# The System Concept and Its Application to Engineering

Erik W. Aslaksen

Sinclair Knight Merz, 100 Christie Street, St. Leonards 2065, Australia

**Abstract**: The system concept is introduced as a mode of description and as a means of handling complexity. Systems engineering is the application of the concept to both management and design within complex projects; the paper focuses on the application to the front end of the design process and the transition from the functional to the physical domain.

**Keywords**: System, complexity, purpose, functional domain.

## 1. Introduction

The word "system" would appear to have originated in ancient Greece, where it related to music and meant a compound interval or a scale or series of notes extending through such an interval, but it is likely that the system concept, i.e. something being both whole and consisting of parts, was also present in other cultures, e.g. in China. In the English language, the word "system" came into use during the seventeenth century, when we find it used with a number of somewhat different, but closely related, meanings. In the early part of the century there is still the meaning of the universe, as e.g. in "In this Round Systeme All", but soon we find it applied to signify an ordered collection, as e.g. in "Man's life is a systeme of different ages" or "The yeare is a systeme of four seasons" [1].

The word "system" is used in all areas of human activity and at all levels; some examples are education system, transport system, solar system, telephone system, Dewey decimal system, weapons system, ecological system, space system, and so on; there is almost no end to the uses of the word "system" that come to mind. But what do people *mean* when they use the word "system"? To what extent is that meaning context-dependent? Is there some part of the meaning that is common to all applications? For our purpose, we only need to recognise that the meaning is either that of an *ordering*, such as in the periodic system, or something consisting of *interacting* parts, and as we shall only be interested in the latter, we shall adopt the following formal definition:

*A **system** consists of three related sets:*

- *a set of **elements***
- *a set of **internal interactions** between elements*
- *a set of **external interactions** between one or more elements and the external world; i.e. interactions that can be observed from outside the system*

We note that there is nothing in this definition that would restrict the elements or the interactions between them in any way.

## 2. Nature and Features of the System Concept

### 2.1 Linguistic classification

To get a better understanding of the system concept, we might start out by investigating how it fits into our linguistic structure. Is it a singular term, i.e. does it refer to an object? In order for "system" to be a singular term, we have to be able to say "x is a system", and then, by letting *x* point to (or reference) particular things, the truth value of the sentence will be either true or false. What is the truth value of the sentence "A car is a system"? The answer would have to take the following form: "Let C be the set of all cars, and S the set of all systems. The sentence is true if and only if C is a subset of S." We have no difficulty in determining C, but what is S? There is no such set; there is no rule that allows us to identify one thing as a system and another thing as not being a system, and therefore we have to conclude that "systems" is not a class of things.

Is it a property? Could it be that in the sentence "This car is a system" we do not mean the cupola "is" as indicating existence, but that "is a system" is the predicate associated with the singular term "this car", as in "This car is blue"? Can being a system be a property of a thing? Can we find a rule that would allow us to determine if a given thing has this property or not? There is no such rule, and therefore we have to conclude that "system" is not a property.

We get closer to understanding the concept if we recognise that when we say "A car is a system", this is an abbreviated mode of expression; what we really mean is "For our present purposes, we shall describe a car in the form of a system". There are many purposes for which it is not necessary to describe a car in the form of a system; e.g. for the purpose of describing a car as an investment object, a traffic hazard, a greenhouse gas emitter, etc., in which case one or a few global parameters are adequate. But if we want to describe its functionality and performance in more detail, the number of variables and their relationships increase rapidly, and as the *complexity* of the description increases, we find it easier to process mentally if we structure the description in the form of a system. So, it appears that a system is what Frege would call a *second-level concept* [2], or what we shall call a *mode of description*; a concept for formulating the concepts the mind uses to process its sensory inputs.

## 2.2 Cognitive basis

That the mind tends to handle complexity in this manner has been a matter of observation for some time, and has led to the realisation that complexity is relative - for example, what is complex to the human mind may be simple for a computer, and *vice versa*. The mind can manipulate objects that are characterised by more than one parameter as entities; that is, it is able to consider the parameters simultaneously rather than sequentially, as a computer normally does. But there is a limitation to this ability; as the complexity of an object increases and the number of parameters exceeds a certain number, the mind finds it rapidly more difficult to consider the object as an entity, and automatically starts to *partition* the parameters into smaller groups and to process them as separate objects. The most immediate evidence of this is language; in order to express something complex, such as a story, we use a limited set of vowels that can be combined to form words, the words are subdivided into groups (nouns, verbs, predicates, etc.) and combined to form sentences, and the whole story is a string of sentences. However, the elements need to *interact*, and in the case of language the interaction takes place in the mind of the listener and is determined by the sequence of the vowels, words, and sentences.
Another example of using the system approach is provided by structured programming. The partitioning of a program into modules is so that it is easier for the human to understand and thereby be able to verify, test, and maintain. To the computer it would make no difference if the program was just one long unstructured file.
The converse of this subdivision of complex entities is the aggregation of simple, or low information content entities, so-called *chunking*, as was first described in the seminal paper by Miller [3]. In this paper, he examined experimental data on absolute judgement, the resolution (or bits in the value of the measure) with which we can characterise stimuli without making errors, and demonstrated that the number of chunks of information that can be retained in short-term (or immediate) memory is about seven, irrespective of the number of bits of information in each chunk, as determined by the dimensionality of the stimuli (limited to the experimental data available), and he attributed this to a process of *encoding*, a learning process through repetition that relies on long-term memory. In the context of engineering, we would call the chunks “objects” or “elements”, and the encoding “information hiding”.
Miller illustrated the idea of chunking by considering a person learning Morse code. At first, every dot and dash is heard as a separate element, then these sounds are organised into letters, then the letters into words, and finally whole phrases. Another illustration of really big chunks is our ability to recognise persons; when we turn a corner and are confronted with Bill, we are able to instantly say “Hi, Bill”.

The picture of the dual processes of chunking and partitioning, as illustrated in Fig.1, will provide an important conceptual foundation for examining the process of engineering in terms of the two dual processes of top-down and bottom up design, in Sec. 5.

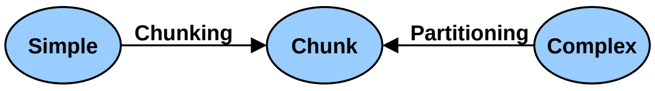

**Figure 1 : The dual processes of chunking and partitioning.**

## 2.3 Application to engineering

To see what the proper use and meaning of the system concept is in engineering, we need to recall our understanding of it as a mode of description of an object. A description of a physical object can be considered from two points of view - *what* we want to describe, and *how* we want to describe it. We never describe everything about an object (this would require an almost infinite number of variables), we describe those features that are relevant to our current purpose, such as functionality, cost-effectiveness, reliability, etc. (or any combination of such features). And we can present the description in different ways, e.g. unstructured - just listing all the parameters and their values in random order - or structured, and one of the ways of structuring the description is as a system, as per our definition of the system concept. This is illustrated in the following figure, Fig. 2.

