The System Concept and Its Application to Engineering

Erik W. Aslaksen

The System Concept and Its Application to Engineering

Springer

Author
Erik W. Aslaksen
Gumbooya Pty Ltd.
Allambie Heights, NSW
Australia

ISBN 978-3-642-32168-9 e-ISBN 978-3-642-32169-6
DOI 10.1007/978-3-642-32169-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012943052

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Introduction

The Subject Matter

The term *systems engineering* has been present within the engineering profession for more than fifty years, it has been the subject of thousands of books and papers up to this day, there is an international organization, the International Council on Systems Engineering (INCOSE), dedicated to develop and promote it, and still there is no agreement, neither within the community of those who call themselves systems engineers nor within the engineering profession in general, about what systems engineering really is. First of all, is it a distinct discipline within engineering, or is it a technique or approach, on a par with Reliability-Based Maintenance Analysis or Management By Objectives, that is useful in performing certain engineering tasks? There are certainly many who see systems engineering as simply an extension of well-established engineering practice into larger projects; an extension aimed at ensuring the successful outcome of such projects in the face of their increased complexity. As such, it belongs to engineering management, which is an integral part of all engineering disciplines, but not a discipline in itself. Under this perspective, systems engineering is similar to and closely related to Quality Assurance, and indeed, the format of the main systems engineering standard, ISO 15288, has much in common with that of ISO 9001.

Second, if it is a distinct discipline within engineering, what is its nature? It cannot be directly compared to such disciplines as mechanical or electrical engineering, as it does not have a foundation in Natural Science. Perhaps it is more akin to industrial engineering? Can systems engineering be taught as a separate undergraduate course, or only at graduate level to engineers with some practical experience? This question, formulated as "What is the intellectual content of systems engineering?" is obviously of central importance to the academic engineering community, and has been the subject of extensive discussions over the last several years. Is there a "core" of knowledge on which a systems engineering curriculum can rest?

Third, what should be included in the systems engineering Body of Knowledge (BoK)? What material can be identified as belonging distinctly to systems engineering, rather than being material from other fields, such as operations research, that is used by systems engineering? The uncertainty about this has made systems engineering vulnerable to exploitation as a marketing platform. Of the numerous methods, processes, frameworks, and models that have been put forward, many would appear to be a reformulation of existing knowledge with the aim of creating a product with which the author can be identified and which can be exploited for anything from academic advancement to commercial gain. This situation, while giving the impression of a field in vigorous development, is, in the opinion of a number of long-time practitioners, a distraction from and detrimental to the development of systems engineering as a well-defined and solidly founded activity within the engineering profession.

The subject matter of this book, which has been on my mind ever since I started teaching an undergraduate subject in systems engineering at the University of Technology, Sydney in 1983, is therefore not just systems engineering itself; it is concerned with the issue of why systems engineering is not more widely embraced by industry and the engineering profession. Even within the company I have been working in for almost twenty years, and which is one of the leading engineering companies in Australia, I have had very little success in establishing systems engineering as a worthwhile activity. The frustration is that most, if not all, of the material and insight required in order to achieve that goal is available as a result of the excellent work already done by a large number of authors; and there is a feeling that it just needs to be viewed from a unifying perspective in order to form a coherent whole. It is like having all the pieces of a puzzle; we know that they do fit together to form a picture, the question is only how. Attempting to create such a unifying perspective is the main subject matter of this book.

There is a further aspect to this issue. While there is this extensive literature describing the multitude of techniques and processes that fall under the general umbrella of systems engineering, there is relatively little information of what to apply and how to apply it to any given engineering project outside the defence and aerospace industry. Perhaps we have been going about this in the wrong way in depicting systems engineering as a (massive) set of processes? Should not systems engineering be seen primarily as a mindset, as a particular way of viewing a complex project, with the choice of processes and techniques following from the understanding gained from this view? This question is explored in the last part of the book.

Background

As mentioned, a great deal of work has been done in developing systems engineering, and selected parts of that work form the background on which this book is based. It is not possible to include more than a fraction of that work, and attempting to do so would only distract from the purpose at hand; this book is not intended as a literature review or a bibliography. The purpose of defining this background is to allow the reader to make up his or her own mind as to its adequacy as a basis for the perspective developed; for anyone interested in studying the history of systems engineering, the book by Hall [1] is a good starting point.

An early description of systems engineering, based largely on experience within the telecommunications industry, was provided by Hall [2]; other early books were those of Chestnut [3] and Johnson *et al* [4]. A much more formal approach, based in part on set theory, was provided by Wymore [5].

A remarkably perceptive overview and assessment of systems engineering in its infancy was provided by Bode [6]. In the Introduction, he states: “There seems little doubt that this approach may help us on many fronts. On the other hand, the “systems approach” is sometimes urged, somewhat uncritically, as a sovereign nostrum for all ills. Thus, one needs to pay particular attention to aspects of the subject that may make it appropriate or rewarding in some situations but not in others.” And in the section headed “Some Misgivings” he adds: “On the other hand, it takes only a little reflection to convince oneself also that techniques and methods of thought developed in the aerospace world may not always be immediately appropriate in other contexts. There are simply too many differences between that world and others.” And he goes on to say: “Many of the tenets of systems engineering seem pretty obvious and unlikely to contribute anything that would not have been discovered anyway.” And at the start of the section headed “Varying Conceptions of Systems Engineering” he states: “It seems natural to begin the discussion with an immediate formal definition of systems engineering. However, systems engineering is an amorphous, slippery subject that does not lend itself well to such formal, didactic treatment. One does much better with a broader, more loose-jointed approach.”

This early part of the development of systems engineering was strongly influenced by the requirements of the Cold War, whose main characteristic was a high-intensity arms race. Weapons and space systems of unprecedented complexity had to be developed in record time, and the organisations required to meet this challenge were correspondingly complex, consisting of numerous contractors. Due to the huge amounts of funds available within the defence budgets to develop systems engineering as a key enabler of technological superiority, systems engineering became almost completely defence oriented. The objectives, the terminology, and the metrics became totally aligned with those of the defence industry, and have remained largely so until today. Furthermore, due to the time pressure of the arms race, the focus shifted from using systems engineering as a design methodology to using it as a management methodology. Such topics as change control, configuration management, technical performance measurement, integrated development teams, and lean systems engineering, just to mention a few, have largely focused on management and control rather than on design, and indeed, the first systems engineering standard, MIL-STD-499A, published in 1974, was entitled *Engineering Management*. An example of this anchoring in the defence industry was the influential textbook by Blanchard and Fabrycky [7].

Complementary to this development of systems engineering, from its roots in telecommunications through the needs of the defence industry, there was a strong movement to develop a more general systems methodology, or systems science, based on General Systems Theory (GST). 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” [8]. This led to the creation of the International Society for Systems Science (ISSS) and later the International Federation for Systems Research (IFSR). However, numerous other roots of such a systems science can be identified, and the book *An Introduction to Systems*

Science, by Warfield [9], one of the central figures in systems science, gives a good overview of many of these antecedents. A couple of well-known references are the books by Laszlo [10], Bowler [11], and Boulding [12]. A more recent book, which gives a good overview of the current state of GST, is the one by Skyttner [13].

While the early development of systems engineering was focused on systems consisting of hardware and software, the importance of the human element in systems and of systems consisting of people, i.e. organisations or enterprises, soon became obvious, and much of the systems engineering literature in the last 15 years has been concerned with this aspect in one way or another. One result of this is that the distinction between systems engineering and systems science is becoming more blurred, and one book which emphasizes this is *Advanced Systems Thinking, Engineering, and Management*, by Hitchins [14].

Structure of the Book

In this book, the task of creating a unifying perspective is approached in a very straight forward manner: In order to understand the meaning and content of the term "systems engineering" it is reasonable to first have a clear understanding of the two components of that term, i.e. "system" and "engineering", and the first two parts of the book are dedicated to this. One could be excused for thinking that this would be a trivial task, essentially looking the word up in a dictionary, but some everyday observations will quickly tell us that this is not so. As a small example, one of Australia's major technology consultancies once stated that the secret of success in delivering large technology-based projects was to get away from "the engineering mindset". Engineering has a long and proud tradition of providing value to society through the development and application of technology, and it is important to see systems engineering as a continuation and further development of this tradition. And the word "system" is used so commonly, with so many meanings and in so many different contexts, that it is not obvious how it should be understood and applied in the context of engineering.

Building on this understanding, the third and last part of the book looks at the application of the system concept to engineering. If the purpose of this application is to allow us to handle complexity more cost-effectively, we need to investigate where and why complexity arises within engineering, and then gain an understanding of why a benefit arises and how it can be maximised. Central to developing this understanding is the concept of a Return on Investment as the point of departure for the top-down development of the functionality of a complex project as a system of less complex, but interacting elements, and as the basis for optimising the design of complex systems.

Intended Readership

This book is intended primarily for engineers. While some of the material will be of interest to practitioners in other professions, such as philosophy, given the current interest in Philosophy and Engineering [15], and business management (from the book's view of engineering projects as investment opportunities), the focus is on engineers with a desire to gain a deeper understanding of the system concept in the context of their profession. They can be practising engineers or students, although the latter would generally have to be graduate students, given both the level of abstraction in much of the material and its appeal to the reader's practical experience. This is clearly not a textbook, in the sense of developing a proficiency in applying the information provided through examples and exercises; it relies on the reader's professional background and maturity.

With regard to its use as supplementary reading in a graduate systems engineering course, it is useful to distinguish between System Centric and Domain Centric courses, as was developed in a paper by W.J. Fabrycky [16]. Under the perspective of that distinction, this book is more suitable within a System Centric program, as it deals mainly with general principles and novel approaches rather than with well established applications of what might be called "classical" systems engineering to specific disciplines or industry sectors.

Acknowledgements

It is a pleasure to acknowledge the diligent review and improvement of the text by Susanne and Carlmax Sturzenegger, as well as their contribution of the examples in sec. A4.1. Wolt Fabrycky contributed a number of significant comments and observations, and his sharing of some of his deep understanding of engineering in general and systems engineering in particular with me is greatly appreciated. And, as always, the support provided by my wife, Elfi, was an essential factor in completing this work.

Allambie Heights, June 2012 Erik W. Aslaksen

References

1. Hall III, A.D.: Metasystems Methodology. Pergamon Press, Oxford (1989)
2. Hall III, A.D.: A Methodology of Systems Engineering. Van Nostrand, Princeton (1962)
3. Chestnut, H.: System Engineering Methods. Prentice Hall, Englewood Cliffs (1969)
4. Johnson, R.A., Kast, F.E., Rosenzweig, J.E.: The Theory and Management of Systems. McGraw-Hill (1963)
5. Wymore, A.W.: A Mathematical Theory of Systems Engineering – the Elements. Wiley, New York (1967)
6. Bode, H.W.: The Systems Approach, in Applied Science – Technological Progress, report to the Committee on Science and Astronautics, US House of Representatives (1967)
7. Blanchard, B.S., Fabrycky, W.J.: Systems Engineering and Analysis. Prentice Hall, Englewood Cliffs (1981); This book is now up to its fifth edition
8. von Bertalanffy, L.: General Systems Theory. George Brazillier (1941)

9. Warfield, J.N.: An Introduction to Systems Science. World Scientific Publishing Co. Pte Ltd, Singapore (2006)
10. Laszlo, E. (ed.): The relevance of General Systems Theory. George Brazillier (1972)
11. Bowler, D.T.: General systems thinking: its scope and applicability. Elsevier, Amsterdam (1981)
12. Boulding, K.E.: The World as a Total System. Sage Publications, Beverly Hills (1985)
13. Skyttner, L.: General Systems Theory - An Introduction. Macmillan Press, Basingstoke (1996)
14. Hitchins, D.K.: Advanced Systems Thinking, Engineering, and Management. Artech House, Inc., Norwood (2003)
15. van de Poel, I., Goldberg, D.E. (eds.): Philosophy and Engineering. Springer (2010)
16. Fabrycky, W.J.: Systems Engineering: Its Emerging Academic and Professional Attributes. In: Proceedings, ASEE Annual Conference and Exposition, Louisville, KY, June 21 (2010)

Contents

PART A
The System Concept

A1 Everyday Use and Meaning

A1.1 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, the word "system" 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 "system" came into use during the seventeenth century, when we find it used with a number of somewhat different, but closely related, meanings. In the early part of the century there is still the meaning of the universe, as e.g. in "In this Round Systeme All", but soon we find it applied to signify an ordered collection, as e.g. in "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, perhaps, 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 General Systems Theory, as discussed briefly in the Introduction. 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. As a result of these developments, coupled 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.

A1.2 Current 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 [2] contains the following entries: 1. complex whole, organization; 2. method; 3. classification. The Australian Pocket Oxford Dictionary [3] 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.

Then there is the area of general systems theory, as mentioned in the Introduction, Within this area there are as many definitions of systems as there are authors on subjects involving systems; 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.

For reasons that will become clear in Parts B and C, we shall use the following, very broad, definition [4]:

*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.*

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,* *i.e. with its own properties that can be defined without reference to the parts.* Let us look at a number of everyday sentences involving the word "system" and see if this holds true.

a) In the tabloid press, we might encounter the sentence "*The health system is in a mess*", and without considering the truth value of this sentence, what is implied and what is understood by the use of the word "system" here? 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 imply 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) Consider the sentence "*Every book can be located within the Dewey decimal 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. books. The system has no output or properties, and we do not use the word "system" to imply that the Dewey decimal system has any meaning unrelated to books. 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 illustrate in the last subsection of this chapter. 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 the systems engineering community about the meaning and properties of systems, and we shall look further at this problem in the next section.

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 be instructed in its meaning and use. This situation is somewhat analogous to the use of a word like “society”; it is very useful when we want to signify something like “an association of humans” without going into specifics. Once we narrow the context to a specific society, such as The Chemical Society or a particular tribal society, we need to be taught what the meaning is and its proper application. Never the less, if we restrict ourselves to the first group above, we shall argue, in Chapter A3, that there is indeed something common to all applications of the system concept; only it is of quite a different nature to what we have been looking for in the above examples.

A1.3 “Systematic” and Ordering

It is sometimes stated that systems engineering is a “systematic” approach to engineering; that is, performing engineering as a process consisting of a number of steps carried out in a fixed order. This reveals a poor understanding of engineering and seems to imply that, prior to the introduction of systems engineering, engineering was a haphazard process. As we shall see in Part B, nothing could be further from the truth; engineering has a proud history of developing very satisfactory solutions to the needs of society in an ordered and efficient manner. The process of engineering, as it developed following the emergence of engineering as a profession at the end of the eighteenth century, is a structured and systematic process. For example, there is a subdivision of the design into phases, such as concept design, preliminary design, and detailed design, with flow-down of requirements from one to the next. And the fact that design precedes construction is true whether we apply systems engineering or not.

The issue here is that such a sequence of work packages or activities is a system only in a very restricted sense. Yes, there is a set of elements and, yes, there are interactions between them, but these interactions mark transitions in time; transitions between increasingly detailed views of the same object. For example, the design stages do not exist at the same time and do not form a whole due to their interactions; they are the whole. It is no different to the days of the week following each other sequentially and constituting the whole week. The identification and naming of stages within the process of engineering emerged as a means of controlling what would otherwise be a continuous, single process with the result emerging as a surprise at the end, and it falls within what is most correctly called engineering management.

The wide-ranging meaning of the word "system" and its use in everyday speech remains a problem for systems engineering in that techniques, processes, and approaches are presented as something new under the banner of systems engineering simply because they display a sequential nature, when they are in fact anchored in engineering management. This leads, on the one hand, to systems engineering sometimes being dismissed as "this is what we already do" or "just common sense"; on the other hand, to a lack of identification of, and focus on, the real nature and benefits of systems engineering. At this point in our development it is probably enough just to be aware of the problem and be sensitive to its implications as we progress; a more in-depth discussion is contained in Chapter C2.

A1.4 Previous Definitions in the Context of Engineering

Before leaving the topic of "everyday use and meaning", it is appropriate to briefly review some of the extensive literature that deals with the use and meaning of the word "system" in the context of engineering, even though we shall only examine the application of the system concept to engineering in Part C. First of all, there are the various standards:

a. ISO/IEC 15288:2002(E): A combination of interacting elements organized to achieve one or more stated purposes. *Note 1:* A system may be considered as a product or as the service it provides. *Note 2:* In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively, the word system may be substituted simply by a context dependent synonym, e.g. aircraft, though this may then obscure a system principles perspective.
b. EIA-632: An aggregation of end products and enabling products to achieve a given purpose.
c. ECSS-P-001A, Rev.1: Set of interdependent elements constituted to achieve a given objective by performing a specified function (IEC 50:1992). *Note:* The system is considered to be separated from the environment and other external systems by an imaginary surface which cuts the links between them and the considered system. Through these links, the system is affected by the environment, is acted upon by external systems, or acts itself on the environment or the external systems.

d. MIL-STD-499B (issued in draft form only): An integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective.

e. MIL-STD-499C (draft): An integrated set of products (to include processes and Government facilities) and personnel which interact with one another in an organized or interrelated fashion toward a common purpose which cannot be achieved by any of the products alone or by all of the products without the underlying organization. The integrated products and personnel fulfil manufacturing, verification, integration, deployment, training, operations, support and disposal functions to provide needed operational capabilities or satisfy objectives. The system products include factory, operational and depot hardware and software (delivered and developer); purchase requirements; manufacturing processes and instructions, verification plans and procedures; deployment plans and procedures; training plans and courses; technical manuals; support plans and spare parts requirements; and disposal plans, instructions and, if needed, equipment. An acquisition program develops, produces, and deploys the products and defines the skill and manpower levels for personnel.

f. IEEE Std 1220-1998: A set or arrangement of elements [people, products (hardware and software) and processes (facilities, equipment, material, and procedures)] that are related and whose behaviour satisfies customer/operational needs, and provides for life cycle sustainment of the products.

g. NASA Procedural Requirements for the Engineering of Systems: The combination of elements that function together to produce the capability to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose.

These definitions clearly have a lot in common, as would be expected, and differ mostly only in the level of detail, but two points are worth noting. Firstly, they all require that the system has a *purpose*, and that this purpose is prior to the system. That is, the system is designed with the *intent* of fulfilling this purpose; this is the defining characteristic of an *engineered system*, as we shall discuss further in Secs. B3.1 and C2.1. Secondly, while it is not explicitly stated, the wordings imply, in varying degrees, that the system is a physical system, in the sense that it consists of such physical elements as hardware, software, processes, personnel, etc. None of these definitions seems to suit a system of thoughts or ideas or concepts, even though engineering is mainly an intellectual activity.

What about definitions in the literature? A small sample will demonstrate that these vary greatly, depending on the viewpoint and/or purpose of the author.

(i) A system is an array of components designed to accomplish a particular objective according to plan. [5]

(ii) 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) [6].

(iii) A system is defined as a set of concepts and/or elements used to satisfy a need or requirement [7].

(iv) 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. [8].

Within INCOSE itself, the SE Handbook [9] defines a system as

(v) 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:

(vi) A system is a construct or collection of different elements that together produce results not obtainable by the elements 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, behaviour 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 [10].

All of these definitions are proper and have their area of applicability, some wider, some more narrow; their differences only point out that the concept of a system and its application to engineering is not entirely straight-forward. But while there is nothing wrong with any of these definitions, they are all completely utilitarian and do not convey any understanding of the nature of the concept nor of what the purpose is of applying it to engineering. But without that understanding it is not possible to know if we are applying it in the most effective manner.

References

1. The early history of the word “system” is taken from The Oxford English Dictionary, 2nd edn. Clarendon Press, Oxford (1989)
2. Collins Shorter Dictionary, HarperCollins (1991)
3. Australian Pocket Oxford Dictionary, 2nd edn. Oxford University Press, Melbourne (1984)
4. Aslaksen, E., Belcher, R.: Systems Engineering. Prentice Hall, Australia (1992)
5. Johnson, R.A., Kast, F.W., Rosenzweig, J.E.: The Theory and Management of Systems. McGraw-Hill (1963)
6. Wymore, A.W.: A Mathematical Theory of Systems Engineering - the Elements. Wiley (1967)
7. Miles, R.F. (ed.): System Concepts. Wiley (1973)
8. Wymore, A.W.: Systems Engineering Methodology for Interdisciplinary Teams. Wiley (1976)
9. Systems Engineering Handbook. In: Haskins, C. (ed.) INCOSE (2006)
10. http://tucson.sie.arizona.edu/sysengr/INCOSE/WhatIs.html

A2 The Philosophical Context

A2.1 The Philosophical Framework [1]

A2.1.1 The Characteristics of Philosophy

In order to understand how the system concept fits into our view of the world and our existence as human beings, we need to make a brief detour and consider the pertinent aspects of philosophy. The first step is to recall its basic structure and nomenclature.

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 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 often 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.

A2.1.2 The Subject Matter of Philosophy

There are, 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. 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 pervasiveness and importance of engineering in modern society.

A2.1.3 The Structure of Philosophy

Although in principle philosophy is about providing an understanding about the connectedness 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 on 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 conceptual 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 items 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 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 *verificationism,* 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 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*

Particulars 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 are commonly 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 following subsection gives a slightly expanded discussion of these issues.

A2.2 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 $x \in X \Rightarrow x \in Y$ (or in words: "x being a member of X implies that x is also a member of 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 "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 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 foray into philosophy - 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 properties of the mind; that is, the languages children learn are the ones that these properties of their minds make them predisposed to learn.

For our purposes, the significance of this 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 language use. In sec. A3.1 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.

A2.3 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, An example of a pure intuition is 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 pre-existing (*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 constitute the entities on which the thinking process operates. This is illustrated in the diagram, Fig. A2.1.

+

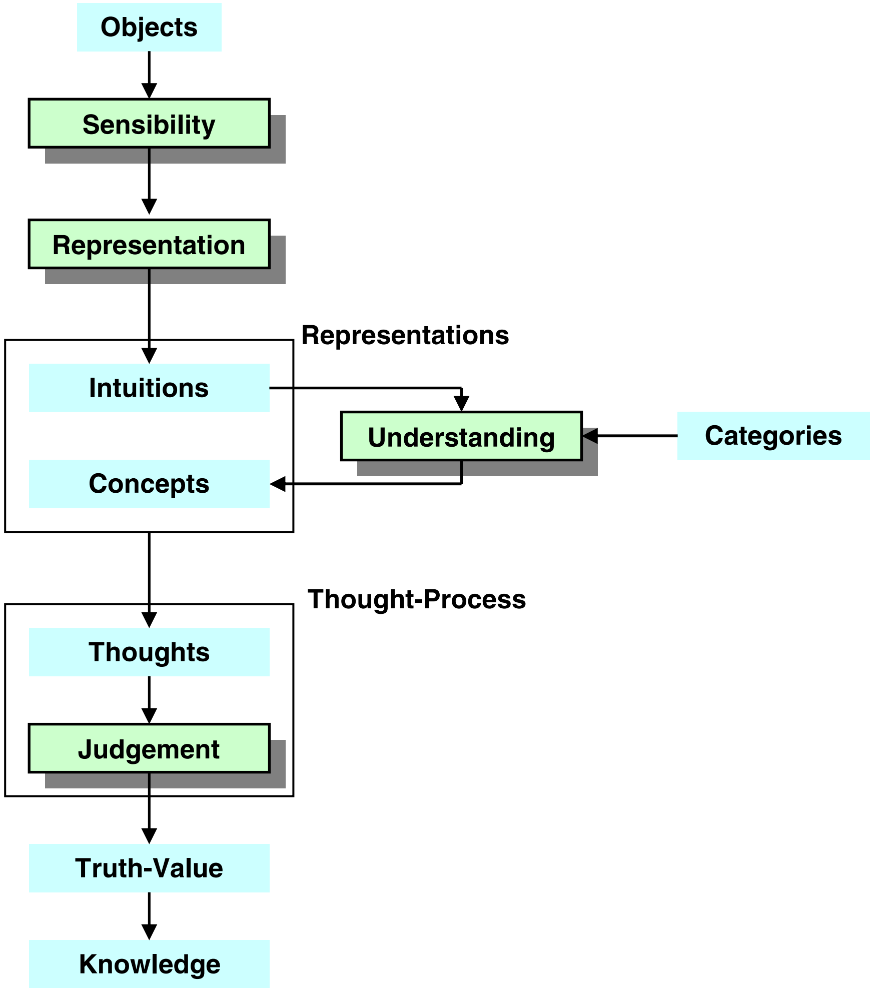


Fig. A2.1 Kant’s view of the thought-process.

A2.4 Abstract Entities and Linguistic Frameworks [7]

In the two preceding sections 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 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 observable 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 rules. The purpose for which the language is intended to be used, for instance, the purpose of communicating 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-no 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. There are two essential steps. First, 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 second, 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.

References

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, `http://www.utm.edu/research/iep/` and the Stanford resource, `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 Geach, P., Black, M.: Translations from the Philosophical Writings of Gottlob Frege, Blackwell, 3rd edn. Oxford (1980). Further discussions of Frege's work can be found in Michael Dummett, Frege: Philosophy of Language, 2nd edn. Duckworth, Harvard University Press, London, Cambridge MA (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 Smith, N., Wilson, D.: 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, `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 Semantics and Modal Logic, enlarged edition. University of Chicago Press (1956), and available online at, `http://www.ditext.com/carnap/carnap.htm`

A3 The System Concept within the Philosophical Framework

A3.1 Categorisation 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 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 if and only if C is a subset of S." We have no difficulty in determining C, but what is S? There is no such set; there is no rule that allows us to identify one thing as a system and another thing as not being a system, and therefore we have to conclude that "systems" is not a class of things.

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 *partition* the parameters into smaller groups and to process them as separate objects. The most immediate evidence of this is language; in order to express something complex, such as a story, we use a limited set of vowels that can be combined to form words, the words are subdivided into groups (nouns, verbs, predicates, etc.) and combined to form sentences, and the whole story is a string of sentences. However, the elements need to *interact*, and in the case of language the interaction takes place in the mind of the listener and is determined by the sequence of the vowels, words, and sentences.

Another example of using the system approach is provided by structured programming. The partitioning of a program into modules is so that it is easier for the human to understand and thereby be able to verify, test, and maintain. To the computer it would make no difference (except perhaps with regard to memory requirements) if the program was just one long unstructured file.

The converse of this subdivision of complex entities is the aggregation of simple, or low information content entities, so-called *chunking*, as was first described in the seminal paper by Miller [2]. In this paper, he examined experimental data on absolute judgement, the resolution (or bits in the value of the measure) with which we can characterise stimuli without making errors, and demonstrated that the number of chunks of information that can be retained in short-term (or immediate) memory is about seven, irrespective of the number of bits of information in each chunk, as determined by the dimensionality of the stimuli (limited to the experimental data available), and he attributed this to a process of *encoding*, a learning process through repetition that relies on long-term memory. In the context of engineering, we might call the chunks “objects” or “elements”, and the encoding “information hiding”, as we shall discuss further in Chapter B5.

Miller illustrated the idea of chunking by considering a person learning Morse code. At first, every dot and dash is heard as a separate element, then these sounds are organised into letters, then the letters into words, and finally whole phrases. Another illustration of really big chunks is our ability to recognise persons; when we turn a corner and are confronted with Bill, we are able to instantly say “Hi, Bill”.

The picture of the dual processes of chunking and partitioning, as illustrated in Fig. A3.1, will provide an important conceptual foundation for examining the process of engineering in terms of the two dual processes of top-down and bottom-up design, in later chapters.

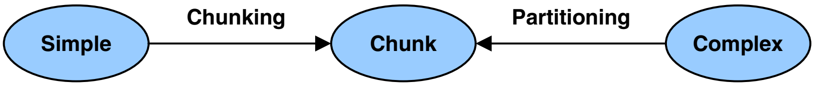


Fig. A3.1 The dual processes of chunking and partitioning.

Based on the observation that the mind prefers to subdivide a complex whole into roughly seven parts, 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, and a very simple (simplistic?) model of memory activity that demonstrates this characteristic is outlined in the following section. But before going into that, it is interesting to briefly consider our attitude to the brain as compared, say, to our attitude to our hand. We understand the capability of the hand, we know how to train it do certain tasks, such as playing the piano, and we know its limitations. And we know what happens if we take an axe to it, but do we know what is the equivalent mental activity? What mental activities can harm the brain, which ones can improve it? What tasks are suited to the brain? To become an operator of a complex machine tool requires a long period of training, but even though its complexity is as nothing compared to that of the brain, how much training do we get in using the brain? We tend to think of “me” and “my brain” as separate, and the brain as something “I” can control, whereas in reality “I” am to a large extent what my brain allows me to be. Just consider for a moment memory, the ability of the brain to store and retrieve data; without that we would have no concept of time, there would be no past and no future, just now. How we experience the world is obviously a function of how our brain is organised, so it is very reasonable to expect the functionality of the brain to provide the definition of what we consider to be complex, and that this is therefore a specifically human characteristic.

A3.2 A Simple Model of How the Brain Handles Complexity [3]

The purpose of this little digression is not to claim any understanding of how the brain is constituted and *actually* works; it is simply to present a model of information processing in engineering terms that can make it *plausible* that partitioning of a complex thought-entity could be an inherent feature of the brain, and so make the system concept more real for engineers.

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 increase 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 time t equals $e^{-t/c}$. The amount of energy required to set or reset (refresh) a unit of information in short-term memory, 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 require the set of variables to be connected; that 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 sub-entities *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 χ = 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

While not a very sharp minimum, it does show a clear minimum at about s = 7.

A3.3 A Linguistic Framework for the System Concept in Engineering

In Sec. A2.4, 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. And we need to always remember that we think in terms of our natural language. No matter what symbolic or specialised "languages" we use, such as mathematics, circuit diagrams, various network diagrams, SysML, and the like, the understanding of what the symbols mean is ultimately expressed in natural language, and it is this linguistic understanding our mind operates with. Linguistics is of central importance to any mental activity, and the application of the system concept to engineering is no exception.

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 dedicated facility most likely yes, but if it is a shared facility 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 telecommunications 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 engineers 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. As the definitions in Sec. A1.3 show, the existence of a boundary is intrinsic to the system concept; the fact that all systems of interest to engineering are *open*, with interactions across this boundary, is a different matter.

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” are 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, Fig. A3.2.

