that systems engineering has several objectives:

- to scope the problem space;
- to explore the problem space;
- to characterize the whole problem;
- to conceive potential remedies;
- to formulate and manifest the optimum solution to the whole problem, that is, the best solution achievable in the situation, constraints and circumstances;
- hence to solve, resolve or dissolve the whole problem

The notion of 'optimum' can be seen, for example, in the Apollo mission designs, where a balance had to be drawn such that all the various parts could be fitted within the volume, shape and mass constraints imposed by lifter capacity, aerodynamics, etc. The optimum solution was one in which all the essential parts were present, in the necessary amounts, and in the best configuration possible within the constraints. The optimum solution was, then, the one best able to achieve the mission.

Regarded from a different viewpoint, this means that an objective of systems engineering is to select, bring together, and cause to interact, parts such that the whole exhibits requisite emergent properties, capabilities and behaviors; methods for achieving this are core to the skill of systems engineering.

The real objectives of systems engineering can be seen in another sense, and at a higher level. Systems engineering seeks to create systems of many different kinds that work harmoniously together in the service of, and to the benefit of, mankind.

Strategies for Solving, Resolving or Dissolving the Problem

It is often considered that systems engineering solves problems by creating new systems, comprised largely of operators working with technology. To be sure, there are many such instances, but there are other ways of solving, resolving or dissolving problems. Some of these are shown diagrammatically in Figure 4.1.

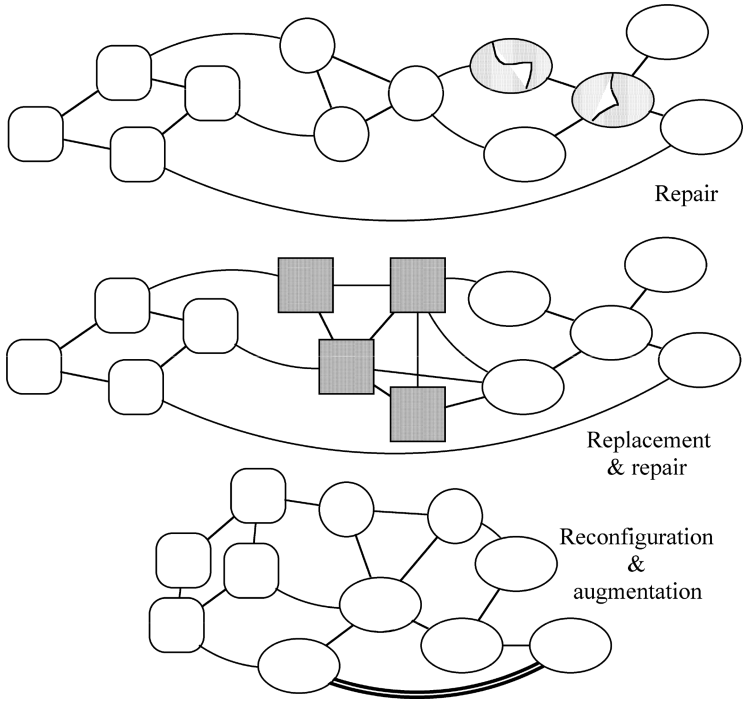

Figure 4.1 Notional strategies for solving problems.

Complex systems may exhibit dysfunctional behavior for a wide variety of reasons. One is that some parts are not working/operating, as they should, as they once did, or as the evolving problem situation requires that they should, e.g., to maintain stability, to meet demand. The simplest, straightforward solution is to repair the inadequate parts. This is shown conceptually in the upper diagram of Figure 4.1, where the shapes are suggestive of part of systems, while the lines are suggestive of paths of interaction between the parts. The parts may be tangible entities, they could equally be activities in a process, they could be people undertaking tasks in a shop, or whatever.