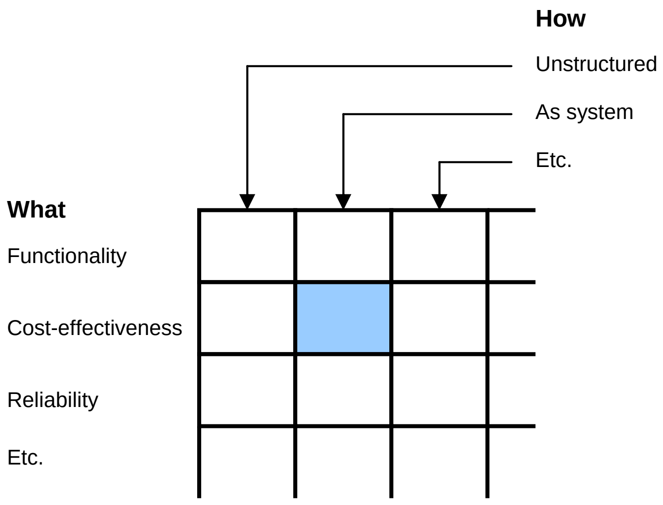

**Figure 2 : The set of all descriptions of an object.**

All the elements in the “as system” column represent systems; the shaded matrix element represents a particular one. So there are many systems associated with any one object.

## 2.4 Some features of the system concept

*Emergence*. Emergence is a central feature of the system concept, and while there is an ongoing debate about the exact nature of this feature, it is often summed up by the statement that "the whole is more than the sum of its parts". That is, the system has properties that are not evident in any of its elements. Or, from another perspective, the properties of the system are determined not only by the properties of the elements, but also by the interactions between them. Most often the properties of interest are in the form of capabilities; that is, the system has capabilities that are not found in any of its elements.

Based on our understanding of the system concept as a mode of description, there is no problem with defining the emergent properties of a system; much of the current discussion, as described e.g. in a paper by Ryan [4], would appear to arise from a lack of clarity regarding the system concept. Consider a given object. If it is described as a single object, it has no emergent properties; all its properties are simply the properties of the object. If we describe it as a system, i.e. a set of interacting elements, then the emergent properties are those that disappear when we turn off the interactions between the elements. Thus, the existence of emergent properties is simply a feature of the system concept; they are not defined by the object itself.

The interest in and research regarding emergence arise from the converse situation; that is, given a set of elements with their capabilities for interaction, how can one predict the properties of the object that results from letting them interact in a particular manner? This question has always been central to one engineering discipline, chemical engineering. One of the fundamental processes in chemical engineering is reaction, in which two or more substances are brought together under controlled conditions to form one or more new substances, and where the new substances have properties that are not found in any of the components.

*Size*. We shall call the number of elements in the system its *size*, and generally denote it by $n$. The size is related to what is often thought of as the *level of detail* of the description of an object, but equating the two can be very misleading, as becomes immediately evident if we consider the case $n = 1$. This is what is called the "black box" description of an object; it is a description limited to the externally observable properties of the object. However, that description can be very detailed and involve hundreds or thousands of variables (as inputs, outputs, and properties) and the relations between them.

The relationship between the value of $n$ and the level of detail arises only within the top-down development of the system; the size of the system (and remember, this is the description of the object) is increased in a step-wise process, from global variables into more and more detailed variables associated with an increasing number of system elements, until all the variables and relations in the original description are accounted for. This process does not (indeed, must not) add anything to the original description; it recasts it in a form that will make the design process more efficient, as is described in Sec. 5.

*Structure.* When we say that a system is a set of interacting elements, it is important to keep in mind that the interactions are purely formal in the sense that they have no properties of their own; *the ability to interact and the properties of the interactions are inherent in the elements*. That is, the potential to form a system is already present in the elements in isolation, and they form a system when this potential is realised and the interactions are "turned on". This "turning on" may take a variety of forms; electrical engineers would probably think in terms of electrical signals and protocols, mechanical engineers would maybe think of flanges and couplings or a flow of hydraulic oil, and chemical engineers would think in terms of bonds, but abstracting from any physical realisation, any one particular interaction is either active or not. This might, at first, seem like a significant restriction; it is not difficult to think of systems where the interactions vary in strength, e.g. between people or groups of people. But in reality it is not a restriction at all; we are simply saying that any such variations in the interactions must be due to corresponding variations in the properties of the participating elements.

Furthermore, we shall consider interactions to be always distinctly between two elements. This is not the only possibility, as a simple example demonstrates. Take the case of a lecturer giving a lecture to a group of $n$ students; we shall consider the interaction to consist of $n$ separate interactions. But we could have introduced a new type of interaction, a one-to-many or broadcast interaction, and represented this case as a single such interaction. This example also makes us aware that we need to allow for the case of the interaction having a *direction*, otherwise we could not represent the broadcast case correctly.

The result of this is that we can represent a system as a directed graph, with the elements as the nodes or *vertices* and the interactions as the *edges* of the graph, and we shall call this graph a representation of the *structure* of the system. In Fig. 3 we show the same set of elements with three different structures; this illustrates the fact that the same set of elements can form different systems or, as it is often formulated (although, strictly, this is not a correct wording), that a system can have a *dynamic structure*.

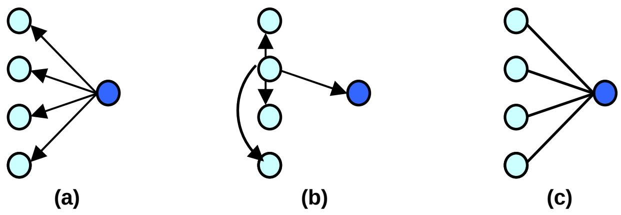

**Figure 3 : Three structures of a set consisting of a lecturer and four students. In (a) the lecturer is lecturing, in (b) a student is asking a question, and in (c) the lecturer is conducting a distance survey of the students.**

Another representation of the structure is in the form of an *adjacency matrix*, **A**, defined by

$$a_{ij} = \begin{cases} 1, & \text{if interaction from element } i \text{ to element } j \\ 0, & \text{otherwise} \end{cases}$$

From this, it follows that the structure is independent of the nature of the elements and their interactions (i.e. as far as structure goes, the elements may be considered identical and indistinguishable), but it is not independent of the number of elements in the system. However, intuitively we would say that all systems with one central element and $n$-1 elements, each communicating with this central element only, as in Fig. 3 (a) or (c), have the same structure. In other words, the concept of structure is *scalable*.