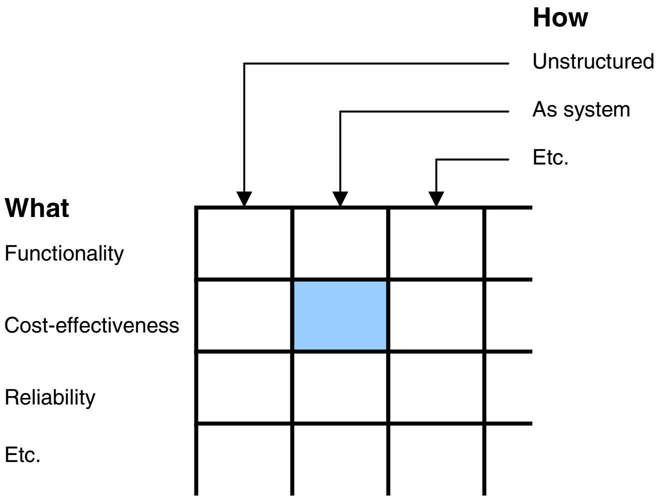


Fig. A3.2 The set of all descriptions of an object

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 engineers 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 the systems engineering community 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; 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.

A3.4 Implications for the Use of the System Concept

We now understand that “system” is a mode of description of an object, and that the purpose of this mode is to make it easier for us to conceptualise and work with an object with a form and/or behaviour that appear complex to us. We also have a general idea of the partitioning process and its boundaries. But what does this all mean as far as applying the concept in practice? Are there any rules or procedures that arise out of this understanding?

The first observation we can make is that there can be more than one system associated with an object; the system is tied to the aspect of the object we want to describe. For example, for a given product, the system describing its composition in terms of physical elements is different from the system describing how it is manufactured in terms of processes, and both are different from a system describing its cost structure in terms of cost codes. These systems interact, e.g. changing the physical composition (architecture) will influence both the manufacturing process and the cost, and so these systems could be considered as subsystems of a “complete” description of the object. The word complete has been enclosed in quotation marks, because we recognise that there is no such thing as complete *per se*; the definition of a boundary is part of every system. However, there is a significant difference between these systems and the system arising out of combining them, which is perhaps best illustrated by taking the case of the physical system. In that system we describe a complex physical object, such as a process plant, in terms of a set of elements drawn from an ensemble of elements of the same type; the set is defined by a rule which allows us to determine whether an element is within the set or outside, and this also defines the boundary. There can be interactions across the boundary, but these are always interactions between elements of the same ensemble. But when we form a system of this system and systems describing other aspects, such as cost, they are not of the same type, and there is no definable ensemble. The boundary created by our choice of aspects is of a very different nature, and there are no interactions across this boundary. In other words, deciding on the completeness of our description of an object, i.e. the degree to which it is a "holistic" description, involves two very different decisions. The first one is which aspects to include, and then, within each aspect, where to draw the boundary, as in whether to consider a single butterfly in South America an element in the system determining the weather in Australia.

Focusing on the description of a particular aspect, we are faced with the issue of how to choose the "best" partitioning. And how do we define "best"? That depends on what we are describing, but as a general statement, the best choice of elements is the one which minimises the interactions between them. However, the partitioning of an aspect of a complex object into a set of elements is usually not done in one operation; it is done in a step-wise, top-down manner so that, according to our insight into how the mind handles complexity, the subdivision in each step results in typically less than ten elements, and the general rule is that there must be only one partitioning criterion at each level. For example, at the first level it might be the contract under which the element will be produced, at the second level it might be by location, and at the third level by functionality. Or, if the aspect is cost, at the first level it might be by project phase, at the second level by activity (e.g. design, construction, operation, and maintenance), and at the third level by cost type (material, labour, financing, etc.).

Finally, an obvious, but still not always implemented rule is that the set of interacting elements (i.e. the system) must be identical to the original object, in the sense of describing the same object. The partitioning process involves design, e.g. in the choice of technology or in the choice of contracting strategy, but at each level the set of interacting elements must fulfil the requirements placed on the element they came from on the level above. The importance of this upward traceability and the anchoring of the process in the original object as the top element are essential features of the system design process, as was set out in considerable detail in an earlier work [4], and we shall return to this (and the other issues raised here) in Chapters A5 and C2.

References

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 Philosophie16, 192–205 (1892), We would perhaps call a second-level concept a class of concepts
2. Miller, G.A.: The Magical Number Seven, Plus of Minus Two: Some limits on Our Capacity for Processing Information. The Psychological Review 63, 81–97 (1956); available online at `http://www.well.com/user/smalin/miller.html` References to subsequent papers can be found at, `http://citeseer.nj.nec.com/context`

3. The model is contained in E.W. Aslaksen, The Changing Nature of Engineering. McGraw Hill (1996). It is based on the concept of information as decrease in entropy and the effort required to maintain a (low) state of entropy in the face of the tendency of the entropy to increase (in accordance with thermodynamics)
4. Aslaksen, E.W.: Designing Complex Systems: Foundations of Design in the Functional Domain. CRC Press (Taylor & Francis) (2008)

A4 Some Features of the System Concept

A4.1 Emergence

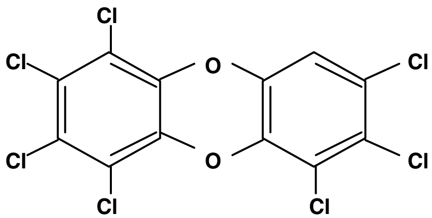
Emergence is a central feature of the system concept, and while there is an ongoing debate about the exact nature of this feature, it is often summed up by the statement that "the whole is more than the sum of its parts". That is, the system has properties that are not evident in any of its elements. Or, from another perspective, the properties of the system are determined not only by the properties of the elements, but also by the interactions between them. Most often the properties of interest are in the form of capabilities; that is, the system has capabilities that are not found in any of its elements. An interesting example was already mentioned in Chapter A3; a sentence can be described as a system, with the words as elements and the meaning of the sentence as the emergent property. Another typical example would be any integrated electronic circuit.

Based on our understanding of the system concept as a mode of description, as we developed it in Sec. A3.2, there is no problem with defining the emergent properties of a system; much of the current discussion, as described e.g. in a paper by Ryan [1], would appear to arise from a lack of clarity regarding the system concept. Consider a given object. If it is described as a single object, it has no emergent properties; all its properties are simply the properties of the object. If we describe it as a system, i.e. a set of interacting elements, then the emergent properties are those that disappear when we turn off the interactions between the elements. Thus, the existence of emergent properties is simply a feature of the system concept; they are not defined by the object itself.

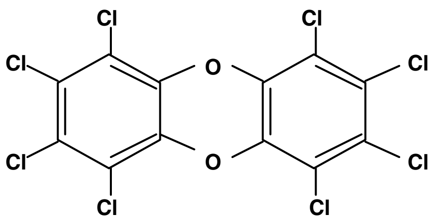
The interest in and research regarding emergence arise from the converse situation; that is, given a set of elements with their capabilities for interaction, how can one predict the properties of the object that results from letting them interact in a particular manner? This question has always been central to one engineering discipline, chemical engineering. One of the fundamental processes in chemical engineering is reaction, in which two or more substances are brought together under controlled conditions to form one or more new substances, and where the new substances have properties that are not found in any of the components. There is now a vast amount of knowledge about chemical reactions that allows a degree of predictability, but the unpredictability of the properties of new substances is particularly striking when it comes to their interactions with living matter, which, by the way, accounts for the cost of developing pharmaceuticals, and two examples illustrate this. The first is two members of a class of substances call dioxins, and the emergent property is the Toxic Equivalent Factor (TEF), a measure of the carciogenic effect of the substance. These two substances, with structures as shown in Fig. A4.1, differ only by a single chlorine atom, but the TEF differs by a factor of 100 [2]. As an aside, imagine this case translated into the realm of teams, with people instead of atoms. Could the addition of a single person to a team of 21 persons result in a change in effectiveness of a factor of 100?

The second example is a herbicide sold under the trade name of Dual (a registered trademark of Syngenta Agro AG, Switzerland). Here, the difference between the two substances is not even a difference in the atoms making up the molecule, but simply a difference in the structure of the molecule (i.e. they are isomers). The structures of the two isomers resulting from a stereoselective reaction are shown in Fig. A4.2, and the herbicide effectiveness, measured at an application density of 500 grams per hectare, is 92 % for the S isomer, whereas that of the R isomer is only 39 % [3].

The existence of such emergent properties is one of the reasons why the system concept is useful; very complex system behaviour can often emerge as the result of letting relatively simple elements interact. (The other reason is the manner in which the brain operates, as already mentioned in Sec. A2.3.)



(a) Heptachlorodibenzodioxin, TEF = 0.01



(b) Octachlorodibenzodioxin, TEF = 0.0001

Fig. A4.1 Two members of the dioxin family.

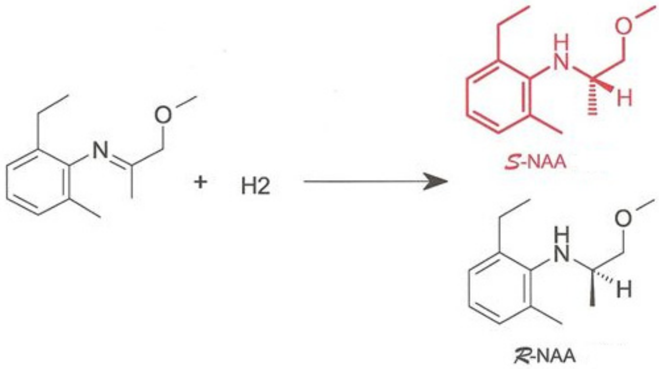


Fig. A4.2 The stereoselective reaction in the S-NAA-process.

An issue closely related to the concept of emergent properties is the fact that properties of a system not present in the individual elements can be completely described using the functional parameters of the elements; no new parameters are required. It may be much more *convenient* to introduce new functional parameters which directly characterize the emergent properties of the system, but these will always be related to, or expressible in terms of, the functional parameters of the elements. A little example from an important type of interaction, correlation or coherence of identical elements, is given by considering a set of identical radiators being combined to provide the service of illuminating a small, remote spot with radiation. From the users' point of view, a most useful functional parameter is beam-width, and they might not even be aware of the relationship between beam-width and the parameters of the individual radiators (relative position and phase angle). - Another example, from a different type of interaction - elastic collision between mass points - is the characterization of a gas in terms of pressure and temperature and, if the "service" of the gas is its ability to absorb and release heat, the functional parameter heat capacity; all of which are related (albeit in a statistical manner) to the parameters describing the individual mass points.

However, there are also emergent properties that are inherent in the system concept itself and not related to any properties of the object being described as a system; they are characteristics of any set of interacting elements, and two such properties are discussed in Secs. A4.4/5.

A4.2 Size and Composition

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

The relationship between the value of n and the level of detail arises only within the top-down development of the system; the size of the system (and remember, this is the description of the object) is increased in a step-wise process, from global variables into more and more detailed variables associated with an increasing number of system elements, until all the variables and relations in the original description are accounted for. This process does not (indeed, must not) add anything to the original description; it recasts it in a form that will make the design process much more efficient. This is described in Chapter A5; here we shall first look at some of the implications for a set of n elements, no matter where in the top-down design process it is located.

A first characterisation of systems arises by observing that sets of elements fall into two main categories; *homogenous* sets, in which the elements are all identical, and *heterogeneous* sets, in which the elements are not all identical. It is important to note that, when we say "the elements are all identical", we mean "identical with respect to the properties under consideration". For example, we might consider the subscribers to a communications network as identical (i.e. all as average or typical subscribers) for the purpose of network design, although they are, of course, all different as individuals. This type of homogenisation by means of averaging is used extensively in the treatment of complex systems (e.g. "per capita" variables in economics, "per vehicle" variables in traffic modelling, etc.).

In heterogeneous sets, the elements are characterised by a set of variables, say u_i, $i = 1$ to m. Each variable may take on values in a continuum (e.g. real numbers) or in a discrete set (e.g. integers), but for the purpose of forming *subsets*, the continuous variables are converted to discrete ones by defining intervals; a typical example would be age groups within a population; within each subset, the elements are considered to be identical. This type of *aggregation* can form a useful step in going from the simplification of a homogeneous system to a heterogeneous one. Consequently, a subset is identified by a vector, $\boldsymbol{u}$, with each component taking on values in a discrete set, and the *composition* of a heterogeneous system is a function on $\boldsymbol{u}$, $\mathbf{Z}(\boldsymbol{u})$, which is restricted to integer values denoting the number of elements in the subset $\boldsymbol{u}$.

A4.3 Structure

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

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

The result of this is that we can represent a system as a directed graph, with the elements as the nodes or *vertices* and the interactions as the *edges* of the graph, and we shall call this graph a representation of the *structure* of the system. In Fig. A4.3 we show the same set of elements with three different structures; this illustrates the fact that the same set of elements can form different systems or, as it is often formulated (although, strictly, this is not a correct wording), that a system can have a *dynamic structure*.

This dynamic behaviour can extend beyond just different interactions between a set of elements to cases where some or all of the elements temporarily turn their interactions off and thereby withdraw from the system, so that the system size is itself dynamic. In the case shown in Fig. A4.3, this would occur e.g. if the lecturer and two of the students were studying their own notes while the other two students were discussing among themselves; the system size is temporarily reduced to 2.

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

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

From this, it follows that the concept of structure is independent of the nature of the elements and their interactions (i.e. as far as structure goes, the elements may be considered identical and indistinguishable), but it is not independent of the number of elements in the system. However, intuitively we would say that all systems with one central element and n-1 elements, each communicating with this central element only, as in Fig. A4.3 (a) or (c), have the same structure. In other words, the concept of structure is *scalable*.

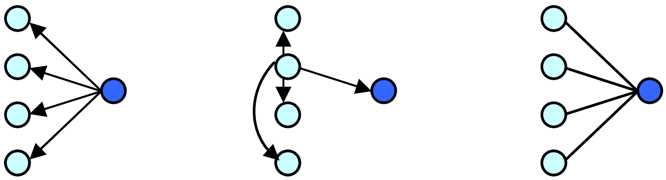


Fig. A4.3 Three structures of a set consisting of a lecturer and four students. In (a) the lecturer is lecturing, in (b) a student is asking a question, and in (c) the lecturer is conducting a distance survey of the students.

The number of different structures increases rapidly with the number of elements, and to get a handle on this we introduce an (n-1)-dimensional vector, **Ω**, called the *system configuration*, as follows:

$$\Omega = (\omega_1, \omega_2, \ldots, \omega_{n-1}),\ with\ \sum_{i=1}^{n-1} \omega_i = n\ , \tag{A4.1}$$

where ω_i is the number of elements supporting i links to other elements. The configurations of a few well-known structures are immediately identified, as shown in Table A4.1.

Table A4.1 The system configuration of four simple structures.

Structure	Configuration
Linear chain	Ω = (2, (n-2), 0, 0, ….)
Closed chain (circle)	Ω = (0, n, 0, 0, ….)
Central element (e.g. broadcast)	Ω = ((n-1), 0, 0, …., 1)
All-with-all (maximally connected)	Ω = (0, 0, …., n)

Intuitively we feel that the *complexity* of a system depends on several factors, including the number of elements, number of different classes or types the elements fall into, and the number of interactions between the elements. With regard to the latter factor, for a given set of elements we feel the complexity should be considered greater the more interactions there are between the elements, and a very simple measure of what we might call *structural complexity* is therefore given by

$$\chi = \frac{1}{n(n-1)} \sum_{i=1}^{n-1} i \cdot \omega_i\ . \tag{A4.2}$$

The values of this factor of complexity for the same four structure listed in Table A4.1 are shown in Table A4.2.

Table A4.2 The structural complexity of four simple structures.

Structure	Structural Complexity
Linear chain	$\chi = 2/n$
Closed chain (circle)	$\chi = 2/(n - 1)$
Central element (e.g. broadcast)	$\chi = 2/n$
All-with-all (maximally connected)	$\chi = 1$

A4.4 Systems and Thermodynamics

It has probably occurred to you that there is a strong analogy between the system concept and the view of matter as consisting of particles, and it is interesting to pursue this analogy for a moment to see if it can benefit our understanding of systems. The idea that matter was made up of small, indivisible particles goes back to Greek philosophers, who called them *atoms*. The concept seems to have been first enunciated by Leucippus and then further developed and documented by his student Democritus [4]. In the 17th and 18th century chemists identified the substances that could not be further separated by chemical means, and the atoms became the constituents of these *elements*. At the same time, physicists were developing the concepts of the macroscopic properties of matter, such as volume, mass, energy, temperature, and pressure, and improvements in measuring instruments and techniques allowed the relationships between these variables to be determined and formulated as empirically justified laws of Nature. In particular, the study of the macroscopic behaviour of gases led to a new subject in physics called thermodynamics, and it was then reasonable to look for a connection between this macroscopic behaviour and the mechanical behaviour of the atoms making up the gas. This connection was made by Maxwell and Boltzmann, resulting in the further subject of statistical mechanics, which was rapidly extended to all three phases of matter (gas. liquid, solid) and to the transitions between them.

There is a close conceptual affinity between systems engineering and thermodynamics, including its extension to statistical mechanics. In both cases, the objective of the methodology is to allow the human mind to explore and understand objects that are intrinsically of a complexity way beyond the capability of the mind to comprehend directly. In both cases, the approach is what is often called *top down*; a step-wise development of understanding starting with a high-level, phenemenological description of the object in terms of a few, directly observable variables and then expanding the level of detail with each step, but in such a manner that the consistency with previous steps is maintained (upwards traceability). That is, a concept on one level is expressed in more detail in terms of a set of interacting concepts on the next level down, and this decomposition of a complex entity into a set of less complex, but interacting entities - the *elements* of the original entity - is the essence of the systems approach. Therefore, as thermodynamics is a relatively mature branch of science and systems engineering is a young discipline, it is natural to try to exploit the conceptual similarities with the aim of advancing the theoretical foundations of systems engineering.

Of course, the objects considered by thermodynamics are very different to those considered by systems engineering. A typical thermodynamic object is a (dilute) gas, whereas a typical object in systems engineering would consist of hardware, software, and people. In thermodynamics the number of elements (atoms or molecules) is typically of the order of 10^{23}, whereas in systems engineering the breakdown into elements would typically stop before reaching 10^4. In thermodynamics, the interactions between the elements are normally relatively simple, ranging from elastic collisions in a gas to the interactions between molecules in a living cell, whereas many of the objects considered by systems engineering involve interactions between humans. In a somewhat simplified summary, we can say that in the objects of thermodynamics the complexity arises from the number of elements involved, whereas in system engineering the complexity lies mainly in the interactions between the elements. Nevertheless, the statistical approach used in the extension of thermodynamics into statistical mechanics can, with appropriate modifications, be used to great advantage in investigating the properties of systems.

The appropriate modifications arise mainly from the most significant difference between the objects of thermodynamics (or physics in general) and those of systems engineering, which is that while the former *exist* as parts of Nature, the latter are *created* by engineers for particular purposes. We study Nature in order to develop conceptual models of its properties and behaviour, and the success of such models is measured in terms of the extent to which they are *true*. In engineering, success is measured in terms of the extent to which the engineered object meets the *intent* of its designer.

There is one issue that surfaces from time to time in the systems engineering community, the issue of whether a system is open and closed, and it is then often combined with a discussion of that issue in thermodynamics; in particular, the fact that the entropy of a closed system can never decrease. In my opinion, this issue is really a non-issue, as every engineered system is an open system by virtue of the fact that it provides a service (in the broadest sense), and to do that it must interact with its environment.

Whether the entropy increases, decreases, or remains unchanged over time is a different issue, and is, of course, dependent on such factors as maintenance and energy exchange. And while thermodynamics and the concept of entropy can be applied to any physical system; this application is not particularly useful in the context of systems engineering. The basic relationship between entropy and uncertainty, which was exploited elegantly by Shannon in his development of information theory, applies in the case of engineering systems also. We can provide a plausible definition of entropy for such systems, as was done in [5], but it is not clear that this leads to any further insight or to relations between this entropy and any other significant system parameter.

A4.5 Coherence

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

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

Comparing the concept of reliability with that of coherence, we realise that whereas the former is perfectly well defined for an element in isolation, the latter makes no sense for an element without reference to the system in which the element is embedded. In Section A4.1 this was expressed by saying that coherence is one of those parameters that emerge from the system concept itself; in a previous publication [6] such parameters were called second-tier parameters, in contrast to such first-tier parameters as reliability and performance. Thus, coherence can be seen as an additional characterisation of performance that emerges as a result of the interactions between the elements. In order to express this in a more quantitative form, we first consider a system's output, i.e. the measure of what it does, to be expressed by a single variable, U_0, and introduce the concept of an element's *contribution* to that variable, denoted by u_i, $i = 1$ to n. However, in contradistinction to the case of a reliability block diagram, where the elements form a series connection of groups of blocks in parallel, the contributions of the elements are always additive, i.e.

$$U_0 = \sum_{i=1}^{n} u_i \; . \tag{A4.3}$$

Now, instead of characterising the contribution of an element to the system output in terms of a single variable, we use a two-dimensional vector, expressed in radial coordinates as $\boldsymbol{u}_i = (u_i, \varphi_i)$. The system output then also becomes a vector quantity,

$$\mathbf{U} = (\mathrm{U}, \Phi) \tag{A4.4}$$

The angular coordinate or *phase angle*, φ_i, expresses the fact that, while an element may be operating at full capacity as a single element, characterised by the *amplitude* u_i, it may not be contributing to the system output to its full capability because it is not aligned with the system objective. And not only is the system amplitude, U, diminished through such a misalignment, but we shall interpret the resultant change in Φ as the system being diverted from its original objective.

The value of Φ characterising the original or desired objective of the system is arbitrary, so it is convenient to set it equal to zero, and take this to be the reference direction. The system output is then given by

$$U = \left[\left(\sum_{i=1}^{n} u_i \cos \boldsymbol{\varphi}_i \right)^2 + \left(\sum_{i=1}^{n} u_i \sin \boldsymbol{\varphi}_i \right)^2 \right]^{1/2}, \tag{A4.5}$$

and

$$\Phi = arctg \left(\frac{\sum_{i=1}^{n} u_i \sin \boldsymbol{\varphi}_i}{\sum_{i=1}^{n} u_i \cos \boldsymbol{\varphi}_i} \right). \tag{A4.6}$$

The *degree of coherence*, ξ, shall then be defined by

$$\boldsymbol{\xi} = \frac{U}{\sum_{i=1}^{n} u_i}, \tag{A4.7}$$

and the *degree of diversion*, ζ, shall be defined by

$$\zeta = 1 - \cos \Phi\,. \tag{A4.8}$$

To investigate some properties if these new system parameters, let us consider the case where the amplitudes of the elements, u_i, are distributed according to the triangular distribution shown in Fig. A4.4.

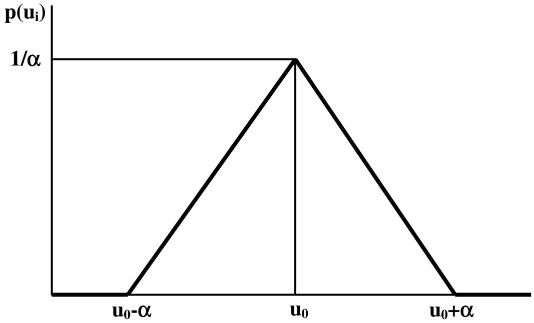


Fig. A4.4 A simple distribution of element contributions.

This very simple distribution reflects the observation that in many systems the majority of elements contribute more or less the same to the system objective, with a small fraction contributing significantly less and a small fraction contributing significantly more. To generate a population of *n* elements with this distribution, let x_i be a random number in the range 0-1, $i = 1, \ldots, n$. If $x_i < 0.5$, then $u_i = u_0 - \alpha(1-(2x_i)^{1/2})$, otherwise $u_i = u_0 + \alpha(1-(2(1-x_i))^{1/2})$.

The element amplitudes are time invariant, but the element phase angles undergo random fluctuations in the range $-\pi < \varphi_i < \pi$ at a constant *phase failure rate,* λ. But there is also an interaction between the elements that tends to align their outputs and which acts as follows: Each element sees the combined output of the other *n*-1 elements, called the *interaction*, and, the phase of each element is aligned with the interaction phase at a rate, the *phase repair rate,* μ, which is determined by the product of the interaction amplitude, E, and a constant *alignment factor,* μ_0. Consequently, the distribution of the phase of an element, φ_i, is given by

$$f(\varphi_i) = a + b\delta(\varphi_i - \Phi) \qquad \text{(A4.9)}$$

where

$$2\pi a + b = 1 \ ,$$

$$b = \frac{\mu}{\mu + \lambda} \ .$$

This system can also, in a very limited fashion, take account of the fact that in many systems the interaction is limited to nearest neighbours or some other, smaller group of elements (just think of people in society, in an organization, etc.). To this end, picture the elements arranged in a ring, and the interaction seen by any one element is obtained by calculating the combined output of the n_0 elements on either side of the element. (In order to account for the case where all the elements interact equally with each other, *n* must be an odd number.)

Finally, we need to introduce one more feature of this system. We have taken the reference direction from which all angles, including Φ, are measured to be zero, but there is nothing in the system definition which makes this the preferred direction of the system output. One possible way of introducing this is that each time the phase angle of an element undergoes a random change in value, this new value is multiplied by $(1 - \varepsilon)$. In terms of enterprises; this can be thought of as a *management effort*; the effort required to keep the work of each participant aligned with the enterprise objectives and to maintain an enterprise culture, but can also be a truly external influence, such as fashion.

Thus, an interpretation of ε is that it could represent the degree of consensus among the members of the group making up the system with respect to their values and purposes. This is a very topical interpretation, for two reasons. Firstly, with all the interest in cellular organisations, terrorist networks, and netwar [7], if we accept this interpretation, our model shows the high leverage one may obtain from a small amount of training or indoctrination in a system setting. Secondly, it correlates with the high importance accorded to developing a company culture and maintaining conformity to this culture in modern management theory and practice; an outstanding example of this is General Electric, where cultural conformity was put ahead of any performance measure [8]. And, thirdly, ε could be interpreted as the influence of Systems Engineering in pulling together the various disciplines within an engineering enterprise.

To investigate the behaviour of this system, a numerical model was developed, in the form of a Visual Basic program, which steps through time by generating random changes to element phases at each step, in accordance with the above equations. The step size is also the *unit of time*. At the start of each step the phase failures are determined; with this interim system state the interaction amplitude and a possible repair or alignment of each element is determined, resulting in the new system state. The program has two modes. In the first mode, the initial state of the system is a random distribution of element phases; the behaviour of the system is then that the element phases become aligned over a period of time, with a duration called the *lock-in time*, T_L. The criterion for determining that lock-in has occurred is that the system amplitude, averaged over a period of 20 time steps, is equal to or greater than the average coherence value (calculated in the second mode) times the sum of the element amplitudes.

In the second mode, the initial state of the system is one in which the element phase angles all have the value zero; that is, the system is fully coherent. This might be the result of a lock-in, or of some other, externally induced alignment. Then, there is a period in which there is a certain fluctuation of the element phases (and consequently of the system phase), but not enough to change the longer term system phase, until the random fluctuations become so large that the system phase angle "flips" to a new value. The duration of the period until such a "flip" occurs is the *time to failure*, T_F, and the criterion for determining that a failure has occurred is that the system phase, averaged over 20 time steps, differs by at least 60^o from the average over the previous 20 time steps.

In the previous publication on the subject matter of coherence cited above [6], it was noted that this temporary breakdown in coherence due to the random fluctuations and the resulting "flips" in the system phase can be observed in some everyday situations, one of them being a meeting to discuss and finalise a position on an issue. Documentation on the issue has been circulated previously, and the issue has been discussed one-on-one with all participants prior to the meeting, with everyone agreeing that a certain general direction would be the preferred one. However, during the meeting, discussing the details of the preferred direction, the opinions of the participants start swinging around in all directions, and suddenly lock on to a direction which is quite different to the one preferred prior to the meeting. No new facts were presented, no new aspects of the issue were raised that had not been raised before; the "flip" is solely a result of the system interaction. It must be that feelings, opinions, attitudes, etc. that are repressed in a one-on-one situation because one wants to appear rational can blossom in a system setting; rationality is no longer the main criterion, it is the good feeling of being in agreement with everybody else.

We may also note, in passing, that a state of chaos, or breakdown of coherence, as a precondition to establishing a new order (as represented by the system phase) has always been well understood by anarchists.

The two variables, T_L and T_F, as well as the two variables defined previously, coherence and diversion, are random variables; each run of the model will give different values of these variables. Consequently, our numerical model must be a statistical model that determines such statistics as mean and standard deviation of *samples* of these variables, and the *sample size*, *s*, determines the accuracy with which we might draw inferences about the underlying distributions. From sampling theory we know that if the mean and variance of the underlying distribution (i.e. of an infinitely large sample) are μ (not to be confused with the alignment rate) and σ^2, respectively, then the mean of whatever variable we are considering has a Gaussian distribution with a mean of μ and a variance equal to σ^2/s. It will turn out that, of our four variables, T_F has the highest value of the ratio σ/μ , and it is of the order of 1. As a result, the standard deviation of the mean of T_F, which we might call the *accuracy* of our model, is equal to $1/s^{1/2}$. That is, for a sample size of 1000, we achieve an accuracy of about 3 %. A single run on a normal PC with 3 GHz clock frequency takes on the average 20 seconds (depending on the parameter values), so a sample of 1000 runs takes about 5.5 hours to complete, and this becomes the limiting factor on the accuracy.

There are two features of our system that should be mentioned. Firstly, the two probabilities λ and μ are measured in units of per unit time, but the unit of time is arbitrary. Thus, we may choose a fixed value of λ, and then consider only the one parameter μ/λ. The fixed value of λ must be small enough to make the step-wise calculation a good approximation of the continuous temporal behaviour of the system, say, $\lambda < 0.01$. Secondly, the parameter μ is given as the product of the interaction amplitude, which again is determined by the parameter n_0, and the alignment factor, μ_0. Now, it is not a priori clear the behaviour of the system depends only on the product of these two rather than on their values individually; on the contrary, from our understanding of sampling theory we would expect the size of the "sample" from which the interaction amplitude is created to influence the stability of the system, and this will turn out to be the case. As a consequence, we will have to deal with the two parameters n_0 and μ_0 individually.