In the central diagram of the figure, the defective or inadequate parts have been repaired, but now the three linked circular parts have been replaced by four linked squares. This represents the strategy of solving the problem by introducing a new system. In general, introducing a new system amounts to replacing an existing one, although it may not always seem so to those creating the new system. So, the team of designers, technicians and engineers concerned with introducing a brand-new management information system (MIS) into a large corporation might not consciously realize that there had been a management information system in place since the corporation formed: it could not have formed or survived without one. It may have been the case that staff that used telephones, filled in ledgers, wrote reports, etc., constituted previous management information systems. The new, shiny MIS will most likely find itself competing with the old, 'handraulic' version, as the corporation executives cling to their experience of tried and trusted methods — and people.

Similarly, a new computer-based accounting system may be taking over from a room full of men with ledgers and quill pens. A new digital news service may be replacing an old analog one, which replaced teleprinters which replaced the old town crier. . . there is, in truth, very little that is new under the sun, and new systems are invariably replacing something that existed before, no matter how different, inadequate and antiquated it may have seemed.

A third strategy for solving system problems is to augment, i.e., to leave the existing system in place untouched, and still performing its role, doing its job. In some arenas, this augmenting might be referred to as 'jury-rigging,' while in others it would simply be increasing capacity by enhancing the degree of parallelism, perhaps. It is represented in the lowest diagram by the additional double interaction link.

A fourth strategy, also shown diagrammatically at the bottom of the figure, might be to reconfigure a problematic system, i.e., to rearrange entities and relationships, but without necessarily altering any of the entities and/or relationships per se. A topical example of this might be the rearrangement of parts in a depot so that picks could be undertaken with less effort and less distance traveled. (Picks refers to the process of collecting all the parts from various bins to fill some 'shopping list,' or order. According to how the bins are laid out, and what is stored in each location, any one pick might take more or less time. The problem, then, is to minimize the time for the 'average' pick, so making the whole process more efficient and timely.)

A second example might consider the practice of arranging all the workstations associated with a particular sequential process in a straight line, so presenting the operators with a clear view of the chain, and enabling them to see their role in the context of all the others. This, so-called 'streamlining' approach minimizes the distance traveled by work-in-progress (WIP), speeds up the end-to-end process, reduces the amount of WIP — which costs money — and so potentially improves both efficiency and effectiveness.

All of these, and more, might reasonably be considered to be systems engineering. Moreover, the replacement strategy might take place in stages, and so effectively be satisficing, with each bite of the cherry moving further towards whole solution. (That approach evidently runs the risk of provoking counterintuitive responses during the ongoing process of providing each part-solution. . . .)

Self-organizing Systems

Some systems are self-organizing, that is, they develop their own internal organization — processes and structure — without external direction, control or influence. The Sun is self-organized into three concentric spheres, with heat energy being progressively transferred from the hot core by radiation and convection, to the surface.

Self-organization can be observed in much humbler circumstances. If a beaker of hot coffee is allowed to settle for several minutes, before carefully adding creamer and sugar then, if the conditions are right, the creamer, instead of mixing, may form into small cells, about the size and shape of an ear of corn (maize). These small cells form into concentric rings on the surface of the coffee. Usually, the effect is fleeting, but occasionally several rings form, before unexpectedly and suddenly, the cell walls collapse and full mixing occurs within seconds.

Human activity systems (HASs) may be self-organizing in some circumstances. Groups of people, 'thrown' together and motivated by fear, reward, or a desire to achieve, have been observed to form into teams to take on different tasks that the whole group feels need to be undertaken. Moreover, unlike managed organizations, such teams changes their size and composition according to the size and duration of the tasks they undertaken with individuals migrating between teams as the need is perceived.

A second example of people self-organizing is to be seen at cocktail parties, or pre-prandial gatherings, where people socialize before moving on to eat. Initially the assembly room is empty. As people arrive, they form into small groups, talking. As more people arrive, more groups form, but the size of each group tends to reach a steady state, so that newcomers do not join a fully formed group, or if they do, a current member leaves to join another group. Eventually, the room is filled with people in equally sized groups, with individuals flitting from group to group ('circulating'). The reason behind this phenomenon appears to be simple: as more people arrive, the background noise of people talking rises. Anyone within a group trying to hear others in the group talking has to be able to hear over the background hubbub. As the group size increases, there comes a point where the diameter of the group is too great to permit audibility, so people will leave to join another group. In support of this hypothesis, group sizes tend to be smaller if there is an orchestra playing; this increases the background noise, such that only smaller diameter groups permit audibility.