*Coherence*. The concept of coherence is perhaps most commonly used in connection with speech or thought; the Collins thesaurus lists the following synonyms of "coherent": Articulate, comprehensible, consistent, intelligible, logical, lucid, meaningful, orderly, organised, rational, reasoned, and systematic. All of these imply some form of *relationship* between the elements of a set, which is also at the core of the system concept. The concept of coherence is, of course, well known in contexts other than speech; to engineers one of the first to come to mind might be *coherent radiation*, where a set of atoms interact to form a system in such a way that their radiation is locked in frequency and phase to produce the (largely) monochromatic radiation which is the emergent property of the system. And if we simplify our view of any system to the extent that it has a single objective (or purpose), and if we, loosely, define coherence as the degree to which the capabilities of the elements contribute to realising that objective, then we could say that coherence is indeed one of the most general characteristics of a system.

In purely technological systems (i.e. no persons involved) the elements either work normally until they fall victim to random failure (as in the case of an electronic circuit), or their performance degrades gradually, often at an increasing rate (as in the case of an engine). But in both cases the degree to which what they do (i.e. their functionality) contributes to the purpose of the system remains unchanged, and the change in performance is expressed in the concept of *reliability*. A varying alignment of what elements do with the system purpose is experienced only in systems where people form part (or all) of the elements; typically, we are focusing on enterprises as the class of systems for which the concept of coherence is useful.

Comparing the concept of reliability with that of coherence, we realise that whereas the former is perfectly well defined for an element in isolation, the latter makes no sense for an element without reference to the system in which the element is embedded. Coherence is one of those parameters that emerge from the system concept itself; in a previous publication [5] such parameters were called second-tier parameters, in contrast to such first-tier parameters as reliability and performance. Thus, coherence can be seen as an additional characterisation of performance that emerges as a result of the interactions between the elements.

## 3. Systems Engineering

### 3.1 Sources of Complexity in Engineering

The driver for the application of the system concept in engineering is the rapidly increasing complexity of the projects, and there are a number of sources of this complexity. The most obvious ones include the *size* of the systems, as exemplified by transportation, power, and telecommunications systems, the *number of interacting components*, as exemplified by a modern car or a computer system, and the *number of disciplines* involved, as exemplified by manned spaceflight. But, more generally, there are two underlying developments which may turn out to be the most important drivers of systems engineering.

The first one is that, for most of the last century, the development of new technology through research was seen as an imperative for developed nations, and there was such an appetite for new technology that almost any new development found an application and a market somewhere. Only in the latter quarter did we start to notice some serious concerns about where all this technology was leading to, and whether its application was always in the best interest of society as a whole. The question started to shift from "can it be done?" to "should it be done?", and the increase in knowledge, both through travel and television, of what was happening in the world outside our own local community made us aware of the fact that we are all sharing the same limited resources and influencing a common environment. It is becoming clear that it is not just a matter of having better technology, it is also a matter of knowing how to apply this technology in the most appropriate manner. This requires an understanding of the interrelation of the application with its environment. While this was always within the scope of engineering, the immediate and direct benefits of introducing a new technology were usually so major that other effects appeared relatively insignificant. Sometimes they were actually insignificant because the scale of the application was initially so small that the side-effects, which are generally dependent on the scale in a very non-linear manner, were also small; at other times they were simply assumed to be small because no methodology existed to handle the increase in complexity involved in a proper assessment. The former reason no longer holds in many cases; for technologies such as the internal combustion engine, irrigation, and power generation the applications have grown to such a scale that what was earlier side-effects have become major effects. The latter reason is no longer acceptable to society, and the legislative framework in which engineering takes place is continually being tightened to ensure that a holistic approach is being taken to determining all the effects of every project over its life cycle. The result is that a whole new dimension of complexity has been added to engineering, creating a strong demand for adopting systems engineering as an intrinsic component of the engineering process.

The second driver, and the one of most interest in the context of this conference, is to be found in the relationship between humans and technology, which in

recent times has started to develop from a purely physical one to one involving cognitive aspects. This development is made possible by the advances in electronic data processing, and the computer itself is the best illustration of this. In the early sixties, the human-machine interface was via the card reader as input device and the line printer as output device; in between the computer operated autonomously. Twenty years later, the advent of the PC allowed a form of dialogue between the user and the machine, and today the development of the interface is about mutual understanding, or cognition. A simple example of this is the auto-correction function in a word processing program.
For systems, this development has meant that the human is no longer outside the system, as a user, but is increasingly an element of the system, and the behaviour of the human is an essential factor in the functionality of the system. As that behaviour is vastly more complex than that of any man-made component, the complexity of cognitive systems is moving system design into a new realm, one in which the application of the system concept will be the dominant paradigm.

## 3.2 A Short History of System Engineering

In order to understand where systems engineering is today, it is useful to see how it arose and what has determined its development so far. It is probably fair to say that systems engineering had its origin in the telecommunications industry.