The model also has two features that need mentioning. The first is that the probabilities of changes taking place in any particular time step are generated simply by comparing the values of λ and μ with a random number in the range 0-1. For this to be valid, the resulting probability must be significantly less than 1; say, less than or equal to 0.4, so that, if we restrict n_0 to 40, then the maximum value of μ_0 is 0.01. The second feature in that in mode 2 the system state is one of complete alignment (or coherence = 1), and there is therefore a certain time period before any failure could take place, even if μ = 0. The duration of this time period will obviously depend on our choice of λ.

The first part of the investigation is concerned with the statistics of T_F, and consequently runs the program in the second mode. Figure A4.5 shows the result of thousand runs for a particular case (51 elements, 20 "nearest neighbours", amplitude variation ≈ 0, failure rate λ = 0.005 per time step, alignment factor μ_0 = 0.005 per time step, and influence factor ε = 0), and the values for the mean and standard deviation of T_F are 33,279 time steps and 39,839 time steps, respectively.

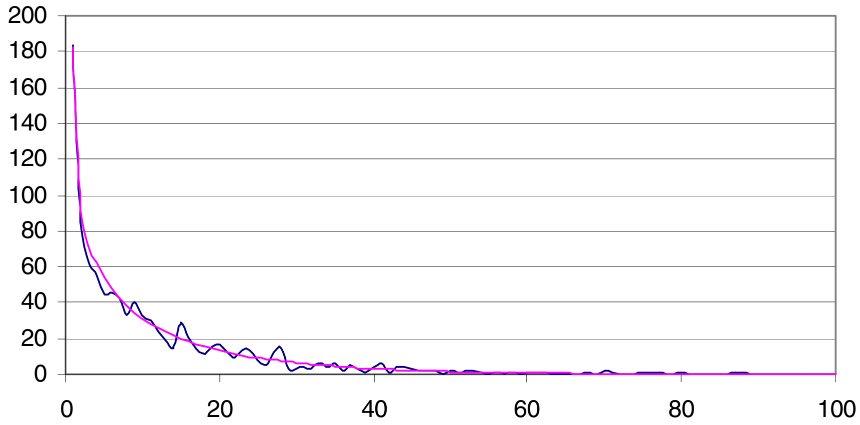


Fig. A4.5 The distribution of T_F for a system with 51 elements, 20 "nearest neighbours", amplitude variation ≈ 0, failure rate λ = 0.005 per time step, alignment factor μ_0 = 0.005 per time step, and influence factor ε = 0. The time scale is in units of 3,000 time steps.

The first thing to note is that, because the mean value and standard deviation of T_F are roughly equal, the standard deviation of the mean value of T_F for a sample size of *m* runs equals T_F divided by the square root of *m*, so that for a sample size of 1000 the standard deviation of T_F is about 3 %.

The second feature of the distribution illustrated by Fig. A4.5 is that it is closely approximated by a gamma distribution, shown as the smooth curve. The gamma distribution is given by the expression

$$f(x) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta}; \tag{A4.10}$$

for $x > 0$, $\alpha > 0$, and $\beta > 0$, and $f(x) = 0$ elsewhere, and where the gamma function may be approximated by Stirling's approximation,

$$\Gamma(\alpha) = e^{-\alpha}\alpha^{\alpha-1/2}(2\pi)^{1/2}\left[1 + \frac{1}{12\alpha}\right]. \tag{A4.11}$$

(Please note that the use of α as a variable symbol here is in deference to common usage, and it must not be confused with the amplitude variation parameter.) The mean and variance of the gamma distribution are equal to $\alpha\beta$ and $\alpha\beta^2$, respectively, so that in the case shown in Fig. A4.5 the values of the distribution parameters are:

$$\alpha = 0.698\ ;$$
$$\beta = 47690\ ; \text{ and}$$
$$\Gamma(\alpha) = 1.3\ .$$

For the case of $n = 51$, $\alpha = 0.$ and $\lambda = 0.005$, the dependence of T_F on the two parameters n_0 and μ_0 is shown in Fig. A4.6.

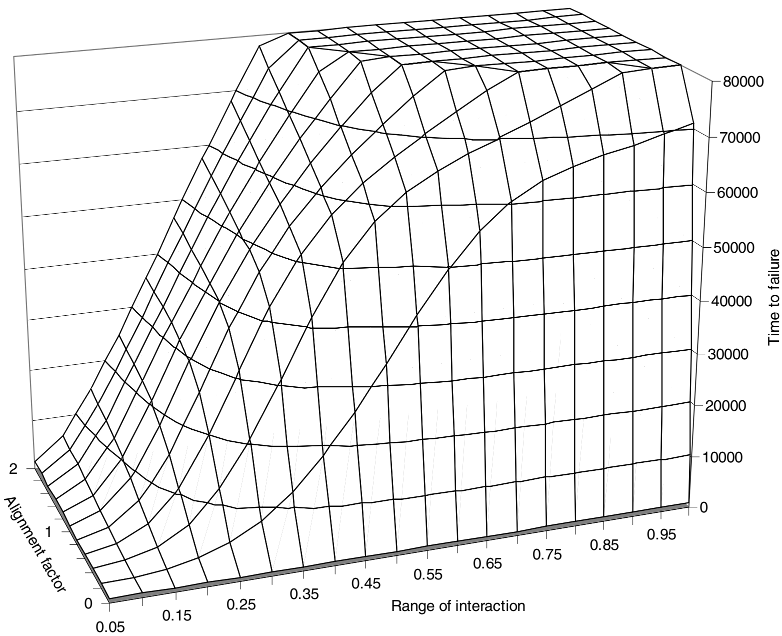


Fig. A4.6 The mean time to failure, T_F (in units of time), as a function of n_0 (in units of $n_0/(n\text{-}1)$, and with a range of 0.05-1) and μ_0 (in units of μ_0/λ, and with a range of 0-2).

Running the model in Mode 2 also produces values of the coherence, averaged over the time to failure, and for the same case as used to produce Fig. A4.6, the results are shown in Fig. A4.7. However, due to the fact that the system starts out in a state with coherence = 1, there is an initial time period in which the coherence decreases from this value to its "equilibrium" value, and the time constant for this exponential decrease is $1/\lambda$ or, in the above case, 200 units of time. For most values of n_0 and μ_0 this time period is very small compared to T_F, and therefore this initial high value of the coherence does not significantly affect the calculated average. The only exception is in the limit of $\mu_0 \rightarrow 0$; the calculated average is 0.31, as shown in Fig. A4.7, whereas the actual value is 0.123. This can be shown by a straight forward Monte Carlo calculation, which yields the result

$$\text{Coherence}\ (n_0;\ \mu_0\!=\!0) = 0.88/n_0^{\,0.5}\ .$$

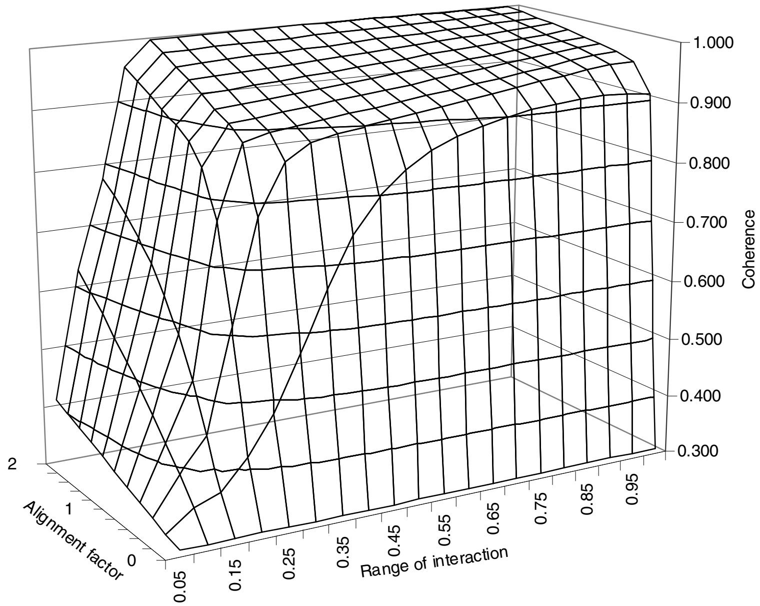


Fig. A4.7 The coherence as a function of n_0 (in units of $n_0/(n\text{-}1)$, and with a range of 0.05-1) and μ_0 (in units of μ_0/λ, and with a range of 0-2).

A4.6 Stability

The concept of stability is perhaps most easily visualised, and most familiar to us, in mechanical terms, such as, for example, a ball, with mass m, resting at the bottom of a parabolic depression, as shown in Fig. A4.8a. Moving the ball away from this equilibrium position results in a restoring force moving the ball back to this position or, in other words, increases the potential energy of the ball; this is a stable equilibrium position. In Fig. A4.8b the ball is resting at the top of a parabolic peak. The slightest disturbance will make it move away from this position; it is an unstable equilibrium position.

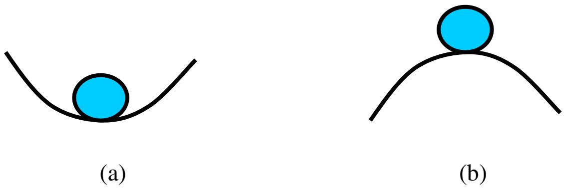


Fig. A4.8 A stable (a) and an unstable (b) equilibrium position.

Another example is provided by a load, *P*, supported by a slender column of length *L*, with a moment of inertia *J* and a modulus of elasticity *E*. If the top of the column is deflected by a small amount *x*, we have the situation illustrated in Fig. A4.9.

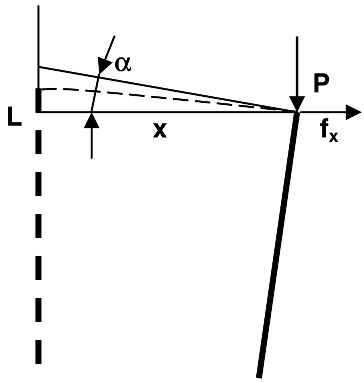


Fig. A4.9 A small deflection, *x*, of a column of length *L*.

The force f_x is composed of two forces, one, f_e, is the elastic force trying to restore the column to its original vertical position, the other, f_p, is the x-component of the force exerted by the weight P:

$$f_e(x) = -\frac{3EJ}{L^3}x \; ; \; f_p(x) = P\boldsymbol{\alpha} = P\frac{3}{2L}x \; ;$$

which are good approximations, as can be found in any first-year mechanics textbook. Consequently,

$$f_x = \frac{3}{2L}\left(P - \frac{2EJ}{L^2}\right)\cdot x \; .$$

For small values of P, f_x is negative, so that it is a restoring force and the perfectly vertical column and its load are in a stable equilibrium. But as P increases, there comes a point where f_x is zero and we have an indifferent equilibrium, and after that it becomes positive and we have an unstable equilibrium. It takes only an arbitrarily small increase in P to transition from stability to instability; it is the proverbial straw that breaks the camel's back.

Both of these cases illustrate the general definition of the *static* stability of a physical system:

A system is in a stable state if any small change to this state results in an increase in the system's energy.

In the first case, the total energy, U, of the system, which in this case is only the potential energy of the ball, increases for any displacement, x, in (a), with $U = m \cdot g \cdot x^2$, but decreases in (b), with $U = -m \cdot g \cdot x^2$. In the second case, the change in total system energy is composed of the change in the elastic energy stored in the column, U_e, which is equals the work done by f_e, and the change in potential energy of the load, U_p:

$$U_e = \int_0^x f_x dx = \frac{3EJ}{2L^3} dx^2 ; \quad U_p = P \cdot x \cdot \alpha / 2 = P \frac{3}{4L} dx^2 .$$

Consequently,

$$U(x) = \frac{3}{4L} \left(\frac{2EJ}{L^2} - P \right) dx^2 ;$$

and so, again, for $P < 2EJ/L^2$, $U(x)$ takes on its minimum value for $x = 0$.

In the case of processes, stability means *dynamic stability*, i.e. that the *rate* at which the process takes place remains stable for small perturbations of the process parameters. An example of a process that can become unstable is the heat generation in a coal stockpile. The coal will tend to absorb moisture, and as this is an exothermic process, heat is generated, leading to a temperature increase limited by the loss of heat to the environment. However, as the temperature increases, the rate of the oxidation process, which also generates heat, increases exponentially, so that the rate of change of the heat generation process is governed by an equation of the form

$$dQ(T)/dt = q - s(T - T_e) + ae^{bT} ,$$

where q is the rate of heat generation due to moisture absorption, s characterises the rate of heat loss to the environment, T_e is the environment temperature, and a and b characterise the oxidation process. This function is shown in Fig. A4.x for three different sets of values of the parameters, resulting in stable, indifferent, and unstable operation.

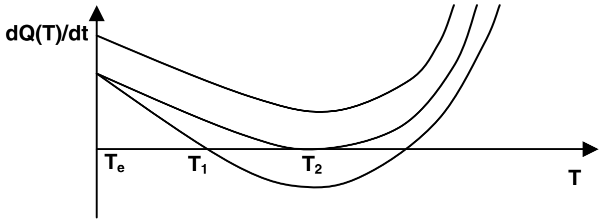


Fig. A4.10 The rate of heat generation, dQ/dt, within the stockpile as a function of the stockpile temperature T, for three different sets of process parameter values. The lower curve shows a stable operating point at T_1, the middle curve has a reduced rate of heat loss and an indifferent operating point at T_2, and in the upper curve the rate of heat generation is so high, and the heat loss so low, that the process has no stable operating point and spontaneous combustion will occur.

Stability is also an important property of systems, and the breakdown of coherence we studied in the previous section is a particular form of instability. If we recall the comments made about the relationship between systems and thermodynamics in Sec. A4.4, we realise that any system represents a degree of order among its elements, as defined by the interactions between the elements, and according to thermodynamics, any such order implies a state of higher internal energy as compared with no order, i.e. a random arrangement of the elements. So, in analogy with the above examples, we would expect our engineered systems to include features that will allow them to turn what would otherwise be an unstable operating point into a stable one, and to see what these features might be, we first identify the forces or events that drive the operating point away from its design point. Firstly, we have the more or less random failures of physical components of all types; electronic, structural, and mechanical. The "more or less" is inserted to remind us that use of "random" here does not mean that each individual component is equally likely to fail in any unit of time; that is only true if the failure rate is constant; i.e. an exponential failure distribution, which is not the case for components that show wear. Secondly, we have failures of software; the errors are inherent in software from its first introduction into service, and reveal themselves as failures under particular data configurations. (A brief discussion of software failure models is contained in [9]). Thirdly, as humans become more and more important as components of engineered systems, we have failures of humans, including incorrect action, delayed action, and no action when a particular action is required.

Handling these three types of failures under the specified operating conditions of the components is what we call maintenance. *Preventive maintenance* aims to eliminate potential failures before they occur, e.g. by measuring wear, or by replacement or retraining prior to rapid increases in the failure rate; *corrective maintenance* rectifies failures on occurrence. As components fail and are repaired or replaced, the performance of the system will fluctuate, and at some point in time it is possible that the performance becomes so poor that we say the system has failed. This leads to the definition of system failure rate, λ, as the inverse of the mean time between such failures and, given the specified operating conditions, which include the requirements on maintenance, the *system reliability*, R(t), is the probability of not having a failure in the time period up until t, and is given by

$$R(t) = e^{-\lambda t} \ .$$

While it would be possible to classify such system failures as instabilities, that would go against common usage for two reasons. Firstly, because there is a definitive long-term average of the system performance, so that in this sense the operation is unchanging and therefore stable, and secondly because both the long-term average and the fluctuations of the system performance are predictable (the latter statistically) as long as the operating conditions remain within their specified limits, and they take on their values as a result of the design of the system. This degree of control is not what we associate with instability; we somehow associate instability of a system with the complete and unpredictable collapse of its performance.

However, this "unpredictability" is due to two very different causes, as can be illustrated by looking at three examples. The first is an electricity transmission grid, characterized by a number of generators, loads, and a network of transmission lines between them. Under normal operating conditions, this grid will absorb failures of individual components, often without any noticeable disruption to supply, sometimes with local, short outages. But if certain load flow conditions occur, either through outages of plant or unusual load configurations, a small, local failure can make the whole system collapse, causing a black-out, as happened in the north-eastern US a number of years ago. There is nothing unpredictable about the system behaviour, it is well known that such systems have a stability limit and how to calculate it; what is unpredictable is the occurrence of *abnormal operating conditions* that will take the system beyond this limit. The ability of a system to perform under abnormal operating conditions is not reliability, but *resilience*, and in the case of systems with a stability limit, resilience is a measure of the distance between normal operating conditions and this limit. Of course, not all systems show instability; in many systems the performance just degrades with distance from normal operating conditions.

The second example is from the field of materials science, and while it is not directly concerned with a system, it is a good precursor to the third example. Normally, when designing a steel structure, the criterion for allowable stresses is tied to the yield stress, typically two-thirds of the yield stress. However, if the structure has to operate under high temperatures, it turns out that there is a second failure mode in addition to instantaneous failure on overstressing, and that is failure occurring as an instability at the end of a long period of slow creep. The *creep rupture* stress, which is considerably less than the yield stress, is a decreasing function of temperature and of the required time to rupture, and if one designs a structure for a creep rupture time of 10,000 h at the operating temperature, then one knows that rupture will occur some time after 10,000 h of operation. At any time during its operation, the structure is in equilibrium with the forces acting on it, but the internal equilibrium, or state, of the structure is slowly shifting ever closer to the boundary between stability and instability, and the "death" of the structure due to this instability is built into the design. There is no issue here of unpredictable operating conditions; the only "unpredictability" is the exact moment of occurrence. (A light bulb is a similar case.)

The third example is a system that maintains itself through interactions with its environment, as is the case with a living organism and, in particular, a human. Throughout the lifetime there is stability in an average sense, with sicknesses and accidents making up the fluctuations, until, at some point, the system goes unstable, and death occurs. In very simplistic terms, such a system consists of two processes, A and B, with A being the externally observable process that does work, gathers food and eats it, defends itself, etc, whereas B is an internal process that uses some of the proceeds from A's activities to maintain A. But B's effectiveness diminishes slowly with time, and due to the feed-back loop with A, there comes a time where the decline in B leads to a decline in A, which again reduces the inputs for B's activities, which results in a further decline in A, and a complete break-down occurs. A simple model of such a system was put forward in [10], and the interesting point about this model is that if fluctuations are ignored, i.e. the system parameters are averages only, then no instability occurs. It is the fluctuations that, due to the non-linearity of the interactions between the two processes, drive the system state slowly towards the system's stability limit.

As engineered systems become more complex and, in particular, the human component becomes prevalent, it is reasonable to expect this latter type of instability to become more prevalent.

References

1. Ryan, A.: Emergence is coupled to scope, not level, Complexity, xx (2007). A condensed version appeared in Insight, the newsletter of INCOSE, vol.11(1), pp. 23,24 (January 2008)
2. Van den Berg, M., et al.: Toxic Equivalence Factors (TEFs) for PVBs, PVDDs, PCDFs for Humans and Wildlife. Environmental Health Perspectives (106), 775 (1998), Quoted in Journal of Toxicology and Environmental Health, Part A 63, 1–18 (2001)
3. Moser, H., Rihs, G., Sauter, H.: The influence of atropisomerism and chiral center on the biological activity of metolachlor. Zeitschrift für Naturforschung 37b, 451–465 (1982) (in German)
4. The entry in Wikipedia is a good place to start, `http://en.wikipedia.org/wiki/Democritus`
5. Aslaksen, E.W.: Designing Complex Systems: Foundations of design in the functional domain. In: Complex and Enterprise Systems Engineering. The Taylor & Francis series. CRC Press, Boca Raton (2008)
6. Aslaksen, E.W.: A model of system coherence. Systems Engineering 6(1), 19–27 (2003), A somewhat abbreviated version can be found in Designing Complex Systems. CRC Press (2008), sec. 8.6
7. Arquilla, J., Ronfeldt, D. (eds.): Networks and Netwars: The Future of Terror, Crime, and Militancy, RAND Publications (2001), It can be downloaded from the Web, `http://www.rand.org/publications/MR/MR1382`

8. Welch, J., Byrne, J.A.: Jack. Headline Book Publishing (2001)
9. Aslaksen, E.W., Belcher, R.W.: Systems Engineering. Prentice Hall (1992), sec. 10.1.6
10. Aslaksen, E.W.: System Thermodynamics: Complexity Emerging from Simplicity. Systems Engineering 7(3), 271–284 (2004)

A5 Applying the System Concept

A5.1 Systems Thinking and the Systems Approach

The application of the system concept to complex matters and, in particular, to complex engineering projects, is often understood in terms of a couple of intermediate processes, as shown in Table A5.1.

Table A5.1 The path leading from the system concept to systems engineering.

The Topic	Its Significance
System Concept	A tool for thinking about complex matters
Systems Thinking	Understanding a complex entity or issue using the system concept
Systems Approach	A technique for solving complex problems using systems thinking
Systems Engineering	Processes for handling the complexity in engineering projects using the systems approach

Systems Thinking is defined by Checkland [1] as an epistemology which, when applied to human activity, is based on basic ideas of systems; Senge [2] states that Systems Thinking is a discipline for seeing wholes. It is a framework for seeing relationships rather than things, for seeing patterns of change rather than snapshots.

Developing an understanding is a step-wise process, and one possible description of this process is:

- Define the *entity*, i.e. the boundaries of the system. This involves defining the *aspect* (e.g. cost, reliability, capability, behaviour) and its *extent* (what it applies to, such as physical or functional elements).
- Define the elements (already understood)

- Define the internal interfaces
- Define the external interfaces (influences)

Often, the complex entity can be characterised as a set of problems or as a problematic situation, and Warfield [3] called it the *problematique*. In that case, the process might look more like this:

- Identify the stakeholders
- Develop a description of the situation and obtain agreement from all stakeholders
- Analyse the situation and identify its elements; i.e. a grouping of problems
- Identify the interactions between the problems
- Determine the strength of the interactions
- Focus on the dominant problems; a smaller system, a simpler situation

There is not a single, agreed definition of "the systems approach". While there is something common to all definitions, the different application of the approach leads to a focus on different aspects.

Simon Ramo [4] thought of it in terms of using a team of cooperating experts in all aspects of the problem; i.e. an integrated approach to problem solving. Senge [*op.cit*], thought of it in terms of enterprise transformation (emphasis on the dynamics). Alex Ryan [5] sees the goal of the systems approach in more abstract terms, as understanding the organisation of ideas and to view problems and solutions holistically.

Let us now, as a preparation for the work in Part C, have a look at some of the implications of applying the system concept to engineering.

A5.2 Representation of an Object

We have argued that the system concept, i.e. viewing an object as a set of interacting elements, is the mind's way of handling complexity. But this still leaves the question of how to choose the elements and their interactions unanswered. Are there any rules or heuristics that can guide us in this regard? To answer this, we need to take a step back and understand what we are dealing with when we say "a complex object". Our understanding of what is meant by "complex" was touched on in several places in the previous chapters, and will be addressed in more detail in the specific context of engineering projects in Chapter C1. For the moment, our intuitive understanding of it as meaning something described by many parameters and relations between them will suffice. The "object" can, in principle, be anything; a piece of hardware, an idea, a software program, a person, etc, but irrespective of the nature of the object, the process of "understanding" essentially means developing a representation in our minds of what the object is, what it does, and how it behaves. If we now look ahead and specialize to objects that are created (of which engineered objects form a subset), where the object and/or its use reflects the intent of its creator, we are considering objects that have a purpose or, as we shall be calling it, that produce a *service* (in the widest sense, including producing a product), and our mental representation of such an object may be visualised as shown in Fig. A5.1.

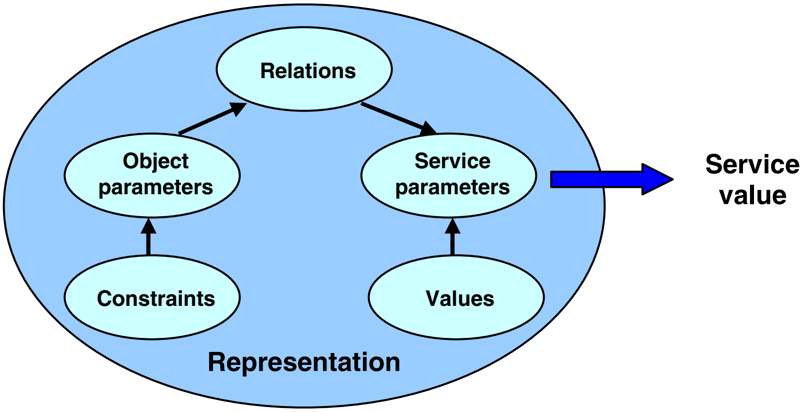


Fig. A5.1 The components of the mental representation of a created object.

A set of parameters and their values are required to describe what the object is, and another set of parameters describe the service produced by the object. The values of the latter are determined by the object parameters through a set of *relations*, this is the set that describes the behaviour of the object. The *value* of the service, which characterises the object's ultimate purpose, is determined by a set of value functions, and the freedom we have to vary the object parameters and thereby optimise the value is limited by a set of *constraints*.

There are a couple of aspects of this representation of an object that require some further explanation and justification. Firstly, of the elements of the representation, only two – the object parameters and the relations - are *directly* related to the object; the others are only *indirectly* related to the object through the person whose mental image this is. (And when we say "person" here, this could also be a group of persons who have, through discussion and consensus building, formed the same mental image of the object.) The constraints depend on that person's view of the conditions under which the object must deliver its service, the service parameters are determined by that person's view of what the object should be doing, and the value functions are in any case completely subjective and determined by the person forming the mental image.

Secondly, the representation in Fig. A5.1 is not unique; in particular, we could have chosen to introduce the concept of the object's performance and the associated set of parameters as a separate element in the representation instead of the set of service parameters, with the latter implicit in the value functions. As it stands, the performance parameters are implicitly included in the set of object parameters. The reason for this choice is, of course, our engineer's view; the objects we are interested in are ones that have a purpose, and as designers what we need to be able to visualise are the relationships between an object's parameters and the service it can provide. That is, the design process starts with a required service, and we then look for objects that might provide this service. So, although the full discussion of the application of the system concept to engineering only takes place in Part C, we are already starting to focus our attention in that direction, and that will be quite obvious in the next two sections, where we look at the two completely distinct classes of objects found in any project to provide a service, but more particularly in engineering projects.

Thirdly, in our mental representation of a given object, the importance of the various components in Fig. A5.1 will vary greatly, and this is determined in part by the nature of the object itself, but also by our current relationship to the object. For example, in the case of a piece of machinery, if we are the users of the machine, our mental picture of it will be mainly in terms of what it can do for us and how we can use it; i.e. its service parameters will dominate, with only a few global object parameters, such as size and weight. But if we have to manufacture the machine, our mental picture will be mainly in terms of the object parameters, as they are embodied in the drawings and specifications. This brings us to the very important realisation that not only do objects have a purpose, but the descriptions of objects also have a purpose. There is not one unique description, or representation, of an object; there are numerous descriptions of varying nature (i.e. balance between the types of components shown in Fig. A5.1) describing different aspects of the object, and for each aspect there are numerous descriptions of varying level of detail (i.e. the number of parameters), forming a manifold of possible descriptions. But they are related through the fact that they all describe aspects of the *same* object, which brings us to the idea that we could structure a complex description as a *system* of simpler descriptions, in effect a system of systems. The elements of such a system should, of course, be as *orthogonal* as possible, in the sense that they should convey different information about the object, much like the axes in a coordinate system, and in the case of physical objects, the choice of such a set of axes is what is reflected in so-called *architecture frameworks*. This is touched upon briefly in A5.4, and some commonly used architecture frameworks within systems engineering will be discussed in Chapter C2.

A5.3 Abstraction and the Top-Down Process

Consider the description of a yet-to-be-designed object in the form of a requirements specification. Such a specification can easily contain many hundred requirements, and is therefore in itself a complex object. If we want to describe it as a system of less complex objects containing, say, seven requirements each, the system would have close to one hundred elements, so how can we make the step from one large element to a system of a hundred interacting elements? It is accomplished in a step-wise process, using what is called *abstraction*, although this name can be misleading, as there is nothing abstract, in the sense of unreal, about the entities created in the process. In the present context, abstraction means "leaving out detail", and the first step in the process is to describe the *purpose* or *function* of the object in terms of a small number of parameters. These parameters may be the most significant of the parameters occurring in the requirements; sometimes they will be commonly used combinations of such parameters, i.e. more global and less detailed parameters, as e.g. cost, if the requirements are given in terms of acquisition cost, operating cost, and maintenance cost. Another such global parameter could be availability, defined simply as the probability of finding the object in a state where it can fulfil its purpose or performing its function.

The next step is then to identify a small number of sub-functions that are necessary and sufficient in order for the object to perform its overall function; each of these again defined in terms of a small number of parameters, and each is again an abstraction in the sense of leaving out details, as compared to the level of detail of the specification. The condition of sufficiency means that the values of the new parameters completely determine the values of the top-level parameters, the condition of necessity means that if any one of the new parameters is removed, it is not possible to determine the values of the top-level parameters in terms of the new parameters. A function (or sub-function), with its parameters, is called a *functional element*.

This step-wise *partitioning* is continued until the lowest level parameters uniquely determine all the requirements in the specification. However, and this is the crucial point, this partitioning is not a simple subdivision, as in dividing a cake into smaller and smaller pieces; that would not lead to any lessening of the complexity. Within each level, the functional elements *interact*, and it is through this interaction that a complex function on one level can be performed by a set of much simpler functional elements on the next lower level; they form a *system*. Which brings us back to the question at the beginning of this chapter, "How do we chose a particular partitioning?", and we now realise that the answer is not a simple one, but consists of some general rules that act more as a guide than as a cook-book recipe, and that applying the top-down systems approach to the description of complex objects does in no way diminish the creative aspect of engineering; it just moves it up a level.

The first rule might be formulated as follows: *Before starting the top-down process, have an unambiguous and complete definition of the object to be described as a system.* If a feature or requirement is missing from the initial definition, the top-down process will not supply it. On the contrary, traceability requires the description in terms of a system to be identical to the original description of the object as a single element. Also, the nature of the elements and the partitioning criteria obviously depend on the nature of the object; a physical object, such as a material handling system, will have very different elements to those of a body of work, such as the work required to create the material handling system.