It is a moot point as to whether, or not, bringing a group of people together and encouraging/motivating them to self-organize could reasonably be described as systems engineering. . . however, there remains a suggestion that self-organizing systems may be well ordered, adapted to the tasks, efficient, flexible, and effective, particularly in changing, or high-pressure, situations.

System of Systems

A system is made up from parts, often themselves systems: they are generally referred to as subsystems. To be a subsystem is to satisfy the definition of system, i.e., to be a whole, and to be made up from complementary, interacting parts. In this sense, then, every system is a system of systems, although the term has rarely been used, since it would be a tautology.

It has been the practice for many years, for instance, to form an avionics system for a modern aircraft by synthesizing it from various extant, off-the-shelf systems: navigation; automatic flight control, attitude sensors, pitot-static sensors and instruments, communications, radars, altitude sensors, stores management systems, etc., etc. These might be made to work as one by

interconnecting them through special interface units and by introducing a common data highway and some central processing system to collect, correlate and coordinate information sources and sinks. Would an avionics system developed in this way be termed a system of systems? It seems unlikely.

In recent years, the term has arisen to mean something rather different. A number of extant, operational systems, often businesses or enterprises, may be brought together under one umbrella, and referred to as a system of systems. But is the term valid?

For a system of systems to exist, there would have to be a whole. The various system parts, effectively subsystems, would interconnect in some degree to create synergistic interactions, such that the parts worked together, cooperated and coordinated their activities with some degree of unity.

So, a system of systems would, to deserve the epithet, display the basic characteristics of any system. In forming a system of systems, by bringing together extant systems, it is unlikely that full and complete interaction would be attained, such as would be necessary to produce optimum emergent behavior. Instead, it is to be expected that there will be limited interaction. This does not mean that an overall system cannot be formed; rather that it is unlikely to be optimal. However, it may well proffer significant advantage in terms of synergy, efficiency, and effectiveness.

An example of a 'system of systems' might be a city-wide integrated transport system, bringing together under one umbrella the previously independent transport systems: intercity railways, suburban railways, underground railways, buses, ferries, riverboats, taxis, coaches, etc. Simply calling disparate transport elements an integrated system would not make it so (politicians take note). To be a system of systems, the various elements would need to be 'harmonized,' in several respects:

- Timetables would be harmonized so that the flow of people was not interrupted by having to wait when disembarking one mode of transport before embarking on another.
- Interchange points between modes of transport would be constructed/improved to facilitate passenger flow and comfort.
- Ticketing and charging would be made seamless, so that passengers could buy tickets that would carry them across all the transport media.
- Capacities of each of the modes of transport would be adjusted to preclude queues forming at the various entry, boarding and interchange locations.
- Possibly, measures would be required to control or reduce road congestion caused by private cars and other vehicles. For instance, deliveries to shops might be permitted only during the night and at weekends, when passenger demand for the integrated transport system was low.

Evidently the various modes of city transport can, in principle, be welded into a whole, and one with emergent properties, too: in this case, properties of reduced mean journey time over the whole system, increased capacity of the whole system, increased simplicity of perception and of usage by passengers, and hence of an overall improved 'commuting environment.'

There are many other examples that might reasonably be termed 'systems of systems,' although we might not think of them in that way. For instance, the criminal justice system can be viewed as a circular pipeline system of systems, consisting as it does of the following systems:

- policing system for the detection and apprehension of suspected criminals
- custody system to hold suspects during investigation
- crime information system
- trial and judgment system;
  - prosecution system;
  - defense system;

- punishment system;
- rehabilitation system;
- probation system;
- policing system for the detection and apprehension of suspected criminals.