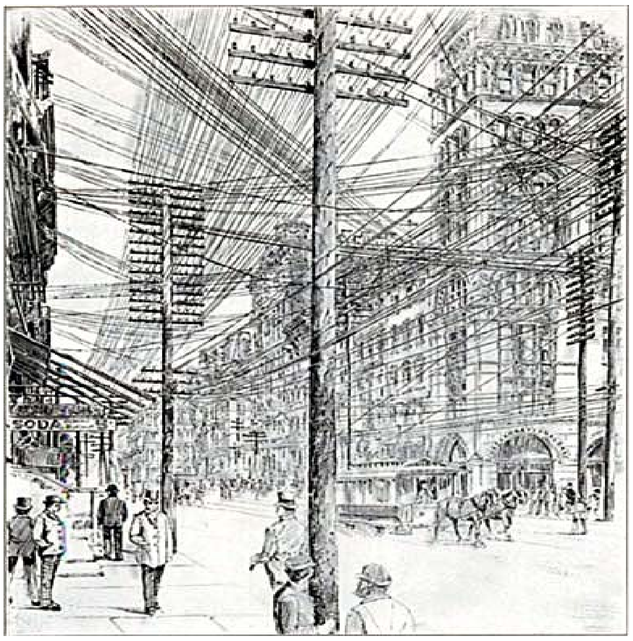

**Figure 4 : An unstructured network.**

As you can see in this etching from New York in the late nineteenth century, something had to change, and the change, as you all know, was the introduction of switching and a hierarchical structuring of the network into levels of exchanges; local exchanges, regional exchanges, national exchanges, and so on. The important point to note here is that the system principle, that is, the description of an entity as a structured collection of interacting elements, applied to the physical object, i.e. the network, only. If we think of a project as having two aspects; the object to be constructed, and the body of work required to create the object, then, in the case of the telephone network there was no need to change the way in which the engineering was done or managed, not least because the work was undertaken all within the same organisation. This was also the main reason or argument for why there had to be a single telephone company within each country; it was well understood that having several companies would introduce an additional dimension of complexity beyond that of the networks themselves.
Now fast forward to the years immediately following World War Two, and the beginning of the Cold War, whose main characteristic was a high-intensity arms race. Weapons and space systems of unprecedented complexity had to be developed in record time, and the organisations required to meet this challenge were correspondingly complex, consisting of numerous contractors. It was therefore quite natural to apply the systems approach, which had been so successfully applied to the telephone system, to *both* the physical system, consisting of hardware and software, *and* the body of work. And systems engineering was born! [6]
Systems engineering is the simultaneous application of the systems approach to both the physical system, or object, and the body of work, or project. While quite different in nature, these two systems are tightly coupled, with much commonality between the System Breakdown Structure and the Work Breakdown Structure.
Well, with this understanding of how systems engineering was born and what drove its development, it is not so difficult to see how it has become what it is today [7]. First of all, due to the huge amounts of funds available within the defence budgets to develop systems engineering as a key enabler of technological superiority, systems engineering became almost completely defence oriented. The objectives, the terminology, and the metrics became totally aligned with those of the defence industry, and have remained largely so until today. Secondly, due to the time pressure of the arms race, the focus shifted from using systems engineering as a design methodology to using it as a management methodology.
Thirdly, the defence industry has a single client; the defence department. And, what is more, this client not only understands systems engineering, but has been the driver in getting it accepted within the defence industry. This is a very different situation to what one encounters in other industry segments, where most clients know nothing about systems engineering, often have their own, well-entrenched approaches to projects, and offer no commercial incentive to contractors with a systems engineering methodology.
Fourthly, and finally, the current version of systems engineering is focused firmly on the manufacturing industry. That is, on a process that takes place in a factory, and that usually produces many items of the same type and therefore emphasizes such activities as product development, prototyping, system integration, and life

cycle support, a process that does not apply to all industries

### 3.3 Recent Directions in Systems Engineering

Over the last decades, two developments have had a significant influence on systems engineering. The first is the importance of software; more and more systems are now classified as software-intensive, and the methodologies developed within software engineering are being adopted within systems engineering [8]. An example of this is the model-based approach and the use of a universal modelling language, SysML. The second development is the greatly increased interest and activity in the area of complexity theory and complex systems, including cognitive and complex adaptive systems [9].

## 4. The Design Process

### 4.1 The Context

Design is one component of the manner in which engineers work; the *process of engineering*. The starting point of this process is a set of stakeholder (or user) requirements. The process converts that set into a complete physical description of the engineered object, which is then implemented and put into operation, and through its operation it provides a performance which fulfils the user requirements to a greater or lesser extent. It is the *intent* of the engineer that the performance of the system will fulfil the user requirements, and the degree to which this intent is achieved, i.e. the overlap of the performance with the user requirements, is a measure of the success of the engineering process, as illustrated in Fig. 5.

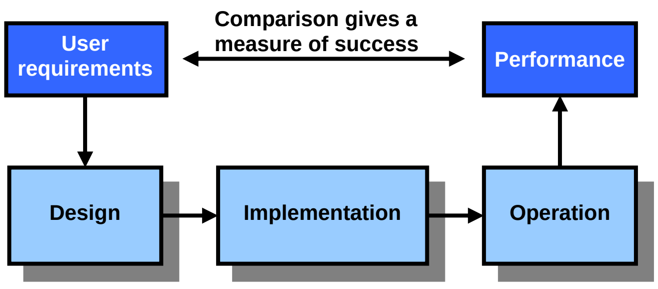

**Figure 5 : How design fits into the process of engineering.**

We see, then, that in distinction to art, where the artwork in general fulfils its purpose simply by being, or by its form, engineered products have to be involved in action over a period of time; they have to *operate*. Operation here has to be understood in a broad sense, including, for example, a bridge operating by carrying traffic, a house operating by providing shelter, and a door handle operating by transmitting the force from the hand to the door. The value created by this operation is very often dependent on the length of time the product is able to remain in its operating state, and during this time, its operating *life time*, it will undergo some form of *deterioration*. In rare cases this deterioration will be so slight as to have no effect on the operation, but for most products it will have a significant impact on the ability of the product to perform its operation, and the products have to be *maintained* in order to perform their intended operations over their life times. As a result of this, and of our definition of success for the engineering process, it is no longer sufficient for design to consider only the functionality of the product, but also its interaction with its environment throughout its life time. Maintainability and supportability have to be considered from the very start of the process, and this increases the complexity of the design significantly.

### 4.2 Top-Down and Bottom-Up

Engineering evolved from a craft to a profession when the knowledge of the craftsman was placed in a formal framework based on natural science, with mathematics as a major component of that formalisation. The manner in which that took place, and in which it has taken place ever since, is through what we shall call *construction elements* (although they are called by many other names, such as components, building blocks, unit processes, etc.). Examples of such construction elements are culverts and soil nails in civil engineering; beams, columns, and plates in structural engineering; bolts, shafts, and bearings in mechanical engineering, capacitors, resistors, and transistors in electronics engineering; transformers, generators, and cables in electrical engineering, and distillation, filtration, and reaction in chemical engineering. Through the introduction of these construction elements, engineering separated into two more or less distinct groups of activities; the development of new elements, and the use of these elements in solving design problems, which can also be formulated as the development of technology and the application of technology.
Now, while there is certainly an important design activity involved in creating new construction elements, this only takes place once per element, whereas the element may be used thousands of times in the design of new applications. As a result, engineering design has become largely the art of choosing and combining such construction elements to achieve a required performance, and this is reflected in the education of engineers. Following a grounding in physics, chemistry, and mathematics, this knowledge is used to give the students an understanding of how the construction elements work and how they are designed and manufactured. But the emphasis soon turns to characterising the elements by their external characteristics, the knowledge required to be able to combine them with other elements to form a design solution. For example, most electronics engineers will look upon a transistor as a 3-terminal device with transfer functions characterised by a few parameters, and it is unlikely they have anything but the vaguest recollection of energy band gaps, doping levels, and the like.
Today we have literally millions of standard elements at our disposal, and the efficiency of the design of physical objects is only possible because of them. Imagine if we

had to design every bolt and nut from scratch each time we needed one!
With this understanding (and simplification), engineering can be presented in the manner shown in Fig. 6, with the construction elements taking on the role of an interface between the two main engineering activities.