The second rule relates to the fact that the purpose of describing an object as a system is to make it easier for us to understand and to handle mentally, and it might be formulated as follows: *Before starting the top-down process, be clear about what you want to use the resulting description for. What is it you want to understand: The functionality? The physical composition? The operational use?*

The contracting strategy? The logistic support requirements? As mentioned at the end of the previous subsection, each one of these *views* of the object will have different elements and different interactions, although they are all related by the fact that they concern the same object.

The third rule is: *There can only be one partitioning criterion for each level within the top-down process.* However, the criteria for different levels can be different. For example, for the construction of a motorway, the first partitioning of the work would be into lots, i.e. the partitioning criterion is the location of the work; at the next level the partitioning criterion might be the type of work, i.e. bulk earthworks, structures, paving, etc.

The fourth rule is: *The choice of partitioning criteria should be guided by the desire to minimise the management of the associated work.* This rule arises because we always want to understand a complex object in order to carry out some activity; this could be design, contract administration, construction, estimating, logistic support, etc, and while partitioning the object into less complex elements makes it simpler to carry out the activity within each element, it introduces interfaces between the elements and a corresponding management overhead. A poorly chosen partitioning can negate the benefit of the system approach; the increase in management overhead means that there is no net simplification. In most cases, this rule can also be formulated as a requirement to minimise the interfaces between the elements, although there can be some trade-off between the simplicity of the interfaces and the choice of elements, e.g. when trying to use Commercial-Off-The-Shelf (COTS) elements.

A5.4 Projects and Products

Without in any way limiting the applicability of a systems as a means of description, the context in which we, as engineers, will mostly be applying it is that of a *project*. A project is a body of work taking place within a predetermined time-frame and with a defined purpose, and consequently we see that there are really two main objects associated with a project; the manifestation of the purpose to be achieved, and the body of work required to achieve it. These two objects are quite different in nature; in particular, the body of work only exists within the time-frame of the project, whereas the manifestation of the purpose usually only reaches its completion at the end of the project and then persists for some time period after that. Clearly, these two objects are also very closely related, with the nature and extent of the activities within the body of work being to a significant extent determined by the purpose, but before we examine this relationship in more detail, let us agree on a couple of definitions.

In line with conventional usage and the above definition, the body of work, as an object, will be called *the project*, and the manifestation of the purpose of the project will be called *the product*. We will not be confused by the fact that a project is often named by its product, such as e.g. “the Big River Bridge project”. Furthermore, we recognise that “product” must be taken in its widest sense and is not restricted to a physical product, as in a substance, a component, or a piece of equipment. For example, the project might be the design of an organisation, and the product is the organisation, i.e. a set of relationships and processes. Another example is the project consisting of teaching a person a skill; the product is that person's ability to perform work requiring that skill.

This last example illustrates an issue that will surface in more than one place throughout this book, and that is the issue of ability or capability. Often the purpose of a project will be worded in terms of providing a product that is able to perform a certain task or, with reference to Fig. A5.1, is able to provide a certain service. Whether the product then, in its life time beyond the end of the project, actually provides that service or not is in that case not part of the purpose.

The difference between "shall be able to provide" and "shall provide" is a significant difference in scope and, above all, in complexity of the project, and it reflects the difference between focusing on the physical product and on the service it is to provide. In this sense it is analogous to the difference between prescriptive management (defining what to do) and Management By Objectives (defining what to achieve), and this analogy leads us to the relationship between the entity responsible for the project (say, producer) and the entity requiring the service (say, user). Whether it is sensible for the purpose of the project to include provision of the service rather than just provision of the ability to provide the service will depend on the knowledge and capabilities of the two parties in each particular case, but there are certainly many cases where the producer is best placed to take all or some of the decisions involved in providing the service. One reason is that the skills needed to provide the service may be quite different to those needed to use the service; another is that the provider often gains experience by carrying out numerous projects for different users, whereas each user may only experience one or a few projects.

A couple of examples should make this clearer. In the first example the service is to cure a sickness or disease. Research is carried out, and a couple of drugs are developed that are effective. With these drugs, the users (patients) are now able to cure the sickness (i.e. to provide the service), but they do not have the skills and experience to chose the most appropriate drug for their individual cases. If the project definition includes providing the service, the medical profession will also decide which drug to use and how it should be administered in each case. In the second example, the service is a product, and the business that needs this product (the user) engages an engineering firm to design a corresponding production facility. The user can now either give the engineer a specification for the design of the facility, which will provide the ability to provide the service, or a specification for the product, leaving the choice of facility *and* decisions about how best to construct and use the facility up to the engineer. If the user has acquired many of these facilities and has lots of experience running them, then the first option is the best (although the engineer would still be able to suggest innovations). But if this facility is a first for the user, the second option is likely to be more successful, as it allows the engineer to draw on experience from previous similar projects.

These examples illustrate three issues involved in the two cases of the scope of a project (i.e. ability to provide or provide). The first is *responsibility*. In both cases the producer is responsible for his actions; the difference is that in the first case, the outcome is almost completely determined by the actions of the producer and the *uncertainty* of the outcome is generally very small, whereas in the second case there are numerous factors outside the control of the producer influencing the outcome, and the uncertainty of the outcome may be substantial. The patient may eat or drink substances prohibited by the cure, forget to take the drug regularly, etc; the user may not operate the plant in accordance with the producer's instructions, may use inferior raw materials, etc. The uncertainty in the outcome leads us to the second issue, *risk*; i.e. the product of the probability of a faulty outcome and its cost impact. In the first case, the provider can ensure that the risk is very small, and it is therefore reasonable and customary for the provider to accept this risk. In the second case, one needs to consider the various components of the risk and the party able to control each one, and in the case of those components outside the control of either party, it should be accepted by the party best able to do so, which is normally the user.

This view of a project and the two cases are generally quite well understood, but despite this, the issue of risk allocation causes considerable problems, and the reason for this is largely the attitude of the legal fraternity when it comes to the third issue, *professionalism*, which might seem surprising in a group that is itself professional. A profession is characterised by a body of knowledge and a defined set of activities to which that body of knowledge can be applied. A member of the profession works within a project by applying *professional judgement* to the tasks involved, that judgement being based on the body of knowledge (which includes not only data and techniques, but also the knowledge of how and when to apply these) and the facts of the particular project. This is in contrast to a trade, where a tradesman applies his *skill* to the work.

Any professional judgement involves a degree of uncertainty, depending on the data available to base the judgement on. Gathering that data may well be part of the professional activity, but again it comes down to a judgement on how much effort to expend on that. In the example of the medical professional, the uncertainty of how the patient will react to the drug can be reduced by doing tests, and there is a clear trend towards reducing the uncertainty by doing more tests, as is reflected in the rising cost of medical treatment. But in the end it comes down to judging what is a reasonable trade-off between cost and risk, and while that is, as a general framework, greatly influenced by society's values and affluence, society should accept that, in each individual case, the person best able to form that judgement is the doctor. However, that the doctor makes the judgement and takes responsibility for it does not mean that the doctor can *guarantee* the outcome, and if the outcome is unfavourable it is not the *fault* of the doctor; it's life, a process subjected to a great deal of uncertainty.

A similar situation exists in engineering, where more sophisticated, extensive, and expensive testing technologies and verification methodologies are developed and applied in order to reduce uncertainty. But despite this, there appears to be a definite trend towards a reduced acceptance of professional judgement, and we can discern at least a couple of reasons for this. A major, if not the major factor is the expansion of the legal profession and our move towards an increasingly litiginous society, particularly within the anglo-saxon legal system, where truth and reason are less important, legal proceedings are more like a joust between the lawyers representing the two parties, and the reality of uncertainty is an unwelcome impediment to a good (and profitable) stoush.

Another factor is that professional standards have been somewhat eroded. The best professionals are better than ever, but the spread in competence has widened, and under the combination of rapidly increasing demand for professionals and the downward pressure on funds for tertiary education, the lower limit of acceptability has declined. As a result, unacceptable failures have occurred, contributing to a lowering of confidence in professional judgement.

So, you might well ask, what has this got to do with systems? The point is that if we accept that the essence of professional activity is the exercising of professional judgement, and that a measure of the quality of that activity is the degree to which it reduces uncertainty in the outcome, then a main aspect of that activity must be the ability to simultaneously consider all the factors influencing the outcome. Or, at least, all the factors that have a significant influence; any such factor left out increases the uncertainty in the outcome. With the rapid increase in knowledge in such professions as the sciences and engineering, the number of significant factors involved in any situation and the relationships between them are also increasing, and with them the *complexity* involved in professional judgements. Trying to mentally juggle all these factors simultaneously is no longer an option, and a top-down system approach is required. In the following two sections we look at some general features of describing products and projects as systems; the special application to engineering is treated in detail in Chapter C2.

A5.5 Application to the Product

Every product has a purpose, and we define that purpose as providing a *service* throughout its *lifetime*. These two entities, the service and the lifetime, provide the starting point in the process of describing a product as a system; they provide part of the *system boundary*. This distinction between a service and the product that provides the service is fundamental, and we shall return to this point several times. Basically, the difference is that while a complex service may be described as consisting of a set of interacting, less complex services, the complex service is just the sum of these smaller services; the interactions do not result in any emergent properties, and there is no reduction in the overall complexity of the objects. The reduction in complexity arises solely with regard to the conceptualisation of the service, and therefore the ability of the mind to work with it.

Leaving any further discussion of the relationship between service and product until the next subsection, let us proceed with the application of the system concept to products. There are several ways to view a product; of which the most common are in terms of what it is, what it can do (i.e. its functionality), how it can be used within a given context (operational capability), and how it needs to be maintained (logistic support). Each one of these can be complex and is therefore a candidate for a description as a system, and each such description is called a *view*. So we have a physical view, a functional view, an operational view, and so on, and, in accordance with our definition of a system in A1.2, each view consists of a set of elements, a set of interactions between them, and a set of interactions with their environment. Each view has a particular structure, or, as it is more commonly called in this context, an *architecture*, so that we have a physical architecture, a functional architecture, and so on, and the process of developing these systems is called *architecting*.

However, these views are not independent; they are related by the fact that they all refer to the same object (i.e. the product). In particular, the functional and physical views are related because the physical elements have to realise the required functions, as shown in Fig. A5.2.

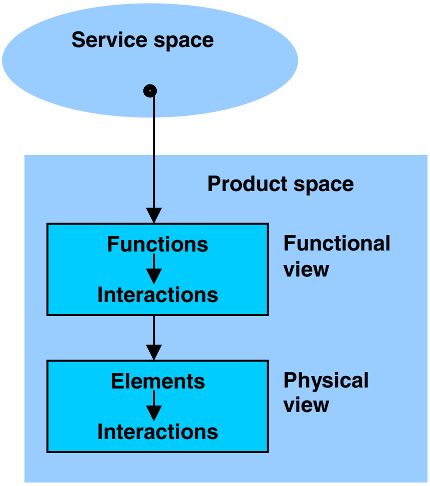


Fig. A5.2 Two views of the product, functional and physical, both of which can be described as systems.

There are several things to note with regard to this application of the system concept to the product. First, we usually have some *choice* when it comes to choosing a set of functions that will provide the service, but for each choice the interactions will be determined by the requirement that the resulting system must provide the service; this is the significance of the short arrows within each view. Second, the same is true of the physical view, but the choice of a set of elements is constrained by our choice of functions. This "form follows function" feature is one reason why system design is not a completely linear process; characteristics of the physical elements (price, delivery time, etc.) may make it necessary to go back and chose another set of functions. Third, there is usually not a one-to-one relationship between functions and physical elements; most often several elements are involved in realising a function, and an element may be involved in realising more than one function. The *allocation* of functions to physical elements can be documented in the form of a matrix, shown below for an arbitrary case with four functions and four elements, where each x stands for a description of the contribution of that element to that function.

	Elements			
Functions	1	2	3	4
a	x	x		
b			x	
c		x		x
d	x			

With the understanding we now have of the system concept, two aspects of this process present themselves immediately. The first is that we have a *choice* of each view; the choice of elements and of the relations between them is up to the person producing the view, subject to the general rules of sec. A5.2 and the requirement that the whole, or external interactions, must remain unchanged, as we remarked on briefly in Sec. A3.4.

The second is related to an important feature of systems we discussed in Chapter A4; emergence. We understand that this feature can only relate to functional systems; in physical systems there can be no talk of the whole being more than or different to the sum of the parts. However, functional elements that by themselves display fairly simple behaviours can, when allowed to interact, together display a complex behaviour, and as simple functions can often (but not always) be realised by simple physical elements, a search for the simplest functional architecture, which we shall discuss in Chapter C3 under the concept of *design in the functional domain*, can be very rewarding.

A5.6 Application to Projects

Recalling our definition of a project as the body of work required to create a product, the elements in a description of a project as a system are always work packages; this is in contrast to the application of the system concept to the product, where the different views had elements of basically different natures, including functions, physical elements, operational elements (i.e. elements of the application of the product within a given context), etc. However, there are also in this case different aspects, and while they are not normally thought of as views, they result in artefacts that look like systems. These aspects are concerned with the means of performing the work, such as human resources, funds, production infrastructure (machines and facilities), and time, and the corresponding artefacts are staff utilisation plans, budgets, equipment allocation, and programs. Each one of these artefacts consists of a set of elements and relations between them, but the elements cannot be chosen; they are determined by the work packages, and each element is an aspect of what is required to carry out the work in a package. This is the reason for the central role of the Work Breakdown Structure (WBS), as will be discussed in much more detail in Chapters C2 and C5.

The final form of the description of the work as a system will, of course, depend on the nature of what is to be created, but the development of the system will always take place in a step-wise, top-down fashion, and in our context of service, product, and project, the first step views the work as taking place in five phases (these phases are not to be confused with the Life-Cycle Stages detailed in the INCOSE Systems Engineering Handbook [6]):

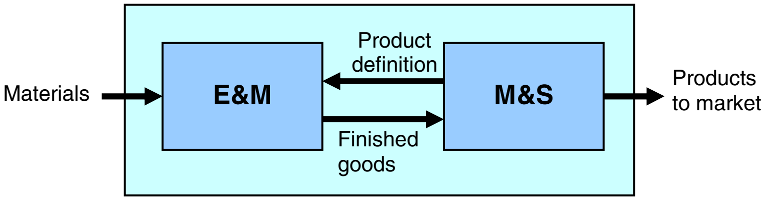
Definition phase	Define and document the service to be provided.
Analysis phase	Determine the functions required to provide this service.
Design phase	Determine the (most cost-effective) product which will realise these functions.
Implementation phase	Realise the product design.
V & V phase	Verify that the product provides the functions and validate that these functions provide the service.

The important feature to note at this point about this subdivision into phases is the change in the nature of the work as we go from the first through the second into the third phase. The definition phase involves working with the stakeholders, and requires approaches attuned to the viewpoints and backgrounds of these stakeholders. Also, it requires the understanding that while the service is described in terms of physical parameters, such as speed of delivery, accuracy, and cost, the service is, as far as the stakeholders are concerned, not tied to a product, even though they may formulate it in terms of a product, because that is what the stakeholders are familiar with. In my experience, dissatisfaction with the performance of a new system will just as often be due to an inadequate definition and understanding of the stakeholders and their requirements as to any shortcoming in the subsequent design process, and this situation must be given due consideration in any application of the system approach to engineering.

In the analysis phase, the service is seen as the outcome of a physical process, and the functions are the elements of this process. However, there are still no physical elements that will provide these functions; these are developed and defined only in the design phase. That is, the product, as considered in the previous section, only appears in this phase. Consequently, there are two very significant *transitions* from service to function to product; transitions in viewpoint, approach, and skills needed. A small example illustrates this.

The owners of an engineering and manufacturing (E&M) facility that until now has been doing contract design and manufacturing decide to acquire their own marketing and sales (M&S) division. The service to be provided by this division is defined as "provide the manufacturing division with definitions of the products to be manufactured and dispose of the resulting products on the market, such that the profitability of the joint operations exceeds that of the contract operation alone". The project is a transformation of the organisation from its present state to one in which it deals directly with the end user market, as shown in Fig. A5.3.

a) Current operation.



b) Future operation.

Fig. A5.3 Adding the marketing and sales service to an engineering and manufacturing operation.

Once the service is defined (and it would normally be defined in more detail than above), the requirements analysis determines what functions are required in order to provide this service, In the present case that would include such functions as acquiring information about and understanding the market and its needs, acquiring information about the competition, providing channels for bringing the products to the market, etc. This analysis phase is followed by the design phase, it which elements that provide these functions are designed. While the functions have no physical form, the elements consist of processes and procedures performed by people with specific skills organised in a particular way, and the systems, consisting of hardware and software, that support these processes. For example, decisions have to be made regarding outsourcing functions or performing them in-house.

Overlaid on, or integrated with, these phases is a process for determining the profitability, or the Return on Investment; increasing the profitability was, after all, the ultimate purpose of the service. In the analysis phase the nature of this process is mostly one of using experience to examine what various functions are likely to contribute to the profitability and so to choose a set of functions that, together, provide the most cost-effective solution. In the design phase, the costs and performance of the elements are actual values, and the accuracy of the profitability predictions (and they are, of course, always just that - predictions) is much better than in the analysis phase.

A second example provides a somewhat different perspective on the relationship between the analysis and design phases. The service to be provided is that of getting clothes laundered, but the requirement is worded in terms of "developing a machine that will give households the capability of laundering their clothes". That is, the means of providing the service has already been restricted by linking it to a particular physical solution; another option would have been to have centralised laundering facilities and a collection and delivery service, but that is now excluded. As a consequence, the functions required to provide this service are also tied to this particular solution and will be formulated in terms of functions of the washing machine, rather than “pure” functions. A function such as “the ability to empty out the water” only makes sense when referred to a washing machine.

We see that there is a dual use of the concept of a “function”; in the one case as “the ability to do something”, and in the other case as “the ability for an object to do something”, and with reference to our discussion in sec. A5.2 we would say that the first case is at a higher level of abstraction than the second. However, as long as we are aware of these two uses of the concept, this should not cause us any problems.

References

1. Checkland, P.: Systems Thinking, Systems Practice. John Wiley & Sons, New York (1999)
2. Senge, P.M.: The fifth Discipline: The Art and Practice of the Learning Organization. Doubleday/Currency, New York (1990)
3. Warfield, J.N.: An Introduction to Systems Science. World Science Publishing (2006)
4. Ramo, S., St.Clair, R.K.: The Systems Approach. TRW Inc. (1998)
5. Ryan, A.: What is a Systems Approach? Journal of Non-linear Science (2008)
6. Systems Engineering Handbook, v. 3.2. INCOSE (January 2010), sec.3.3

PART B

Engineering

B1 A Short History

B1.1 Introduction

Engineering can be characterised as the activity of, on the one hand, combining scientific knowledge with practical experience to develop new technology and, on the other hand, applying this technology to meeting needs expressed by society. Therefore, given the tremendous increase in both scientific knowledge and accumulated experience in the period over which we have written records, say, the last 5,000 years, it is not surprising that the nature and extent of engineering has changed dramatically in this period. And due to the accelerating nature of this development, it is also not surprising that engineering as we know it today, as a profession with an extensive scientifically-based body of knowledge and a defined academic profile, is of relatively recent origin, certainly less than 200 years. However, throughout the whole historic period, the work of the people we can loosely identify as engineers (even if part-time) has exhibited the dual characteristics of using acquired knowledge to develop solutions to construction problems, and then organising and directing the effort required to carry out the construction.

Our purpose in delving briefly into the history of engineering is to obtain a better perspective on the profession as it is today, therefore we shall confine ourselves to developments in Europe, the Middle East, and North America, as it is from these origins that modern engineering arose. Certainly there were engineering activities in other parts of the world, such as the building of the Great Wall and the numerous canals in China, the mining and smelting of copper in the Andes, and the building of the temples in the Yucatan, but these activities had little or no effect on developments in Europe and often occurred later than corresponding activities in Europe. An exception to this were certain inventions, such as the manufacture of gunpowder and paper, but even in these cases the industrial exploitation of these inventions through engineering took place in Europe.

B1.2 The Beginning

At the outset of the 5000 year time period we are considering, the construction activities were centred around irrigation, simple structures such as compacted earth walls, and the production of simple tools and weapons. But from then on, the rate of development increased, both of technology and of the application of it.

More sophisticated tools, such as metal chisels and mallets, appeared, and new devices, such as levers and pulleys were constructed and combined in various forms to allow various tasks, such as lifting water or grinding corn, to be carried out more efficiently and to allow animals to substitute for humans in providing muscle power.

As an aside, it is interesting to note that while the invention of the wheel is always cited as a major technological breakthrough, it was really the invention of the bearing that was most significant. Wheels, in the form of logs, had been used to reduce the friction in moving heavy objects, such as massive blocks of stone, from early times, but attaching the wheel to the moving object only became practical by using a bearing with a much smaller diameter than the wheel to reduce the work due to friction, showing a basic, or intuitive, understanding of the concept of work as the product of force and distance.

Simple processes for mining and processing ore appeared, and with them increasing skills of working metals, first gold, then copper, tin, lead, and then, from somewhere around 1000 BC, iron. Metals became the underpinnings of some of the ancient societies, such as Egypt (gold), the Second Hittite Empire (iron), and the city-state of Athens (silver) [1].

These developments in technology were paralleled by the increase in the size of the construction projects undertaken. The pyramids in Egypt, each of which might have taken 20 years to build with a workforce of 10,000 or so, are the most well known of these large projects, but there were many others, including the city of Memphis itself and large dams and canals in Mesopotamia [2]. Such construction projects involved a considerable skill in planning, organising, and logistic support, and so we see that what today is loosely called "project management" was always an intrinsic part of the engineer's capabilities. Not only that, but in order to exercise these capabilities, the engineer had to be a person of considerable standing and authority, often an official such as a governor or administrator, so that the engineering was often only one part or aspect of his work.

In the period up to about 300 AD, engineering made great progress, mainly in the region around the Mediterranean and the Middle East [3]. Water wheels driving increasingly sophisticated plant for grinding corn and raising water became commonplace, and around 250 BC a piston pump made of bronze, with the piston turned on a lathe, was designed and produced by Ctesibius in Alexandria. Other well-known engineers in this latter part of the period were Philo of Byzantium, Hero of Alexandria, and Vitruvius in Rome (although the latter would be called an architect today).

More complex building structures appeared, supported by technological advances in the production of such construction elements as mortar, bricks, tiles, quarried stone, etc. Outstanding examples of this are the Roman aqueducts and the Coliseum.

A central development in this period was no doubt the increasing skill in mining and processing of iron ore, and in the processing of iron into tools and devices. The process of hardening by carburisation and heat treatment was fully understood and developed in this period, and such tools as saws with set teeth became available.

Throughout this period, engineering was basically a craft, in that it was based on experience handed down from generation to generation; there was little or no scientific basis for the designs and processes. An exception to this was the understanding of the workings of such basic components as the lever, the wheel (or pulley), the wedge, and the screw, based on geometry, and masons used a few geometric figures as the basis for their designs. For example, Archimedes could calculate leverage on this basis.

In the period from 300 to 1100 AD Europe, or the Christian part of the western world, saw a marked decline in intellectual activity, and a falling off of standards in such public services as sanitation and water supply. Paris and London in the year 1100 were worse off than Rome in 100 AD with regard to street maintenance and sewers, and would remain so for several hundred years. But in the part of the world conquered by the Arabs and generally identified as the Islamic world, technology and science had a golden age, and the resurgence that started in Europe around 1100 and led up to the Renaissance owes a great debt to Arab and Persian culture and peoples for preserving and continuing the scientific and engineering tradition from classical times. From northern Mesopotamia to Spain, there was a thriving scientific and technological culture, and much good engineering was carried out to meet the demands of such large cities as Baghdad and Cordoba.

B1.3 The Renaissance

The Renaissance marked a turning point and the beginning of a great increase in the speed of development of engineering and the science on which it depended, and the reasons for this were principally

a. the end of the stifling and autocratic control of education and intellectual activity exercised by the Church throughout the Dark and Middle Ages;
b. the rediscovery of classical science through the contact with Arab culture;
c. the discovery and exploration of new parts of the world (partly as the result of advances in shipbuilding);
d. the development and rapid growth of printing; and
e. the gradual acceptance of the vernacular as a means of written communication, replacing Latin.

The period from 1450 to 1750 saw a proliferation of machinery and equipment to support the demands of a growing and increasingly wealthy population (even if this wealth was very unevenly distributed). Water wheels powered sawmills producing the timber for buildings, ships, and various structures, bellows for smelting furnaces, trip-hammers for forges, and pumps for water supply. Windmills, in use since the start of the millennium, proliferated and were used both for grinding corn and pumping water. The casting and machining of iron produced a variety of equipment, from pumps to cannons, and such "standard" construction elements as shafts, bearings, gear wheels, and pulleys.

Perhaps the most significant feature of this period, in the context of engineering, was the formalisation of the activity of design and its separation from the work involved in implementing it. While the engineer had to have a first-hand understanding of how things were constructed, so that his knowledge was still largely craft-based, the details of the object to be constructed were defined prior to any construction taking place in the form of drawings and written instructions (often as notes on the drawings). The constructed objects became an expression of the intent of the engineer rather than of that of the craftsman; the creativeness involved in an object was transferred from the craftsman to the engineer, and the measure of the craftsman was changed from his ability to create new objects to his skill in producing what the engineer had designed and his ability to do so efficiently. It was the beginning of the process of turning crafts into industries and of the Industrial Revolution, and with it the beginning of engineering as we understand it today.

However, before we go on to considering the Industrial Revolution and what we might call the modern era of engineering, some central features of engineering, both as it was then emerging as a distinct activity and as it is as a profession today, can be illustrated by looking at the work of one man, Leonardo da Vinci [4]. The historical facts are well known: Born in 1452 as the illegitimate son of a notary in the town of Vinci, near Florence, he served an apprenticeship in the workshop of the famous artist Verrocchio in Florence, became an accomplished draftsman, painter, and sculptor, skilled in all the processes and techniques underlying these arts, and became acquainted with science and engineering through interaction with some of the leading practitioners of the day, such as the physicist Toscanelli. At the age of 31, with confidence in his own knowledge and skills in a wide range of fields, including both arts and science, he went to Milan to work for the ruler, the Duke Lodovico Sforza, and stayed there for 16 years. Following the French invasion of Milan in 1499, Leonardo fled to Venice, but in the next 16 years he alternated between Florence, Milan (after the French were driven out again), and Rome, until settling down in France in the service of Francois I for the last three years of his life until his death in 1519.

Leonardo da Vinci is best known as a painter, although only a relatively small part of his efforts were spent on painting and only 15 paintings are definitely attributed to him. By far the greater part of his efforts were directed at science and engineering; he had an insatiable curiosity and desire to understand how things, both mechanical and animate, worked, and his broad and very solid training, coupled with his inventiveness, allowed him to be at the leading edge in a number of areas, including the foundations of painting (as in light and perspective), astronomy, anatomy, mechanics and machinery, structural design (e.g. bridges and domes), and civil construction (e.g. fortifications and canals). He was a universal genius, often referred to as the archetype of the Renaissance Man, but from our perspective the most interesting and significant characteristics of his work are the following:

a. His studies and his understanding were based on observation; form was everything. If he could see how something was made up of its parts, and could describe, in the form of drawings and notes, what those parts looked like and how they fitted together, he was satisfied he understood it. He did very little experimentation, and his grasp of mathematics was not sufficient to allow him to create abstract models or theories. His few ventures into mathematics were confined to geometry, as the theory of form.

b. His approach to engineering was that of an artist. His main objective was to express his ideas and inventions, which he did in the form of brilliant drawings and sketches. Once the problems were solved in his own mind, turning them into working reality, which involved such tedious and time-consuming activities as obtaining funds and organising and supervising the work force, was of much lesser interest, and so most of his projects and inventions were never realised (although this was sometimes also due to circumstances beyond his control). Even documenting his work adequately was seen as a diversion from the constant activity of creation, and many of his valuable studies remained as unfinished collections of notes.

c. To the extent that he was obsessed with form, he neglected functionality. Take as an example his flying machines; none of them exhibited the functionality of actually being able to fly (which could have been demonstrated by simple calculations), but they were elegant and ingenious depictions of how various components would have to be formed and interrelate in order to represent observed characteristics of flight (e.g. the flapping of wings). In short, they were works of art.

Leonardo da Vinci exhibited, to an extraordinary extent, one of the dual characteristics of an engineer; that of having the creative ability to see how technology could be applied and further developed in order to provide solutions to given problems. But he lacked the complementary one of planning, organising, and supervising over the long run the activities required to convert a design concept into a successful reality. Not for him the long slog of placing block upon block over 20 years to build a pyramid. Due to his fame, his work as an engineer is perhaps the most striking example of the dilemma that is inherent in engineering (and engineering education); how to resolve the conflict between creativity and inventiveness on the one hand and the purposeful application of existing knowledge and technology to ensure a cost-effective realisation on the other. Somewhat crudely, what is needed in a successful engineer is a balance between the genius and the plodder, and Leonardo was all genius. As a result, his influence on engineering was almost zero.

The period from about 1550 to 1750 was, as far as engineering is concerned, dominated by three developments; the increasing reliance on reason instead of on belief and superstition, the increasing reliance on experimentation and measurement as the arbiters of theoretical propositions, and, of course, the advances in papermaking and printing, which made books more affordable and greatly accelerated the spread and influence of this new knowledge. Based on the advances in science, exemplified by such names as Simon Stevin, Galileo Galilei, and Isaac Newton, engineering started to develop a substantial body of knowledge and a curriculum for an engineering education. The first engineering schools appeared in France at the end of this period, starting with the artillery school at Fère-en-Tardenois near Aisne in the early part of the 18^{th} century, then the Ecole National des Ponts et Chaussées in 1747, and finally the Ecole Polytechnique in 1794, and a certain degree of specialisation emerged [5]. In particular, mechanical engineering and the design of machinery became distinct from civil engineering.

These developments in the foundations of engineering were accompanied by equally significant developments in their application. New and improved ways of forming and machining materials, as well as advances in mining and metallurgy, resulted in both more sophisticated designs and more efficient production, so that cost-effective machinery and implements found widespread application in factories of all kinds as well as in agriculture and construction. All that was needed for the Industrial Revolution to take off was an efficient source of power (not subjected to the restrictions of wind and water power), and this was provided by James Watt and his inventions, which saw the steam engine transformed from the very inefficient Newcomen engine into its modern form within a couple of decades.