The policing system appears both at the start and the end of the list of systems: many wrongdoers are recidivists, going around the criminal justice cycle not once, but many times–hence, a circular pipeline. Although this overall process of crime, detection, punishment, probation. . . is well understood to be a whole system, it is generally not treated as such, or even viewed as such, by governments. That this is true is evinced by the fact that increased spending in one area, e.g., rehabilitation, would reduce spending in others, e.g., recidivism, and hence the numbers in the whole loop. Which perhaps explains why the various elements of the cycle are paid for generally out of different government purses, and why the whole is treated only as separate parts.

Despite this, the whole does have emergent properties: the number of supposed wrongdoers in the whole system; the sum cost of the whole (which is generally not publicized); the removal from society of 'undesirable elements,' with the consequent feeling of security and justice that society supposedly enjoys. Evidently, the criminal justice system is not so much separate from society, but a controlled and regulated compartment within society.

Other examples of the notion 'system of systems' are to be found in defense, where an army might be made up from infantry, armor, cavalry, engineers (sappers), intelligence, etc., etc. Clearly, the whole is made up from complementary, interacting parts, and the whole exhibits emergent properties, capabilities and behaviors of strength, coordination, synergy, mobility and many more, recalling the principles of war. . . . So, an army is a whole, and is a system within any sensible definition. That this is true would be particularly apparent if one of the principle 'subsystems' were absent, e.g., an army would hardly be an army without infantry.

There are various arrangements where the term 'system' does not appear justified, however. Several like enterprises may be categorized with each other under some umbrella title, or holding company. Is this a system of systems? It cannot be described sensibly as a system if the parts do not interact, do not cooperate, do not coordinate, and do not, together, exhibit properties of the whole. That is not to suggest that there is no potential advantage; there may be some economic or marketing advantage, for instance. However, the arrangement would not sensibly be a system as such.

Similarly, it is to be expected that any system of systems worthy of the tautology would exhibit emergent properties, capabilities and behaviors of the whole, i.e., not exclusively attributable to the rationally separable parts, i.e., the constituent systems.

Why is it important to identify whether or not a grouping of systems forms a system of systems, or not? For the important reasons that, if the grouping is really a system, then it has the potential to be in harmony with its environment, to adapt to that environment, to be more than the sum of its parts in terms of capabilities, performance, efficiency, effectiveness, to exhibit synergy, homeostasis, dynamic stability, to evolve, and so on. To mis-classify, say, a family of systems as a system of systems, is to misunderstand, and to form unreasonable expectations of capability, performance and outcome.

Bottom-up Integration

It is possible, and often practicable, to synthesize technological systems by bringing parts together and making them work together through interfaces. This is sometimes called 'bottom-up integration.' For instance, the average personal computer (PC) can be formed in this way, using various parts

that are readily available in the market. By choosing between variants of available parts, PCs with different capabilities — processing speed, capacity, overall size, etc. — can be created.

Bottom-up integration, as the example suggests, is a useful, pragmatic approach to synthesis where the parts are understood, where they do not adapt significantly to each other when interacting, and — in short — where the synthesis process employs the so-called building-block approach.

For instance, a new radar might be synthesized by bringing together a transmitter, a receiver, an antennae, etc., each of known and well-understood characteristics, and so producing a new radar.

Bottom-up integration is not without risks, which can become evident when, for example, software packages are employed such that only part of a whole package is required. A particular application might require only one module of, say, a complex word-processing program. Counterintuitive problems may subsequently arise when that module is used as part of another system; one reasons for such problems might be that the module depends for its good operation on interactions with other parts of the original world-processing program, no longer accessible. In such a case, it would be prudent to include the whole of the program, even though only part of it was apparently needed. . . .

In the general case, interactions between the parts, which are expected to be linear and consistent, may result in counterintuitive results. Emergent properties, including potentially undesirable ones, are due to interactions within and between the parts, including energy, information and timed interactions. Sometimes the building-block approach can overlook such interactions, such that the 'whole wall' may not behave as anticipated. Owners and would-be builders of PCs may be painfully aware of the limits, as well as the advantages, of the building-block approach.

