A Critical Examination of the Foundations of Systems Engineering

Erik W. Aslaksen, INCOSE Fellow
Sinclair Knight Merz
100 Christie Street, St. Leonards, NSW 2100
Australia

Abstract *This half-day tutorial consists of two main parts. In the first part, we establish a common understanding of a philosophical framework suitable for examining the foundations of systems engineering, and then proceed to look at the system concept within that framework. We ask such questions as: Is it a thing? Is it a quality of a thing? Does it exist within space-time? Is it a universal? And if so, what type of universal? In finding answers to these and similar questions, we shall develop a clear understanding of the system concept through its relationship to other concepts and, above all, of why it is such a powerful concept and of how it must be employed to exploit that power.*

In the second part, we look at the application of the system concept to engineering and endeavour to establish a framework for systems engineering. That is, what features must be common to all systems engineering methodologies? What distinguishes systems engineering from other types of engineering?

The purpose of the tutorial is to provide INCOSE members with a depth of understanding of these basic issues, which will allow them to judge for themselves what the merits of a particular methodology are relative to their application, and hopefully also to eliminate some of the misunderstandings and confusions that occasionally creep into our discussions. The tutorial does not promote any particular methodology, and is of no direct practical use whatsoever.

Contents and Organisation

The tutorial is structured into four 50 minute sessions, with a ten minute break between sessions. It is intended that each session would consist of approximately 40 minutes lecturing and 10 minutes discussion, but the approach to this tutorial is meant to be flexible and adaptable to the desires of the participants.

INTRODUCTION

Systems engineering is a relatively new discipline and one still very much in its early development stages. Significant work has been done on defining the methodology and developing process models, and there is a great deal of activity in extending the methodology into a variety of areas. New variants and associated modeling tools are being put forward at an increasing pace, and organisations such as INCOSE are promoting the very real, practical benefits of the methodology. But in this development rush it is easy to lose sight of the underlying, basic nature of the system concept and the entities and principles associated with it, and there is the danger that new developments can become disconnected from this kernel of knowledge, thereby losing both much of their purported benefit and their legitimacy. They may use all the right words, but do not reflect an understanding of the meaning and significance of these words.

In the established sciences, such as mathemathics, physics, and chemistry, there is a vast body of work that is concerned with the philosophical aspects of the science and answering basic questions about existence and our knowledge of that existence. Now, engineering is different from these sciences, and whereas science is about truth, engineering is about usefulness, and in a general manner of speaking, anything that is useful and cost-effective is good engineering and does not need any philosophical justification. That may be so, but, aside from ethical issues, which are not relevant in the present context, this is only a partial view of engineering; a view limited to the outcome of engineering, or what we might call a "black box" view. If we look inside that "black box", the core activity is a creative process, an activity that takes place in the minds of engineers. That is where philosophy comes in; understanding how the mind works, the nature of ideas, and the boundaries of reason are central issues in philosophy.

In this tutorial we seek to provide a critical examination of the philosophical foundations of the system concept and its application to engineering. It is not about promoting any particular process view or any tools - it is not about doing, but about understanding. It is also not an overview of the current state of systems engineering nor the history of its development, as it is assumed that most of the participants in this tutorial have a strong background in systems engineering. After a short introduction to philosophy, where we review the definition of the concepts and terms we shall be using in order that we all "sing from the same sheet of music", we take a thorough look at the system concept; where and in what context it originated, and how it developed into its current form. We then investigate the different contexts in which the concept is used today and the meaning attached to it in those contexts, and attempt to place the concept within the framework of philosophy. Is it a thing? Is it a quality of a thing? Does it exist within space-time? Is it a universal? And if so, what type of universal? In finding answers to these and similar questions, we shall develop a clear understanding of the system concept through its relationship to other concepts and, above all, of why it is such a powerful concept and of how it must be employed in order to exploit that power. For the rest of the tutorial we specialise to the context of interest to us, systems engineering, which allows us to provide both a much sharper definition of the concept and a more detailed framework for its application.

The second part of the tutorial looks at the implications of our new-found understanding for systems engineering, i.e. for the design of systems. Irrespective of the particular approach taken and the models and tools used to support it, we can identify certain features that should be present and certain rules that should be obeyed in order for any approach to be characterised as systems engineering. However, there are also features which we can choose to include or not, and it is these that will form the subject matter of a discussion to which it is hoped that the participants will provide significant contributions.

PART A - SYSTEMS AND PHILOSOPHY

1 A VERY BRIEF OVERVIEW OF PHILOSOPHY (1)

1.1 The Characteristics of Philosophy

The Collins dictionary defines philosophy as : a. Pursuit of wisdom. b. Study of realities and general principles. c. System of theories on nature of things or on conduct. d. Calmness of mind. While all of these characteristics, with perhaps the exception of the last one, apply to the purpose we have in mind, we can also formulate the characteristics of philosophy by saying that philosophy attempts to answer questions distinguished by their *abstract* and *ultimate* character. Abstract means without reference to concrete circumstances or to any particular physical reality. Abstraction is something we engineers in general have a great deal of difficulty with, and even when we employ abstract concepts, we really see in our mind's eye a particular physical entity. It is the old issue of knowing what the solution is before we have examined the problem, and while this is often a strength, it becomes increasingly a weakness as the complexity of the problem increases.

Causality is an ingrained part of our human nature; when we observe something happening, we are certain that something caused it to happen. And equally ingrained in our nature is the need to know what the cause is; uncertainty is the worst state for the mind to be in. We must have an answer to every question and an explanation for every event. Sometimes they are *rational* explanations; that is, explanations that are subject to verification by measurement or logic. But if we cannot find a rational explanation, we would rather accept an irrational one, based on belief, rather than remain in a state of uncertainty. However, as can be experienced when dealing with small children trying to come to grips with the world around them, the explanation of one event just leads to a new question about the cause of that event, and so on in a chain of questions and answers that usually ends with the answer "Because that's just the way it is". Philosophy is concerned with the existence of ultimate questions, such as "Is there an ultimate cause?", and while we shall not be concerned with that particular question, the method of enquiry that leads to it will be important to us.

1.2 The Subject Matter of Philosophy

There is, in principle, no boundaries to the subject matter of philosophy; the philosophical method of enquiry can be applied to anything and everything, from religion and morals to the philosophy of science. The emphasis and the tools used (e.g. linguistics) may vary considerably form one area to another, but common to all philosophical enquiry is that it uses *reason*, i.e. the capacity of our minds for maintaining, linking, and analysing thoughts using a particular schema that we call *logic*.

There would not appear to be an established area of philosophy that could be called "philosophy of engineering", but there is no reason why there should not be. Engineering is seen as a very practical, down-to-earth activity, very much about specific, physical objects, but if we examine it more closely, we recognise that the essence of engineering is creativity. Engineering takes place in the mind of an engineer, so the many questions about the mind, the rules by which it operates and the identity of the objects on which it operates, are highly relevant to engineering. And one can certainly pose the "ultimate" question "What is the purpose of engineering?", a question which should be of considerable interest when we consider the prevasiveness and importance of engineering in modern society.

1.3 Structure of Philosophy

Although in principle philosophy is about providing an understanding about the whole of things, an integrated account of the world in which all truth will be harmonised, in practice this is too daunting a task, and its practitioners tend to concentrate of a particular area, which then develops somewhat of an identity of its own. At a first level of partitioning, philosophy can be divided into pure and applied philosophy. In the first of these, the philosophical activity is independent of any application-specific theories and concpetual frameworks, such as one finds in physics, medicine, and religion, just to name a few. In the second, it is exactly to these theories and frameworks that the philosophical enquiry is directed.

On the next level of partitioning, pure philosophy can be subdivided into the following branches:

a. *Logic*. This is the study of reasoning, the process by which the mind is able to reach conclusions about the relationships between items of knowledge. The process works in two "directions"; when reaching a conclusion about combining existing item of knowledge to form a new item it is called *synthesis*, when discovering new items of knowledge by dissecting a known item of knowledge into its constituent parts it is called *analysis*.

b. *Epistemology*. This is the study of knowledge; of what it means to know something, what it is possible to know, how knowledge is generated in the first place, its relationship to perception, and so on. Within epistemology, one often distinguishes three differing viewpoints:

- *Empiricism*, closely related to *verificationsim*, with experience being the basis of all knowledge and understanding;
- *Rationalism*, with knowledge being generated by reasoning alone, independent of any experience; and
- *Realism*, closely related to *idealism*, with knowledge about an object having a reality independent of the physical object being thought about.

c. *Metaphysics*. This is the theory of being; of what exists and what is meant by existence. The things that exist make up our *ontology*, so that we can say that metaphysical activity generates an ontology in order to achieve a cogent description of reality. Or, in other words, the ontology contains the things that philosophy is about, and this leads to a further classification. On a first level of partitioning, the two main groups are:

- *particulars*; and
- *universals*

Particular fall into two main groups - physical particulars and abstract particulars. The former are the "things" of everyday life - a shoe, a book, a car - and within these there could be further subdivision according to the degree of *substantiality*, in the sense that a brick is more substantial (or delineated) than a heap of sand. Examples of abstract particulars are numbers and sets.

Universals can be grouped into four main categories:

- *properties*, such as colour or size;
- *relations*, such as "greater than" or "between";
- *kinds*, such as "human" or "suspension bridge"; and
- *mass terms*; such as "energy" in the question "how much energy?".

d. *Ethics and aesthetics*. The theory of value, of what is good and what is bad.

e. *Semantics*. The theory of meaning and truth. This refers to our use of language; the theory uses linguistics as a basis, but goes beyond it to look at meaning in the sense of reference and logical implications. Because of their importance to our purpose, the next subsection gives a slightly expanded discussion of these issues.

1.4 Sentence Structure, Reference, and Meaning (2)

We are all familiar with the subject-predicate type of sentence structure, such as “Lucy is tall”. Here “Lucy” is the subject term and “is tall” the predicate term; the sentence predicates tallness of Lucy. This type of sentence structure is perhaps the most important one for our purposes, but there are other structures. A sentence such as “All cars have wheels” does not fit the subject-predicate structure, because “all cars” is not a subject. To make sense of this sentence, and to extend the analysis of the logic of a sentence to all types of sentences, we introduce the concept of a *variable*, say x, and the notion of sets. Let Y be the set of all objects that have wheels and X the set of all cars, then the meaning of the sentence is that $x \in X \Rightarrow x \in Y$. This same approach applies to the subject-predicate structure; if X is the set of all objects that are tall, the meaning of the sentence is that there exists an x that is identical with Lucy and $x \in X$. It also applies to a sentence like “Lucy exists”; the meaning is that there exists an x such that x is identical with Lucy, or $\exists x : x = \text{Lucy}$.

Returning to the subject-predicate sentence; now that we have looked at it meaning in terms of a variable, we should look at the components of the sentence. First, the subject-term. A name like Mary is a member of a class called *singular terms*, which also includes such items as “a man” or “the Prime Minister”, and all singular terms *refer* to objects (we shall return to the nature of objects later). What exactly is meant by “reference”? We shall be satisfied with our intuitive understanding of it as the relationship that holds between a singular term, such as “Lucy”, and Lucy herself. We realise that the main purpose of language is to be able to refer to objects and make statements about them.

Now to the predicate term. When a predicate is combined with a singular term, it makes a statement about the singular term, and the complete sentence has a *meaning*. Meaning has two dimensions, a *sense* and a *reference*. The sense of the sentence “Lucy is tall” is the idea that someone can be tall, the reference is the truth-value of the sentence - either true or false, depending on the object (person) to whom the singular term refers. In mathematical terms, the concept of “tall” is a function from object to truth-value of the sentence.