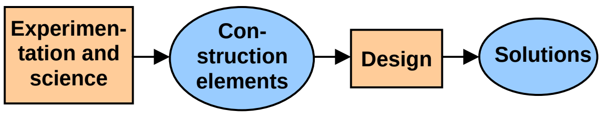

**Figure 6 : A representation of engineering, consisting of two groups of activities and two types of artefacts.**

Taking the picture presented in Fig. 6 at face value, design becomes a matter of picking the right subset of construction elements and combining them in the right way. That is, the design process is one of *synthesizing* a new object out of the set of construction elements, such that this object will provide the required service.
From that initial set, the design progresses through a step-wise, iterative process, with each step consisting of three basic activities:

a) Select a set of elements and combine them into a new object by selecting how they interact.
b) Determine the performance of this object.
c) Decide how to close the gap between the performance of the object and the required performance.

This iterative process is shown diagrammatically in Fig. 7; this figure also indicates where one enters and exits the process. While the first selection of elements is normally from an initial set based on experience, the designer is, in principle, free to include any available element in an iteration, and the process ends when the performance is tolerably close to the required performance.

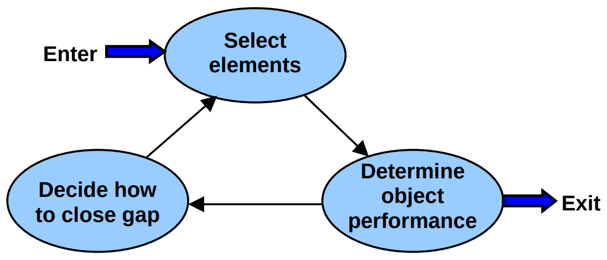

**Figure 7 : The iterative design process**.

For simple objects, or objects similar to ones that have been previously produced, this design process is a straight forward *bottom-up* process; it synthesizes new objects out of a set of existing construction elements, and the proof of correctness occurs only when the new object is created, through a test procedure which establishes that the performance of the object does indeed meet the original requirements. For more complex objects, the design process, i.e. the synthesis step, requires considerable skill and experience on the part of the designer, and it is likely that the outcome of the test is that the object does not quite meet the requirements. A revision of the design is required before a repeated test establishes that the requirements have been met. As the complexity of the object increases, more and more iterations are required to reach a satisfactory result, and the design process becomes increasingly *inefficient*.
With our knowledge of the application of the system concept, an obvious solution to this problem of inefficiency is to somehow subdivide the process into a number of sub-processes, each of which results in a simpler object, but such that when all these simple objects are brought together and allowed to interact, they form an object that satisfies the original, complex requirements. That is, we precede the bottom-up process by a *top-down* process, in which the functional requirements are analysed and partitioned into interacting subsets of functional requirements, or *functional elements*, each of which is simple enough to be efficiently realised through a bottom-up process.
The bottom-up and the top-down processes are compared in Fig. 8, and Fig. 9 (on the next page) illustrates how the focus of the design moves from the bottom-up to the top-down process as the complexity of the project increases.

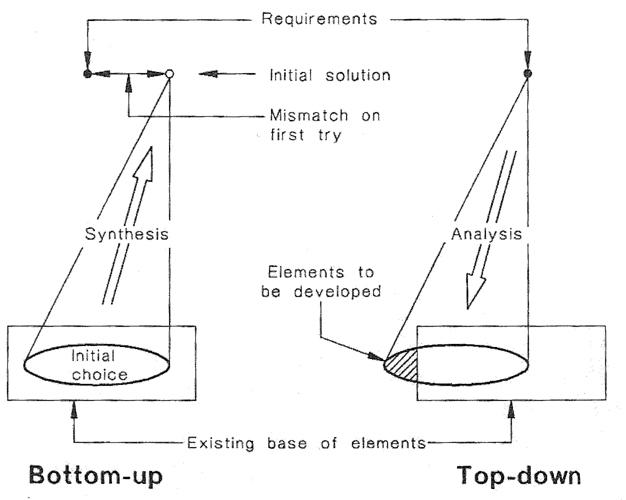

**Figure 8 : Comparison of the bottom-up and the top-down design process. The upper part of the figure is the requirements space, the lower the space of physical elements.**

## 5. The Functional Domain [11]

### 5.1 Some Definitions

Engineering design operates with *descriptions* of engineered objects and, as we all know, these descriptions fall into two types - descriptions of what objects *are* (their physical substance), and descriptions of what the objects *do* and how well they do it (their behaviour). The former consists of such data as size, shape, material specification, surface finish, etc., presented in the form of drawings, schedules, etc., and is what is needed by someone who

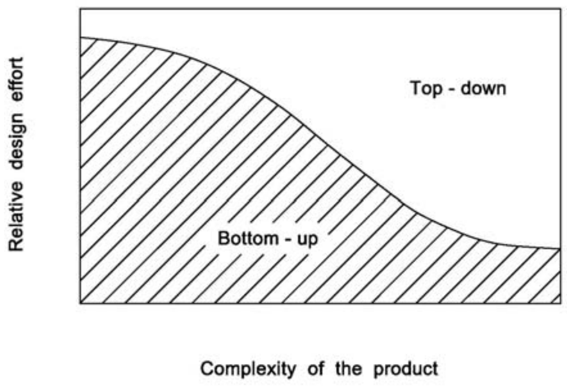

**Figure 9 : The relative effort expended on bottom-up and top-down design as a function of the complexity of the project. [10]**

wishes to reproduce (manufacture) the object. The latter consists of performance parameters, such as load capacity, reliability, processing speed, etc., and their values. This description is, for example, what is needed by users to determine if the object will meet its requirements (i.e. fulfil its intended purpose). Both descriptions refer to the same entity - the engineered object, and are concerned with physical aspects of the object - substance and behaviour.

Definition 1: The *physical domain* is the set of all physical descriptions of engineered objects.