However, bottom-up integration, properly planned and executed, may reasonably be viewed as systems engineering. Some organizations go to considerable lengths to produce discrete building blocks that function linearly and predictably, which have standard interfaces, which do not adapt to inflows or outflows, and which exhibit, for most practical purposes, immutable transfer functions. Their view of systems engineering might be that the system they are creating is not so much the individual parts, but more the standardization and interchangeability of parts so that they can be readily put together in various ways to achieve different objectives, i.e., they are creating a construction system. And it seems to be a valid viewpoint. . . one with which the architects and builders of the Great Pyramid, with their regular, rectangular blocks of limestone, would have sympathized.

Completing the 'Whole' of Systems Engineering . . .

Earlier, it was noted that there was more to Apollo than the mission system and the mission control system. The whole system included the ability to 'self develop,' or evolve over an extended series of missions, from the first orbital steps through to the full-blooded Moon exploration missions. If we look at major defense and other national projects, we may see that the initial system as delivered is sometimes obsolescent at delivery; or, if not, may soon become so. Large projects and complex systems take time, sometimes decades, to conceive, design, develop, make, test, prove, deliver and work up. During this time, the original problems that the project was designed to solve might well have morphed; the operational environment may well have changed, too. So, the delivered solution system has a good chance of being outdated and outmoded at the point of delivery.

It is also the case, as with Apollo, that there is much more to be delivered in conjunction with a typical mission system (an aircraft, a ship, a tank, or some other platform). Additionally, there will be

a concurrent need for a maintenance and servicing system; a repair and replacement system; a crew selection, training and continuation training system; and so on.

Not so apparent, however, but just as real, there has to be a system in place for reviewing the original problem, as now morphed, and evolving the design of the solution system such that it is better able to address the problem as it presently is, not as it was. In defense and other government circles, the outcome of such deliberations is generally a midlife update, a refit, or similar. The facility that undertakes this re-examination of the problem, redesign and rebuild is conventionally considered to be separate from the mission system and all the support systems; but; should it be separate, or is it more properly part of the whole?

For the whole system to include within it the ability to adapt and evolve would be more in keeping with the organismic principle, i.e., would result in a more unified whole. This unified whole would be open, it would exhibit homeostasis, it would adapt to its environment (i.e., exhibit intelligent behavior), and it would in principle have increased longevity, increased harmony both internally and externally, and emergent properties, capabilities and behaviors that kept in line with those required to address the evolving problem.

The concept of system viability is illustrated in Figure 4.2, where a system is seen as viable if it is able to sustain itself. An open system is internally dynamic as it receives inflows, distributes the inflowing resources, converts some of them to work, disposes of waste, and creates product, etc., as outflows.

For these archetypal processes to execute, the interacting parts within the system have to work in some degree of harmony, to cooperate and coordinate: this is marked as synergy in the figure: no synergy, no viability.

To remain viable, a system also has to survive, which generally implies that it has to be able to avoid, neutralize, or accommodate external threats.

Similarly, since it has working parts, it has to be able to maintain itself in the face of internal threats, i.e., detect, locate, excise and replace parts and/or interactions that are not acting as they should.

It has, too, to maintain some level of dynamic stability, or steady state, marked as homeostasis in the figure, affording a balance between inflows and outflows. And finally, it has to be able to evolve in line with a changing environment, with changing threats and opportunities. acronym

As the figure shows, these five aspects form the acronym S-MESH (pronounced 'es-mesh;' it is catchier): they form a necessary and sufficient set of features for a system to remain viable. Looking at the whole solution system in this way, it becomes evident that 'maintenance,' does not maintain the mission system alone; additionally, it maintains all the other (sub) systems, including the maintenance and servicing systems (sic), the training and recruiting systems, and (what might