This can be summarised as follows:

a. For a singular term, its sense is the understanding that it refers to an object, and its reference is the understanding of what particular object it refers to.

b. For a predicate term, its sense is the the understanding that it refers to a concept, its reference is the understanding of what particular concept it refers to.

c. For the sentence, its sense is our understanding of the relationship between the singular term and the predicate term. It is composed of the sense of the singular term and the sense of the predicate term, but it is more than just the sum of them; *the sense of a sentence is an emergent property.* The interaction between the terms is governed by the rules of *syntax*. - The reference of a sentence is its truth-value.

To finish off this subsection, we need to briefly consider the work of Noam Chomsky, as it has, in a certain sense, provided the inspiration for the basic idea behind this tutorial - that there is a close connection between the system concept and its pervasiveness and a certain feature of the mind (3). Chomsky developed a rigorous description of a language in terms of its *grammar*, consisting of a *lexicon* and a set of *rules*, with the latter subdivided into a *syntactic component*, a *semantic component*, and a *phonological component*, and then showed that all languages have certain features in common, i.e. there would appear to be a basic, universal grammar, and all languages are a variation or transformation of this basic grammar. He then drew the conclusion that this is so because of innate properies of the mind; that is, the languages children learn are the ones that these properties of their minds make them predisposed to learn.

The significance of this for our purposes is that Chomsky analysed an aspect of human capability or behaviour and from the results of this analysis drew conclusions about the mind; until then it had mostly been the other way around. That is, assumptions about features of the mind had been used to explain linguistic knowledge and laguage use. In the second section of this tutorial we shall show how a conclusion can be drawn about a feature of the mind by analysing the use of the system concept, and we will also use the fact that a language is itself a system.

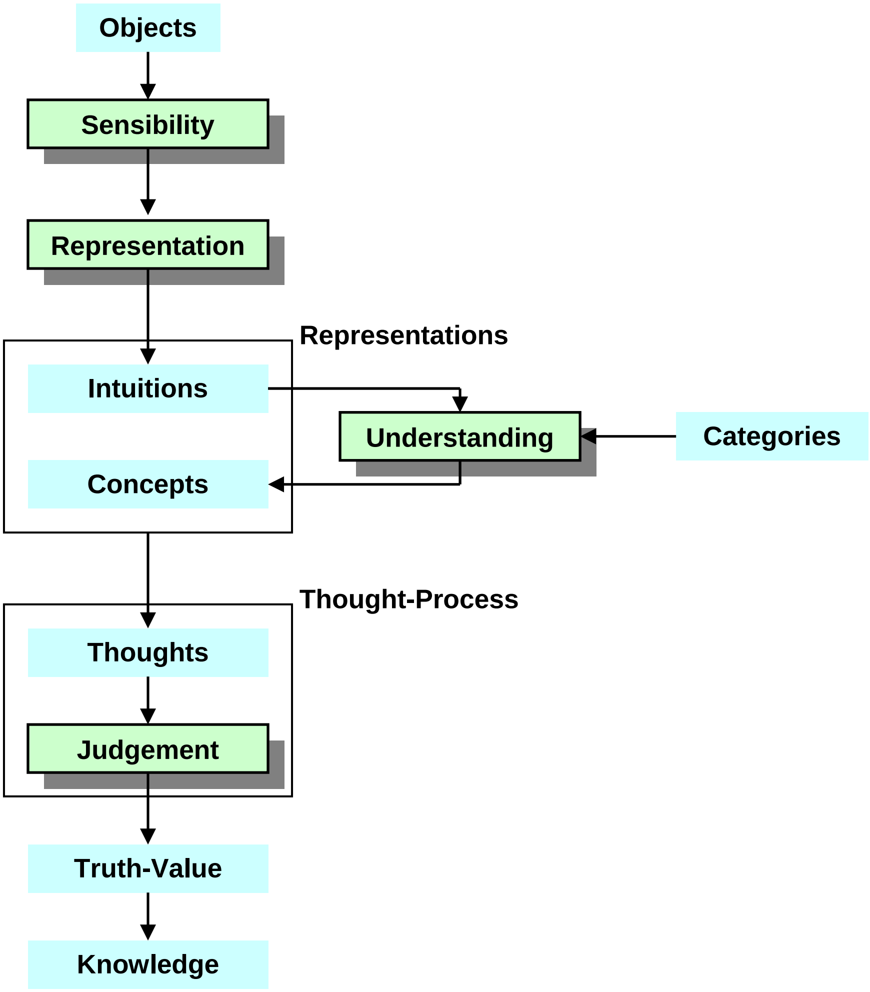
1.5 The Philosophy of Mind (4)

In the previous subsection we introduced the words "understanding" and "concept". These words do not refer to anything in the world outside us; they refer to our mind. "Understanding" is a process that takes place in the mind, and "concept" is something produced by that process and manipulated by it. The philosophy of the mind must be central to philosophy, in the sense that understanding the properties of steel must be central to designing steel structures. We are still at the very beginning of understanding how the mind works, and besides the basic question of to what extent the mind can be used to explore the mind, there is the question of whether the mind and its processes are purely physical or whether there is something "in addition" to the physical; a mental level of existence. It has been, and still is, one of the central issues in philosophy, but in the next subsection - the last in this section - we shall show that it is not really relevant to what we are about. However, before doing so, let us just take a quick look at what is perhaps still the best insight into the nature and working of the mind - the one provided by Immanuel Kant (5).

Two of the main faculties of the mind are *representation* and *understanding*. When we observe external objects through our senses, the faculty of representation turns these *sensations* into *empirical intuitions*. In addition, the faculty of representation is able to generate *pure intuitions*, intuitions to which there corresponds no object that is perceived through the senses, such as an angel. The faculty of understanding processes the intuitions and generates *concepts*. Concepts are classes of intuitions, and when we say we understand what an object is, it means that we know to which concept it is related. In that sense, a concept can also be considered to be a rule for the reproduction in imagination of a set of intuitions. In the words of Kant (roughly, my translation): "Without sensibility no object would be given to us, without understanding no object would be thought. Thoughts not related to intuitions are empty, intuitions without concepts are blind. It is, therefore, just as necessary to make our concepts sensible, that is, to add the object to them in intuition, as to make our intuitions intelligible, that is, to bring them under concepts. - The understanding can intuit nothing, the senses can think nothing. Only through their union can knowledge arise."

The major step taken by Kant was then to argue that concepts are not formed in an arbitrary manner; they have certain characteristics or modes, which Kant called *categories*, which are preexisting (*a priori* of any sense input) in our mind. Categories are to concepts what space-time is to (empirical) intuitions. Intuitions exist within a space-time framework; concepts exist within the framework of the categories. Kant postulated five categories (6): Reality, magnitude, substance, cause, and wholeness, and it is the latter that is of particular interest to us. The ability of the mind to conceive of a number of interacting or related intuitions as a whole is an intrinsic ability, not something we learn through experience.

Both intuitions and concepts are *representations*, which constitue the entities on which the thinking process operates. This is illustrated in the following diagram.



1.6 Abstract Entities and Linguistic Frameworks (7)

In the two preceeding subsections we have seen that, whether we are considering the thought process or the formation of sentences, the entities we are using or processing refer either to a physical object or to an abstract concept. The question of whether these abstract entities have any real existence (e.g. as in Plato's ideas) or not has been and still is a central one in philosophy, and because we already suspect that the concept of a system will be an entity of the abstract kind, this question could be very relevant to our present quest to place the system concept within the framework of philosophy. However, we shall adopt a point of view put forward by Rudolf Carnap, which effectively allows us to sidestep the question for our purposes.

The argument advances in a number of steps. In the first step, we recognize that when we wish to speak (or think) in a language about a new kind of entities, we have to introduce a system of new ways of speaking, subject to new rules, and we shall call this procedure the construction of a *linguistic framework* for the new entities in question. The second step is to recognise that we must distinguish two kinds of questions of existence: On the one hand, questions about the the existence of entities of the new kind *within the framework*, which we might call *internal* *questions*; on the other hand, questions concerning the existence or reality of *the system of entities as a whole*, which we might call *external questions*.

The third step includes, in effect, a two-component definition of reality, and is best illustrated by a couple of examples. Firstly, the simplest kind of entities dealt with in the everyday language - the spatio-temporally ordered system of obsevable things and events. That is, what we above called empirical intuitions. Once we have accepted the thing language with its framework for things, we can raise and answer internal questions, such as “Is there a white piece of paper on my desk?” and “Did King Arthur actually live?”. These questions are to be answered by empirical investigations, and the concept of reality occurring in these internal questions is an empirical, non-metaphysical concept. To recognize something as a real thing or event is to succeed in incorporating it into the system of things at a particular space-time position so that it fits together with the other things regarded as real, according to the rules of the framework.

The external question would be the reality of the thing world itself, and this is something completely different. To accept the thing world means nothing more than to accept a certain form of language, i.e. to accept rules for forming statements and for testing, accepting, or rejecting them. The acceptance of the thing language leads, on the basis of observations made, also to the acceptance, belief, and assertion of certain statements. But the thesis of the reality of the thing world cannot be among these statements, because it cannot be formulated in the thing language.

The decision to accept the thing language is usually not a deliberate, cognitive one, because we have all accepted the thing language early in our lives as a matter of course. However, it will nevertheless usually be influenced by theoretical knowledge, just like other, deliberate decisions concerning the acceptance of linguistic or other ruules. The purpose for which the language is intended to be used, for instance, the purpose of communicationg factual knowledge, will determine which factors are relevant for the decision. The efficiency, fruitfulness, and simplicity of the use of the thing language may be among the decisive factors. And the questions concerning these qualities are indeed of a theoretical nature. But these questions cannot be identified with the question of realism. They are not yes-mo questions, but questions of degree. The thing language in the customary form works indeed with a high degree of efficiency for most purposes of everyday life. This is a matter of fact, based upon the content of our experiences. However, it would be wrong to describe this situation by saying “The fact of the efficiency of the thing language is confirming evidence for the reality of the thing world”; we should rather say instead “This fact makes it advisable to accept the thing language”.

The second example is the system of natural numbers, a system that is of a logical rather than a factual nature. The linguistic framework for this system is constructed by introducing into the language new expressions with suitable rules:

a. Numerals like “five” and sentence forms like “there are five books on the table”;
b. the general term “number” for the new entities, and sentence forms like “five is a number”;
c. expressions for properties of numbers, e.g. “odd” and “prime”, relations, e.g “greater than”, operations, e.g. “plus”, and sentence forms like “two plus three is five”; and
d. numerical variables, “*m*”, “*n*”, etc. and quantifiers for universal sentences, such as “for every *n* …” and existential sentences, such as “there exists an *n* such that …”, with the customary deductive rules.

Here again there are internal questions, such as “Is there a prime number greater than a hundred?”, but the answers are found not by empirical investigation based on observation but by logical analysis based on the rules of the new expressions. The reality of numbers within the framework simply means that the set of numbers is not empty; the external question of the existence of numbers prior to the framework cannot be answered by analysis.

In summary, the acceptance of a new kind of entities is represented in the language by the introduction of a framework of new forms of expressions to be used according to a new set of rules. There may be new names for particular entities of the kind in question, but some such names may already occur in the language before the introduction of the new framework. The two essential steps are, firstly, the introduction of a general term, a predicate of higher level, for the new kind of entities, permitting us to say for any particular entity that it belongs to this kind, e.g. as in "red is a *property*" or "five is a *number*", and secondly, the introduction of variables of the new type. The new entities are values of these variables, and with the help of the variables, general sentences concerning the new entities can be formulated.

1.7 Notes to this Section

1 A very good introduction to and survey of philosophy is Roger Scruton, *Modern Philosophy*, Mandarin (Reed Int'l Books Ltd), London 1996, and this section is indebted to that book. Another well-known book is Harald Höffding, *A History of Modern Philosophy*, Dover Publications, Inc., New York 1955, and a very readable introduction is Will Durant's *The Story of Philosophy*, Simon and Schuster, New York 1953. Two useful Internet references are the Internet Encyclopedia of Philosophy at http://www.utm.edu/research/iep/ and the Stanford resource at http://plato.stanford.edu/.