We now come to a central point in our development of the application of the system concept to engineering, - the realisation that, in addition to the two types of descriptions of engineered objects identified above (of substance and behaviour), there is a third type of description, and it arises as a consequence of the requirement for an engineered object to provide a service. Because of this requirement, it is always possible to abstract from what an object does and how well it does it and describe only the service it is intended to provide, completely disassociated from *how* it does it and from any physical aspects of the object. This is best illustrated by means of a simple example - removing the cork from a bottle of wine. That is the service required by the user; the physical solution provided by engineers may take many different forms, such as a simple cork screw, a cork screw combined with some mechanism for extracting the cork using a smaller amount of force, a thin, hollow needle attached via a valve to small pressurised gas cylinder, a two-pronged device which is inserted between the cork and the bottle, and so on. For any one of these we can give a description of what the object is, by means of drawings, etc., that would allow it to be manufactured without any knowledge of what its purpose is. We could also give a description of how it works, e.g. in the case of the simple corkscrew by means of such parameters as how many turns are required, what torque is required, how much force is required to extract the cork, etc., and this would allow us to determine if it actually fulfils its purpose, i.e. meets the service requirements. These two descriptions would, of course, both be different for different solutions, but all the solutions have in common their purpose, the service they are intended to provide, which we shall call their *functionality*. As a formal definition of functionality we shall adopt the following:

Definition 2: The *functionality* of a physical object is its intended capability for interacting with its operating environment.

The word "intended" expresses a very significant difference between physical and functional descriptions; every statement in a physical description can be verified by an examination of the physical object, whereas the functionality of an object depends on the *intention* of the designer, which again is determined by the requirements of the stakeholder group. The functionality cannot, in general, be deduced from looking at or performing measurements on an object, and there is not necessarily any functionality inherent in a physical object, i.e. disconnected from the intention of its designer. Deducing the functionality of a physical object is, of course, what we call reverse engineering, and its accuracy will depend on what additional information is available.
A description of functionality will generally require a number of variables and a number of functions defining the relations between these variables, in order to fulfill the many clauses of a typical requirements definition document. These variables and functions can often be grouped according to distinct subsets of the stakeholder requirements, and it is therefore possible to regard a description of functionality as made up of a number of individual parts, each one describing some *aspect* of the functionality. Examples of aspects are the capability of generating earnings (cost-effectiveness), the capability of providing continuity of service (availability), the capability of surviving in a given environment (reliability), and the capacity for producing its service (size or rating).

Definition 3: A *functional element* is a description of one or more aspects of the functionality of a physical object, and consists of a set of variables and a set of functions between them, as well as any values required of the elements of these two sets.

Definition 4: The *functional domain* is the set of all functional elements.

A functional element is a description, but it is very different in nature to the two types of descriptions we are used to (substance and behaviour). It describes neither what a thing is nor what it does; it does not describe anything physical, nothing that can be measured with physical measuring equipment. What it describes is an idea, a desire, an outcome, a *capability*, something we can imagine before there is any physical object which would produce it. However, it is important to realise that, while the functional element itself is an abstraction, the purpose

or service it describes is very much in the physical domain, so that the functional parameters are normal, physical parameters. For example, an element whose functionality is to generate electric energy could be characterised by such parameters as power rating, conversion efficiency, etc.
Now, what does the functional domain look like? First of all, a description of functionality can be viewed as consisting of two parts - the set of functional parameters and the values of these parameters. The functional parameters are the equivalents of the properties in the physical descriptions. That is, just as one property of a car is its colour, and the value is e.g. red, a functional parameter is reliability, and the value is e.g. 0.998. Secondly, just as a physical object can be described in more and more detail, so can a purpose or a service. As a result, functional elements are characterised both by the type of service they describe and by the level of detail, i.e. the number of parameters used in the description.
The relationship between a functional element and an engineered object is similar to the relationship between a class and an instance of the class. But whereas a class is a collection of its instances, the functional element is not a collection of engineered objects. The classes, as collections of engineered objects, exist in the physical domain, and such a class consists of all the engineered objects that are intended to provide the same service. There is a one-to-one mapping between such classes and functional elements, but the two exist in two different domains. *A functional element is not a description of the performance of an engineered object*. This is illustrated in Fig. 10. The relationship between the two domains emphasizes something we all know, but that tends to be forgotten - our task as engineers is to give people the means to do things, to carry out activities and achieve objectives; the hardware and software only constitute our solution to that task.

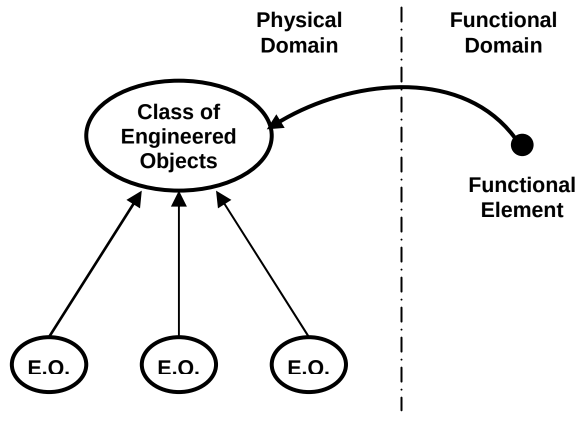

**Figure 10 : The relationship between the functional and the physical domains.**

As defined above, a functional element consists of a set of variables and a set of functions between them. But the set of variables can be divided into subsets that have quite marked differences, and it is recognizing and understanding these differences that allow the concept of a functional element to be further developed and made more precise.
To this end, we first have to introduce the concept of the *service*:

<u>Definition 5:</u> The *service* provided by a physical object is the immediate purpose of its operation, its *output* to the subset of the stakeholder group usually called the *users*.

<u>Definition 6:</u> *Functional parameters* are the parameters describing a service.

Referring to the definition of "aspect" and the example of power generation above, the service provided by a power station is the supply of electrical power. Its rating is a functional parameter, whereas the actual value of the power flow at any time is a variable characterizing the external demand.
The second subset is the one consisting of the additional variables describing interactions required in order to provide the service, such as power, materials, manpower, waste heat to the environment, etc. As all these other variables used to describe the functionality are only there because they are required by the functional parameters to be there, they may be called *dependencies*. They are all related to the functional parameters through the set of functions which forms part of the definition of a functional element and describes the behavior of the element.
Together with the functions between them, these two sets completely describe the functionality of the element, i.e. the intended interactions with the rest of the world, and constitute what we might call the functional variables (or simply variables, when we are considering functionality). However, there is a third subset, consisting of those variables that, while they do not describe intended interactions, describe necessary interactions with the rest of the world. These are normal (i.e. non-functional) variables, usually describing those characteristics of the environment in which the object operates that affect the values of the functional parameters, such as interest rate, state of technological development, availability of trained manpower, and political stability, just to mention a few that are different from those that spring first to an engineer's mind, such as temperature and humidity. Their values are given by the environment and cannot be changed by the functional element. (This restriction is in reality only satisfied as an approximation, albeit most often a very good one, as any interaction implies the involvement of both parties to the interaction.) The variables in this third subset could be called *influences*, and a symbolic representation of a functional element could therefore be as shown in Fig. 11.