B1.4 The Industrial Revolution

The work of James Watt (1736 – 1819) is the best, and certainly most famous, example of the change that was taking place in engineering at that time [6]. Until then, the engineer had been the creator of individual pieces of equipment or individual constructions, much as an artist created individual works. He either had his own workshop, or he worked closely with workshops that produced his designs. And indeed, this was at first the case with James Watt, who had his own, small workshop attached to the University of Glasgow until he made the first significant improvement, the separate condenser. But then, in partnership with Matthew Boulton, he turned the manufacturing of steam engines into a process which showed all the characteristics of a modern industrial manufacturing process. He broke the plant down into separate components, standardised these components to the extent possible, designed separate processes, tools and jigs for their manufacture, and considered the life-cycle of his products by providing some level of spare parts.

The Industrial Revolution, which is generally considered to have occurred in the period between 1750 and 1900, give or take a few years at either end, contained, besides its enormous social and economic evolution, two developments that were particularly significant for engineering. The first of these was the increased understanding of the properties of material beyond the purely mechanical; in particular chemical and electrical properties, and the exploitation of these properties, through inventions, for practical purposes. Electrical conduction was discovered in 1729, and the first primary cell (an electrochemical process) was developed by Volta in 1799. The telegraph had a rapid rise in the first half of the nineteenth century, with the first transatlantic cable in operation by 1858, and the commercial production of induction motors commenced in Germany around 1890 [7]. Chemical processes, exemplified by the production of alkali for making soap, emerged on an industrial scale [8], with the accompanying necessity for understanding and handling such material properties as corrosion resistance.

The second development was the rise of an industry dedicated to manufacturing engineered products, essentially various forms of machinery and components, which were then used in a second level of manufacturing to produce end user goods, such as textiles. Engineering became a main pillar of the national economy, resulting in a new level of professionalism including design for reliability, maintainability, and constructability, and an increasing level of standardisation. The importance of the latter cannot be overstated; it resulted in a hierarchical structuring of the industry, from the level of materials, such as sheets, rods, and profiles for various metals, through the level of components, from the simplest, such as all types of fasteners, up to more complex ones, such as bearings and valves, and then up to the level of equipment, such as a steam engine or a weaving machine. This development was, in effect, the beginning of systems engineering, and the processes and procedures that made up this structured approach to engineering were expanded, refined, and given a certain intellectual foundation within the academic engineering curriculum, and became an intrinsic part of what was considered good, professional engineering.

A person that straddled this period and the following one, and who in many ways exemplified the transition from craftsman to professional engineer, was Thomas A. Edison (1847 – 1931). With no formal education, his keen, enquiring intellect and his capacity for hard work made him one of the great inventors, and his good business sense ensured that these inventions (there were about 1000 of them) became the cornerstone of an industrial empire. His preferred method was experimentation and trial-and-error; he was good at building apparatus, both for experiments and for production, and his West Orange laboratory was full of every conceivable material collected from all over the world [9]. As a craftsman he was not concerned with the theoretical foundations of his inventions, but he would not give up until they had been developed to the stage where they were useful and cost-effective products, and in this important aspect of engineering he was without peer. As engineers, he and Leonardo da Vinci were complete opposites.

A contemporary of Edison's who epitomised the modern, professional engineer was Gustave Eiffel (1832 – 1923). A product of the excellent French education system, this structural engineer is best known as the creator of the Eiffel Tower in Paris and for his structural design of the Statue of Liberty in New York, but he and his construction company designed and built numerous bridges and buildings all over the world. His designs were the result of state-of-the-art structural calculation methods, and his manufacturing of structural components was so precise that very little adjustment had to be performed when they were assembled on site. His professionalism was also reflected in the attention he gave to workplace management and safety; the Eiffel Tower was constructed without a single fatality. He was a pioneer in advancing the understanding of wind loading on structures, and developed and maintained his own research facilities, including wind tunnels, until his death at 91, but his intellectual curiosity also drove him into exploring other technologies, above all wireless telegraphy, for which he used his tower as a base station [10].

B1.5 The Last Century

The following period, from 1900 to 1950, saw the emergence of technology-based inventions and innovations, such as the vacuum tube, synthetic materials and plastics, and, finally, the semiconductor, and their conversion into mass-produced products at an increasing rate, among these the automobile, the aeroplane, and the radio. Engineering now had two increasingly distinct aspects; the development of technology and the application of this technology. Engineered products were found not only in the industries that produced end products, such as the textile industry, but they were now end products in themselves. In 1850 nobody had a steam engine in their house; by 1950 many people had an internal combustion engine in their garage. And all this despite two Word Wars and the Great Depression. (Or perhaps not despite of, but because of; demonstrating that peace and social stability do not necessarily promote creativity, as Orson Welles remarked, somewhat ungraciously, with regard to the Swiss and the cuckoo clock.)

With a view to our interest in systems engineering, two developments are particularly interesting, In chemical engineering, the concepts of unit processes and unit operations emerged and quickly became accepted as the basis for engineering chemical plants. Complex processes, converting raw materials into finished products, were shown to consist of combinations of the members of a relatively small set of processes; only the values of parameters of each process, such as throughput and temperature, would vary from plant to plant. A plant was viewed as a system of elements arranged in a particular manner, i.e. forming a particular structure. However, it was recognised that despite the simplification this afforded through specialisation and standardisation, maximising the overall plant cost-effectiveness required a holistic view, and this was achieved through a model of the overall process, expressed in terms of an accepted symbolic modelling language. We could therefore say that chemical engineers were the first to embrace systems engineering; they just did not call it that.

The second development came through the rapid spread and general acceptance of the telephone. While the telephone itself is a fine piece of equipment, it is useless on its own; its value lies in being able to be connected to a vast number of other telephones, i.e. in being part of a system. As a result, the theory of networks received a great deal of attention, with such associated subjects as graph theory and queuing theory. Many of these ideas about the flow of information within a network would then be applied to the flow of information within the large defence projects that followed the transition from hot to cold war. The partitioning of the work into a set of work packages, the structuring of this set so as to optimise the probability of a successful outcome, and the use of network concepts and a corresponding graphical language (CPM, PERT, or Gantt chart) are generally seen as the beginning of systems engineering as a separate activity. Because this took place within the defence industry, and was heavily promoted and supported by the US Department of Defense, this early formulation of systems engineering focused on project management within the Department's acquisition framework, and that set the direction of systems engineering development for the next several decades.

This brings us to the final period in the history of engineering, from 1950 until today; a period characterised by the space race and arms race, by the PC and by mobile communications, and by the increasing interaction and overlap between engineering, biology, and genetics, and all of it underpinned by an almost incredible acceleration in the development of new technology. In ever smaller and faster semiconductors and integrated circuits, in electronics at optical wavelengths, in new materials and substances and their manufacturing methods; a list almost without end. But perhaps the most significant feature of this period was the emergence of software and software engineering as a new technology, and today it is claimed that more than half of all engineering hours are spent on software engineering. For various reasons, among them the close connection between software and mathematics, especially formal logic, and the fact that the non-material nature of software seemed to free its practitioners from the previous constraints and discipline of engineering, software development took off in a euphoria of uncritical trust in the capabilities of the individual brain, with "gurus" leading the way into the promised land. Only after numerous failures and the prospect of a crisis with serious economic consequences was it generally realised that software development required a controlled and structured approach just as much as engineering did, and so all the features of professional engineering, such as modularisation, standardisation and reuse, structuring, the use of formal, graphical languages to support the design process, etc. were reinvented by the software community as part of the new discipline of Information Technology (IT), albeit in a format tailored to the non-material nature of software.

For systems engineering, the rise of IT was a mixed blessing. On the one hand, the use of computers with the appropriate software allowed many of the systems engineering processes to become cost-effective for smaller projects and so gain wider acceptance. A significant proportion of the vast effort that went into advancing and structuring IT also benefited the further development of systems engineering, particularly in formalising processes and approaches that, until then, had rested on a heuristic foundation. On the other hand, software engineering came into engineering out of left field, so to speak, with very little connection with the tradition and discipline of engineering, and so, by associating systems engineering with software engineering, systems engineering tended to drift away from mainstream engineering. A recent example of this is the 2008 version of ISO/IEC 15288 [11], which, in my opinion, is a step backward from the 2002 version, as far as systems engineering is concerned.

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

References

1. Gregory, C.E.: A concise history of mining. Pergamon Press, NY (1980)
2. Turner, R., Golden, S.L. (eds.): Great engineers and pioneers in technology
3. Hill, D.R.: A history of engineering in classical and medieval times. Open Court Publishing Company, La Salle, Ill (1984); This well-known and excellent account covers the period from 600 BC to 1450 AD
4. There is a vast literature concerned with the life and work of Leonardo da Vinci, and quite a bit of it is available on the Internet. Perhaps the best starting point is simply Wikipedia, `http://en.wikipedia.org/wiki/Leonardo_da_Vinci`
5. Channell, D.F.: A history of engineering science, Garland, New York (1989)
6. Hills, R.L.: James Watt. Landmark Publications, Ashbourne (2002)
7. Atherton, N.E.: From compass to computer: a history of electrical and electronics engineering. San Francisco Press, CA (1984)
8. Furter, W.F. (ed.): History of chemical engineering. American Chemical Society, Washington, DC (1980)
9. Baldwin, N.: Edison, inventing the century. Hyperion, New York (1995); A discussion of Edison's West Oragne laboratory can be found in Hollister-Short, G., James, F.A.J.L. (eds.) History of technology. Mansell, London (1991)
10. Harvie, D.I.: Eiffel – The Genius Who Reinvented Himself. Sutton Publishing (2004)
11. ISO/IEC 15288, Systems engineering – System life cycle processes, 1st edn. 2002, 2nd edn. 2008

B2 Characteristics of Engineering

B2.1 Role in Society

Emerging out of the short historical overview in the last chapter is a view of engineering that highlights a number of characteristics. The most immediate or obvious is that the *purpose* of engineering is to be *useful* by creating products and services that meet needs expressed by society (or any subset thereof). To that end, engineering uses scientific knowledge, but judges that knowledge not as to whether it represents *truth*, but as to whether it is useful in achieving the purpose of engineering. For example, whereas both the expanding universe and the first law of thermodynamics are true, the value to engineering of the first piece of knowledge is zero, but that of the second very great.

Another central characteristic of engineering is its *creativeness*. In its vision statement, *The Engineer of 2020: Visions of Engineering in the New Century*, the US National Academy of Science opened Chapter 1 with the statement: "Engineering is a profoundly creative process. A most elegant description is that engineering is about design under constraint." This statement is significant in that it distinguishes the creativity of the engineer from that of the artist by introducing the constraint of the competition for limited resources. However, it is also a very limiting statement. Design is not an end in itself; it is only one, albeit very important, part of engineering, and designing something without the intent of having it realised and achieving its purpose is a form of intellectual masturbation.

In the next two chapters we shall focus on these two important characteristics. But before addressing these traditional characteristics, there are a few other aspects of the profession that we need to recognize and understand, They are concerned with how engineering relates to its environment, and as the first one we consider a sometimes contentious aspect of the profession; its wider role in society, in particular with regard to leadership. On the one hand, engineering approaches the needs of society in a value-neutral manner, focusing on developing a clear, mutual understanding of those needs and of the value society places on meeting them. As a professional activity, engineering (as distinct from individual engineers as members of society) does not embody any value judgment beyond what is expressed in the legal framework within which engineering operates. Engineers design and produce weapons that cause enormous suffering, machines that destroy the environment, and factories that produce harmful products, such as cigarettes, but the design methodology and the approach to problem-solving are the same as for any other object. In particular, as I shall argue shortly, the basic objective of engineering is to provide the most cost-effective solution to the design problem, irrespective of the purpose of the project. This adherence to a code of professional performance has analogies in the medical profession. For example, a medical doctor who will obey the rule of not assisting in death, no matter what suffering this may cause the patients or their families. Another example is provided by a doctor's obligation to provide the same professional care to a saint as to a sinner or an evil person. The dispassionate and even-handed approach by engineers to their work is probably a significant contributor to the superficial image of engineers as "grey" and as technocrats best kept working away out of sight.

On the other hand, there is a movement for engineers to take more of a leadership role in business and public policy [1], and a particularly eloquent and interesting case is presented in a recent book by Patricia Galloway [2]. She argues that the leadership position engineering had in the 19^{th} century has been eroded in the 20^{th} century, and engineers are in the process of being viewed as technicians, as commodities. If engineers are to establish themselves as leaders in solving many of the world's most pressing problems and compete successfully in a global workplace, they need to broaden their skills beyond the traditional engineering subjects. The interesting part of the case is that, while her argument for the roles engineers should play and in which they could make substantial contributions is well put, as is the call for additional skills, the reasons why it has not been happening are less clearly explored. In particular, she compares the high esteem in which the medical profession is held by society with that of engineering, failing to note that this comparison would in fact, were it carried out in more depth, undermine some of her assertions about where engineers need to change their view of the profession. The first of these relate to globalisation; engineers need to be able to operate and compete successfully on an international market, taking account of the varying cultural and political circumstances. But is not engineering already more international than medicine? Engineers can practice freely in most countries and do work abroad for part of their career, whereas physicians are constrained by national licensing and competency requirements.

The second assertion is about the need for greater understanding of and involvement in politics; too few engineers move into high political office. But are there more physicians in politics? Isn't the issue rather that the democratic system promotes an intellectual level in the elected leadership commensurate with that of the average of the population? China has a much higher proportion of engineers in the national leadership team, which has correctly perceived that internal stability is their main concern during this catch-up phase of the development of their society (as compared with the developed Western nations), well ahead of democracy.

The third assertion is that for engineering to be accorded the same status by the public as the medical profession, engineers need to broaden their skills in the direction of business, management, and people skills. But do physicians have greater skills in these areas? Physicians are much less involved in management and business than engineers.

No, the reason for the discrepancy in status between the two professions is to be found in a very different direction; in the employment structures of the two professions. The medical profession does not (at least in theory) allow an employment relationship between physicians, as this is perceived as enabling a conflict between professional and business interests. Most physicians work either alone or in partnerships, or then as employees of institutions in which the management should have no influence on the medical standards. In the engineering profession it is exactly the existence of the employment of engineers by other engineers that has led to engineers being turned into commodities, as Galloway rightly perceives is happening. In their roles as engineering and production managers in industry, engineers have been more than willing to turn their fellow engineers into obedient cogs in a production process in order to increase their outputs and profits. And universities and the engineering societies have been willing accomplices in this process, meeting the call from industry for large numbers of readily employable technical workers by lowering their professional standards. The fact is that, at least in Australia, a large proportion of today's engineers work as technicians simply because the education of competent technicians (as exemplified by the German title "Ingenieur HTL") has been abolished for political reasons. The institutes of technology were happy to be converted overnight into universities (why be a teacher when you can be a professor?), and the engineering society gained a lot of fee-paying members.

It is, in a way, ironic that an American should lament this development and call for a change, as it is primarily the Anglo-Saxon world that has been driving that development. In the English language, "engineer" was originally the driver of a steam locomotive, and there was always a fuzzy boundary between the tradesman and the professional. In Europe, the engineer was always well respected and generally in possession of academic qualifications second to no other profession, and most of the captains of industry were engineers. And as for adopting a global view, we can only hope that Galloway's plea will result in US industry finally adopting the international system of units.

The issue of the engineer's place and image in society has been a focus in several books by S.C. Florman [3], and if his very thought-provoking observations can be summarised, it would be that *engineering is a core component of our culture*. It is right here among us, constantly working away at developing and improving the very infrastructure in which our existence as people in a modern society takes place. It is not something on the periphery, and of the various components of a strategy to handle our problems (others being e.g. economics, politics, and religion), engineering is still the most promising. But for all that, it is practically *invisible*. This is really the core of the problem rather than status; the latter would automatically follow if the visibility of engineering were in relation to its contribution.

As an aside, but related to Galloway's comparison of engineering with medicine and law, the two other components of our culture are the rule of law and the value of human life. Lawyers (and judges) are clearly seen as upholding the first of these, and physicians are equally clearly associated by the community with the second. However, the third component, as experienced by the community, is not engineering, but technology. The community is fully aware of the central role technology plays in our way of life and in our culture; what they don't understand is that what they see is the result of engineering employing technology.

There are many reasons for this lack of visibility, among them the lack of business-related subjects in the engineering curriculum, as discussed by Galloway, but one important reason is the many engineers do not want to be visible or the get involved in the business aspects of their projects. Having obtained their degrees, all they want is to practise what they have learnt, earn a reasonable salary, maintain the respect of their fellow engineers, and be promoted on the merit of their technical work alone. They form what Florman calls “the rank and file of engineering”, and this should not be seen as in any way a derogatory classification. They form a very valuable and effective component of the workforce of a modern society, and instead of harping on their lack of business skills and leadership qualities and diluting their education with subjects that will be of little value to them, they should be accepted for what they are; highly skilled technical specialists performing very demanding work. The engineering profession, and engineering education in particular, has to come to terms with the fact that as the driver of a rapidly advancing technology that is having an increasingly important influence on all aspects of our society, engineering itself has to change. A long time ago it was recognised that there had to be a subdivision into disciplines based on the area of science that formed their basis; now it is time to recognise that, in addition to the degree of specialisation, there has to be a further dimension to the characterisation of an engineer, the depth of, and degree of involvement with, technical knowledge, and that these two dimensions are necessarily related. As impossible as it is to have an engineer specialised in fracture mechanics to also be a specialist in semiconductor device design, it is to have such a specialist also be a leader in guiding and applying technology and providing the interface to the stakeholders and society in general.

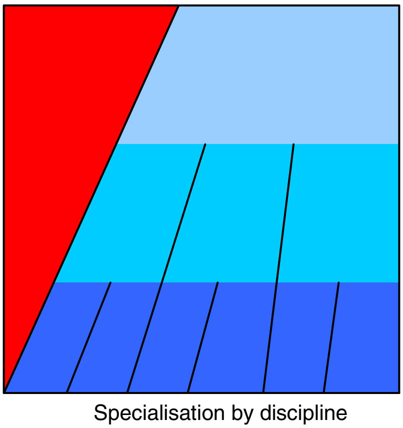


Fig. B2.1 A diagrammatic illustration of the two coupled aspects of the engineering profession. The vertical axis indicates decreasing depth of knowledge and level of detail; the horizontal direction indicates the degree of specialisation, i.e. the subdivision into disciplines. The red triangle represents the non-engineering, or business-related, knowledge.

Figure B2.1 is an attempt to illustrate these two coupled aspects of the profession, and by structuring it in this fashion the desired improvement in visibility and effectiveness could be attained. Rather than all engineers playing the same role with regard to interacting with the rest of the community, this is a graduated role (as represented by the red triangle in Fig. B2.1); for some it is a negligible role, for some it is a major role, and it is the responsibility of the latter to ensure that engineering has the appropriate visibility.

Of course, this requires a restructuring of engineering education, with the curriculum being flexible enough to cover the whole area in Fig. B2.1. One approach to designing such a curriculum is to consider it as a system, with the individual subjects being the elements and the resultant capability of the individual engineers being the emergent property of the education system. Each subject has to be designed with regard not only to its own, local subject matter, but also to its position in the curriculum structure, i.e. with regard to what other subjects it interacts with, so as to provide the required interaction. The essential point here is to recognise that the effect of a particular subject on the educational outcome for any one student will depend on what other subjects are contained in the overall program for that student; the subject will reinforce aspects of previous subjects and provide a basis for understanding aspects of subsequent subjects. Each point in the area of Fig. B2.1 is covered by a particular combination of subjects in a particular order, but the interaction is more detailed than just the ordering. For example, a mathematics subject will, of course, be taken in the first semester, before the students need that knowledge in later subjects, but the examples chosen will relate to those subjects, the problem-solving methodology used (if not explicitly) will be the one to be developed throughout the whole course, and so on.

B2.2 Code of Ethics

Every engineering society seems compelled to have a code of ethics, and ethical behaviour is considered to be an important characteristic of engineering. But is there really a separate code of ethics for engineers, different to one that applies to every human being? An answer to this question is more easily found if we rephrase it and ask "Is there anything that should be added to what society considers to be the normal code of ethics for it to adequately cover the profession of engineering?", in which case the answer is "yes". As with any profession, its members possess specialised knowledge, and it would be unethical to withhold that knowledge if doing so would be harmful to society. This is discussed in the book by S.H. Unger, *Controlling Technology: Ethics and the Responsible Engineer* [4], in the context of the problem of democratically controlling technology for the benefit of humanity, and he argues that this cannot be accomplished entirely from outside the profession. Uninformed control, even with the best of intentions and based on valid general principles, can lead to highly undesirable outcomes, and engineers therefore have a duty to play an active role in directing the development and application of technology by providing complete, accurate, and understandable information to all affected parties.

There is probably not a great deal of disagreement about the general content of a code of ethics for engineers; as always, the devil is in the detail [5]. First of all, decisions about the development and application of technology are seldom black and white; there is usually a balance between benefit and cost (in all its guises), and there is always an element of uncertainty or *risk*. Even though we are subject to risks of various sorts every day, it is a concept society has great problems coming to terms with, both with regard to basic nature (which requires some understanding of probability theory) and with regard to what is an acceptable value in a given case (or even the fact that there is a finite value). Secondly, engineering takes place within a commercial framework (possibly with the exception of engineering within public bodies), so that the livelihoods and careers of engineers are tied to the commercial success of the companies they work for, and that success may be dependent on many factors besides engineering. There can therefore be considerable pressure on engineers to provide a lower risk estimate than they would otherwise do. But, perhaps more frequently, it may not even be a question of a risk to humans or the environment; it is purely a question of commercial risk, and this creates a real dilemma.

On the one hand, if commercial imperatives dictate the necessity for substandard engineering and thereby a significant risk of the product not meeting its required performance, should the engineers walk away from it, or should they go along with it after expressing their assessment of the situation and disassociating themselves from the commercial aspects? And, in the latter case, to what extent should they go public with their assessment, it this might in itself harm the prospect of success? A hypothetical, but not unrealistic case is the following: A small company, with little engineering expertise, has bought the rights to commercialise the patent for a particular process, and they engage a highly reputable engineering firm to provide the engineering services required on a time and expenses basis. As the company has few reserves and a modest balance sheet, they look for investors that would become partners and fund the development of the patent into a commercially viable process, but this proves unsuccessful. Then an opportunity arises to form a joint venture with a manufacturing company to build a plant utilising the process. With that contract in hand as proof of the viability of the process, the company could go the market and raise funds to finance the engineering and its part of the construction cost, and in that manner practically bootstrap itself into a major operation. However, a condition of the joint venture partner is that construction of the plant must start within twelve months. The engineer explains to the company that in order to meet that deadline they would have to compress the normal process development and plant design processes by making a large number of assumptions and also reducing the review and quality assurance processes to a minimum, making it a very risky proposition from the point of view of the performance of the plant. The company instructs the engineer to go ahead and cut whatever corners are necessary to meet the deadline, staying, of course, within the law and all applicable standards. The company also offers the engineer a sizeable bonus for completing on time.

What should the engineer do? Walking away would mean loosing a very profitable job and, possibly, a long-term relationship client with ongoing work. The engineer could accept the assignment, on the condition that its assessment of the risks is acknowledged in writing by the company, but then what about the other parties involved? Clearly, the joint venture partner does not realise the extent of the risk and the immaturity of the process, and the investors would not have a clue, but alerting either of them would scuttle the whole plan. We could say that they should be commercially astute enough to look after themselves and, if necessary, get independent technical advice, but perhaps they consider that the involvement of the highly reputable engineering firm is assurance enough. A failure of the venture would certainly not do the reputation of the engineering firm any good, no matter how well protected it might be in a strictly legal sense.

The problem is that business operates under somewhat different rules within its own community than it does (or should do) towards the community at large. For example, duty of care towards the community requires that a company takes active steps to ensure that the community understands the implications of using its products, whereas when the company sells one of it plants to a competitor, it only has to make all the data available; it is up to the buyer to do a due diligence on the data. The seller is not obliged to make the buyer aware of any problems. So, to what extent should engineering be part of this business process and conform to the same rules? There is clearly a potential conflict here between the loyalty toward the employer or client and the responsibility to the community, and some would see engineering stand above any business considerations in its duty to "provide complete, accurate, and understandable information to all affected parties", as stated above.

On the other hand, many of us feel that it is only by becoming more involved in the business aspects of the enterprises to which our work relates that engineering can fulfil its potential as a major player in shaping our future, as reflected in the two works, by Galloway and Florman, cited above. And indeed, as the short historical overview in the previous chapter hopefully demonstrated, engineers have traditionally been heavily involved in business; it is only in the last few decades that this involvement has become an issue, as discussed in the previous section. In my opinion, any attempt to use a code of ethics to set the engineer aside from the world of business is misplaced, will have little or no effect on the position of engineers in society, and is diverting attention away from the important aspects of the issue.

B2.3 The Process of Engineering

A whole complex of characteristics is contained in the manner in which engineers work; the *process of engineering*. The starting point of this process is a set of stakeholder (or user) requirements. The process converts that set into a complete physical description of the engineered object, which is then implemented and put into operation, and through its operation it provides a service which fulfils the user requirements to a greater or lesser extent. It is the *intent* of the engineer that the service produced by the system will fulfil 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 engineering process, as illustrated in Fig. B2.2.

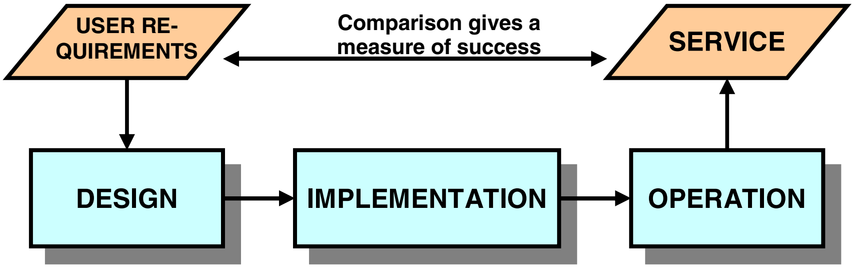


Fig. B2.2 How design fits into the process of engineering.

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

For engineering to be useful, the products and services (which we shall include under products) created by engineers (and, to a large extent, also by architects) must display a number of aspects that are quite different in their natures, and it is only by having a thorough understanding of these aspects and their interrelations that we can obtain a holistic view of the profession and then, in the third part of this book, see how the system concept can be brought to bear in handling the complexity involved. The first and most obvious aspect is that the products must do what the users or, more broadly, the stakeholders require. That is, engineering is always in response to stakeholder requirements, even though in some cases these may not have been explicitly stated. For example, when Edison developed the electric light bulb, there was no requirements specification for such a device and no stakeholder group requesting its development; there was a demand for a light source, which at the time was fulfilled by gas and kerosene lamps. It was

Edison the inventor that perceived how that demand could be met in a new and much improved manner by the electric light bulb, and Edison the engineer that solved all the practical problems that made the light bulb a commercial success. So, we recognise that invention is only a part of engineering if it is in response to a need, explicit or implicit; otherwise it might more properly be considered tinkering.

However, there is a more fundamental issue involved in meeting stakeholder requirements. In most cases, there will be a degree of contradiction, tension, and incompatibility between individual stakeholder requirements. Not only the obvious conflict between cost and performance, but between the project in the narrower sense of an investment opportunity and the rest of the community, and in this regard the role and responsibility of the engineer is a much more controversial one. To what extent is the engineer responsible for ensuring that all stakeholder requirements are considered in a balanced manner, and what is the definition of "a balanced manner"? With regard to the first of these two questions, the trend is definitely toward greater accountability for the engineer; hiding behind "the Client only requested me to look at these aspects" is getting more difficult and unacceptable. The situation is evolving towards that encountered in accounting, where it is not acceptable for an auditor to say "the Client only wanted me to look at this part of the accounts", and as a result there is both an additional element of risk and an increase in complexity (measured by the number of issues to be considered simultaneously) involved in the requirements definition part of the process. The complexity will be addressed in the third part of the book; the risk aspect has led to an involvement of lawyers in the engineering process that would have been unthinkable a couple of decades ago.

The view of the process of engineering provided by Fig. B2.2 demonstrates that design needs to consider not only the object to be designed as such, but also the operation of that object through its lifetime and the environment in which it will operate, and we shall look more closely at what this means for the design activity in Chapter B4. However, while design is the core activity of engineering and the one in which the system concept plays its most prominent role, engineering includes a number of other activities that are required in order to see a an engineered object through its *life cycle stages.* There is no universally agreed definition of how the life cycle of such an object is divided into stages, but a common one is the following [6]:

a) Concept Stage
b) Development Stage
c) Production Stage
d) Utilisation and Support Stage
e) Retirement Stage

Throughout the life cycle a number of activities or *processes* are carried out [7], of which the main groups are:

a) Design
b) Estimating

c) Production
d) Procurement
e) Operation
f) Maintenance
g) Logistic Support
h) Design Management
i) Project Management

Each of these processes is again a project that produces a service and goes through its own life cycle, and each one needs to be designed, optimised, implemented, and sustained. There are obviously functional relationships between these processes, in the form of their services flowing between them, but in addition there is also a temporal dimension to these relationships. One of the benefits of systems engineering, which sees these processes as interacting elements in an overall system, is the realisation that, once these interactions are properly understood, many of the activities can be carried out concurrently. For example, both the maintenance process and the logistic support process can be (and should be) designed concurrently with the main object.

The extent and detailed nature of each of these processes vary from stage to stage, but most importantly, they also vary depending on the contracting strategy that is adopted throughout the life cycle. The totality of the work to be carried out is almost always subdivided into a number of contracts that are assigned to one or more companies (or divisions within the same company), and the work required to execute one such contract is what is termed a *project*, and that is the relevance of this term in the Project Management process. In this view, the process of engineering consists of a number of projects, and project management is the overarching management of all the other activities within each project.