2 The philosophy of language had its beginnings with the works of Gottlob Frege, in particular with his 1892 paper *Ueber Sinn und Bedeutung,* or "On Sense and Meaning", in Zeitschrift für Philosophie und philosophische Kritik, 100 (1892), pp.25-50, translated in P. Geach and M. Black, *Translations from the Philosophical Writings of Gottlob Frege*, Blackwell, 3rd ed. Oxford (1980). Further discussions of Frege's work can be found in Michael Dummett, *Frege: Philosophy of Language*, Duckworth, London, and Harvard University Press, Cambridge MA, 2nd ed. (1981).

3. The seminal work is Noam Chomsky, *Syntactic Structures*, Houton & Co., S'Gravenhage (1957), 2nd printing 1962. Chomsky's ideas are futher developed in his books *Language and Mind*, enlarged edition, Harecourt Brace Jovanovich, Inc. (1972), and *Rules and Representations*, Columbia University Press, New York (1980). A very good review is given in N. Smith and D. Wilson, *Modern Linguistics*, Penguin Books Ltd. (1979).

4. A major work is that of G. Ryle, *The Concept of Mind*, Hutcheson, London (1949); a more recent book with an extensive bibliography is G. Rey, *Contemporary Philosophy of Mind*, Blackwell, Cambridge MA and Oxford (1997).

5. The central work is Immanuel Kant, *Kritik der reinen Vernunft*, originally published in 1781, with a second edition in 1787. A later complete reprint is that published by Th. Knaur Nachf., Berlin, and the standard English translation is that by N. Kemp-Smith, Macmillan Press Ltd, which can also be found online at http://humanum.arts.cuhk.edu.hk/Philosophy/Kant/cpr/. A very good discussion of Kant's ideas is given by R.P. Wolff, *Kant's theory of mental activity*, Cambridge MA (1963).

6. Kant originally listed twelve categories, organised into a Table of four groups of three categories each, but further consideration and analysis of Kant's comments on his categories have concluded that it is more appropriate to consider there to be only the five listed here. For example, Kant had subdivided the category "magnitude" into three concepts - unity, plurality, and totality.

7. Rudolf Carnap, *Empiricism, Semantics, and Ontology*, Revue International de Philosophie 4 (1950), reprinted in the Supplement to *Meaning and Necessity: A Study in Sematics and Modal Logic*, enlarged edition, University of Chicago Press (1956), and available online at http://www.ditext.com/carnap/carnap.htm.

2 THE SYSTEM CONCEPT

2.1 A Brief History (1)

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, for reasons which will be discussed in the next Section, 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 Latin, it retained the meaning of a music interval, but was also used to signify “allness” or “wholeness”, as in “the universe”.

In the English language, the word came into use during the seveteenth 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 used to signify an ordered collection, as e.g. in “Mans life is a systeme of different ages” or “The yeare is a systeme of four seasons”, or “Aristotle is more noted for his order, in bringing Morality into Systeme, ... and distinguishing vertues into their several kinds, which had not been handled Systematically before, ...” And in astronomy one now speaks of “The systems of the world, i.e. Ptolemaick, Tychonick, and Copernican”. Another couple of examples of a similar meaning (i.e. ordered or systematic) are “System is a treatise or body of any Art or Science” and “That there might no vice be wanting to make his Life a systeme of Iniquity”.

A use that stands on its own is “System and Hypothesis have the same Signification, unless, perhpas, Hypothesis be a more particular System, and System a more general Hypothesis”.

One of the first instances of where the interaction of the parts is a specific feature of the system concept is in Hobbes (1651), “By Systemes, I understand any numbers of men joyned in one Interest or one Business”.

In the last century there was a great upsurge in the interest in systems, which has continued with increasing intensity into the present day. On the one hand, this was driven by the realisation that “everything is interconnected” and that the compartmentalisation into (non-interacting) specialist areas of knowledge has some serious limitations; this is the basis of what is known as General Systems Theory (2). On the other hand, the size and complexity of man-made objects required a new approach to their design and development, and this became what is now Systems Engineering (3). As a result of these developments, couupled with the popularisation of technology and science, the word “system” has become a very common word in everyday language, and we need to take a quick look at this common usage before we turn to its use in the more restricted context of systems engineering.

2.2 Everyday Use and Meaning

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? These and similar questions, all relating to the use of the word “system” in everyday language, need to be given careful consideration if we are to achieve a clear understanding of the underlying system concept itself before specialising to the engineering context.

Let us first see what the dictionaries say. The Collins Shorter Dictionary (HarperCollins 1991) contains the following entries: 1. complex whole, organization; 2. method; 3. classification. The Australian Pocket Oxford Dictionary (Oxford University Press, Melbourne, 2nd ed. 1984) contains the following entries: 1. complex whole, set of connected things or parts, organized body of things, set of organs in body with common structure or function, method, organization, considered principles of procedure, classification.

It would appear that the common part of these definitions could be expressed by saying that for something to be characterised as a system, it would have to *consist of parts that are related in some way so as to allow us to perceive it as a whole*. Let us look at a number of everyday sentences involving the word "system" and see if this holds true.

a) The health system is in a mess.

In this sentence, the word "system" refers in a general way to everything that is related to providing health services, so while no parts are directly identified, the use of the word implies that the whole, the health system, consists of many parts. We could be led to say that the meaning of the term "the health system" is the set of all objects whose primary function is to provide some aspect of health services, but then saying that it is in a mess would impy that there is something wrong with all members of the set, which is not what we want to imply. The individual doctors, nurses, ambulance drivers, etc. may be doing a fine job, but by using the word "system" we want to emphasize that it is the output, i.e. the health services, that are unsatisfactory. Thus, while the relationship between the parts is mainly one of "belonging to", it is more than that; in this case "system" is more than a set, and there is some form of interaction between the parts. So if we allow interactions as a type of relation, then our above characterisation of a system applies.

b) Every atom has its place in the periodic system.

Here the word "system" has the meaning of "order" or "taxonomy". There is no interaction between the parts, and the parts themselves are not physical entities, but classes or types of physical entities, i.e. atoms. The system has no output or properties, and we do not use the word "system" to imply that we view the periodic system as a whole. So our above characterisation of a system does not apply in this case.

c) There seems to be no system in the way these taxes are levied.

In this sentence the word "system" has the meaning of "rule" or "order" or "lawfulness". There is no implication of parts being viewed as a whole, but it does imply that "system" would contain certain relationships, e.g. between income and tax. So again, our characterisation does not quite apply.

d) The whole system is rotten.

Whenever the word "system" is used in this way, no matter what "the whole system" refers to, the details of what belongs to or makes up "the system" are left unspecified, but the meaning is always that whatever the system *does* is "rotten". Here again we see the difference between set and system; a set does not *do* anything, a system often does (but not always, as b) shows). We would have to say that our characterisation does apply in this case.

So, is it possible to find something in the meaning of the word "system" that is true in all uses of the word? It does not seem to be, and this points to a core problem in systems engineering - that the word "system" is used so frequently and so loosely that it has lost much of its value. The value can be brought back in by defining the meaning in a particular context, such as engineering, which we shall do in the following subsection. But before doing so, we should note two things. Firstly, as the above examples show, the uses and meanings of "system" fall into two distinct groups. Both groups consider a system to consist of a set of elements, but in the one group - the one that we shall be interested in and consider the “real” meaning of “system” - the elements are interacting and form a whole that has properties that are not found in any of the elements (the emergent properties), whereas in the other group, the elements are not interacting, and the whole is just the sum of the elements. In this latter group, “system” is more or less synonymous with “ordering”, and may be considered a degenerate version of the first group, in the sense that the interactions in the first group are identically zero. Getting these two groups confused is one of the most common problems in discussions within INCOSE about the meaning and properties of systems.

Secondly, as always when restricting a concept to a particular context, we have lost its general applicability and convenience. In order to operate with this concept, we have to have been instructed in its meaning and use. This situation is somewhat analogous to the use of the word “God”; it is very useful when we want to signify something like “whatever one believes in” without going into specifics. Once we narrow the context to a specific religion or sect, we need to have been taught what the meaning is and its proper application.

However, if we restrict ourselves to the first group above, we shall argue, in section 3, that there is indeed something common to all applications of the system concept; it is just of quite a different nature to what we have been looking for in the above examples.

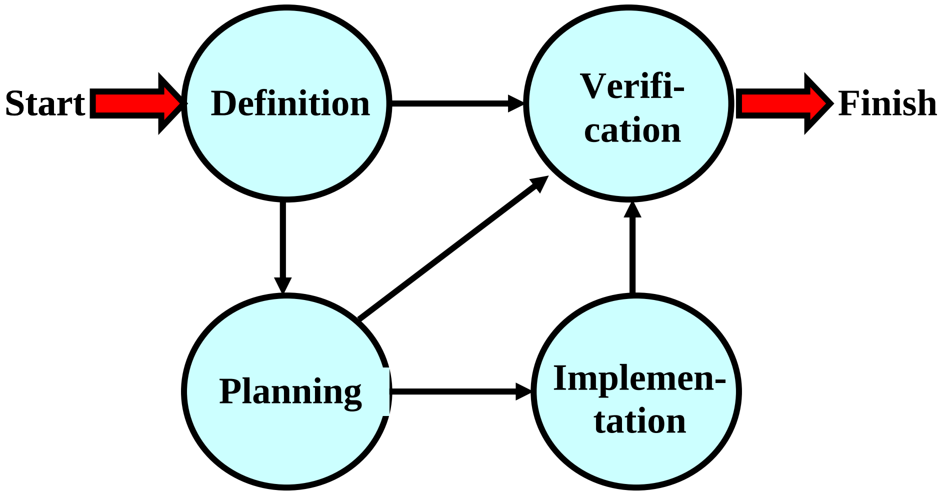
2.3 The System Concept In the Context of Systems Engineering

As we saw in the historical overview, engineering began to turn its attention systems during the 40s. But this was not from any particular interest in the system concept itself; it was from the very practical necessity of finding a word that would characterize the increasingly large entities that had to be considered as a whole in order to obtain the desired performance. Words such as “element”, “component”, “module”, and “equipment” had relatively well defined meanings, but what should we call entities that clearly consisted of a number (often large) of interconnected pieces of equipment?

But it was not only the size of the entities that needed a new word to describe it, it was perhaps above all their *complexity* that required a new approach to their design and description. In all areas of professional activity - the law, medicine, business, and engineering - the level of *complexity* of the subject matter has been increasing, and there are at least three major reasons for this. One is simply the accumulation of knowledge, another one is the increasing interrelation between all elements of society - individuals, political, ethnic, and social groups, and geographically separate groups - due mainly to greatly increased means of communication. And a third one is the explicit inclusion of humans as individuals into processes where they were previously either ignored or only considered in an implicit, cursory manner.

A first response to this increase in complexity, as a means of handling the sheer mass of data and knowledge, was the emergence of *specialisation* and of the specialist that knew more and more about less and less. However, this subdivision into isolated areas of knowledge carried with it its own limitation - the narrower the focus, the less likely it was to provide a true representation of reality, because in reality, everything is interconnected. A second, and more sophisticated, response was to overlay the specialisation with a layer of activity, which, to any desired extent, provided a connection between the specialisations, and the central concept of this layer was that of a *system*. The system concept is a means of handling complexity (rather than reducing it, as specialisation does), and a system is a view of a complex object as consisting of a set of less complex objects, the system *elements*. But, in order for the set to represent the original object, the elements need to be *interacting*; the original object is represented partly by the properties of the elements and partly by their interaction. And that is the core of the system concept. Were they not interacting, they would simply form a collection of the less complex objects.