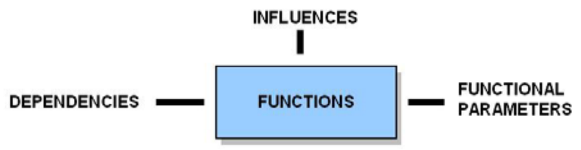

**Figure 11 Symbolic representation of a functional element.**

## 5.2 Design in The Functional Domain

Our interest in the functional domain, and therefore also the reason for the foregoing very brief introduction to it, is that the design of a system generally starts with a set of functional requirements. What the stakeholders want is primarily a service; the physical system is the engineers' solution to providing that service.

So, a complex system is a system designed in response to a complex set of functional requirements, i.e. a very large functional element, and this element forms the point of departure for the design effort. But, as was discussed in sec. 4.2, making the transition from the functional domain into the physical domain becomes more inefficient as the size of the functional element increases, and we are less likely to achieve anything like an optimal solution. To overcome this problem, we would like to express the large functional element as a system of smaller, simpler functional elements, and to make the transition into the physical domain for each one of them, arriving at a system of physical elements.

The process of partitioning a large and complex set of functional requirements into subsets is what we might call *design in the functional domain*. This process starts with a simple statement of the basic function, or purpose, of the system, and then proceeds in a stepwise, top-down fashion by at each step, and for each functional element, choosing a set of functional elements that will realise the purpose, and flowing down the requirements (i.e. the values of the variables) from the preceding step into the variables of the new functional elements. Such a step in the process has some important features: First, there is almost always a choice of the new set of elements; the purpose of the "mother" element can be realised in different ways. That was illustrated in the case of the "uncorker"; another example is the functional element which provides a laundering service. This functionality could be provided by having a laundering facility in each home, or by providing central laundering facilities and a collect-and-return service; the breakdown into smaller functional elements would obviously be very different in the two cases.

Second, the elements must be *interacting*; it is through the interaction that a set of simpler elements can have a complex functionality. Again, there is a choice of the interactions that are active between the elements, but, as was discussed following Definition 4, these interactions are physical interactions and therefore, once chosen, flow directly into requirements on the physical elements.

And, finally, there is the requirement that, at the end of the step, there is a verification that the choice of elements and their interactions actually provide the functionality of their "mother" element. In effect, that means building a model of the "mother" element in terms of the new elements, and therefore each step of the top-down design can be seen as building a more detailed model of the functionality.

The crucial part of this process is, obviously, being able to identify the initial, or top element, the one that results from stripping away as much detail as possible from the initial large and complex set of functional requirements. Seeing that this set is a different one in each case, it would at first appear that this top element would also be different in each case. But, on the other hand, as we strip away detailed requirements, sets that differed only in these requirements become identical; that is, this less detailed set is common to all these cases. It is then not unreasonable to ask if there is some requirement (or small set of requirements) that is common to all cases. The answer to this question can be developed as follows:

The creation and operation of any system must involve an expenditure of resources in some form, such as labor, energy, information, and materials, and even forms which are not normally or easily measured in monetary terms. But would anyone incur a cost without any prospect of a return? The return might not be directly in terms of money; it can be in the form of personal well-being, absence of illness, peace of mind, a sense of achievement, and so on, but indirectly, as with the cost, this can, in principle, always be measured in monetary terms.

The return can only occur once the system has been created and put into operation, so that the expenditure must come before the return and is therefore an *investment*. All engineering projects are subject to this cycle of investment, creation, and return, and as there are many possible projects, all competing for a finite set of resources, the fundamental purpose of the engineering process is to maximize the return on investment, in the sense that while all other considerations may be ignored in a process of simplification, this one cannot be ignored or simplified away without making the design process irrational. Or, in other words, the constraint of competition for limited resources is the essence of the process of engineering; that which makes it fundamentally different from a science. Engineering is not about truth, but about cost-effectiveness, with the effectiveness being judged by the stakeholders. Consequently, the functional element representing this fundamental purpose, Return on Investment (ROI), can be said to be an *irreducible* element, and it is not dependent on the particular system, but is, because it has its genesis in the process which creates all systems, universal to all systems.

## 5.3 The irreducible functional element.

In order to produce a return on the investment, any system must put its functionality into operation, and that operation is described by the values taken on by the functional parameters. The design identifies the value each of the parameters needs to take on in order to meet the requirements of the stakeholder requirements; they are the *nominal design values* for the particular system. The degree to which the system actually meets these requirements can, in principle, be characterized by a single variable, the system *Quality of Service*, or QOS, which is a function (often the weighted average) of the degree to which the individual parameters meet their nominal design values. This single performance parameter is denoted by S, and while it is called "quality", it may in fact often be mostly, or even wholly, a measure of the quantity of whatever the system produces. It just depends on what is the most important feature of the system's functionality, what Hitchins calls Prime Directive [12].

As already stated, the creation and operation of any system must involve an expenditure of resources. This expenditure may take many forms, but in the context of engineering, they must all be converted to monetary terms, e.g. dollars, and together they shall be termed the *cost* of the system, denoted by C.

No system can continue to operate, producing a service and incurring costs (or expending resources), without receiving something in return which sustains the operation. This shall be called the *revenue*, and no matter in what form it is received, it must, within the context of the process of engineering, be expressed in monetary terms. As with the costs, this may not always be a simple matter, as is exemplified by such a return as "quality of life" in e.g. a foreign aid project; this is part of the vast increase in complexity that occurs once engineering goes beyond the traditional boundaries of objects and outcomes directly describable in terms of physics. The revenue may be denoted by R.

All of the above takes place within a time period; no system can perform any service in an instant of time, nor can it be created instantly. The time period over which all activities associated with a system take place is its *life cycle*; in the terms of a living organism, this is the time period between conception and death. The duration of the life cycle, which may just be called the *life* of the system, will be denoted by L, and measured in units of time. Both the cost and the revenue are referenced to the beginning of the life cycle, i.e. they are the sums of the Present Values of costs and revenues, respectively, incurred throughout the life cycle.

These two parameters L and S, and the two variables R and C, are necessary and sufficient for expressing the concept of Return on Investment, denoted by Q,

$$ROI = Q = 100\left[\frac{R(S,L)}{C(S,L)} - 1\right]\%,$$

and constitute what we shall call *the basic set*.