To demonstrate how this works, and how it influences the role of design within a project, consider a particular project, such as the creation of a new production facility. That is, in the context of the system life cycle, as defined above, the project encompasses only the concept, development, and production stages. These are considered to consist of a number of sequential phases, with the completion of each phase constitutes a major milestone or gate, and within each phase one can subdivide the work further, typically into planning, execution, and control. The number of phases and their description depend somewhat on the particular project delivery methodology adopted; a fairly common one is the following:

Concept Stage	Business case development Concept development Feasibility study
Development Stage	Project planning Preliminary design Detailed design

Production Stage	Procurement Construction and integration Testing and commissioning

Let us now assume that this project has been let by the Owner as a single turn-key contract to company X, then it is possible, but highly unlikely, that X would do all the work in house. It is much more likely that X will subcontract parts of the work to other companies, so that each company now has a project of its own, and let us first examine the following scenario:

> X is a developer who looks after the financing and stakeholder management, but otherwise has no resources of its own.
> Y is given the subcontract to provide overall project management on behalf of X.
> Z is given the subcontract to carry out all the work in the Concept Stage.
> U is given the subcontract to carry out all the design work in the Development Stage and provide design support during the Production Stage.
> V is given the subcontract to carry out all the construction in the Production Stage, including testing and commissioning.
> W is given the subcontract for procurement of all equipment and services not covered by the above subcontracts (e.g. production equipment to be installed in the facility).

In this case, the work in the two contracts X and Y falls entirely into the group of project management processes. The work in contract Z, which is limited to a single stage, consists mainly of design and estimating processes, with some procurement, logistic support, and engineering management work and, of course, an appropriate amount of project management. Contract U spans two stages and is almost entirely design and design management, with a small amount of project management, and contract V, which is limited to a single stage, consists of production processes and associated project management (which here is called construction management). Finally, contract W spans the development and production stages and consists of procurement processes and associated project management.

Consider now a second scenario, where the Owner first lets one contract, Contract X, for the EPCM (Engineering, Procurement, and Construction Management) of the facility. During the development stage the EPCM contractor then lets, on behalf of the Owner, one or more contracts for the equipment and services required in the production stage. In this case, Contract X spans all three stages and contains all the processes, including some of the production processes.

And finally, a third scenario is that the Owner lets a contract, Contract X, for most of the work in the concept stage (i.e. similar to contract Z in the first scenario, but without the procurement processes) and then, based on that concept design, Contract Y for all the work in the development and production stages, a so-called Design and Construct (D&C) contract. The contractor then usually lets a subcontract, Contract Z, for the detailed design.

What would cause the Owner to choose one or the other of these scenarios? Firstly, the skills and experience in the Owner's organisation; the first two scenarios do not require the Owner to have any experience in creating new facilities. Secondly, the degree to which the Owner's requirements for the new facility are finalised; the first scenario requires these requirements to be completely finalised and the Owner's scope for influencing these as the overall project progresses is severely limited by the high cost of variations. The second scenario provides considerable flexibility in the interaction between Owner and EPCM contractor throughout the overall project, and in the third scenario the Owner is in complete control until the concept design is finalised and can have any degree of involvement in the development of that design, but has very limited opportunity for change after that. And thirdly, the Owner's perception of the capabilities and competitiveness of the different players; for example, if an appropriate contractor is available, a D&C contract can be the most cost-effective way to go.

For our purposes, and in particular the discussion of design in Chapter B4, the important observation to be made with regard to these three scenarios is that the design processes are quite different in the various projects. The differences are summarised in Table B2.1 below.

Table B2.1 Design processes for three different contracting strategies (scenarios).

Sce	Con	Sta	Input	Main Characteristic	Output
1	Z	C	Fully developed stakeholder requirements	Meeting requirements at lowest cost	A fully developed, stand-alone concept design as a contract deliverable
1	U	D	Concept design and stakeholder requirements	Review concept design and respond to Contractor Y	A fully detailed design, ready for construction
1	U	P	Detailed design	Maintain balance between Contractors Y, V, and W, while defending the detailed design	Realisation of the designer's intent
2	X	C	Initial stakeholder requirements	Finalising requirements while optimising cost-effectiveness	Concept design as an internal interface to detailed design

Table B2.1 (*Continued*)

2	X	D	Concept design	Continue to optimise cost-effectiveness while liaising with suppliers	Detailed design as integrated part of RFT
2	X	P	Detailed design and construction contractor's tender	Work closely with contractor(s) to optimise cost-effectiveness	Successfully operating facility
3	X	C	Initial stakeholder requirements	Finalising requirements while optimising cost-effectiveness	A fully developed, stand-alone concept design as a contract deliverable
3	Z	D	Concept design	Minimise cost by close attention to construction methodology	Detailed design as internal interface to construction
3	Z	P	Detailed design	Minimise construction cost by adjusting design	Achieve completion certification

Table B2.1 show that not only are the design processes different, as indicated in the column "Main Characteristics", but the interfaces between the processes and their environments are also quite different, as indicated in the columns "Input" and "Output". Much of the complexity in design comes from these interfaces, and how we handle this complexity therefore depends on which particular process we are considering. We shall return to this issue in Chapter C2, when we discuss how systems engineering has been developed largely within a particular contracting strategy, the acquisition strategy of the US Department of Defense, and how this has been a barrier to applying the systems approach in non-defence industries.

B2.4 First Step towards an Engineering Ontology

Over the last two decades there has been a significant increase in publications on the subject of ontology [8]. But this use of the word is somewhat different to its traditional use in philosophy, where it is concerned with the question of existence and what exists; the current interest is about explicit specification of conceptualisation, about the vocabulary we can use to speak about a particular domain of interest. Or, conversely, a given ontology defines the objects that can be represented by its concepts, the universe of discourse defined by that ontology.

The rise in interest in ontologies and, indeed, the emergence of ontologist as a distinct profession [9], appears to be driven by two main factors. One is the obvious fact that natural language is not semantically definite; the meanings of words and sentences depend on the context in which they are used. In particular, they depend on the area of professional specialisation, and with the rapid increase in knowledge and consequent increasing specialisation, there is an accompanying need for correspondingly specialised sub-languages.

The other factor is the desire to make language machine-processable; to allow a computer to "understand" the meaning of a text and thereby both to enable a much richer interaction between humans and machines, and to exploit the capabilities of the computer to store and search for information. An example of where this factor is driving developments is the Semantic Web and the OWL Web Ontology Language [10].

The developments motivated by both of these factors has a common base in linguistics, philosophy, and logic, and a body of knowledge is developing that underpins the emerging professional specialisation of ontologist. Despite this, it should not be overlooked that there is a significant difference between the two factors, which is best seen by introducing the concept of ontological commitment, the agreement to use a shared vocabulary and rules of grammar in the communication between *agents*. Then, the first factor is primarily concerned with humans as the agents, whereas the second is primarily concerned with humans and computers as the agents. This difference will be important in the following development, as our focus will be on the first.

We start our development by defining the domain and scope of the ontology, which means answering some basic questions [11]:

What is the domain that the ontology will cover?
What is the ontology going to be used for?
For what types of questions should the information in the knowledge base provide answers?
Who will use and maintain the ontology?

The answer to a) is obviously "engineering", so we must examine the concepts involved in our understanding of engineering.

As to question b), we assume the answer to be "to provide a common vocabulary for communications about engineering", thus encompassing both communications within the engineering community as well as communications between engineering professionals and the wider community. For communications strictly within the engineering community, other, more specialised ontologies and models may be appropriate, as was discussed in a recent paper [12].

Answers to question c) will emerge as the ontology is develops, but basically the questions are variations on the question "How do I apply engineering to this project?". And the answer to question d) is simply "the engineering community" and one or more organisations within that community.

In developing the engineering ontology, we need to adopt a top-level categorisation as a starting point. A number of such top-level categorisations have been put forward, starting with Aristotle's *Metaphysics*, [13], and some of the most discussed ones today are Sowa's Diamond [*op.cit*], the IEEE's Standard Upper Ontology (SUO) [14], OpenCyc Upper Ontology from Cycorp Inc. [15], and Basic Formal Ontology (BFO) [16]. A common top-level categorisation of the things that exist in reality is shown in Fig.B2.3; this ontological sextet is from [17], but with the formal-ontological relations amended by adding the relations between properties and processes, making properties symmetrical with regard to substances and processes.

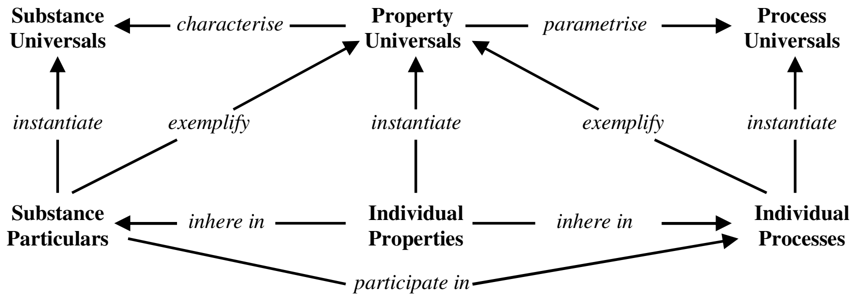


Fig. B2.3 The ontological sextet and the formal-ontological relations.

However, as it stands, this categorisation is not entirely suited as a basis of an engineering ontology; the problem being that the concept of "substance" must go beyond the narrow meaning of something that has mass. To understand what this wider meaning is, we need to take a step back and consider an ontological structure that, at its highest level, has only two categories, "continuants" and "occurrents". A continuant is anything that exists *as a whole* at any point in time when it exists, as opposed to an "occurrent", which only exists as a whole over a period of time. The category of "occurrents" includes what we shall call "processes". The category of "continuants" can be divided into the subcategories of "independent continuants", and "dependent continuants", and the latter category includes what we shall call "properties" and "descriptions".

The issue that now arises is that engineering deals with "independent continuants" that take on different forms in different parts of the engineering process (see later). For example, we can talk about a crankshaft *in general* (as part of an interface between linear and circular motions), we can write a specification for a *specific* crankshaft, and we can design a crankshaft that meets the reqirements of that specification, all without there being any physical entity that is a crankshaft. And then there is the physical realisation of the design; a real crankshaft. Somehow, the *concept* of a crankshaft must encompass all of these forms.

Another example is the concept of stakeholders; we can talk about stakeholders in general, we can specify the stakeholders (set requirements on what type of persons should be included), we can design (choose) a set of stakeholders that meet the specification, and finally, the actual stakeholders, as a group of persons, may appear at a meeting.

To handle this, we shall use the word "objects" as the top level category of "independent continuants", and assign to objects the property "form", which can take on the four values "inherent", "specified", "designed", and "realised". As a result, our top-level categorisation is as shown in Fig. B2.4.

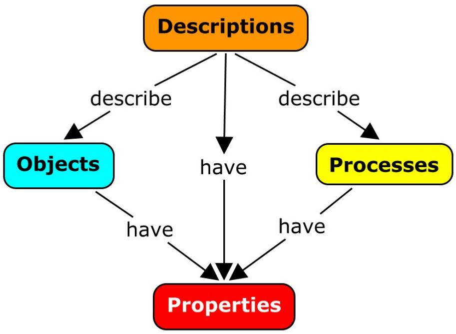


Fig. B2.4 The four top-level categories of the engineering ontology.

What is "engineering"? What do we understand by this word? A definition, such as "the discipline, art and profession of acquiring and applying technical, scientific, and mathematical knowledge to design and implement materials, structures, machines, devices, systems, and processes that safely realize a desired objective or invention" [18] is too general to provide a point of departure for developing a deeper understanding of what engineering is. In particular, in order to develop a common vocabulary, we need to understand what engineering *is about*, in the sense of the *things* we speak about in engineering. The totality of these things is what is used to express the engineering Body of Knowledge; it is obviously going to be a very large and complex vocabulary, and we will need some form of systematic approach in order to develop it. The approach used here is the same used in developing a system description of a complex object; a step-wise, top-down development from the general to the specific, with an increasing degree of detail in subsequent levels and defined relationships in each step.

The starting point is to distinguish a sub-category or *class* of processes, which might be called the class of professional projects; an instantiation of this class is defined as follows:

a. It is performed by people (the *practitioners*)

b. It has a *purpose* defined by a group of people (the *stakeholders*)

c. It is performed within a *timeframe*, starting with the definition of the purpose and ending when either the purpose is deemed to have been achieved or the attempt to achieve it is abandoned.

d. It has a *resource base*, from which the resources required to achieve the purpose is extracted.

e. It has a *knowledge base*, from which the knowledge of how to apply the resources is extracted.

Many instances of processes do not fall within this class, such as the change of seasons, erosion, and the processes taking place within stars, but equally it includes a wide range of processes outside of engineering, such as medicine, dentistry, and architecture. The process of engineering is a subset of this class, distinguished in part by the nature of the resource base and that of the knowledge base, in part by tradition, but that does not need to influence our development of an engineering ontology. If the ontology serves our purpose of providing the basis for a common vocabulary within the engineering profession, that is all that matters. That it may also apply to other professions or be useful for them is irrelevant.

The practitioners of the process of engineering are the *engineers*.

An instantiation of the process is a *project*, and, by considering the purpose, it is possible to distinguish two broad classes of engineering projects,

- projects that utilise the existing resource and knowledge bases (often combined under the concept of *technology*) to meet a *need* expressed by all or a part of society; and
- projects that increase the resource and knowledge bases.

Or, in other words, projects in the first group *apply* technology in order to meet requirements imposed by entities or people who are generally not engineers, and it is these *stakeholders* that are the judges of project success; whereas projects in the second group *develop* technology using that part of the knowledge base that is provided by science, and their success is judged generally by other engineers. Let us agree to call these two groups of engineering projects *application projects* and *development projects*, respectively. There is not a sharp boundary between these two groups, and there will be many projects that contain sub-projects of both types, but because the application of the system concept is much more important in application projects, we shall focus our attention on this class only, as was already implied by the description of the process of engineering in the previous section.

However, before doing that, it is appropriate to note that this distinction of the two types of engineering projects has a significant influence on some of the characteristics we discussed, such as the role in society and the code of ethics. That arises out of the fact that development projects are, to an extent, shielded from direct interaction with society outside of the project; the group of direct stakeholders is very limited. As a result, engineers in development projects have less cause to be concerned about the wider implications of their work and are, in this regard, closer to scientists.

As we shall see shortly, the stakeholders contain a number of distinguishable entities or groups of people, but for all of them, their requirements on a project are of two types. One, requirements on what the project must *do* for them, and two, requirements on the physical characteristics of what is created by the project, i.e. what the result must *be*, such as the colour of products. These latter requirements require no further engineering (although, as boundary conditions, they may influence the engineering), so in the following we will only explicitly consider the former, the *functional* requirements. (In Sec. B4.2 we will come back to the types of requirements.)

With the above understanding, we can describe the process of engineering as the creation of an object that, during its operational lifetime, provides a *service* that meets the need. The object is a physical entity, but otherwise not restricted, and typical examples are a factory, a car, a public transport system, a bank, a communications system, and a bridge. The service is correspondingly unrestricted, for the above examples it would be to provide a product, mobility, information exchange, and overcoming a natural obstacle (river, gully, etc.). In terms of the ontology sextet in Fig. B2.3, the object is a substance, and as engineering involves a large number of different things that would fall into this main category, we shall call the object that provides the service the *plant*. (This is not an ideal name for something that could consist mainly of a group of people or of a single device, but I have not found anything more appropriate.)

This distinction between the service and the plant that provides it is essential; the outcome of the project is judged by the stakeholders on the service provided; the plant is the engineer's solution to providing the service, and obviously there can be many different plants that provide the same service. The distinction also immediately subdivides the timeframe of projects of the first group (application projects) into two main *stages*, the *creation stage* and the *operation stage*, and together they form the first level of subdividing the timeframe into a *life cycle*.

We can now also identify subsets of the stakeholders. Firstly, the need expresses the requirements of a subset of the stakeholders, the *users*, and the expression of the need as a set of requirements is therefore often called the *user requirements definition* or simply the *user requirements*.

Secondly, there is one further group that is always present, as the following argument shows: The creation of the plant and its operation must require an expenditure of resources; the value of these resources, referred to the end of the creation phase, is the *cost* of the project. And because some of this expenditure occurs before there can be any revenue from the service, it is in the form of an *investment*. Therefore, there is always a group of the stakeholders that provide this investment, the *investors*, and they do so in the expectation of getting a *return* on this investment. It occurs in the form of a *revenue* during the operating phase, referred also to the end of the creation phase (or the beginning of the operating phase). The revenue results from the users paying for the service, and is determined by what *value* the users put on having their requirements fulfilled and by the degree to which the service fulfils them, which is a measure of the plant's *performance*.

It is important to emphasize that the concepts of cost and revenue, as they are used in this argument, need to be understood in a very generalised way. The resources expended may range from volunteers donating their time to the expenditure of natural resources or a reduction in the quality of life, and the revenue may range from personal satisfaction to national security. But no matter who provides the investment, they will always want to maximise their return on it, and so if we abstract from all the particulars of projects and thereby generalise the purpose of the project more and more, *maximising the return on investment (ROI) is the common purpose of all engineering projects*.

Or, in other words, we view engineering projects generally as investment opportunities; the need defines the opportunity for a particular project. The return on investment will be denoted by **U**.

Finally, the stakeholders will normally encompass a much wider group of people, such as government bodies, special interest groups, and others affected by the service without actually using it; they may all be grouped under the umbrella of the *community*. The corresponding requirements take on a variety of specific forms, such as protecting the environment, promoting social justice, supporting political stability, and advancing economic development, but in order to treat them within the process of engineering, they must all be associated with a corresponding cost or revenue. Consequently, their requirements can be subsumed under the requirement to maximise **U**.

In the above, we have defined a number of new concepts; some of them are included in Table B2.2 below.

Table B2.2 Definitions of some central concepts in engineering.

Name	Definition	Synonyms/Examples
Project	An instantiation of engineering.	job; endeavour
Timeframe	The duration of a project	
Purpose	A description of the intended outcome of a particular project.	goal
Stakeholders	The set of entities or people that determine or influence the purpose.	project sponsors, system end users, community groups
Need	The subset of the purpose defined by the users	
Users	The subset of the stakeholders that provide the revenue	market
User requirements	The expression of the need as a set of requirements.	requirements definition

Table B2.2 (*Continued*)

Investors	The subset of the stakeholders that provides the finance for the project.	
Community	The subset of the stakeholders that are not directly involved in project activities, but that influence or are influenced by the project.	
Service	The activity that is intended to meet the need, and results from operating the plant.	providing a product; providing a sensation; entertainment;
Plant	The physical object that results from an engineering project and that provides the service.	product; infrastructure; facility; organisation; team; equipment; device
Life cycle	The subdivision of the timeframe.	
Creation stage	The first stage of a two-stage life cycle, starting with the first effort (cost) attributable to the project, and ending when the physical object starts to provide the service.	development stage
Operating stage	The second stage of a two-stage life cycle, starting when the physical object starts to provide the service, and ends with when no further effort (cost) is attributable to the project.	production stage
Resource base	The resources engineers can draw on to perform engineering.	materials; components; labour; facilities; tools
Knowledge base	For engineering, the accumulated knowledge of the engineering community. For a project, that part of the knowledge relevant to performing the project.	Body of Knowledge; standards; textbooks; publications
Technology	The combination of the resource base and the knowledge base.	
Development project	A project whose purpose is to add to the existing technology	applied research; experimentation; investigation

Table B2.2 (*Continued*)

Application project	A project whose purpose is to apply existing technology to meet a need.	
Artefact	An item produced directly by engineers as part of a project as it goes through its various stages.	document; drawing; specification; model; data

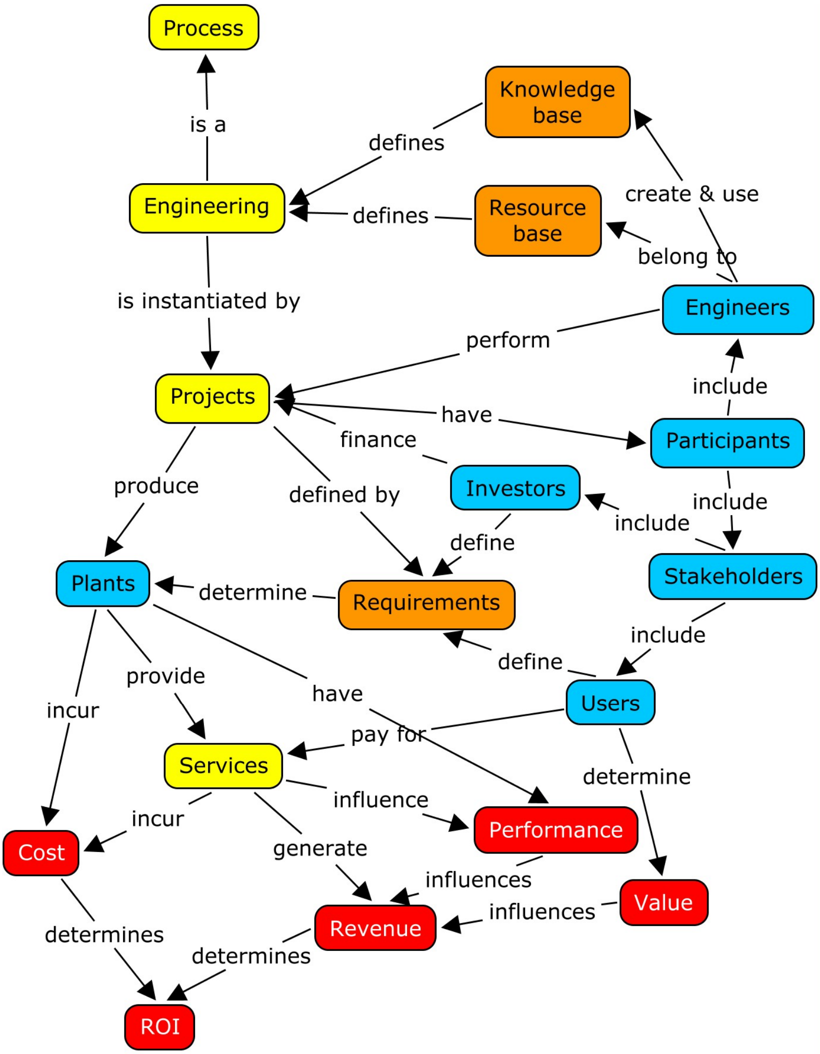


Fig. B2.5 Map of the concepts involved in a high-level view of engineering and the relationships between them [15]. With reference to Fig. B2.4, objects are coloured blue, processes yellow, descriptions tan, and properties red.

The above view of engineering and, in particular, of application projects, can be expressed in a concept map, as shown in Fig. B2.5 on the previous page.

The concept of a project's life cycle, with its constituent stages, as introduced in the previous section, appears throughout the engineering literature, but it is important to understand that it is being used in two different ways and for two different purposes. The first and most immediate is the temporal subdivision of the project, with the stages following each other in a sequential fashion and with *gates* forming defined transitions from one stage to the next. The purpose of imposing this structure on the project is to improve its *management*, and the definition of the stages and their gates are adjusted to suit aspects of the management approach, such as the contracting strategy. In particular, the gate between the creation stage and the operation stage becomes a matter of definition for each project.

The second use of the concept is to provide a high-level structure to the set of processes involved in engineering projects, and it is this use that will be of most interest to us in developing an ontology. The stages are then classes of these processes, which we shall call *activities*, with the temporal aspect only a subordinate feature of these processes and their relationships, and as these classes will feature in our ontology, we will have to agree on this further subdivision of processes:

- Processes
  - Projects
    - Development projects
    - Application projects
  - Activities
    - Technical activities
      - Design
      - Implementation
      - Operation
      - Maintenance
    - Support activities
      - Management
      - Community consultation
      - Procurement
      - Etc.

The four core engineering activities, which are also called *technical processes* [20], are defined in Table B2.2.

Table B2.2 The four basic classes of technical activities forming the life cycle of a project.

Name	Definition	Synonyms/Examples
Design	The process of converting a set of stakeholder requirements into the information package that will allow the plant to be produced and operated successfully throughout its lifetime.	requirements elicitation and definition; exploratory and feasibility studies; concept design; preliminary design; detailed design. (A very good treatment of engineering design is given in [21], albeit with the scope of the illustration limited to mechanical engineering.)
Implementation	The process of converting a design information package into a plant.	production, manufacturing, construction, testing, and commissioning.
Operation	The process of realising the ability of a plant to produce its intended service.	materials handling; processing.
Maintenance	The process of maintaining the plant in a state where it provides its intended service in the most cost-effective manner.	preventive, corrective, and adaptive maintenance; decommissioning and disposal; and through-life support (e.g. training, spare parts provisioning, etc.).

In addition to these four classes of activities, there is another set of classes of activities that operate throughout any engineering project, and some of the major *supporting activities* are indicated above. However, while these are of great importance to the outcome of engineering projects, they are not specific to engineering projects, and their ontology, in the sense of a structured, common vocabulary, should take this wider user community into account.

B2.5 Two Additional Characteristics of Engineering

I would like to conclude this brief (and selective) discussion about characteristics of engineering by considering two characteristics (or, perhaps better, aspects) that, while they have no direct consequences for the application of the system concept to engineering, as set out in Part C, have significant indirect influence on how we formulate that application and how it is perceived. And they are both, in turn, strongly influenced by some of the characteristics we have discussed; in particular, the specialisation of engineering into disciplines and the embedding of engineering in projects as what we called the process of engineering.

The first of these is engineering as a social activity. This is not the role of engineering in society, as discussed in the first section of this chapter, but the social aspects arising out of the interactions between the participants within engineering projects and also between members of the engineering community in general. Any profession relies on the interchange of ideas and information between its members, in the form of meetings, conferences, and publications, and is defined by an accepted framework for its activities and a shared Body of Knowledge on which work within this framework is based. So, *a priori* there are grounds for considering a profession as a social activity. Furthermore, viewing a profession as a social activity is not without its precedents; in particular, there has been a considerable amount of work published on science as a social activity [22]. We would expect that many of the same considerations would apply to engineering, and a useful approach is to compare engineering with science in this regard, and to see which characteristics apply to both and where there are differences.

Science, or any particular branch of science, such as physics, takes place within a set of concepts and what Thomas Kuhn calls a *paradigm* [23]. According to him, it is the acceptance and support of such a paradigm that is, to a large extent, responsible for the efficiency of scientific work. A similar view can certainly be taken of engineering; each discipline operates within an accepted set of standards and by utilising a common, proven technology base, and there are professional bodies maintaining a degree of order and promoting the exchange of information.

A difference becomes immediately apparent when we consider not the existence of paradigms in both cases, but the manner in which they change. Kuhn thought that a scientific paradigm would remain stable and be defended by its community until such a time as the discrepancies between the paradigm and observation became impossible to ignore. That would then be the start of a period of unrest and uncertainty that would culminate in a "revolution"; a sharply defined change to a new paradigm, most often triggered by a discovery that could explain some or all of the discrepancies. The changes in engineering paradigms are much more gradual, and the main reason for this difference is that science provides explanations of the observed world in terms of theories, and as there can be more than one theory explaining a set of observations, there is an element of belief, or world view, in accepting a particular theory. A change in a scientific paradigm therefore involves a *rejection* of the old theory. Changes in engineering paradigms, which are essentially changes in the accepted technology and all that goes along with applying these changes, simply *add* to what is existing without the need to reject any of the previous paradigm as wrong (although no longer useful). When electromagnetism was discovered, it added a whole new discipline and a rapidly developing technology to engineering, but it was not necessary to reject any of the existing engineering knowledge. And when the transistor was invented, it was the start of a massive addition to electrical technology, and although this gradually replaced such existing technology as the vacuum tube, it was not necessary to reject anything.

Engineering is much more pragmatic that science. If something is useful, use it; if something better comes along, then use that. This attitude is a strong influence on the social structure of engineering. In particular, as usefulness is most often (but not always) reflected in commercial success, *the reward system in engineering is indirect* in the sense that it does not depend only (or even mainly) on the engineering effort itself (i.e. as in design), but depends also on a number of other activities associated with engineering projects, such as financing and marketing. This is reflected in the rankings produced by various organisations, e.g. such lists as "The 100 Most Influential Engineers", where, with few exceptions, "influential" relates to commercial success, not to influencing the development of engineering. In science, a leader is someone at the forefront of the development of the science; in engineering "leadership" is almost invariably commercial leadership.

The second characteristic is the use and importance of languages and linguistics. This was discussed with regard to the system concept in Sec. A3.3, it was an inherent aspect of developing an engineering ontology in the previous section, it underpins the important topic of requirements definition in Chapter B5, and will appear again in Sec. C4.4. If we define a language roughly as a means of expressing intellectual content, and consisting of a set of symbols representing concepts and a set of rules for combining them, then, as engineers, we are familiar with a number of languages. Of greatest importance is, of course, our natural language (for simplicity, and without prejudice, we will assume that the natural language is English). The central importance of English is a reflection of what it is to be a human being in general; much of our thinking is done in terms of English (the internal conversation), and speech gives us the ability to communicate those thoughts directly between humans.

However, the nature of English presents us with some problems. Firstly, the meaning of a word is in most cases not unique, as consulting a dictionary will demonstrate. For example, the word "structure" would tend to imply something entirely different to a structural engineer (constructed structure) than to a chemical engineer (molecular structure). Secondly, the meaning of a sentence, its semantic interpretation, is further complicated by the fact that it is dependent on the wider context in which it is embedded and on the background of the reader. In addition to the literal meaning, an inherent property of the sentence that is independent of the context in which it finds itself, there are the pragmatic implications, information conveyed by the sentence when it is combined with all other knowledge available to the reader at the time of reading [24]. This raises a problem, both for the use of English within engineering, where it tends to reflect the particular engineering discipline (as is true within science), and for the embedding of engineering within a project, which requires communication with numerous stakeholders outside of engineering [25].

To overcome this problem of ambiguity or indeterminism of communication within engineering, we have developed a large number of specialist languages, although we might not normally think of them as such. The oldest and best known type of engineering language is the engineering design drawing, which comes in numerous variants, from architectural drawings to circuit diagrams, more recently programming languages, and now such general languages as typified by UML. These languages all have a common purpose; to improve what we might call the "cost-effectiveness" of the language, where the effectiveness is the accuracy of the expression and cost is the effort involved in both generating it and using it (i.e. cost to both sender and receiver). English is unbounded; there is no limit to the accuracy we can achieve if we only make the expression long and detailed enough, but the effort is correspondingly unbounded (as is demonstrated by the cost of legal work).