There is, in principle, no limitation on the type of complex objects the system approach can be applied to. If the object is a complex physical object, the system elements are physical subsystems, such as the hull, the propulsion, the steering, the services, and the fitout in the case of a ship. If the object is a complex body of work, the elements are tasks, and to illustrate the application of the system approach, consider such a complex body of work, consisting of a vast number of individual activities, all interacting in some way. We first group these into a small number of main groups of activities, typically definition, planning, implementation, and verification, interacting as shown in the diagram below. Already at this first level we can take some tentative decisions, such as allocating a duration to each group, allocating resources and determining who is going to be responsible for each group, etc. In a next step, each of these groups are again subdivided into smaller work packages and take more detailed decisions, and so on, until we arrive at a large set of tasks, but where each task is not more complex than that we can conceptualise it and define it as an entity in all its detail.



We are now in a position to define *systems engineering.* When we use the term “engineering”, we generally mean a structured approach to creating something that will meet predefined requirements in a cost-effective manner. Engineering therefore involves two entities - the object that is going to meet the requirements, and the body of work involved in creating it. For any significant project, both of these entities are complex, and they are obviously intimately related. *Systems engineering is the simultaneous application of the system approach to both the object and its creation.*

2.4 A Definition of “System” (4)

For the rest of this tutorial, we shall focus on the system concept in the context of systems engineering, but it will be obvious to you that much of what we develop will be relevant to a much wider field of applications. However, we shall also use this restriction to engineered systems, and thereby our exclusion of such systems as e.g the solar system or the decimal system, to state that all systems have a *purpose* and that they have been *designed* by the engineer with that purpose in mind.

Within that restriction, and in order to give our deliberations the necessary degree of precision, we need to agree on a definition of a system. Throughout the development of systems engineering, there have been numerous definitions of what a system is, a more or less random selection of a few will demonstrate the range of precision and applicability:

> *A system is an array of components designed to accomplish a particular objective according to plan.* (Johnson, Kast, and Rosenzweig, 1963)

> *A system is a set*
>
> $$Z = \{S,P,F,M,T,q\}$$
>
> *where:*
> *S is a set not empty*
> *P is a set not empty,*
> *F is an admissible set of input functions with values in P,*
> *M is a set of functions each defined on S with values in S,*
> *T is a subset of R containing 0,*
> *q is a function defined on F x T with values in M such that q is onto and:*
>
> - *The identity mapping I is in M, and for every f in F,*
> $$q(f,0) = I$$
> - *if f is in F, and s, t and (s+t) are in T, then*
> $$q(f,t)q(f,s) = q(f,(s+t))$$
> - *if f and g are in F, and s is in T, and f(t)=g(t) for all t in R(s), then*
> $$q(f,s) = g(f,s).$$
>
> (Wymore, 1967)

> *A system is defined as a set of concepts and/or elements used to satisfy a need or requirement.* (Miles, 1973)

> *To define a system it is necessary to define the inputs; it is necessary to define the states; it is necessary in some cases to be explicit about the outputs, although this is sometimes arbitrary; and finally, it is necessary to describe how the system changes state in terms of its input and present state. The output of a system is any function of the state of the system. Each state of the system must contain all the information necessary to compute the desired output of the system at any time.* (Wymore, 1976).

Within INCOSE itself, the SE Handbook defines a system as

> *An interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, people, information, techniques, facilities, services, and other support elements,*

and the Fellows have adopted the following definition:

> *A system is a construct or collection of different elements that together produce results not obtainable by the lements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce systems-level results. The results include system level qualities, properties, characteristics, functions, behavior and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected (Rechtin, 2000).* http://tucson.sie.arizona.edu/sysengr/INCOSE/WhatIs.html

All of these definitions are proper and have their area of applicability, some wider, some more narrow. For reasons that will become apparent, we shall use the following, very broad, 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.*

The internal interactions take place between elements, the external interactions take place between the elements and the external world. In both cases, the *possibility* of supporting the interaction must be a characteristic of the element; the set of interactions is merely a formal or logical set denoting which of these possibilities are actually used in a particular system. That is, the interactions do not express any additional functionality not already inherent in the elements.

The above definition does not conflict with any of the other definition; it was chosen to allow the widest possible applicability, while at the same time be precise enough to be useful in an operational sense.

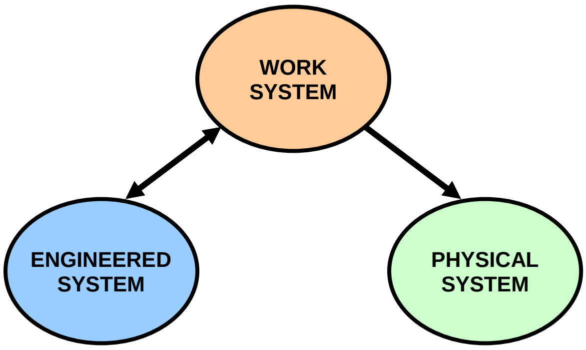
There is one point that needs to be made about our definition of a system, and that is the use of the term *set*, and in particular, as it is used in the set of elements. One possible definition of a set would be any collection of elements, such as {Paris, 4, grey}, where the elements have nothing in common except that they are elements of this (arbitrarily) defined collection. Our definition will be more precise (and more commonly accepted in mathematics), in which the elements all belong to a *universe* of elements, such as the natural numbers {1, 2, 3, …}, and the set is defined by a *property* which selects its elements from the universe, such as e.g. the set of even numbers.

2.5 Types of Systems and Relations Between Them

A first classification of systems is afforded by looking at what we apply the system concept to. As we saw in the subsection on the definition of systems engineering, we can apply it to a physical entity, as in “the radar system” or “the production system”. In this case the elements of the system are physical - they are things that exist in space-time and have properties - but otherwise there is no limitation on what can be an element, it can be hardware, software, people, etc., and the interactions between them can take on any form that these elements can support. In systems engineering, the physical system is most often referred to simply as “the system”, but in this tutorial we will need to distinguish between various applications of the system concept, so we shall continue to call it “the physical system” (but more about this later, when we shall see that it is not a system at all).

We can also apply the system concept to the body of work required to create a physical entitiy. This is what is often called “the project”, but we shall call it “the work system”. Here the elements are work packages, and the interaction between them is either a transmission of information (as e.g. between two design stages) or a handover of physical components (as e.g. on an assembly line). It is not correct to say that work “exists” in space and time; it is “done” in space and time. It is only the result of work that exists, and therefore there is a very close coupling between work and its result. We often define a work package in terms of its result, as in “the work required to produce the construction documtation for the bridge” or “the work required to construct the bridge”.

There is a third type of entities to which we can apply the system concept, and to identify it we need to look more closely at the role of documentation in the systems engineering process. During the phases leading up to implementation (e.g. definition, analysis, and design) we produce various items of documentation that serve only to provide the information required to implement the system; once the physical system is up and running, they have no further purpose. During implementation we also produce items of documentation that are required as part of operating and maintaining the physical system, such as manuals and spare parts provisioning documentation. These latter items are elements of the physical system, but the former actually constitute a distinct type of entities, and applying the system concept to them results in what we might call "the engineered system". This type of system, which is the type of greatest interest to systems engineering, is the subject of the last of our four sections. It is obviously closely related to the other two types of systems, and the relationships are indicated in the following figure:



2.6 Notes to This Section

1. The early history of the word "system" is taken from The Oxford English Dictionary, Second Edition, Clarendon Press, Oxford 1989.

2. General Systems Theory can be said to have its beginning with the work of Von Bertalanffy from 1932 onwards, and the publication of his book "General Systems Theory", by George Brazillier in 1941. This lead to the creation of the International Society for Systems Science (ISSS).

   Some well-known references are:

   Laszlo, E., ed., *The relevance of General Systems Theory*, George Brazillier, 1972

   Bowler, D.T., *General systems thinking: its scope and applicabiity"*, Elsevier, Amsterdam, 1981.

   Boulding, K.E., *The World as a Total System*, Sage Publications, Beverly Hills, 1985.

   A recent book, which gives a good overview of the current state of GST, is the one by Lars Skyttner, *General Systems Theory - An Introduction*, Macmillan Press, Basingstoke, 1996.

3. 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.

   Three other standard references are:

   Blanchard, B.S. and W.J. Fabrycky, *Systems Engineering and Analysis*, Prentice-Hall, 1981/90 (2nd ed.).

   Chestnut, H., *Systems Engineering Methods*, Wiley, 1969.

Johnson, R.A., F.W. Kast, and J.E. Rosenzweig, *The Theory and Management of Systems,*McGraw-Hill, 1963.

Sage, A.R., *Methodology for Large-Scale Systems,* McGraw Hill, 1977.

4. There are as many definitions of systems as there are authors on subjects involving systems. In General Systems Theory some of the (very broad) definitions are

"A system is anything unitary enough to deserve a name", Paul Weiss (biologist);

"A system is anything that is not chaos", Kenneth Boulding;

"A system is a structure that has organized components", West Churchman.

The definitions quoted in the text are from the following books (in addition to the one by Johnson *et al.* referenced in 3.):

Miles, R.F. (ed.), *System Concepts,* Wiley, 1973.

Wymore, A.W., *A Mathematical Theory of Systems Engineering - the Elements,* Wiley, 1967.

- , *Systems Engineering Methodology for Interdisciplinary Teams,* Wiley, 1976.

- , *Model-Based Systems Engineering,* CRC Press, 1993.

3 THE SYSTEM CONCEPT IN THE FRAMEWORK OF PHILOSOPHY

3.1 Nature of the System Concept

We now have, on the one hand, an understanding of how the word "system" is used both in daily language and in the narrower context of systems engineering and, on the other hand, an understanding of a philosophical framework into which all manifestations of human activity must fit in some way. We should, therefore, be able to address the fundamental question: *What is the nature of the system concept within that framework?*

a. *Is it a singular term?* 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 iff $C \subset S$." 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.

b. *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 (as in a.), 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 answering our initial question 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 we might suspect that the system concept is, in some way, a reflection of the way our mind works, and I would venture to make the following statement:

> *The system concept is a practical manifestation of Kant's view of how the mind processes information and forms concepts (i.e. general descriptions). In particular, the core of the system concept - viewing the whole as the result of interacting parts - is nothing but an application of Kant's fifth category, the ability to see the whole as made up of parts.*

So, if we accept this statement, we can say that a system is what Frege would call a *second-level concept* (1), or what we shall call a *mode of description*; a concept for formulating the concepts the mind uses to process its sensory inputs.

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 group 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 which 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 (except perhaps with regard to memory requirements) if the program was just one long unstructured file.

Besides the observation that the mind tends to subdivide complex entities, it can be observed that the number of parts into which the mind prefers to subdivide a complex whole is in the range 4 - 10 (2). We could now speculate that there might be a physiological explanation for this behaviour, in the same way that Chomsky postulated that the universal structure of grammar had its explanation in the functionality of the mind. In the following subsection we propose a very simple (simplistic?) model of memory activity, which demonstrates this characteristic.

3.2 A Simple Model of How the Brain Handles Complexity (3)

Let us first introduce the concept of a *unit of information*; this may be information either about a parameter or a variable, or about the relationship between any two variables. The size of such a unit, measured, for example, in bits, we do not know, but it is not important here. To store such a unit of information must involve a physical change to the brain in the sense of local ordering of some sort; this means an increas in entropy. Therefore an amount of energy has to be expended. It is generally accepted that memory is of two kinds, long-term and short-term, and, as we are most interested in thinking processes, we shall consider only short-term memory. Let the short-term nature of the memory be characterized by a *decay constant*, c, and let us (for simplicity) assume that the failure model is one of constant failure rate. Then, if the unit of information is set in memory at time $t = 0$, the probability of it being intact (or available) at thime t equals $e^{-t/c}$. The amount of energy required to set or reset (refresh) a unit of information in short-term memeory, divided by the decay constant, shall be called the *characteristic power level* and be denoted by ε, measured in watt.