The two parameters S and L appear only implicitly, but that does not make them any less necessary. If the cost and the revenue did not both depend on the quality and the duration of the operation; well, then, why not ask for perfection and that it last forever?

We can now define a very special functional element, which we shall call the *irreducible element*, as that element which is common to all engineered objects and that has the unique property of unifying functionality and purpose. It can be represented graphically as shown in Fig. 12, with the functional parameters (i.e. those parameters that enter into the definition of the functionality) on the right-hand side, and the functional variables (i.e. those parameters describing the interaction with the outside world required in order for the element to provide its functionality) on the left-hand side.

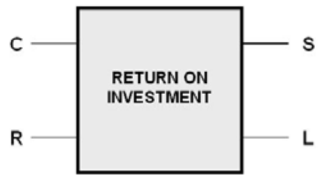

**Figure 12 : The irreducible element Return on Investment.**

## 5.4 Value, the missing link.

In order for the Return on Investment to provide the point of departure for the design process, there is a "missing link" that needs to be established; the connection between service and revenue. That connection is provided by the value concept; the revenue is determined by what the users perceive the value of the service to be to them, essentially what they are (or would be) willing to pay for the service. Clearly, the value of a given service is not an absolute measure; it exists only in relation to a particular user group. Indeed, this dependence of the value on the user group provides an alternative definition of the user group - the group of all persons who have a say in determining the value of the service provided by the system.

Assigning a value to a service is not unproblematic, and one encounters both the view that engineering should be strictly science-based and not concerned with value judgements, and the view that it is almost immoral to put a (monetary) value on all services. Both of these views miss the point that the value of a service is not determined by engineering, but that without defining the value, design of complex systems, where there is always a trade-off or optimisation process involved, ceases to be a rational process. *The concept of value provides the link between the engineer and the user.*

In terms of starting the design process with the irreducible element, the value is expressed in terms of a value function, W(S). The function W(S) is often a highly non-linear function; for a number of reasons there is a relatively narrow range of S in which W(S) increases rapidly with increasing S [13]. Below this range, the quality of the service is so poor that it is basically useless. Above this range, an increase in S brings hardly any further increase in the value; it is a region of saturation or "overkill", as shown in Fig. 13 on the next page. In the region of interest it is therefore often possible to linearise the function, and this fact is used in a very simple design tool that standardises the first step in the top-down design process and provides a framework for developing further functional elements. This tool, which is fully transparent in the form of an Excel workbook, can be downloaded from www.gumbooya.bigpondhosting.com.

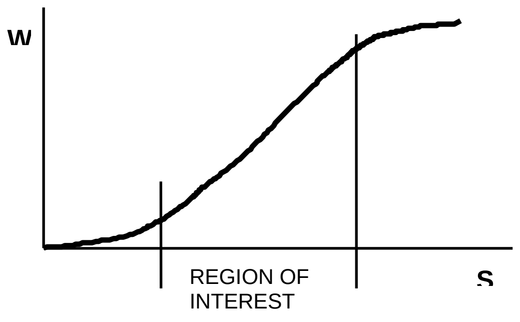

**Figure 12 : The value function W(S).**

## 6. Conclusion

The system concept is closely related to our cognitive processes; it is a reflection of how the brain handles complexity. Its application to engineering started in telecommunications in the early part of last century, but mainly through its application in the defence and aerospace industry has it developed into its current form of a process-oriented methodology for managing complex projects, generally under the name of systems engineering.

However, starting from the realisation that there are two distinct aspects to the complexity of systems – the complexity of the functional requirements, i.e. what the system must do, and the complexity of the solution, i.e. what is required to meet the requirements – the central idea presented in this paper is that the complexity of a set of functional requirements can be reduced by applying the system concept in a top-down process prior to starting the classical design. This process, design in the functional domain, operates with systems of functional elements, and it is suggested that the initial element is always the same, one representing the Return on Investment, as maximising the ROI should be the common purpose of every engineering project.

## 7. References

[1] The Oxford English Dictionary, Second Edition, Clarendon Press, Oxford (1989).

[2] The distinction between objects and concepts, and the subdivision of concepts into first- and second-level concepts are discussed by Frege in "On Concept and Object", Vierteljahrsschrift für wissenschaftliche Philosophie, 16 (1892), pp. 192-205. We would perhaps call a second-level concept a *class* of concepts.

[3] Miller, G.A., *The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information,* The Psychological Review, vol. 63, pp. 81-97, 1956, available online at www.well.com/user/smalin/miller.html. References to subsequent papers can be found at http://citeseer.nj.nec.com/context.

[4] Ryan, A., *Emergence is coupled to scope, not level*, Complexity, xx, (2007). A condensed version appeared in Insight, the newsletter of INCOSE, vol.11, no.1, January 2008, pp. 23,24.

[5] Aslaksen, E.W., *A model of system coherence*, Systems Engineering, vol. 6, no. 1, 2003, pp. 19-27. A somewhat abbreviated version can be found in *Designing Complex Systems*, CRC Press (2008), sec. 8.6.

[6] A good overview of the early years of systems engineering is given in the report by Hans Bode, 'The Systems Approach', in *Applied Science - Technological Progress*, report to Committee on Science and Astronautics, US House of Representatives, 1967.

[7] One of the most well-known texts on systems engineering is that of B. Blanchard and W. Fabrycky, *Systems Engineering and Analysis*, Prentice Hall, currently in its 4th Edition (International Edition), 2008.

[8] Boehm, B., *Unifying software engineering and systems engineering*, Computer, March 2000, pp 114-116.

[9] Complex systems is such a wide and diversified field that no single reference provides a good entrance. There is a journal devoted to systems of simple components, but with complex overall behaviour, the *Journal of Complex Systems*, and there are two well-known research institutes, the Santa Fe Institute, and the New England Complex Systems Institute.

[10] This and the previous figure are from the article by E.W. Aslaksen, *Coming Up? Going Down!*, Engineering World, August 1991, pp. 18-22.

[11] The material in this section is largely an extract from the author's recent book, *Designing Complex Systems – Foundations of Design in the Functional Domain*, CRC Press, Boca Raton, 2008. A much more detailed development of the functional domain can be found there.

[12] Hitchins, D.R., *Systems Engineering: A 21st Century Systems Methodology*, Wiley, 2007.

[13] The value concept and the value function are discussed in more detail in Ch. 7 of an earlier work by the author, *The Changing Nature of Engineering*, McGraw Hill, Sydney, 1996.