This development has a number of consequences. The development of each of these languages is based on a common condition: the applicability is restricted to a defined context, i.e. interest or user group and subject matter, and so their use tends to create problems in communications between engineering disciplines. But, more importantly, their inappropriate use in communications between engineering and the rest of society, both within projects and in general, is one reason for why engineering and its importance is poorly understood by society. This leads me to the observation that if we view them all under the unifying concept of a system, it would be possible to define the core of systems engineering as a search for the most appropriate language for a given engineering task. In this view, the problems involved in the relationship between technology and society appear as language problems.

There is also the issue that any engineering language is defined in terms of English. For all the desire to escape into a purely logical or mathematical formulation, engineering deals with the "real world", with physical objects and with people, and any symbols and concepts we use to describe and operate on these will, at some point, have to be defined using English. As a result, there is no escaping the issues connected with English, as mentioned above; it is just a matter of handling them in the best manner..

Finally, any description in any language, be it of a problem situation or of a solution, can be considered to be a model of reality, and in this sense it is really doubling up to be speaking of a modelling language. All languages express conceptual models of aspects of reality, with the aspects being such things as cost, reliability, safety, etc., and with only natural language being able to cover them all. Conversely, a model exists within a particular language; in order for a model to convey information from creator to user it is necessary for both of these actors to be proficient in the language. Therefore, when we make observations about language, we are also making observations about models.

References

1. The issue of leadership, the distinction between management and leadership, and the role of engineering in society is treated in D.H. Pletta. The Engineering Profession: Its Heritage and Its Emerging Public Image. University Press of America, Lanhem, Md (1984); The need for engineers to have a much closer involvement with the business aspects of their products was emphasized in E.W. Aslaksen. The Changing Nature of Engineering. McGraw Hill, Australia (1996); And, as an example, Engineers Australia has created a separate Centre for Engineering Leadership and Management in order to promote these aspects of the profession
2. Galloway, P.D.: The 21st Century Engineer. Am. Soc. Civil Engineers, Reston (2008)

3. Florman, S.C.: Has explored the nature and content of engineering, its relationship to the environment in which it is practiced, and its relationship to other areas of human intellectual endeavour, in a series of books: Engineering and the Liberal Arts: A technologist's guide to history, literature, philosophy, art and music. McGraw Hill, New York (1968); The Existential Pleasures of Engineering. St. Martin's Press, New York (1976); Blaming Technology: The irrational search for scapegoats. St. Martin's Press, New York (1981); The Civilized Engineer. St. Martin's Press, New York (1987); The Introspective Engineer. St. Martin's Press, New York (1996)
4. Unger, S.H.: Controlling Technology: Ethics and the Responsible Engineer, 2nd edn. John Wiley &Sons (1994)
5. An extensive bibliography on the subject of ethics in engineering has been compiled by W.P. Vann, `http://ethics.tamu.edu/ethics1.htm`
6. ISO/IEC 15288, Systems engineering – System life cycle processes, Annex B (2002)
7. One of the main approaches to project management is that developed and maintained by the Project Management Institute and documented in the Project Management Body of Knowledge (PMBOK), `http://www.pmi.org/Resources/Pages/Library-of-PMI-Global-Standards.aspx` Another well-known, but more specific and detailed approach is PRINCE2, and a comparison of the two approaches can be found, `http://www.aipm.com.au/resource/A%20comparison%20of%20Prince%202%20vs%20PMBOK.pdf`
8. A search on "ontology" in Google gives an immediate impression of the recent activity in this area. Some of the prominent authors are Tom Gruber, `http://www.tomgruber.org/writing/ontology-definition-2007.htm`, John Sowa, `http://www.jfsowa.com/ontology/toplevel.htm`, Barry Smith, `http://ontology.buffalo.edu/smith/BSCV.pdf`, and Deborah L. Mc Guinness, `http://www-ksl.stanford.edu/people/publications.html`.
9. Discussions and presentations at Ontology Summit (2010), `http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2010`
10. `http://www.w3.org/TR/owl-features/`
11. Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology, `http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html`
12. Aslaksen, E.W.: Languages and Models in Systems Engineering. In: 4th Asia-Pacific Conf. on Systems Engineering, Keelung, Taiwan (October 2010)
13. Ross, W.D.(ed.): Aristotle's Metaphysics. Oxford University Press (1958)
14. Standard Upper Ontology Working Group, IEEE P1600.1, `http://suo.ieee.org`
15. Cyc Upper Ontology, `http://www.cyc.com/cyccdoc/upperont-diagram.html`
16. Basic formal Ontology (BFO), `http://www.ifomis.uni-Saarland.de/bfo`
17. Munn, K., Smith, B.: Applied Ontology – An Introduction, vol. 9 in the series Metaphysical Research, Reicher/Seibt/Smith/von Wachter (eds.), ontos verlag (2008), `http://nemo.nic.uoregon.edu/wiki/images/8/88/978-3-938793-98-5_Munn_Ontology.pdf`
18. `http://en.wikipedia.org/wiki/Engineering`
19. This map was created using the tool Cmap, freely available from the Institute for Human and Machine Cognition (IHMC), `http://www.ihmc.us/`

20. ISO/IEC 15288:2008, Systems and software engineering – System life cycle processes, ch. 4 of Systems Engineering Handbook, v. 3.2, INCOSE (2010)
21. Pahl, G., Beitz, W., Feldhusen, J., Grote, K.H.: Engineering Design, 3rd edn. Springer, London (2007)
22. Godfrey-Smith, P.: A good introductin to this subject matter is the book. Theory and Reality. Uni. of Chicago Press (2003)
23. Kuhn, T.S.: The Structure of Scientific Revolutions, 3rd edn. Uni. of Chicago Press (1996)
24. Smith, N., Wilson, D.: Modern Linguistics. Penguin Books Ltd. (1979)
25. This issue is the subject of two papers by E.W. Aslaksen: Requirements Definition – A Plea for a Return to English, Proc. Thirteenth Int'l Symposium of INCOSE, Arlington, VA (July 2003), and ref. 12 above. Some of the material in this section is taken from the latter (with some modifications).

B3 Usefulness – The Purpose of Engineering

B3.1 Definition and Measure

At the start of the previous chapter, we identified usefulness as a main characteristic of engineering. We all probably understand what this means in a general way, but how can we define this attribute in such a way that we can *measure* it? Our earlier, very general definition of usefulness as the degree to which the results of the engineering activity meets the stakeholders' expectations is well and good, but when we consider the myriad of activities engineers are engaged in, from teaching to research, design, management, maintenance, etc, we see that the expectations can be very different in nature and therefore require very different measures. What we need to do is to apply the systems approach and, as a first step, abstract from the details of the particular activities in order to identify what they have in common. No matter what the requirements are, meeting them must have a *value* to the person or group of persons who raised them and, at least in principle, the common measure of value is a monetary one. The idea of putting a monetary value on the outcomes of all engineering activities is controversial, and assigning an actual numerical value to this parameter can be even more difficult, but it is in identifying and confronting these difficulties that we can hope to make some headway into developing this central characteristic of engineering.

Consider the simple, generic case of an engineer (which we shall take to include a group or firm of engineers) being engaged by a Client to design a product that will meet a set of requirements. The most immediate way of defining the value of the engineering would be as the amount the Client is willing to pay for it; this amount is determined by the two opposing forces of supply and demand. If there is little work around, and the engineer is on the verge of starvation, he will take on the job for a very small fee; if there is more work around than he can possibly handle, he will increase the fee to the maximum he estimates the Client will accept. On the other hand, the maximum amount the Client is willing to pay depends on what he believes he will be able to charge for the product when he puts it on the market. Both of these factors are valid influences on the value of the engineering. The scarcity of engineers arises because of the cost and hard work that is required in order to become a professional engineer with the experience to carry out the design. And the value that society (in this case the part of society making up the market for the product) puts on the product is dependent on the engineer's skill in meeting the requirements.

However, this simple example also highlights a couple of major problems associated with putting a value on engineering. Firstly, the demand for engineers relative to the supply is determined by factors outside the control of engineers, generally economic factors such as the demand for local products and foreign exchange rates, but also on such factors as political stability and climate. For example, as a result of the resource boom in Australia, driven mainly by demand from China, starting salaries for engineers in the resources industry doubled within a few years. So, although the work these engineers did was not different to what engineers in this industry did a few years earlier, its value, measured on what industry was willing to pay for it, had increased. And these engineers could also be considered to be more useful, as meeting a demand is the essence of being useful. This approach to valuation reduces engineering, more or less, to a pure commodity, without any intrinsic value.

Secondly, the price the Client receives for the product depends on a number of factors, such as marketing, distribution channels, and competing products, that are not under the control of the engineer, as is also most often true of the supply chain determining the cost of the product. So, while engineering may have a significant influence on the commercial success of the product, it is frequently not the main factor, and using the commercial success of the product as a measure of the value of the engineering is therefore problematic.

These two problems provide an indication of where we need to look for a better understanding of the usefulness of engineering and how to measure it. Basically, engineers see themselves as problem-solvers through the application of technology. Given a problem, we accept the challenge and use our knowledge and experience to find the most cost-effective solution. In most cases, particularly given the routine nature of many of these problems, these are solutions that satisfy all parties involved. But sometimes, and increasingly, the solutions do not satisfy all parties affected; not because they do not solve the problems, but because the problem formulations were inadequate, in particular in that they did not define the *contexts* in which the problems arose and the *values* attached to solving the problems. Therefore, there is nothing wrong with defining the usefulness of engineering as the degree to which the engineered solution resolves a given problem; the issue is that ensuring the adequacy of the problem definition must be part of the engineering. This wider understanding of the problem within its context is what Warfield [1] calls the *problematique*, and he discusses both the necessity for this understanding and a methodology for achieving it in considerable detail. We shall return to this in Chapter C1, where we develop it as an essential part of applying the system concept to engineering; for the moment let us simply note that such up-front work may be required in order for our definition of usefulness to be valid.

With this understanding, we restate our measure of usefulness as “the value of the result of the engineering activity to the stakeholders”, and consider some of the main implications of the characteristic of striving to be useful.

B3.2 Commercial Framework

In order to be useful, engineering must be embedded in a commercial framework; it is only through its interaction with the various activities within that framework, such as financing, production, marketing, and organizing, that engineering can result in value to the stakeholders. In this it differs from the two other professions of medicine and law, as already noted in Sec. B2.1. The value of the practice of medicine is intrinsic and not dependent on any further results. For example, if a physician saves the life of a person who afterwards turns out to be a serial killer, nobody would think of suggesting that the physician could have saved the lives of those victims if he had not saved the killer's life. Similarly, upholding the law, and assisting people to stay within the law, are activities that are considered to have an intrinsic value, no matter what the further consequences are. For example, if a lawyer finds a loophole in the law that allows a business person to escape conviction for actions that have caused other people damage, nobody would blame the lawyer if this person continues to cause damage.

Engineering does not have any such intrinsic value. As engineers, we know what good engineering is and can recognize a great design when we see one, but even such a design has, in itself, no actual value, at most a potential one. It has no effect and changes nothing by just being; this only happens when it is realised, and that realisation of a design can only take place within a commercial framework. Consequently, for engineering to achieve its potential for being useful, engineers need to understand and be involved in the realisation process, i.e. participate in the business process. As we saw in the historical overview, this has always been the case, and it is only since WW2 that the lack of integration of engineering and business has become an issue. It probably had its origin in the British social structure, where engineers never had the standing their colleagues on the Continent enjoyed, but the main driver seems to have been the rise of "business management" as a separate profession in the US and the focus of this profession on the marketing and selling aspects. Of course, technology was still at the core of the post-WW2 development in the US, and a company like Hewlett-Packard was a shining example of first-class engineering in the broadest sense, but the main factor in making the US economy so dominant was the fuelling of a rampant consumerism through superbly organized and highly effective marketing. Big cars, Coke, and McDonald (and, later, Microsoft) were the ubiquitous symbols of what became known as Yankee commercial imperialism, and it was as a result of this success that a crack started to develop between engineering and its commercial framework, with engineers seen as the "backroom boys" turning the crank to supply the "real" business with a never-ending stream of products on demand. It was a myopic "push" industry, with limited concern for what society really needed and for what the long-term effects might be, and so the involvement of engineers, with their analytical and dispassionate approach, was not required and even unwelcome.

In recent years this has all been changing, with increasing concern for the way in which technology is shaping our society and with increasing realisation of the complexity resulting from our intervention in Nature and the competition for scarce resources, and so it is appropriate to re-examine the role of engineering within its commercial framework. It might at first appear that this framework varies too much between organisations in different fields, such as manufacturing, construction, consulting, maintenance, certification, etc, but on closer examination, and employing our now sharpened abstracting skills, we can identify a few common features. One is that any one of these organisations encompasses the same main disciplines, albeit with very different proportions. In addition to engineering they are sales, legal, production, financing and business management, as illustrated in Fig. B3.1.

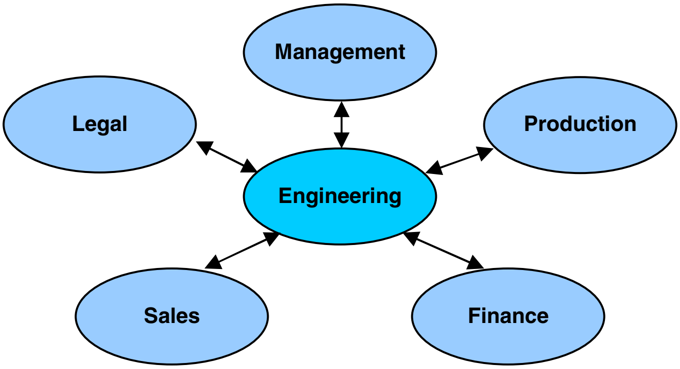


Fig. B3.1 The main disciplines making up the commercial framework of a project.

Engineering has been placed in the centre in Fig. B3.1, not just because it obviously plays a central role in an engineering project, but also because it illustrates what can easily happen, and has often happened; the engineering function gets isolated from the project's environment. Sales deals with the clients, finance deals with the equity and debt providers, production deals with the supply chain, legal deals with liability and the world of intellectual property, and management deals with recruiting, politics and image. Engineering sits as in a walled garden in the middle of this, getting problems and requirements thrown over the wall, and throwing solutions back. The solutions are generally very good as far as solving the problems and meeting the requirements as presented, but there is the catch; "as presented" is as the problems and requirements are understood by the functions acting as interfaces to the various stakeholders. Not only can distortions take place in transmission, which leads to engineering providing solutions that do the wrong thing perfectly, but each interface puts its own emphasis on the information, and so makes it very difficult for engineering to make optimal trade-offs.

Rather than this one-way flow of requirements to engineering there should be a two-way flow of information; a proper dialogue leading to an understanding of the context in which the requirements arise on the part of the engineer and an understanding of the implications of the requirements for engineering on the part of the other function. For example, in the case of the interface between engineering and sales, participation in the discussions with the customer base allows engineering to understand the needs behind the requirements, why the customers desire particular product characteristics and what they are worth to them. It then also allows engineering to explore alternate ways of meeting the same needs, proposing approaches and solutions that neither the customer nor sales would know were possible.

In the case of the interface with finance there is a similar need for dialogue; the finance function needs to understand the sequence of work packages within engineering and the resulting timing of the demand on funds, and engineering needs to understand and accommodate the realities of attracting investment.

There is no suggestion here that engineering should take over the other functions, or that engineers should become proficient in them; it is a matter of understanding, not of being able to perform these functions. Each of the functions in Fig. B3.1 requires specialist skills and experience, and attracts people with certain characteristics. But through effective interactions between the functions the contribution of each one to the overall performance of the enterprise is greater than if each one focuses solely on its own area of expertise. The performance of the enterprise is an emergent property of the system of individual functions, and we know that as such it is dependent on the structure of the system, in this case the organisation, and on how well the individual interactions operate.

The consequence of this for engineering is that its usefulness, according to our definition, will depend not only on its intrinsic capabilities, but strongly on its ability to interact effectively with the other functions within the commercial framework.

B3.3 The Contractual Framework

The discussion of the commercial framework in the previous section probably made us think of a manufacturing company, where engineering is integrated into the company organisation and the various interfaces are fixed by that organisation, and that is also the environment in which the majority of engineers work. However, there is also a significant sector of engineering that operates in a different commercial environment; it is made up of the consulting engineering firms, large and small. This sector of engineering services mainly the construction industry, as distinct from the manufacturing industry, and the products are often large, one-off objects, such as a motorway, railroad, power station, dam, port, hospital, factory, etc; generally what falls under the heading of infrastructure. In this sector, work is carried out within projects rather than within companies, the management style is project management rather than corporate or line management, and the commercial framework is provided by contracts rather than by a corporate structure. The importance attached to the design of corporations, their structure, processes, and culture, is now supplanted by the importance of choosing the best *contracting strategy*, and contract management becomes an important part of the management of a project. Of course, contract management plays an important role in the manufacturing industry also, but the contracts are between the manufacturer and the client, not within the manufacturing organisation itself.

As a result of this, and referring to the discussion in the previous section, it is clear that the usefulness of engineering within the construction industry will depend on the contractual framework, and this can be best demonstrated by considering a few of the most common contractual arrangements. However, before doing that, it is useful to keep in mind another difference between the manufacturing industry and the construction industry. In the manufacturing industry, the engineering associated with a new product is a complex process, consisting of numerous steps forming an integrated development process, including prototyping and many different types of tests, and with a cost that may be many times the production cost of a single item. Also, new products often depend on leading edge technology, so that a significant portion of the risk exposure lies within engineering, further emphasizing its importance.

In a typical infrastructure project, the cost of engineering is less than ten percent of the total cost. There is little if any development and prototyping, and the greater part of the engineering work is prescribed by standards and codes. Exploratory work and preliminary design often focuses on geotechnical investigations and surveys.

The traditional contracting type is for the Owner to effectively act as overall project manager and first issue one or more competitive contracts for the design. It can be a single contract, but often there is one contract for the concept/preliminary design and associated costing, as an input to the feasibility study and the decision to proceed with the investment, and then another contract for the detailed design and technical support during the tendering for one or more construction contracts. Because the design is completely detailed at the point of tendering for the construction, it allows the construction contractors to be very competitive and refine their bids to the n-th degree, not having to allow anything for uncertainty as far as the design goes. It also allows the Owner to select specialised contractors for separate parts of the job, such as piling, excavation, road-works, structures, and the like. This is therefore potentially the contracting strategy that will result in the lowest cost, and it allows the Owner to decide every detail of the project.

However, it also has a number of potential problems associated with it. Firstly, it requires the Owner to have considerable experience in the overall project management and, in particular, in formulating the contracts appropriately, and the Owner needs to have the corresponding capacity. Secondly, the onus for providing continuity throughout the project and ensuring that no part of the project falls into gaps between contracts rests with the Owner. Thirdly, once a contract is let, the Owner’s only means of influence is through contract variations, which are generally very costly, so that every detail should be decided *before* the construction contracts are let. And finally, the Owner accepts the risk that the finished product will not meet the Owner’s needs.

What is, in many ways, a variant of this traditional contract type is the Engineering, Procurement, and Construction Management (EPCM) contract, where the Owner engages an engineering consultant to carry out these tasks, but where the majority of the risk still rests with the Owner. For that reason, it needs to be based on trust and on a relationship between the Owner and the Engineer, but it relieves the Owner of having to have any capability and capacity with regard to these tasks, while still having the ability to have a strong and ongoing influence on the project.

Another often used contracting strategy is for the Owner to let a Design and Construct (D&C) contract, based on a concept design incorporated in a Brief. Because this allows the design to be optimised with regard to the construction methodology (and cost), it can be very cost-effective as far as the construction goes, but it allows the Owner very little influence on the design after the tender design is accepted and so requires a very good Brief, and the quality of the product may be uncertain. It allows the Owner to transfer the completion risk (time, cost, and performance) onto the contractor, but at a cost.

A variation on the D&C contract is to add the requirement for the D&C contractor to also maintain the facility for a certain length of time. This has the advantage, from the Owner's point of view, of ensuring that the quality of the facility is appropriate. A further variation on this is for the contractor to build, own (and maintain), and operate the facility for a certain length of time, before transferring it to the Owner, a so-called BOOT contract. This is most common for such major infrastructure works as e.g. motorways and desalination plants, but would not be found in manufacturing plants, where the Owner has to be in charge of the daily operation.

Finally, a more recent development which has been gaining acceptance, and one that can, in principle, be applied to any of the above arrangements or variations on them, is the concept of *alliance* or *relationship* contracts [2]. In an alliance, the Owner is a participant, along with other Non-Owner Participants (NOPs). which include the main contractors. This has arisen from the desire to reduce the litiginous aspect of the construction industry, and rests on two main pillars. One is an equitable scheme for sharing the "pain and gain" of the project, defined not only in purely monetary terms, but also in terms of an extensive set of Key Performance Indicators (KPIs) that include e.g. the handling of community concerns. The other is a commitment for all contract participants to work together as an integrated team on a "best for project" basis, rather than pursuing the narrower interests of each participant. In principle this is a good approach; working as a team focused on a common goal can bring significant gains in efficiency and innovation. In practice there are some obstacles, as is evidenced by the fact that a significant proportion of alliance projects fail to deliver the expected results [3]. For example, one or more of the participants are often government bodies that have legislative or statutory requirements and limitations on their behaviour that are fixed and not able to be modified to suit "best for project" requirements. But perhaps the greatest problem is the inability to switch between a competitive, traditional corporate culture in one project to a "best for project" culture in the next project, and with the emphasis on "branding" and "corporate identity", it is difficult to submerge this temporarily in an alliance. This problem is present in the decision-making throughout the project, but nowhere is it more evident than when determining the Target Outturn Cost (TOC) [4], the yardstick against which performance is measured in monetary terms, in the early phase of the project. Each participant tries to gain the best starting position before the "real" cooperation begins.

B3.4 Engineering as a Commodity

We noted earlier that if the value of engineering is determined strictly on the basis of supply and demand, it is essentially a commodity. But would it be so terrible if this were the case? It might at first be a bit off-putting for an engineer to be placed in the same category as a lump of coal or a barrel of oil, but in both cases it is true that the value is tied to the demand. And if it is a demand that will be increasing, being a commodity in demand should not worry us. What we should be concerned about is how we can improve the value of this commodity and thereby increase the demand.

In the first instance, and most importantly, it is crucial for engineering to be seen as a process, as a sequence of activities, and for the buyer of the engineering service (i.e. the commodity) to understand that the value of this process is determined not just by how well the individual activities are performed, but on the sequence in which they are performed and on the interaction between them. It is the complete process that should be treated as a commodity, and its value is an emergent property of the interacting activities. That is, the process of engineering is a *system* of activities, and the sum of the values of the individual activities is generally much less that the value of the process as a whole. All too often this understanding is lacking, and as a result engineering is not allowed to provide its full potential value.

There are several factors that contribute to this situation. One, and perhaps the major factor is the blind belief in the benefits of a "competitive market"; it is often considered good commercial practice to subdivide the engineering within a project into separate stages and/or work packages and put each one separately out to tender. The result is a number of barriers to the flow of information, both in time throughout the duration of the project and between disciplines at any one time, but also often an absence of coordination and of an understanding of the "big picture". Choosing a relationship contract, as discussed in the last section, as the arrangement within a stage does little to change this myopic view, it only provides a better working relationship between the participants in that stage. And the fact that some entity, such as the Owner or Principal, is a participant through all stages does little to improve this situation, because once a contract is let, any further direction or guidance from that body is inhibited by the threat of costly variations and a shift in responsibility, which would negate what were the supposed benefits of the competitive arrangement in the first place.

There are, of course, as always two sides to this issue; the engineering community must bear some of the responsibility for this disjointed approach to engineering within a project. The narrower and more discipline-specific an engineering contract package is, the lower the risk. With standards, previous experience, and accepted practices for checking and review, the probability of a reasonably competent engineering organisation getting the design for a normal road bridge or high-voltage substation wrong is very small. The technical risk in a project lies mainly in the *integration* of such relatively simple components within the context of the overall project. The more the required outcome of the project depends on the interactions between the components rather than on the performance of the components in isolation, or in other words, the more the required performance is an emergent property, the greater the complexity of the engineering task and with it the risk. As a consequence, engineers have not been altogether unhappy to be engaged on a piece-meal basis; while this limits the use of their talents, it also limits the risk.

B3.5 The Value of Systems Engineering

Finally, with a view to the application of the system concept to engineering in Part C, a given that systems engineering has been around for quite some time and is now being increasingly accepted throughout the engineering community as the preferred approach to complex projects, we could ask how systems engineering has added to the usefulness of engineering, and how this can be measured. For a number of reasons, this turns out to be two very difficult questions to answer except in a qualitative and largely anecdotal manner, and Eric C. Honour, President of Honourcode Inc, has dedicated considerable effort to this issue. In one study [5], he collected data on 44 projects as part of his ongoing efforts to document the value of systems engineering, and one illuminating result is the relationship of a quantity called Development Quality (DQ) as a function of the proportion of the systems engineering effort spent on a project, as shown in Fig. B3.2.

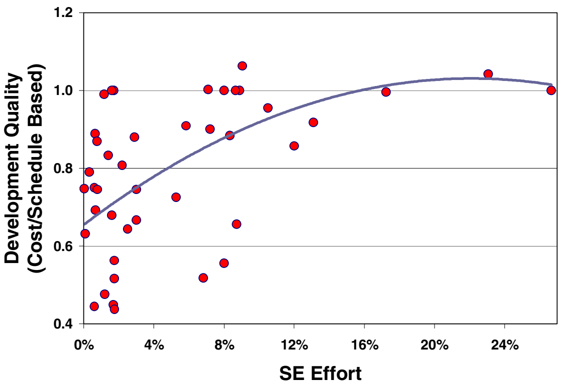


Fig. B3.2 Development quality as a function of systems engineering effort (reproduced with permission from [5]).

In this figure, DQ is defined as the inverse of the average of the actual cost (AC) to planned cost (PC) and actual schedule (AS) to budgeted schedule (BS), or

$$DQ = \frac{2}{AC/PC + AS/BS} ,$$

and the systems engineering effort is the product of the actual cost of performing the traditional systems engineering task and a measure of the quality of that performance, expressed as a percentage of the project cost up to delivery of first article, not including production costs.

Instead of calculating the mean value of the data points in Fig. B3.2, as presented by the curve in that figure, we could look at the variance and the maximum value as functions of the systems engineering effort. The former is steadily decreasing, whereas the latter remains practically constant, with a value of around 1. This supports the view that "classical" systems engineering, as it is presented in the majority of textbooks and practiced in the defense and aerospace industry (from which most of the data in Fig. B3.2 was taken), does not add any technical knowledge that is not already present in engineering; if the engineering is done properly, the outcome will be the same with or without systems engineering. What systems engineering adds is a methodology for handling complexity and thus ensuring that the engineering *is* done properly as the complexity of the projects increases. From this point of view, systems engineering is similar to quality assurance: it is possible to achieve high quality without the existence of any formal quality system (and there are numerous examples of this). What a quality system does is ensure that the quality is high every time; it reduces the variance of the engineering process.

This view of "classical" systems engineering is the main motivation behind this book. It is my belief that, in the drive to improve the quality of the engineering process, a main feature of applying the system concept is being overlooked, and that is in enabling the transition from the functional stakeholder requirements to requirements on the engineered object that, through its operation is to meet these requirements, to be made explicit. It extends the engineering process from converting a specification into an operating system to converting requirements on a service into a system providing that service by adding what I call *design in the functional domain* – the subject of Part C. The data in Fig. B3.2 expresses how efficiently the specification was converted into a first article; it does not necessarily reflect to what extent the system provided the desired service.

Eric Honour has continued his research into the issue of the effect of applying systems engineering, and his most recent results, which are included in his PhD thesis, further support the conclusion that "classical" systems engineering leads to better projects (cost and schedule), but not to better systems (technical performance) [6].

References

1. Warfield, J.N.: An Introduction to Systems Science. World Scientific Publishing (2006)
2. Project Alliancing – Practitioners' Guide, issued by the Government of Victoria, Australia (2006)
3. Kiers, D.: Alliance Competence: Key Capabilities for Success. APPEA Journal, 811–822 (2001)
4. Target Outturn Cost: Demonstrating and Ensuring Value for Money, Alchimie Pty. Ltd. (2004), http://www.alchimie.com.au
5. Honour, E.C.: Understanding the Value of Systems Engineering. In: Proc. 14th Annual Internat'l Symposium, INCOSE (2004), http://www.incose.org
6. Honour, E.C.: Dissertation at the University of South Australia, Contact E.C. Honour for relevant information and material (2012), http://ehonourhcode.com

B4 Design and Creativeness

B4.1 Overview

In Chapter B2 we identified creativeness as a main characteristic of engineering; in this chapter we shall examine that assertion in more depth and develop a detailed understanding of the relationship between engineering and creativity. To do this, we need to look at the process of engineering from a point of view that differentiates activities on the basis of their creative content, and that is the subject of the next section. We can then, in the further sections, focus on those activities where creativity is a major factor. But first, let us be certain that we have a clear understanding of what we mean by "creativeness".

Dictionaries give such definitions of the verb "to create" as "bring into being, give rise to, make, produce", with synonyms such as "coin, compose, concoct, design, formulate, invent, originate". So while it can simply mean the production of something, it generally has the connotation of originality and uniqueness. For example, we would not say "the factory created 10,000 units in the month of May", but we would say "the company created over twenty new models this year". However, newness or originality implied by using "create" can be subtle. For example, we would probably not say "the bricklayer needed a day to create the wall" even though there might not be another wall exactly like it, but we do say "she always creates such a harmonious atmosphere" even though it is obviously a repetitive act. Somehow, creating implies a mental effort, and this latter example shows that what is created does not have to be a physical object; it can be a feeling, an idea, a concept, etc.

The aspect of originality and uniqueness becomes more apparent when we consider the adjective, "creative", which, besides being defined as "having the ability or power to create", is also defined as "Characterised by originality and expressiveness". And when we characterise something as "very creative"; the "very" refers to the degree to which the object is original and unique. (By the way, Fowler's *Modern English Usage* has the following to say about "creative": " It has been aptly called a `luscious, round, meaningless word`, and said to be `so much in honour that it is the clinching term of approval from the schoolroom to the advertiser's studio`.")