We need next to describe the thing or things we think about. Such a *thought-entity* could be described as consisting of N variables and M relations between them. Because we shall (for simplicity) assume that, within a thought-entity, a relation is always between two variables, the maximum value of M is $(N^2 - N)/2$. On the other hand, for the thought-entity to be a real entity, we should requirre the set of variables to be connected; the is, if the entity is represented as a graph, there should be a path between any two variables. This implies that M cannot be less than N - 1. The relations also represent information, and in the absence of any contrary indication, we will assume that the description of a variable (what it is, its value, etc.) and the description of a relation (which two variables it connects, the functional relationship between the variables, etc.) each represent one unit of information, so that a thought-entity contains N + M units of information.

The value of M compared to N is a measure of the complexity of the thought-entity, but for this simple demonstration of the model we shall assume the lowest complexity possible, i.e. M = N - 1, or a linear structure.

We shall need one further model parameter - the *assurance level*, α. This is the probability of not having a single unit of information failure within a thought-entity, and is given by

$$\alpha = \left(e^{-\frac{t}{c}} \right)^{N+M}$$

where t is now the duration of the refresh cycle. That is, the rate at which the information units are refreshed equals $1/t$, which is proportional to the total power expended, and we find that this power, P, is given by

$$P = (N+M)\varepsilon \frac{c}{t}$$
$$= -\frac{\varepsilon}{\ln \alpha}(N+M)^2$$

Let us now see what happens if we subdivide, or *partition*, the thought-entity into *s* related sub-entities and postulate that operations on the thought-entity (or thought-processes involving the entity) can be converted into operations both on the subentities *and* on a entity consisting of the sub-entities as single variables and the relations between these sub-entities. The power needed to keep this reformatted thought-entity intact, P', depends on how the partitioning is done, and for simplicity, let us assume that it is uniform, i.e. that each of the *s* sub-entities contains the same number of variables, $n = N/s$. Then

$$P' = -\frac{\varepsilon}{\ln \alpha}\left[\frac{1}{s}(2N-s)^2 + (2s-1)^2\right]$$

and if we define the *reduction factor*, χ, by $\chi = P'/P$, and as an example choose N = 24, we obtain the following relationship between χ and *s*:

s	2	3	4	6	8	12
χ	0.483	0.317	0.241	0.188	0.192	0.288

3.3 A Linguistic Framework For the System Concept

In the first section of this tutorial, we saw that introducing a new concept or, what is effectively the same, introducing a known concept into a new area, as is the case with introducing the system concept into engineering, requires us to develop a linguistic framework for the concept in the context of the area of application. To this end we consider a few examples of how the word "system" is used within systems engineering.

a. *The early warning system cost in excess of $5 billion.* In this sentence, the noun phrase "the early warning system" has the meaning "the parts required in order to achieve the early warning function", but it is an imprecise meaning because what is required in order to achieve the early warning function is undefined. In a general manner we understand it to include all the parts whose primary function is to participate in achieving the early warning function, but where should we draw the *boundary*? Is the facility used to train the operators included in the cost? If this is a dedicted facility most likely yes, but if it is a shared facilty most likely not.

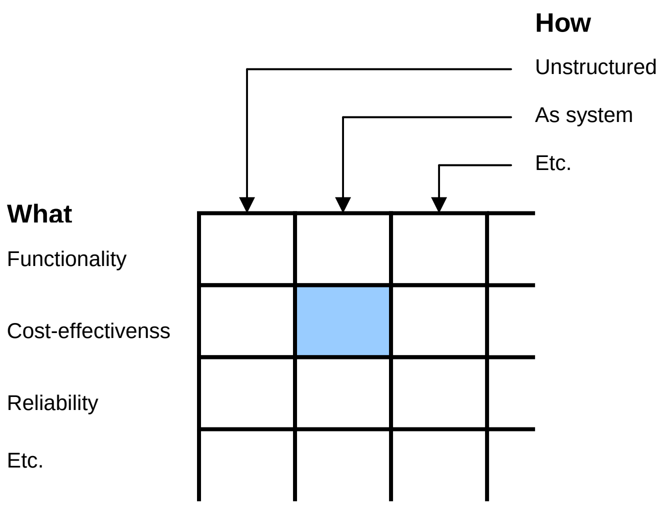
b. *The telephone system is good value for money.* When ordinary persons utter this sentence, they have no idea of what is included in "the telephone system"; its meaning is "whatever is required in order to allow me to use the telephone". If a telcommunications engineer utters the same sentence, she or he has a very much more detailed understanding of what is meant by "the telephone system", but even then it is not very precisely defined; two enigneers could easily have different opinions of what is included in "the telephone system".

These two cases demonstrate that when the phrase "the x system" is used in this manner, its meaning is one of inclusion of what is related to x. But the understanding of what is included may be highly context dependent, and if we want to convey a particular content, using such general statements is inadequate. The idea, sometimes floated in INCOSE discussion groups, that the absence of a defined boundary makes the system concept more "holistic", is clearly erroneous.

c. *A car is a system.* As we discussed earlier, in this sentence, the cupola "is" cannot be the "is" of existence, meaning that "car" and "system" reference the same object; we now know that "a car" and "a system" are two completely different types of entities. Strictly speaking, this sentence makes no sense, but in practice, the term "a system" is used here as an adjective (even though "system" is not a property), and its meaning is closely related to that of "complex". In view of Sec. 3.1, the meaning of "a system" as an adjective is "too complex for the mind to handle efficiently as a single entity".

While these uses of the word "system" is perfectly all right and useful in daily language, and even within engineering in a general sort of way, the meaning is much to imprecise for its intended use in the systems engineering context, i.e. as an entity on which we can operate and carry out some design activities.

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,



The set of all descriptions of an object

where the shaded matrix element represents one system. All the elements in the "as system" column represent systems, so there are many systems associated with any one object.

Thus, we now understand that the meaning of the phrase "the x system" is "the description of certain properties of the object x in the system mode", and the proper use of this phrase in systems engineering presupposes a definition or common understanding of what constitutes the object x, and what properties we are considering.

With this understanding, we also see that the truth value of the sentence "x is a system" is TRUE if x refers to a description in the system mode of an object, and FALSE otherwise. So the truth value of the sentence "A car is a system" is FALSE within the context of systems engineering, and this confronts us with one of the central issues in systems engineering today - a situation somewhat analogous to that of a Christian Scientist with appendicitis. On the one hand, most systems enigineers recognize that the daily language use and meaning of the word "system" is too general for it to be useful as the key element of an engineering discipline, and so many of the discussions within INCOSE working groups are really just misunderstandings about or different opinions about the meaning of "system". These discussions often end with a frustrated "it all depends on what you mean by a system", and there has even been suggestions that we should abandon the word "system" and coin a new word with a precisely defined meaning. On the other hand, the daily language, imprecisely defined meaning of "system" is so prevalent and useful, and the use of a sentence like "A car is a system" so ingrained in us, that it would be neither beneficial nor practically possible to abandon it.

The way out of this dilemma is neither to abandon the general language use nor to coin a new word (4); it is to accept the fact that many words have a context-dependent meaning. It does require us to be more disciplined in our use of language and, above all, it requires a conscious effort to think in terms of functionality, i.e. what a physical object does, rather than in terms of what it is. The word "car" immediately conjures up the image, in our minds eye, of the physical object, with four wheels etc.; what it should conjure up is the description of its functionality, i.e. the capability of transporting a small number of people and meeting certain performance criteria. This immediate connection between a word and the physical object it refers to is the greatest barrier to lateral thinking in design, and it is, of course, also at the root of our problems with the word "system". Instead of associating "system" with a description, we associate it directly with the physical object to which the description refers.

The rest of the tutorial is concerned with examining this issue in more detail, and to try to develop some general concepts and rules for how th manipulate them that can eliminate some of the current problems with the system concept and its application within systems engineering.

3.4 Notes to This Section

1. The distinction between objects and concepts, and the subdivision of concepts into first- and second-level concepts is 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.

2. The seminal work in this area is a paper by G.A. Miller, *The Magical Number Seven, Plus of 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.

3. A slightly expanded version of this model is contained in E.W. Aslaksen, "The Changing Nature of Engineering", McGraw Hill, 1996.

4. The meaning and use of the word "system" is a subject of recurring debate in INCOSE, and there have been suggestions (e.g. within the IEWG) that maybe we should create a

new word to convey the specific meaning relevant to systems engineering. The consensus was (thankfully) that this was not a good idea.

PART B - IMPLICATIONS FOR SYSTEMS ENGINEERING

4 THE PHYSICAL AND FUNCTIONAL DOMAINS

4.1 Some Initial Definitions

4.1.1 Engineered Objects

Aa discussed briefly in Sec. 2.4, engineering is about creating entities that serve a purpose; they provide a *service*. These entities range from very simple ones, such as a bolt, to very complex ones, such as a transport system, but they all have in common that in order to serve their purposes, they must be physical in nature. We shall call such entities, created by the process of engineering for a particular purpose, *engineered objects*, defined by

Definition 4.1: An *engineered object* is an entity consisting of one or more inseparable material parts, all connected together by definite physical relationships, and designed to provide a service.

Engineered objects obviously make up a part of what we in Sec.1 called the "thing world" and as such have *names*, which identify them as belonging to classes of objects, such as houses, cars, resistors, capacitors, etc.

4.1.2 Descriptions

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 wishes to reproduce (manufacture) the object. The latter consists of performance parameters, such as load capacity, reliabilty, 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. fulfill its inteded purpose). Both descriptions refer to to the same entity - the engineered object, and are concerned with physical aspects of the object - substance and behaviour.

At this point we need to make a quick detour into a subject that has seen much debate within INCOSE over the last years - requirements and their properties and classification. It is quite common to subdivide the requirements about an object's behaviour into two classes; functional requirements that specify what the object must be able to do, and performance requirements that specify how well it must do it. While there is nothing wrong about this subdivision, it is really no more significant or necessary than if we were to subdivide the physical requirements into two classes, the requirement on what parameters must be used to describe the object (such as weight and size, if they are important), and requirements on their values. For our purposes here (and perhaps for INCOSE in general?), let us agree that the physical characteristics of engineered objects are described by two classes of parameters - *substance parameters* and *performance parameters* - and their values, and that the term *functional parameter* is reserved for what we shall now discuss.

We now come to a central point in our development of the foundations of systems 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 provided, 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 fulfills 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*.

Definition 4.2 The *functionality* of an engineered object is its purpose, or the service it is intended to provide.

Definition 4.3 The functionality is described by a set of *functional parameters* and their *values*, which together make up the *functional requirements*.

Definition 4.4 A *functional element* is a set of functional requirements.

A functional element is a description, but it is very different in nature to the two types of descriptions we are used to. 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 parmeters as power rating, conversion efficiency, etc.

We also note that a functional parameter may not be a number, but a function of the environment in which the functionality is provided. For example, in the case of the power generating element, we might require the power rating, say Q, to be no less than the following function of the air pressure or installation elevation, *h*,

$$Q = Q_0 - Q_1 (h/2000)^2.$$

Thus, a functional element describes not only the service to be provided, but also the conditions under which it is to be provided. This example also illustrates how the level of detail of the description of functionality can be increased, resulting in the functional element containing a greater number of parameters.

The concept of functionality is not new, and is implicit in aspects of natural language, such as when we associate an object with a named class of objects. When we identify an engineered object by naming its class, we immediately provide a very substantial part of the description of what it is. For example, when we identify it as a car, we are saying that it has wheels, doors, an engine, a steering mechanism, seats, etc. The name also provides some part of the description of how it behaves, such as accelleration, braking, uses fuel, etc. It also says something about what it does, e.g. transport people. But this same service is provided by a bicycle, a bus, a train, and aeroplane, etc., so that what identifies an object as being a car is *mainly* its physical characteristics. However, if we take another example, such as identifying an engineered object as a stimulant, the name is almost completely related to the service provided by the object, as the physical realisation could be a solid, a liquid, or a vapour, and of a wide range of chemical compositions. So, we recognize that the nature of the name of an engineered object can lie anywhere between a description of the functionality only and a description of physical characteristics only.