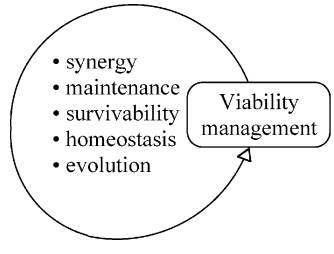

Figure 4.2 Viability management. Initial letters may be formed into a useful viability acronym — S-MESH.

be called) the evolution management system, etc. (Maintaining the maintenance systems raises the specter of Gödel's Incompleteness theorem, implying that there may be limits to how completely this can be achieved; however. . . .) And this wider 'whole' would interact with other systems within the environment which would supply resources, dispose of waste, etc., to maintain dynamic stability. The whole can be seen as organism-like, existing in harmony with its environment. . . .

There is a case, then, for the whole solution to any problem to include within itself, not only the ability to adapt, but to evolve as well, so that the evolving solution system can solve, resolve and dissolve the continually morphing problem in the changing environment. This process of evolution would be further facilitated if system conception, design and implementation had this potential for evolution in mind throughout. Including the ability to adapt and evolve within the organismic whole suggests that the whole may, in the final analysis, be self-organizing, adaptable, optimal, flexible, enduring, in harmony with its environment, and intelligent.

Summary

Systems engineering has been around since before the Pyramids were built. It was not recognized and codified as a discipline until the 20th century, however. Recognizable examples of systems engineering include the air defense system for England during the Battle of Britain, and Apollo, which set the style and substance of practical systems engineering.

World War II provided a major stimulus for the development of systems engineering, with defense and military systems becoming more complex, and with the concomitant need to understand and manage that complexity. A host of well-publicized disasters demonstrated to the world that complex systems could be problematic, and that there was a serious need to understand systems, counterintuitive behavior, and how best to manifest 'new' systems that were able to resist threats both from without and within.

The systems approach was seen as the best way to understand, and systems engineering was seen as the best way to solve, resolve or dissolve complex issues and problems in virtually all walks of life: sociotechnical systems, social systems, healthcare, industry, agriculture, ecology, economics, international peacekeeping, etc., etc.

Arguments have waged back and forth about the substance of systems engineering since the 1950s: is it applied systems science, or is it engineering? Such arguments arise since solutions to many problems include technological parts, within sociotechnical systems such as defense, policing, air traffic management, and the like. Engineering organizations may then conceive and design technological elements of such socio-technical systems, and so believe that they are 'systems engineering.'

On the other hand, the systems approach, of regarding a system only and always in the context of being open to, and adapting to, its environment, is at the heart of the conception and design process, as is the concept of emergence. It seems that there may be, in essence, two different ways of conceiving, designing and developing technological solutions:

- the classic engineering approach, decomposing the whole and synthesizing the whole from its parts such that the whole precisely equals the sum of the parts;
- and the systems approach, regarding the solution as open, adapting to environment, dynamic and potentially exhibiting requisite emergent properties, capabilities and behaviors — where the whole can be greater than the sum of its parts.

The wide variety of definitions and descriptions of systems engineering conceals a singular philosophy behind them all. Despite differences in language and application, they concern

themselves with wholes, finding answers to whole problems, analysis and synthesis, innovative design, management of complexity, integrity, quality, risk management, planning and decision-making. These considerations suggest a catch-all definition of systems engineering:

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

The objectives of systems engineering are seen as being to solve, resolve or dissolve problems (suggesting that systems engineering may have more in common with doctoring, and detective work, than with many other disciplines). In so doing, systems engineering tasks itself with conceiving and manifesting the optimum solution, i.e., the best solution in the circumstances, where best implies most able to undertake the mission of the system, making best use of available resources. The optimum solution will be one where the parts combine to create requisite emergent properties, capabilities and behaviors, i.e., where the whole is greater than the sum of the parts; emergence offers more from given resources.

Self-organizing systems occur in the natural world — the Sun is self-organizing; so, too, is the solar system. Human activity systems may similarly be self-organizing; that is people may form themselves into groups, teams, etc., adopting different roles and tasks, even reconfiguring themselves as the whole group seek to achieve some goal, defend against some threat, etc.