4.1.3 The Domains

We now have two main types of descriptions - physical and functional. Let us define two corresponding sets, or what we shall in this case call *domains*.

Definition 4.5 The *physical domain* is the set of all physical descriptions of engineered objects.

Definition 4.6 The *functional domain* is the set of all functional descriptions associated with engineered objects; i.e. the set of all functional elements.

Considering first the physical domain, we could (in priciple and highly idealised) describe every engineered object by giving the type and location of every atom in the object, and in addition describe what each object does by means of transfer functions, flow diagrams, PI diagrams, etc. But that is not what we do; we describe engineered objects in terms of types and classes of construction elements or *components*, such as resistors, capacitors, bearings, shafts, nuts, bolts, beams, walls, footings, etc.. Within each type, it is generally sufficient to specify only a small number of parameters or *properties* that are a mixture of form and function descriptors, such as weight, size, colour, power rating, dynamic behaviour, etc., and if the type is standardized, the number can be very small, even just one (e.g. the size of a hexagonal metric nut). *It is only the use of standardized components that makes physical design (and construction) practical.* As a consequence, the physical domain has a *structure*; in one part are, for example, all the descriptions of fasteners, within that part are parts for descriptions of screws, rivets, nails, etc., and within these parts further subdivisions of description types, and so on. Another grouping might be descriptions of electronic components (capacitors, resistors, semiconductors, conductors, etc.), another one mechanical components (bearings, shafts, gear wheels, etc.), and so on. Of course, there will be groups of descriptions of much larger objects, such as cars, planes, and so on.

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 we saw in the case of the power generating element above). 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, and this will allow us to investigate the structure of the functional domain in Sec. 4.4. But first we have to be absolutely clear about the relationship between the two domains, and then look briefly at where we now stand with regard to the two central issues in this tutorial - design (in Sec. 4.2) and systems (in Sec. 4.3).

4.1.4 The Relationship Between the Physical and Functional Domains

From the examples in the previous section, we see that the relationship between a functinal 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 colletion 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 service provided by an engineered object.*

The relationship is illustrated by Fig. 4.1. In this figure, note that the relationship arrow points from the element to the class; we must have the element before we can define the class. This relationship 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. Until recently, the focus of engineering was on creating the means; even relatively simple objectives required major technology developments, and the complexity resided in the development of that new technology. That situation is changing, so that in many areas today we have more technology than we know what to do with, and the complexity has shifted from providing the means to understanding and defining exactly what the objectives are.

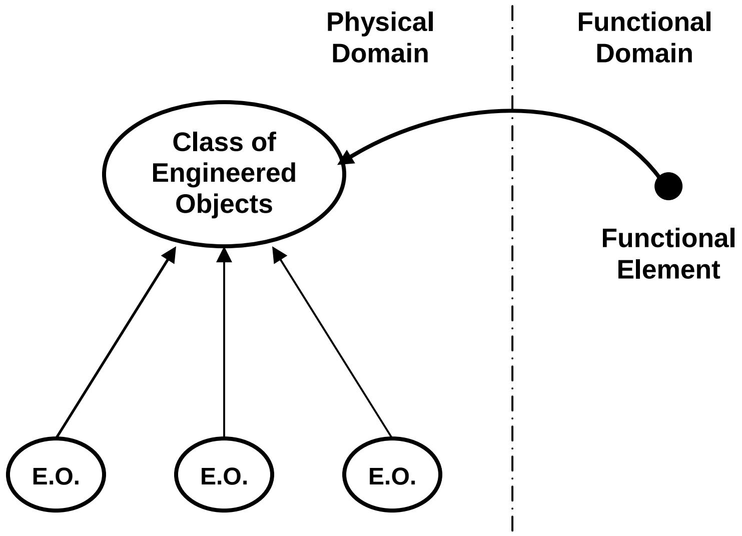


Figure 4.1 The relationship between functional elements and classes of engineered objects.

4.1.5 <u>The Parameter Space</u>

There is a further aspect to the relationship between the functional and physical domains. Consider a given functional element, then there corresponds to it a set of functional parameters (and their values). Consider further an engineered object that has been designed to meet the requirements of this particular functional element; it is described in terms of a set of performance parameters (and their values) and a set of substance parameters (and their values). The functional parameters and the performance parameters are of the same type in the sense that they describe *doing*, or *actions*, rather than the *being* described by the substance parameters, and in this sense they are members of the same collection of parameters, which we shall call the *parameter space*.

In order to be able to test the degree to which the engineered object meets the functional requirements, we shall demand that the set of performance parameters includes the functional parameters.

There is, of course, also the space of all substance parameters, but as that is of no interest in the current context, there is little likeliehood of confusing the two spaces; this is the reason we have chosen to call the space of functional and performance parameters "the parameter space" rather than inventing a special name, such as "the action parameter space".

4.2 The Design Process

In order to examine the implications of our new-found understanding for systems engineering, we have to briefly consider the basic activity within systems engineering - designing engineered objects using system descriptions of the entities arising in the design process. We shall consider the design process to start with a set of functional (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 service which fulfills the user requirements to a greater or lesser extent. It is the *intent* of the designer that the service produced by the system will fulfill the user requirements, and the degree to which this intent is achieved, i.e. the overlap of the service with the user requirements, is a measure of the success of the design process, as illustrated in Fig. 4.2.

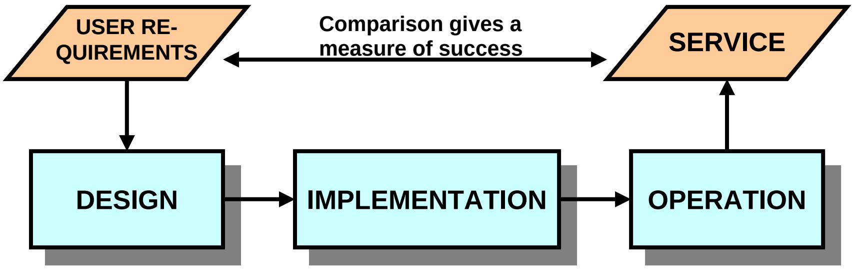


Figure 4.2 How design fits into the process of engineering.

Now, we know that user requirements consist of three types of requirements - functional requirements, performance requirements, and substance requirements. The latter just flow straight through the design process without requiring any design effort (except for allocation). The performance requirements already relate to a physical object and can therefore go more or less straight into a system specification, whereas in the case of the functional requirements, we first have to decide on a physical object to relate them to. (A statement like "The system shall ..." implies that there is already a system.) That is, we have to chose a basic system archtecture (technology, components, interactions). This is illustrated in Fig. 4.3.

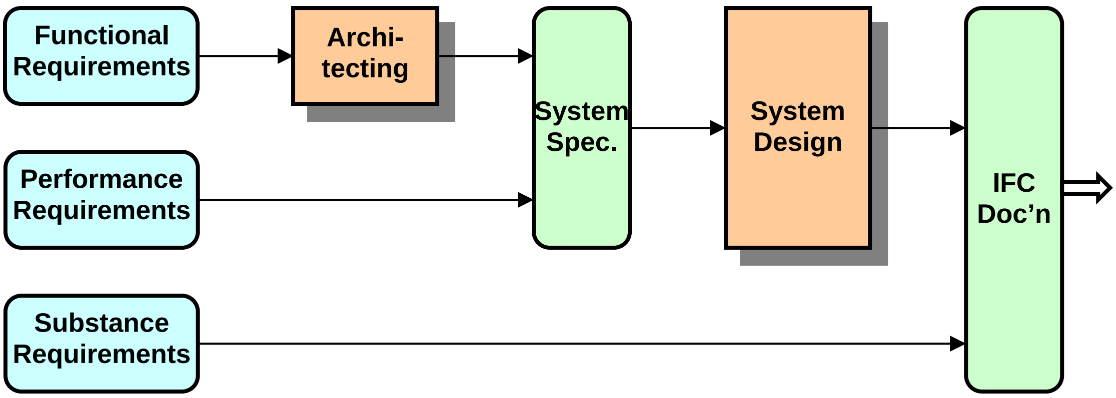


Figure 4.3 A simplified view of how the three types of user requirements fit into the design process. IFC means "Issued For Construction".

There are a large number of methodologies for carrying out the architecting and design of (complex) engineered objects (1). We shall not be concerned with any of these methodologies here, but it is probably fair to say that in all of them the most difficult step is the one called “Architecting” in Fig. 4.3, i.e. the conversion of functional requirements into requirements on a physical architecture, or *the transition from the functional into the physical domain*. We need to realise that the “classical” eingineering design methodology is a bottom-up methodology; one of sythesizing a solution from a set of known construction elements. That is still the predominant feature of an engineering education; from physics and chemistry we learn how the basic elements function. For electical engineers these are resistors, capacitors, inductors, transformers, semiconductors, etc.; for mechanical engineers they are shafts, bearings, gear wheels, etc.; and for civil engineers they are embankments, slopes, beams, trusses, etc. The synthesized solution is then tested, either by simulation or by building a prototype, and if the performance does not quite meet the requirements, the elements are adjusted to achieve the required performance. Many of these solutions are repeated so often that they themselves become construction elements; for example, capacitors and inductors become filters, semiconductors become counters and gates, and the most advanced example of this evolution of construction elements is probably the microprocessor.

However, as the requirements and the synthesized solution become increasingly complex, it becomes increasingly unlikely to get anywhere close to the right performance on the first try and, even worse, increasingly more difficult to decide how to change the elements in order to move the performance toward that required. In short, the process becomes increasingly inefficient.

If we look at some of the defining documentation on systems engineering (2), we see that almost all of the material relates to operations on physical entities, as exemplified e.g. by the *List of Requirements* in EIA-632. And indeed, that same standard defines a system as “an aggregation of end products and enabling products to achieve a given purpose”; so it is clearly focused entirely on the physical domain. But how do we know that we have considered all possible architectures? And how do we prove that a particular one is the optimal one? There are many aspects to these issues, such as previous experience and boundary conditions, but it is certainly true that the difficulty of making the transition from the functional domain into the physical domain increases with increasing complexity of the user requirements. And so we are led to ask ourselves if there would be some way in which we could partition the user requirements into interacting subsets, each of lesser complexity, and then carry out the conversion for each subset? In other words, would there be any benefit in introducing the system concept into the functional domain, and if so, what would this look like?

Another way to look at this is to first consider how some systems, made up of relatively simple elements, can have properties that are complex and, at least initially, unexpected. An small example of this is how a few capacitors and inductors can give rise to a bandpass filter. So, if we had initially required the properties of a bandpass filter, how would we have deduced that the elements we needed should have the properties of capacitors and inductors?

4.3 Systems in the Functional Domain

We could perhaps imagine that introducing the system concept into the functional domain would be trivial; we only have to use the system mode of description. That is not the case; the system mode of a functional description, while still conforming to our definition of a system, has some quite special features, which arise mainly because of the special nature of the interaction of the elements.

When we apply the system mode to the description of an engineered object, the description is structured into descriptions of elements and of the interactions between them. Firstly, these elements are in themselves engineered objects, and they are related to the original object in a form of a *taxonomy* (or, these days, often called ontology, although that meaning of the word "ontology" is somewhat different to the one we introduced in Sec. 1). For example, if we want to describe a radar as a system, the elements might be antenna assembly, transmitter, receiver, processor, and display. In the taxonomy of engineered objects, these elements are on the level of "equipment", which is lower that the level "system", but higher than the level "module" (e.g. power supply or amplifier). The existence of this taxonomy, and of many standardised components within it, is an immense aid in doing design.

Secondly, the interactions between the elements are of a physical nature, i.e. an interchange of something physical, such as a flow of energy, an interchange of information, a flow of matter, etc., and as a result of these dynamic events, the *behaviour* of the whole may display features not found in any of the elements. Conversely, the complex behaviour of the whole may be described by the behaviour of set of much simpler elements and their interactions.

In the case of functional elements, this is all quite different. Firstly, there is no established taxonomy of *doing*, as opposed to the above taxonomy of *being*. Or, a taxonomy of actions, as opposed to a taxonomy of things. For example, for a substance to be called a wine (3), it must have certain properties, such as colour, alcohol content, be made from grapes, etc., and depending on the values of these properties, we obtain a taxonomy of wines, such as red wines, rosé wines, and white wines. If we take an action, such as amplification, it also has properties, e.g. gain, harmonic distortion, efficiency, etc., but specifying values for these properties does not yield a taxonomy of amplification, and does not really make much sense at all, unless we know what the amplification refers to.

The issue here is that a functional element is of the form "the ability to perform an action in relation to something", and the partitioning of the action into subactions is dependent on the "something". A simple example may help to illustrate this. Let the functional element be "To prevent an enemy aircraft from reaching the shores of Australia." The action of preventing could be partitioned into the four subactions detect, identify, track, and destroy. However, if the functional element was "To prevent my cows from breaking out of the paddock", that subdivision would not make much sense. So, it is clear that, should we attempt to build such a taxonomy of actions, it would have to be specific to the application to a greater or lesser extent. (Or, in other words, it would have to be a set of application-specific ontologies.)

Secondly, there is no question of behaviour; for there to be behaviour, there has to be a thing that behaves. The functionality is the *desired result* of the behaviour of the yet to be engineered object. Functionality is the description of a capability, it is something that *is*; it is not something that happens. Functionality is very close to the meaning of a sentence (or a paragraph, or a story); just as the meaning of a written sentence is something that is in the sentence and is perceived by the reader, functionality is perceived by the users. Consequently, the interaction between functional elements is not the flow of anything physical; it is in the nature of a relationship, of "belonging to".

The little example of the air defence system is an illustration of this. The "interaction" between the four subactions, that which makes them into a system rather than just a collection of four actions, is that they relate to the same thing; in this case the enemy aircraft. Or, to pursue the analogy with a sentence, the situation is similar to that of the difference between different persons reading one word each of a sentence and one person reading all the words; only in the latter case do the words form a sentence that has a meaning (this is the system), and the interaction takes place in the brain of the person reading the words. The brain perceives them as belonging to the same sentence.

But if we could develop such a taxonomy of actions, why would we do it? What would be the benefit? Well, exactly the same as with standardised components; reuse of existing design and existing models. The sort of decomposition described above, which could be called *design in* *the functional domain*, is being carried out under the name functional decomposition and analysis, and is used predominantly in defence capability development. Modelling the functional elements is extremely time-consuming and therefore costly, and as a result this approach, which should be a cornerstone of systems engineering, is hardly ever applied to commercial projects.

4.4 Structure of the Functional Domain

4.4.1 Introduction by Means of Examples

A functional element has the following basic format: *A description of the ability to perform an action in relation to something*. A few examples are:

A description of the ability to
a) prevent enemy aircraft from reaching the shores of Australia
b) prevent my cows breaking out of the paddock
c) move containers
d) move persons

The first two have the same action, “prevent”, as do the last two, “move”, and we could consider classifiying FEs acording to the action involved. If that is sensible remains to be seen; on the one hand a) and b) would appear to be very different FEs, but on the other hand any FE concerned with preventing something from happening would be characterised by how often the something would happen if there was no prevention, in what fraction of events the prevention was successful, etc.

These examples also show that the nature of the “in relation to somehing” depends on the action. In the case of “prevent”, it is of the form “something from happening”, i.e. something from taking an action, such as “enemy aircraft reaching the shores of Australia” and “cows breaking out of the paddock”. In the case of “move”, the “in relation to something” is simply naming whatever is to be moved, but in this case the FE could be further developed by specifying “between points A and B”, or “between a set of nominated locations”, etc.

Also, in the above examples, the actions were simple, in the sense of being expressed by a single verb. In general there could be a set of actions, auch as “doing A, then B, and then C to something, while at the same time doing D to something else”. What makes this a single FE is, as remarked earlier, that the actions refer to the same entitiy and/or to the same period of time.

Finally, a FE will also, in general, have *parameters*, describing how well the action is to be carried out. Typical parameters would be how fast, how reliably, how cost-effectively, etc. the action is to be carried out. That is, we have the parameters themselves, such as speed, reliablilty, and cost-effectiveness, and then their required values (or ranges of values).

So, we see that FEs can be classified, and thereby the functional domain given a structure, according to a number of aspects, and that the “position” of a FE within the functional domain could be compared to the position of a point in a multi-dimensional space. Let us now try to make this somewhat more precise.

4.4.2 The Structure of Functional Elements

As noted above, a FE has the format “A description of the ability to perform an action in relation to something”; let us see how such an ability is expressed by considering a specific example, as shown in Fig. 4.4:

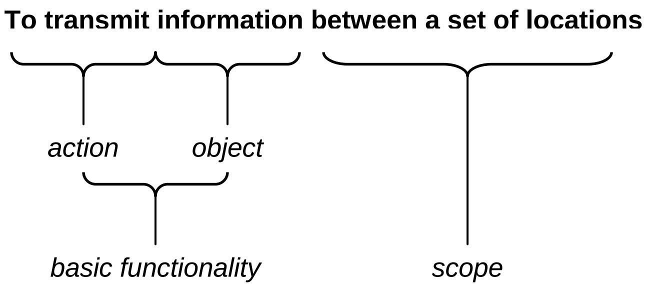


Figure 4.4 The components of a functional element.

From this example we deduce that a FE consists of three components, which contain distinctly different types of information:

(i) The *action*, i.e. the verb that describes the basic function of the element;
(ii) the *object*, which describes what is being acted on; and
(iii) the *scope*, which specifies the scope of the action.

The action and the object together describe the basic functionality (i.e. without qualification as to its scope), and this basic functionality defines a *class* of FEs.

These three components are not completely independent; there are relationships between them which have to be taken into account. The central component is the action, so let us first see how the other two components relate to the action. First of all, the object is restricted to those objects to which it makes sense to apply the action. For example, it would make no sense for the functionality to be “to transmit a street”, whereas it would make sense for it to be “to transmit a disease”. So, to the action “transmit” there corresponds a set of objects (which are, of course, always nouns) that can be “transmitted”.

Secondly, the scope of a particular action may vary over a range of qualifications, but they are restricted to qualifications that make sense for the particular aciton. For example, “between two locations” (dedicated) and “from one location to a set of locations” (broadcast) both make sense for the action “to transmit”, but make no sense for the action “to store”.

These relationships are illustrated in Fig.4.5 on the following page. However, within each of the sets shown in that figure, we can discern further structure. For example, within the set of objects associated with the action “to transmit”, we have major groupings, such as “information” and “disease”, within “information” we would perhaps subdivide into audio, text, video, and machine readable, “text” could perhaps be further subdivided into messages and documents, and so on. - Within the scope it is also possible to create structures; an obvious example related to the action “to transmit” is a grouping into fixed and mobile locations.

But what about the actions themselves? Can they be structured in any way? That would provide the top level (or levels) of any taxonomy of FEs, and in the next subsection we look at this very briefly.

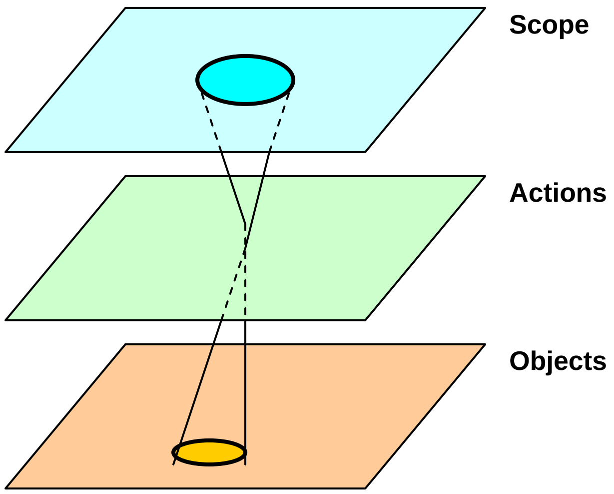


Figure 4.4 The components of a functional element.

4.4.3 A Taxonomy of Actions

A functional element will, in general, consist of a number of actions. For example, in the case of the air defence system we identified four actions - detect, identify, track, and destroy. Each one of these actions may again contain several more detailed actions, and so on, until we arrive at a set of basic actions that can not be reduced any further; we might therefore call them *irreducible* actions. They are to the functional domain what standardised components are to the physical domain, e.g. resistors, capacitors, and inductors to passive electrical circuits.

Developing a taxonomy of actions is obviously a major undertaking, but a start might be to note that irreducible actions seem to fall into three major classes:

1. Those that constitute a *translation in space* (e.g. transmit, transport, move, hoist, relocate, etc.)
2. Those that constitute a *translation in time* (e.g. store, retrieve, delay, etc.)
3. Those that constitute a *transformation* or change (e.g. process, combine, reduce, extract, correlate, deduce, infer, etc.)

Within each of these classes it would be possible to find further subdivisions, but they would have to have a practical value, and that can only be established as the development and use of FEs become more common.

4.4.4 Properties and Parameters

Associated with a FE are two two distinct types of variables. On the one hand, we have the variables that determine the nature of the FE. In the case of our example, this would include variables such as the number of locations in the set, the nature of the locations (e.g. fixed or mobile), etc., as already discussed. This type of variable, which generates subclasses and instances of FEs, may be called a *property* of the FE. Properties characterise *what* the FE does.

On the other hand, there are variables that characterise *how well* the FE does it. In the case of the example, these would include error rate, capacity, and delay (or latency). This type of variable may be called a *functional parameter*.

The functional parameters that are significant for a particular class of FEs will emerge as the result of defining a useful taxonomy of FEs, a task that will require further study and, above all, a concensus among systems engineering practitioners. However. in the next section we shall see that there are functional parameters that apply to all FEs, so-called *universal parameters*, and as a result we can extend our picture of the structure of a FE from the three components shown in Fig. 4.4 to include the relationship between components and variables, as shown in Fig. 4.5.

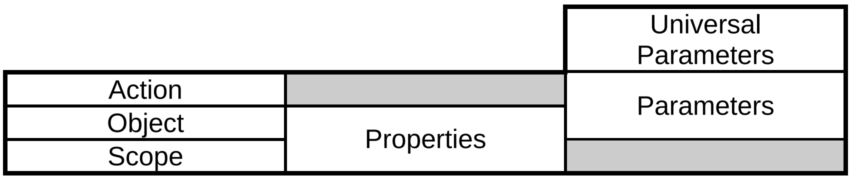


Figure 4.5 Illustrating the relationships between the components and the three types of variables associated with a FE.

4.5 Notes to This Section

1. It is not possible to list all the books and articles that have developed and described the process of design within the systems engineering paradigm. A standard refererence is the book by B. Blanchard and W. Fabrycky, *Systems Engineering and Analysis*, Prentice-Hall, Upper Saddle River, NJ (1998), now in its third edition; but a few of the ones focusing more specifically on a design methodology are:

   a. Wymore, A.W., *Systems Engineering Methodology for Interdisciplinary Teams,* John Wiley (1976).

   b. - , *A Mathematical Theory of Systems Engineering: The Elements,* (1967)

   c. - , *Model-Based Systems Engineering*, CRC Press, Boca Raton, FL (1003)

   d. Chapman, W.L., A.T. Bahill, and A.W. Wymore, *Engineering Modeling and Design*, CRC Press, Boca Raton, FL (1992)

   e. Buede, D., *The Enginering Design of Systems*, Wiley, New York, (2000)

   f. Warfield, J., *Science of Generic Design*,

2. There are anumber of Handbooks and standards describing and defining what is understood by systems engineering. Perhaps the first one was MIL-STD-499, first issued in 1974 as Rev.A under the title “Engineering Management” and then later, in 1992, as Rev.B (draft) with the title “Systems Engineering”. The draft was abandoned in favour of the emerging industry standards, including the following:

   a. IEEE 1220, *Systems Engineering*, Institution of Electrical and Electronics Engineers,

b. EIA-632, *Processes for Engineerin a System*, Electronic Industries Alliance, Arlington, VA, USA, 1999.

c. ECSS-E-10, *Systems Engineering*, consisting of ECSS-E-10A, *System Engineering*, and ECSS-E-10-01, *Systems Engineering Process*, ESA Publications Division, The Netherlands, 1996.