System of systems is a relatively recent concept, referring to the bringing together, under one umbrella, of a number of companies or organizations to form a larger whole. If the term is to have any meaning, then this larger whole should, presumably, exhibit the characteristics of a whole, i.e., the open parts should interact and complement each other such that the whole is greater than the sum of its parts.

Instances of groupings referred to as 'system of systems' do indeed conform with these aspects of a system — a city's integrated public transport system for instance — while others seem to be more groups, classes, or families of systems, rather than systems of systems: the difference is significant. A system of systems may reasonably be expected to exhibit requisite emergent properties, capabilities and behaviors, while associations and families may not; moreover, the 'design' of a system of systems is practicable using the same notions, methods and practices as for any other complex open system.

Bottom up integration refers to the practice of synthesizing technological solution systems from engineered parts, which are generally designed not to adapt to each other, although power, information and signal may pass between them. The parts do not interact, as such, but transform, and the whole is intended to be precisely the sum of the parts. This is a pragmatic approach to the construction of a variety of different technological systems from a basic set of building blocks, and is employed both in brown goods and white goods manufacture, amongst many others. This, too, may be regarded as systems engineering, in that the system that is being created is not so much the product or artifact, but is a construction system, not unlike Lego™ or Meccano™.

Finally, a look is taken at what should constitute a whole in the context of delivered systems which are expected to have unlimited existence, such as some defense systems, companies and corporations, etc. The delivered set of systems for, say, some defense project, may include the mission system (e.g., a new fighter aircraft) plus the ground support facilities (maintenance, servicing, repair, etc.) plus recruiting and training of both mission personnel and support personnel, not forgetting trainers of the trainers, and so on.

Importantly one other element may be added to create a viable whole — the system for evolving the whole, i.e., for examining the original problem and the way in which it is changing, for conceiving revised mission system and support system designs and for implementing those designs. If this element, let us call it the evolution system, is included in the whole, delivered solution

system, then that whole need have no limit to its existence, since it will be able to adapt, and accommodate change, as it arises. Similarly, it will be able to 'redesign' itself, 'reconfigure' itself, and so can evolve its emergent properties, capabilities and behaviors. In principle, by adding the evolution system to complete the viable whole, that whole becomes organic, self-organizing, an intelligent system of systems, enduring, continually optimal, and in balance with its environment: worthy goals, indeed.

Assignments

1. Develop coupled definitions of a system, and of systems engineering, without including any mention of method, process, sphere of application, etc. Define engineering, using the same rules. Identify areas of potential overlap between systems engineering and engineering as you have defined them. Discuss.
2. Systems engineering supposes there to be a best solution to any problem. Would a best solution consider the potential cost of that solution, or is the concept of 'best solution' confined to that which solves, resolves or dissolves the original problem most precisely and completely. Justify your view.
3. a) Systems engineering is applied systems science.
   b) Systems engineering is simply engineering done properly.
   c) Systems engineering adopts the systems approach.
   d) Bottom-up systems integration is systems engineering.
   e) Emergent properties are to be avoided — they indicate poor engineering. . . .

   Discuss these statements, identifying an appropriate Weltanschauung and context for each, together with likely motives behind, and validity of, each statement.
4. Explain what you understand by the term 'viable system?' What properties, capabilities and behaviors would a viable system necessarily possess/exhibit? Would you consider a viable system to be — necessarily — successful, effective, reliable, durable? Explain, with examples.
5. Define and illustrate the term 'system of systems,' with examples from your experience both of groupings of systems that warrant the term, and others that do not. Explain the differences. Do you consider the differences to be important, or not? Justify
6. Using the catch-all definition of systems engineering given in this chapter, identify the generic activities/steps that you would consider were needed to manifest an optimum solution, starting with a given problem space. You should identify no more than twelve steps, no less than six. Explain your rationale; justify your result in terms of completeness. Is the process you have used Cartesian Reduction? Explain and justify your response.