Supporting guides/handbooks include:

d. *Guide to Systems Engineering*, DSMC, Fort Belvoir, MD, USA, 199

e. *Systems Engineering Handbook*, Version 2.0, INCOSE, 2000

3. Wine as an example of how to develop an ontology (in the sense of taxonomy) is used in the very readable paper *Ontology Development 101: A Guide to Creating Your First Ontology*, by N.F. Noy and D.L. McGuinness, Stanford University CA, USA, available from www.ksl.stanford.edu/.

5 SOME PARAMETERS OF FUNCTIONAL ELEMENTS (1)

5.1 Applicability of Functional Parameters

The parameters that describe the requirements to be satisfied by a FE will, of course, depend to a large extent on the specific FE, and the more detailed the FE is, the more parameters it will have that are unique to that FE. However, because FEs can be ordered into a taxonomy (to some extent, at least), we expect the applicability of a parameters to vary from one parameter to the next, and may extend from a class to a subclass to perhaps a single FE. From the point of view of our examination of the foundations of systems engineering, an obvious question must be: Are there any parameters that apply to all FEs?

The answer would appear to be a definitive "yes", and in the following subsections we define and discuss a few such universal parameters. Universal parameters have at least two important functions: Firstly, they allow comparison between different policy options and investment programs. Secondly, in a top-down design methodology, they provide a simple means of checking traceability; because they appear in every FE at every level of the top-down partitioning of a single (very large and complex) FE into a system of smaller FEs, traceability becomes simply a matter of allocation.

5.2 Return on Investment

Before an engineered object can start to provide its intended service, it must be created (designed, developed, manufactured, etc.), and that creation must represent a *cost* of some sort. And because this cost is incurred before the object is created, it represents an *investment.* Consequently, it must be true that *every engineered object requires an investment.*

However, would anyone make an investment without any hope or expectation of a *return* of some sort? That return does not have to be directly in monetary terms, but might in the first instance be measured in terms of well-being, self-realisation, strategic advantage, etc. And the greater the return for a given investment, the better. Or conversely, if the desired return can be obtained with a smaller investment, that must also be a better solution. It therefore follows that *Return On Investment (ROI) is a parameter common to all engineered objects, and maximising its value must be an objective of every engineering project.*

In order for ROI to be a useful functional parameter, it will have to be defined in such a manner that it can be measured. Now, one might take the position that ROI is a matter for the business community and should not be of concern to engineers, and that is indeed the position taken by many, if not the majority, of engineers. Unfortunately, that position has seen engineers, as a profession, relegated from leaders of industry and society to "backroom boys"; technicians turning the crank in solving detailed, technical problems. It is also a position that is completely at odds with one of the main aims of systems engineering - taking a holistic view that encompasses the needs of all stakeholders. If we recognize ROI as a fundamental property of every engineered object, then it should receive a commensurate amount of attention from the systems engineering community.

There would be no in-principle difficulty with providing an operational definition of ROI; the dificulty is in getting agreement within the systems engineering community. Here is an example of such a definition:

a. The object life cycle shall consists of three phases:
   - (i) The *creation phase,* with duration TC;
   - (ii) the *operation phase*, with duration TO; and
   - (iii) the *decommissioning phase*, with duration TD

b. All costs and revenues shall be referenced to the point in time which is the end of the creation phase and the beginning of the operation phase, called the *reference point.* A vcost or revenue discounted to the reference point shall be called *Present Value (PV)* of that cost or revenue.

c. All costs and revenues shall be in reference point dollars (i.e. no inflation or CPI change).

d. The *discount rate, p,* shall be equal to the cost of money (i.e. the debt interest rate).

e. The PV of all costs accrued during the creation phase, CC, shall be covered by the sum of Equity (EQ) and Debt (DD).

f. The Debt would normally be repayed during the operating phase, and the repayments are treated as part of the operating cost. In any case, the debt shall at any time be no greater than the assured realisable value of the assets at that time.

g. Let CO be the PV of all costs accrued in the operating phase, CD the PV of all decommissioning costs, RO the PV of all operating revenue, and RD the PV of all (if any) revenue realised during the decommissioning phase.

h. The ROI shall be defined by

$$ROI = (RO+RD-CC-CO-CD)/(TO*EQ) .$$

But, you will ask, how can we talk of cost and revenue of functional elements that have no physical substance? The answer to the revenue issue is given in the next subsection; regarding the cost, it is important to remember that in a FE we are dealing with *intent* for a class of engineered objects, not actual values of existing plant. The main purpose of introducing a measure such as ROI is to set *target values* for the costs, which can then be allocated as the design process prgresses.

5.3 The Value Function

The revenue resulting from the operation of the engineered object, i.e. from providing the service to the users, will depend on the *quantity* and on the *quality* of the service. In general, the revenue will be the product of the quantity of the service provided, multiplied by the *value* per unit of the service. The value is a function of those functional parameters that make up the quality of the service, and this *Value Function* is a fundamental functional parameter of any engineered object. Without it there can be no proper optimisation of the design, and the concept of value provides the link between the engineer and the user.

Of course, introducing the concept of value immediately begs the question: Value to whom? The answer to that should be intrinsic to the definition of the service; a service is always provided *to* somebody, to a *user group*.

If we realise that the value function is a fundamental parameter of systems engineering, why are we not paying more attention to it? It’s all too difficult and complex? But is it not a central claim of systems engineering to be a methodology for handling that sort of complexity? Taking a holistic approach, and reconciling the differing requirements of a diverse group of stakeholders?

These questions, and the answers to them, lie at core of what I believe is one of the most important issues in systems engineering today - our handling of the interface between engineering and the users of our engineered products. Despite our professed recognition of the importance of the users and of the fact that the complexity of many of our systems arises mainly from the human element, our methodologies and our current concerns are all about the hardware and software, and about the processes for creating these. Systems engineering promised to be the paradigm that would allow engineers to take a leading position in guiding society in the application of technolgy, in being able to grasp and handle the full complexity of diverse stakeholder requirements, including financial, economic, and environmental aspects. That promise, it would appear, has been all but forgotten.

The simplest relationship between functional parameters and value is given by an expression of the form

$$W = W_0 \prod_1^m \left[1 + c_i \left(\frac{x_i}{x_i^0} - 1 \right) \right] .$$

Here x_i^0 is the nominal value of the functional parameter x_i, W_0 is the nominal value of the service, and c_i is a measure of the importance the user group assigns to the i-th functional parameter. The reason why this expression has universal validity is to be found in the S-shape of the function expressing the dependence of value on any one single parameter. This function, say w(*x*), is often a highly non-linear function; for a number of reasons there is a relatively narrow range of *x* in which w(*x*) increases rapidly with increasing *x*. Below this range, the quality of the service is so poor that it is basically useless. Above this range, an increase in *x* brings hardly any further increase in the value; it is a region of saturation or "overkill", as shown in Fig. 5.1. In the range of interest, a linear approximation is always a reasonable one.

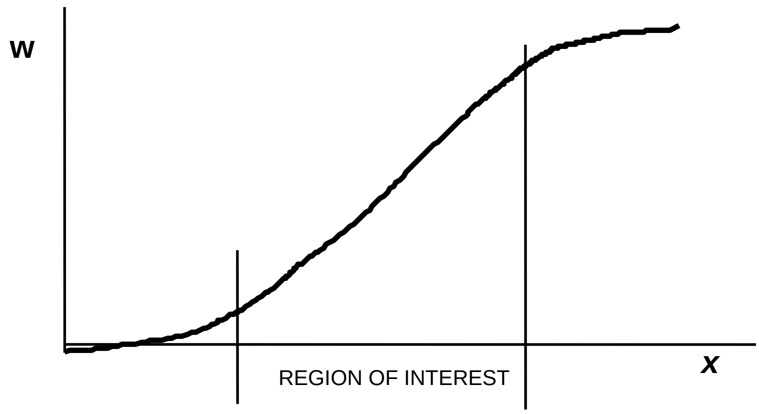


Fig.2.3 Typical form of the value function w(*x*).

5.4 Failure, Reliability, and Availability

Reliability and availablity have always been important concepts in systems engineering, and it is easy to show that they are as universaly applicable in the functional domain as they are in the physical domain, but with two differences. Firstly, when we talk of reliability and availablity in the functional domain, it does not apply directly to the functional elements themselves, but to the action (service) they describe. Secondly, a physical failure mechanism is obviously not a concept that applies in the functional domain.

The central concept is that of failure. We shall say that a functional element has failed if the value of its service falls below a certain level. The full meaning of this “shorthand” is “Any realisation of the functional element as an engineered object will have failed if its service is less than a certain value".

The definition of reliability then follows in the normal fashion, but again, when a functinal element contains a requirement of the form "The reliability R(T) shall be no less than 0.95", the full meaning of this "shorthand" is "Any realisation of the functional element as an engineered object shall have a probability of remaining in its operating (i.e. non-failed) state under specified conditions for a period T of no less than 0.95".

A similar interpretation holds for availability. A requirement that the availability of the service shall be no less than 0.99 means "Any realisation of the functinoal element as an engineered object shall have a probability of being in its operating state of no less than 0.99".

At this point it might be appropriate to note what has hopefully been clear all along - functional elements cannot be stochastic in nature. That is, the values of the functional parameters are ordinary numbers, not distributions. It is only the actual performance of an engineered object that is of a stochastic nature, with the parameters describing the performance being distributions.

5.5 Notes to this Section

1. Some of the ideas presented in this section were previously discussed in E.W. Aslaksen, *A Leadership Role for NCOSE*, Proc. 4th Int'l Symposium, National Council of Systems Engineering, San Jose, August 1994, and in E.W. Aslaksen, *The Changing Nature of Engineering*, McGraw-Hill, 1996.

6 SUMMARY

In this critical examination of the foundations of systems engineering, our first results relate to the system concept itself:

A. The system concept is a manifestation of a basic feature of the mind - the ability to perceive a set of interascting elements as forming a whole, and vice versa. It is what Kant termed a category and what, within semantics, Frege called a second-level concept.

B. A system is a mode of description.

C. The system concept is the mind's way of handling complexity. The complex behaviour of the whole can be viewed as the (emergent) result of the less complex behaviour of the interacting elements.

We then saw what differentiates systems engineering from the rest of engineering:

D. Systems engineering is the simultaneous application of the system mode of description to the engineered object and to the work of creating it.

Finally, looking more closely at how we do systems engineering, we obtained some further results:

E. There are three different descriptions related to an engineered object:

   (i) A description of what the object is; its *physical characteristics*.

   (ii) A description of what the object does; its *behaviour* or *performance.*

   (iii) A description of the service the object is intended to provide; its *functionality*.

   The first two make up what we call the *physical domain*, the last type of description constitutes the *functional domain.*

F. The system mode can be applied to all three of these, but the natures of the systems (i.e. the elements and the interactions between them) differ considerably.

G. One of the most difficult, if not the most difficult, step(s) in the process of engineering is the transition form the functional into the physical domain, especially as the complexity of the requirements increases.. Our aim, as systems engineers, would be to reduce the complexity in the functional domain before attempting to carry out the transition. What we need is a methodology for partitioning a functional element into a system of elements.

H. The foundation of such a methodology would be a population of standard functional elements and their parameters, in complete analogy to what we already have in the physical domain.