The two nouns, "creativeness" and "creativity", which appear to be interchangeable, follow simply as "the ability to create".

It would be tempting to say that creativeness is a peculiarly human trait, but that would be simplistic. Consider a set of distinguishable elements, each of which can be combined with other elements in a predetermined fashion. (This could e.g. be the set of all electronic components.) The task is to achieve a certain objective (e.g. performance) by combining two or more elements from this set. In principle, this task could be performed by a computer by trying out one combination after the other and selecting the one which best met the objective, but it does not take a very large set for this to become practically impossible. So we say that the designer who, through a combination of experience and intuition, comes up with a combination that meets the objective is creative, even though the solution may not be unique (and there is no way of proving that it is the best).

Perhaps the most striking example of non-human creativeness is Nature. Through a process of random mutations and selection in a competitive environment Nature has, over time, created innumerable new species. In this case there is no physical entity that is responsible for or directs this creativity, it is inherent in the process of life itself, and from our systems perspective, we might say that this creativeness is the ultimate example of an emergent property.

These two examples demonstrate the difficulty with defining and measuring creativeness. Clearly, coming up with a new combination of existing objects can be very creative, as in combining the known facts about light propagation to come up with the special theory of relativity, but, on the other hand, it can be largely routine, as in designing an amplifier for a new frequency band or a bridge using standard pre-cast beams; the degree of creativeness depends on how similar it is to previous work. Somehow it must also depend on such factors as how clearly discernible or sharply defined the elements are, how complex they are, how large the set is, and how numerous the possible combinations are. For example, in two areas that we would connect with very high creativeness, composing music and writing literary work, the sets of elements (notes and words) are relatively small, but the sets of possible combinations are essentially unbounded.

The physicist David Bohm defines creativity as founded on the sensitive perception of what is new and different from what is inferred from previous knowledge [1], and he illustrates this by the way a child creates new concepts by trying something out and seeing what happens. This is contrasted with the learning process in school, which is largely by repetition rather than by discovery, and how this to some extent inhibits our ability to see something new and original. He also comments on beauty as the connection between art and science, and how beauty is related to operation of the mind through the concept of structure; all very much in line with what we discussed in the first part of this book and Kant's idea of the categories giving a structure to the concepts we develop. He quotes Cezanne's statement of art as a harmony parallel to that of Nature, emphasizing that art is about relationships and structure, not about the symbols themselves. We can paraphrase this by saying that beauty consists of the relationships that can be easily grasped by the mind.

An aspect of creativeness that is particularly relevant to engineering is the extent to which it is purposeful. H.E. Gruber [2] argues that creativeness is always purposeful, even though chance may often play a role, and that the insight that leads to a creation is seeing the relevance of something from another context to the purpose at hand. In another paper in the same book, D.B. Wallace considers what traits make or contribute to a person's creativeness, and emphasizes that while we can look at the existence of absence of various traits in creative persons, it is the particular composition and interaction of traits that makes the person's creative ability unique. We would say that creativeness is an emergent property of a set of traits, and Ertas and Jones [3] identify the following three as central to this set:

- Risk taking (anything new implies a risk)
- Challenging authority and procedures
- Preferring the complex and difficult (i.e. a challenge)

A different, and one might say more utilitarian, view of creativeness is presented by R.K. Sawyer [4]. It is not possible to measure creativeness as a personal trait, this was tried in the 1950s and 60s and abandoned. What is important and can be measured is the creativeness of groups of specialised individuals working together; again, creativeness as an emergent property, but this time of a set of individuals. He devalues the importance of originality, using the performing arts or even translation as examples, and states that the demand for originality in art is less than 200 years old. In this I believe he mixes art and craft and the transition from the latter to the former, much as we saw a transition from craft-based to science-based in the case of engineering in Part A. He sees creativeness as a sort of market transaction between producers and consumers, and considers that for an idea to be creative, it must be *appropriate*, recognised as socially valuable in some way to the community. That is, creativeness is fundamentally a social concept, and this can be illustrated by Fig. B4.1.

The interpretation of this figure is that a person must first absorb all the relevant previously accepted creations (this constitutes the Domain) before he or she can be creative, and acceptance is determined by the Field (the relevant part of society). In this sense, then, a study of creativeness becomes a study in social anthropology, and specialising this to engineering, we could speak of engineering anthropology.

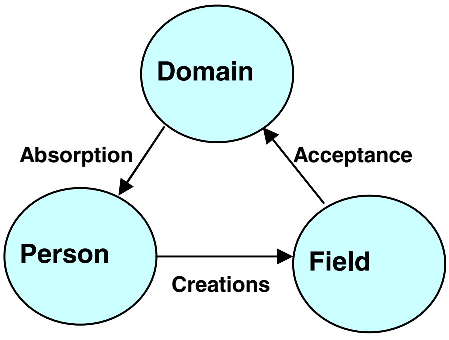


Fig. B4.1 A social view of creativeness.

This brings us to the process of innovation, which K. Holt [5] defines as encompassing the use of knowledge for the generation and practical application of new and viable ideas. The inclusion of "viable" reflects the generally accepted difference between invention and innovation, in that the latter includes the commercialisation and thereby the acceptance by the market of the new idea. He then connects innovation closely with creativeness by citing D.W. Taylor [6] to the effect that creativeness is that thinking which results in the production of ideas that are novel and worthwhile. Again, a very utilitarian view of creativeness.

If we now try to distill this all into an understanding of creativeness that we can take forward into a consideration of its applicability to engineering, we find that there are three distinct views, expressed here as questions:

- What is the nature of creativeness?
- What enhances it, what inhibits it?
- How can it be measured?

With regard to the first question, we would say that creativeness should be purposeful. Fulfilling a purpose it what gives substance to a creation, whether this be inducing a feeling of pleasure, making people think about an issue, or provide a solution to a problem, just to name a few possible purposes. We also recognise that while chance and serendipity may play a role, creativeness generally involves hard work. When we look closer at what appears to have been sudden insights, or so-called Eureka moments, we find that they were in reality preceded by considerable periods of thinking and collecting information about the issues. And finally, we agree that it is founded on knowledge; every great insight and invention is clearly imbedded in the knowledge and technology available at the time.

The answer to the second question follows, to some extent, from the answer to the first. Creativeness is enhanced by interaction between people involved in creating, whether this be by direct interaction, such as in artist communities, or by access to the work of others, as evidenced by the direct connection between the rate of discovery and invention and the extent and availability of written (or printed) material. Creativeness is also enhanced by the attitude of society to change and new ideas; how it accepts them and values them and the people who create them. The converse of this is clearly demonstrated by the so-called Dark Ages in Europe, the period from 900 to 1300, when the all-powerful Catholic Church monopolised education and publishing and suppressed all intellectual activity outside its own, narrow boundaries.

The third question is perhaps the most difficult one, as it tries to connect a multi-faceted, loosely defined, and largely mental activity with the very practical activity of measuring. Given the ephemeral nature of the creation process, perhaps it is better to measure creativeness by the outcome of the process rather than by any parameters of the process itself? That means reformulating the question as: What is the creative content of a piece of work? This question does not seem to be any easier to answer if we consider a work of art; for example, why is a sketch by Picasso more creative than a laboriously painted landscape in the local art exhibition, and how would you quantify the difference? In art, part of the answer is in newness, or being different, which is perhaps why so many weird things are presented as art these days, but in both science and engineering this is not very relevant, because on the one hand, newness and originality is a *sine qua non*, if this is not present, it is simply plagiarism in science and production in engineering, and on the other hand, newness itself is of little or no value. For example, a painting consisting of a uniformly coloured surface and with the title “Hidden” to indicate that there is something (presumably more interesting) underneath is hanging in a well-known art gallery, whereas taking the innards out of a car engine and presenting it as a new design would not go down well. If we restrict our attention to engineering, then we come back to the issue that creativity must somehow be measured by its *impact*, by the change that results from it.

So, how do we measure this impact? This formulation of our original question raises a number of further issues. Firstly, over what time frame should the impact be measured? Some very creative designs and inventions had their main impact many years after their inception and, for example, the impact of printing with moveable type was felt over centuries. Secondly, the impact may not be direct, but lead to other developments that have great impact. The transistor and the laser are both examples of this, and if we consider today’s personal computer, which undoubtedly has a great impact on our lives, how would we apportion this impact to all the engineering activities that went into making it possible? Thirdly, the impact may not be beneficial, or at least not to all who are impacted by it, as in the case of the development of a new weapon. This should not be an issue in judging the engineering effort, as long as the impact is the intended one, but in fact it often does influence society’s judgement. It is possible to be evil and creative at the same time, as we already touched on in Chapter B2.2. Finally, even when (and if) we have agreed on all the parameters that characterise impact, there remains the problem of quantification.

To progress this whole issue of how to measure creativeness and turn it into something we can use in our further discussion of engineering, we can narrow the focus even more when we recall our ontology development in Sec. b2.4, in which we defined all engineering activities to take place within *projects*, i.e. within a body of work that has a defined scope and definite beginning and end. In Ch. C3 we shall argue that with any project we can associate a Return on Investment, and we now define the impact of an engineering activity within a project as the contribution it makes to this Return on Investment. This approach does not by any means solve all problems with measuring creativeness, but it allows us to make some assessment of the extent of creativeness in the various types of engineering activities.

B4.2 A Taxonomy of Engineering Activities

For our purposes in this chapter, and with reference to what was said earlier about the aspects of engineering, it is convenient to introduce the (highly simplified) taxonomy of engineering activities shown in Fig. B4.2.

The first subdivision, into developing technology and applying it, recognises the fact that, while engineers are most definitely involved in developing new technology, much of the effort in developing the technology on which engineering relies is provided by scientists, such as physicists, chemists, and metallurgists, as well as by mathematicians. And while there are certainly significant creative aspects to science, this creativity is directed towards understanding Nature. Its driving force is our inherent curiosity and need for certainty, and the ultimate criterion is truth.

Turning scientific discovery into technology is heavily dependent on engineering and the creativeness of engineers, and recognising the two aspects of developing new technology, scientific discovery and engineering development, is the reason for the two boxes in the second subdivision in Fig. B4.2. The transistor is a good example of these two aspects. The ability to modulate the current flowing in a semiconductor was the scientific discovery, based on knowledge of semiconductor physics; developing semiconductor technology required vast amounts of engineering effort in crystal growing, manufacturing under vacuum and ultra-clean conditions, doping, lithography and etching, and bonding and packaging, and there was certainly a lot of creativeness involved in overcoming the many obstacles on the way. However, in conformance with the stated focus of this book, the creativeness we are interested in relates to finding new applications of technology to meet expressed needs, and there is a subtle, but important difference between the creativeness involved in the two cases. In the case of technology development, the creative activity takes place wholly within engineering; the problems are posed by engineers, and the solutions are used by engineers. In the case of applying the technology, the needs arise outside of engineering, and the products are used (and judged) outside of engineering. A significant part of the creativeness lies is making the right connection between the need (or the problem space) and the available technology (the solution space), and it is often true that the major contribution to the complexity of a project lies in this interface with the non-engineering stakeholders. It was also with an eye to this second case that we settled on contribution to the Return on Investment as a measure of creativeness, as the return on the investment in developing a new technology comes only when it is applied.

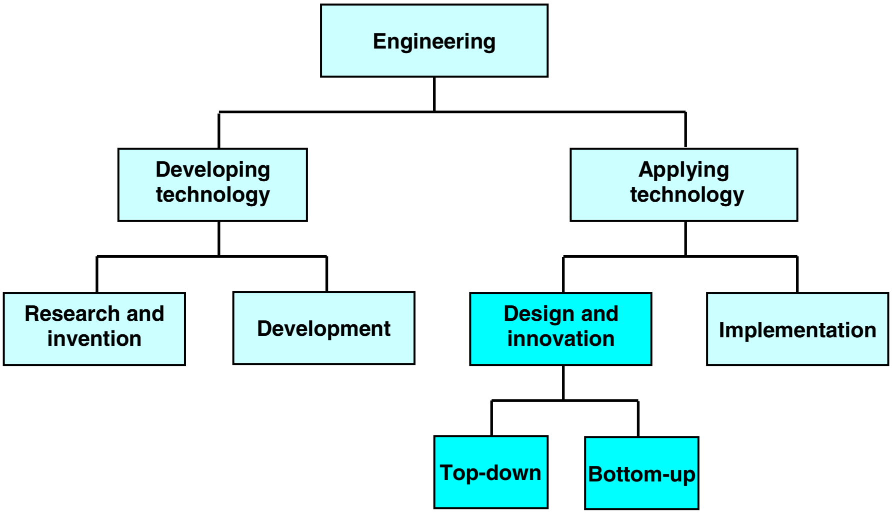


Fig. B4.2 A simplified taxonomy of engineering activities, focusing on the creativity in the design activities.

Given this focus on application, the essence of engineering is to create new products and the equipment and processes needed to produce them, and this creativeness has two sides to it. On the one hand, there is the *design* activity, which develops a description of the object and which is probably where creativity becomes most visible. To support this design activity, engineering has developed a very large body of standard components and of knowledge based on previous projects. The balance between repetitious application of this body of knowledge and the creative component of the work varies greatly from project to project; being able to judge where the right balance lies on a particular project is one of the characteristics of a good engineer. On the other hand, there is the activity of converting this description into a product, an activity which requires vision, leadership, and a whole set of management and business skills. This is also a part of the engineer's creativeness, just as the creativeness of a sculptor involves both being able to visualise the sculpture and then having the skill to carve the marble, or the creativeness of a painter involves creating the image in the mind and having the skills to draw and paint.

The design activity consists of two sequential steps that, while they may have a significant degree of overlap with regards to approach and process, are still quite distinguishable, and to see how this comes about, let us take a look at the requirements that are the starting point of the design process, the *stakeholder requirements*. At the start of an engineering project, the stakeholder requirements are contained in a variety of documents, such as a Requirements Definition Document (RDD), Conditions of Contract, Special Conditions of Contract, and so on, and from a *contractual perspective*, the requirements can be grouped as follows:

(i) A statement of the purpose of the project;
(ii) a set of requirements on the object to be created to fulfil this purpose; and
(iii) a set of requirements on the work required to create that object.

The third group, while potentially of great significance to the project, is addressed mainly through the project management and usually has little, if any, relevance to the design process, and we shall not consider it any further in this chapter.

From a *design perspective*, it is most convenient to consider the requirements in the two first groups above, which are the contents of the RDD, to consist of three types of requirements - functional requirements, performance requirements, and material requirements. Under *functional requirements* we shall understand requirements on what the stakeholders want to achieve, on the capability they need, without any reference to any plant. In particular, and with reference to our initial development of an engineering ontology in Sec. B2.4, this includes the requirements on the *service* and the conditions under which it is to be provided. Under *performance requirements* we shall understand the performance required of the plant which is to provide the service, and the *material requirements* are e.g. surface treatment requirements, weight and size limitations, etc. Some of the material requirements may just flow straight through the system design process without requiring any design effort (as is indicated by the direct arrow in Fig. B4.3), but others will enter into the systems design process through a process of *allocation*. The performance requirements already relate to a plant 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 plant to relate them to. (A statement like "The system shall ..." implies that there is already a system.) That is, we have to choose a basic system architecture (technology, components, interactions). This is illustrated in Fig. B4.3.

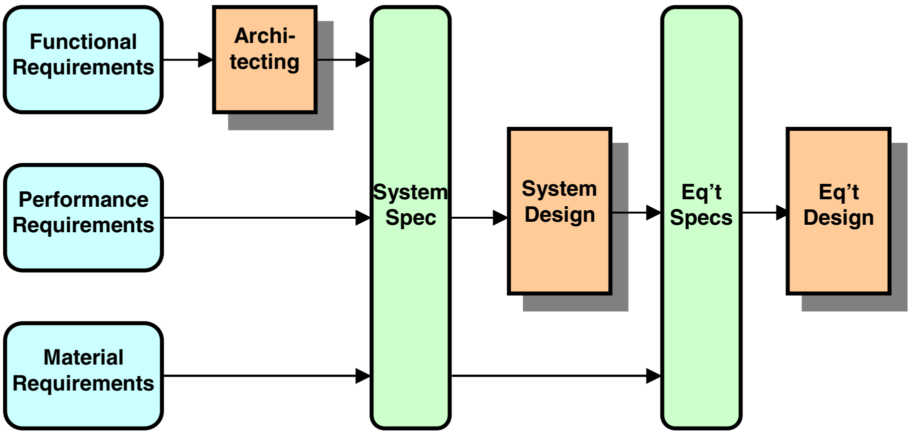


Fig. B4.3 A simplified view of how the three types of stakeholder requirements fit into the design process. Here Equipment Design includes all hardware and software design.

In Fig. B4.3 we have introduced a few new terms, i.e. Architecting, System Specification, and System Design, and in doing so we have pre-empted the development of the application of the system concept to engineering, which forms the subject matter of Part C of this book. Given the consideration of the system concept in Part A and our common use of the term in daily language, this should not cause any problems in the context of the present discussion. However, it might be appropriate to make a comment about the definition of the System Specification, as there is some confusion about this document within the systems engineering community. Such standards as e.g. ISO/IEC 15288 and EIA 632 do not actually define this document, and neither does the INCOSE *Systems Engineering Handbook*. We shall define it as the totality of requirements on the system, expressed with reference to a particular system architecture. This is essentially the definition contained in DoD-STD480A and DoD-HDBK-248A, "a document which states the technical and mission requirements for a system as an entity, allocated requirements to functional areas (or configuration items), and defines the interfaces between or among the functional areas". System design then encompasses the activities which result in individual specifications for each system element, such as trade studies and optimisation, and equipment design converts the requirements in these specifications into manufacturing/production/construction requirements.

As a result of the view presented in Fig. B4.3, we see that the requirements and the design activities needed to process them can be separated into those that relate to the service and those that relate to the plant. In architecture this two-sidedness of stakeholder requirements is expressed as Form and Function, and there is often a certain tension between them under the constraint of cost. While there can be some tension of this nature in engineering, it is not usually significant; for example, there is no reason why a car that fulfils demanding functional requirements cannot also be beautiful without any significant cost penalty. So, while there are numerous examples in engineering where there is tension between functional and physical characteristics; just think of any aircraft or satellite component, where there is always tension between functionality and weight, under the constraint of cost, it is true that in most engineered objects form (as in shape) is subjugated to function. Engineering design is in general focused on achieving a required functionality at the lowest cost.

The relative importance between the two groups of requirements, and the level of detail to which they are expressed, varies tremendously depending on the nature of the project. In the case of a capability development project, (ii) is of vanishing importance, whereas in a project to develop a piece of equipment (i) is effectively a single statement, "The purpose of the project is to develop the equipment specified in (ii)". There is a continuous process of development, from the first formulation of the need to be able to provide a service (in the most general sense) to the data required to produce the systems and equipment that will provide this service, and it may involve a number of separate projects. This process can be seen as consisting of two distinct design activities. The first is the analysis of the requirements on the service, what we have called the functional requirements, in the process called architecting in Fig. B4.3, and which constitutes a *transition* from the functional domain to the physical domain. The second activity, the design in the physical domain, is the traditional design activity that includes detailed design, trade studies, value engineering, constructability issues, etc. Again, the relative importance of these activities in a particular project will depend on the relative importance of the three groups of requirements above, and in most projects today only the second activity is important, because the requirements are formulated completely in the physical domain. That is, the requirements on the service are formulated as requirements on the performance of a piece of equipment or a system of several pieces of equipment; the choice of the physical realisation, i.e. the transition into the physical domain, is done outside the engineering process and usually in a poorly documented process, based mostly on previous experience. This severely undervalues the fact that engineering is a creative activity and that creative thinking relies on the ability to abstract from what is in order to imagine what could be.

Consequently, when we speak of "the purpose" or of "the service requirements" in the present context, it must be understood as what these would be were they formulated without reference to a physical realisation, they are what we have defined as the functional requirements. And by "project" we shall understand the whole sequence of activities starting with the formulation of the functional requirements, even though this may in reality encompass a number of individual projects and contracts, not all of which would necessarily be considered engineering in the traditional sense.

From Fig. B4.3 it is obvious that this grouping of the requirements makes design a two-step process. The second step, which is the one engineering education has traditionally focused on, ends with the physical specification of an object, the engineered object, or what we in Se. B2.4 decided to call the *plant*, to such a level of detail that it can be demonstrated that if the object is realised (i.e. manufactured or constructed) in accordance with this specification, it will meet all stakeholder requirements. This specification may consist of text, drawings, computer models, and whatever else is required for the realisation without any further decision making or design effort.

The point of departure for this step, and therefore the end result of the first step, is a definition of the plant in terms of what it should *do* and how well it should do it, the conditions under which it must be able to do it (what is shown as System Specification in Fig. B4.3), and any direct, physical requirements on the object, such as type of materials to be used (or avoided), surface finishes, weight and size restrictions, etc. The process of getting from the start to the finish of this second step has been developed and refined over the last couple of centuries, and we shall look at it in more detail in the next chapter. Here we just note that its main characteristic is a synthesis in terms of known construction elements, and for this reason it is often called the *bottom-up* process, this also serves to distinguish it from the process in the first step, the *top-down* design process, with which we shall mainly be concerned in this chapter.

One reason for focusing on the first step of the design process in this chapter on creativity is that not only is the structure of the process different in the two steps, but the knowledge and skills required are also significantly different, and the development of technology has led to the creative aspect of design being shifted more and more into the first part. A second reason is that this shift implies a shift in engineering culture, away from an entrenched culture - a way of thinking about engineering design - that is completely anchored in the physical world. Design invariably means design of a physical object, models are always models of the behaviour of physical entities, optimisation means varying the parameters of physical entities to achieve a required performance, and so on. And it does not matter if it is software or hardware; e,g, in object-oriented design, the objects are physical objects, such as persons or bank accounts, and their attributes are physical attributes, such as names, addresses, account balances, etc. To use the analogy with mathematics; engineering is at the stage mathematics would be in if the manipulations were limited to physical objects, i.e. two oranges and three oranges equalled five oranges, two cows and three cows equalled five cows, but the abstract relationship 2+3=5 did not exist. As engineers, we are making things hard for ourselves by insisting on a physical representation, and we are limiting the impact we could have on industry and on society in general. In the case of the RDD, the business world is not primarily interested in the physical realisation, and by insisting on expressing the mutually agreed objectives in terms of the parameters of a physical realisation we are aggravating the problem of the contextual implications of English (ref. the previous discussion in Sec. B2.5).

B4.3 The Functional Domain

Let us at now look at the idealised situation where, at the outset of the design process, there is nothing but a set of stakeholders' requirements on the outcome they desire as a result of the project to which the design belongs. This could possibly be as general and high-level as a single paragraph, but it could also be a document of several hundred pages, describing the service and its interaction with the environment in which it is to exist in minute detail. However, in either case the requirements say nothing about the physical characteristics of the plant that will eventually provide the service (this is the idealisation). With reference to Fig. B4.3, this means that there are no performance or material requirements, and another way of saying this, and one that is more common in the defence industry, is to say that the requirements specify only a *capability* that the stakeholders want to acquire as a result of the project.

The document containing the requirements, the RDD, can therefore (in this idealised case) be considered to describe a completely *abstract* entity; one that has no physical existence, neither as a design nor as a plant. However, the RDD does relate to the plant in the sense that it is the intention of the designer that the plant will meet the requirements by interacting with its operating environment. This leads to the following two definitions [7]:

Definition 1 The *functionality* of a plant is its intended capability for interacting with its operating environment.

The word "intended" has been included in order to exclude such incidental interactions as a bull becoming enraged by a red sports plane making an emergency landing on its pasture; this is not part of the plane's functionality. On a more serious note, the word "intended" expresses a very significant difference between physical and functional descriptions; every statement in a physical description can be verified by an examination of the physical object, whereas the functionality of an object depends on the *intention* of the designer, which again is determined by the requirements of the stakeholder group. The functionality cannot, in general, be deduced from looking at or performing measurements on an object, and there is not necessarily any functionality inherent in a physical object, i.e. disconnected from the intention of its designer. Deducing the functionality of a physical object is, of course, what we call reverse engineering, and its accuracy will depend on what additional information is available.

<u>Definition 4</u> The *functional domain* is the set of all functionalities.

Thus, in this (idealised) view of engineering, *each project starts with the formulation of a functionality.*

The functionality 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 is not tied to any particular physical plant. 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 functionality as a concept is an abstraction, the service it describes is very much in the physical domain, so that the functional parameters are normal, physical parameters. For example, an element whose functionality is to generate electric energy could be characterised by such parameters as power rating, conversion efficiency, etc.

The relationship between functionality and engineered objects is an indirect one; the direct relationship of the functional domain and the physical domain is between functionalities and classes of engineered objects, as illustrated in Fig. B4.4. In this Figure, note that the relationship arrow points from the element to the class; we must have the functionality before we can define the class. The class is the set of objects that have been engineered in order to meet the requirements of the functionality.

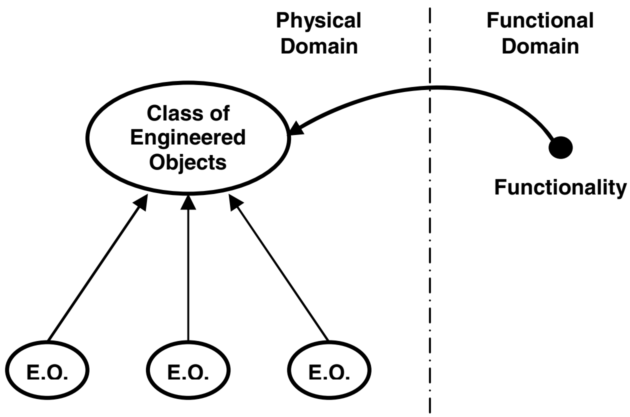


Fig. B4.4 The relationship between the functional and physical domains.

This understanding of the relationship emphasizes that the concept of functionality, as defined above and as illustrated in Fig. B4.4, differs basically from the way the term functionality is used in daily language 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 acceleration, 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 a class of engineered objects can lie anywhere between a description of the performance only and a description of physical characteristics only, and the performance part is commonly termed the functionality of the class of objects. These objects form a class because their actual performances have something in common, i.e. something determined after the fact of their creation, whereas those objects forming a class in the sense of Fig. B4.2 are those that were designed with the *intent* of meeting the requirements of the functional element.

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 a 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 may describe 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.

B4.4 Creativeness and Design in the Functional Domain

The first step in the design process ends with the transition from the functional domain into the physical domain, and that transition may be considered to take the form of a *mapping* of an area in problem space into an area in solution space. The problem is how to meet the need expressed by the initial stakeholders, and the area in problem space is a description of that particular problem in terms of a set of relevant parameters and their values. The area in solution space is a description of a physical object that, through its operation, will provide a service meeting that need, albeit that it is a high level description, referred to as an architecture or a concept design. Finding a good mapping (not to mention finding the best one) always requires creativeness to some degree because, even if the solution in the end turns out to be an existing one, the problem will always have some element of uniqueness. However, in many cases there is a body of work that has to be (or should be) completed before we look for a mapping, and it is here that there may be the greatest need for creativeness.

To see what this body of work encompasses, let us start with a very simple picture of what the first step involves, as shown in Fig. B4.5. In this view, the box labelled "Definition" represents the work required to formulate the need, as expressed by the initial stakeholders, into the definition of a complete and self-consistent definition of the service which will meet that need, i.e. into what we have called the functionality. This work is also sometimes called requirements elicitation, in the INCOSE *Systems Engineering Handbook* it is called the Stakeholders' Requirements Definition Process, and a detailed process is developed in [8]. The resulting document, which includes the definition of the functionality, is often called the Requirements Definition Document, or RDD.

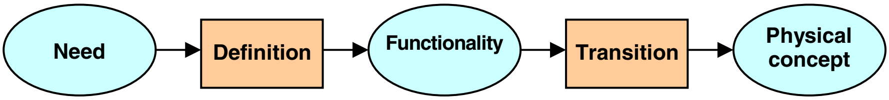


Fig. B4.5 Subdividing the first step of the design process.

The box labelled "Transition" represents the work involved in carrying out the mapping from the functional domain into the physical domain, and despite that, as a transition, this involves both domains, we shall call this work *design in the functional domain*. (The reason for this will become clearer in Part C.) The result of this work is, per Fig. B4.3, documented in the System Specification, and referring to the same figure, it would appear that architecting, design in the functional domain, and transition are all synonymous. The reason for keeping all three of them, and the distinction between them, will be explained in Part C; here we want to focus on the transition aspect, because the ability to make such a transition from one domain into another is a major, if not *the* major, characteristic of creativeness. It is what a painter does in transforming an idea or a vision into a painting, it is what a composer does in transforming an idea or a mood into a score, and it is what Gruber [2] called *insight*, or seeing the relevance of something (that exists in another context) to the purpose at hand. However, while in art there may be little to restrict or guide this transition, in engineering there is a definite restriction on the relationship between a functional element and its physical realisation, arising from the fact that the service is always defined in terms of real, measurable, physical parameters. This may be illustrated by a very simple example: The required service is that of decorking a bottle, and the basic physical parameter is the force applied to the cork; it must exceed the friction force holding the cork in the bottle. The physical realisation then becomes the means of generating that force, and as we know, there are many different ways of doing that, including a two-pronged device that can be inserted between the cork and the bottle, a thin needle with an attached gas bottle, and, of course, the good old corkscrew. So, the transition takes place via the physical process of applying a force to the cork, and this can be generalised, as shown in Fig. B4.6.

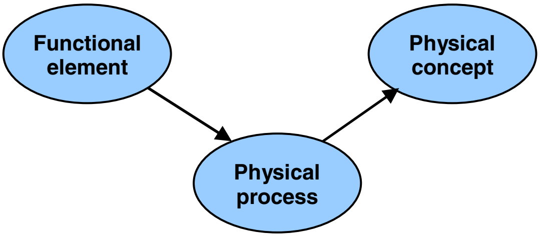


Fig. B4.6 The transition from the functional to the physical domain takes place via the physical process involved in providing the service.

In general, the service will not be provided by a single, simple, physical process, and this means that the functionality will be complex, requiring many parameters and relationships between them. It then becomes increasingly difficult to make the transition and to find a physical realisation that provides a satisfactory service, and even when one is found, it is difficult to ascertain if it is an optimal solution. Our understanding of the system approach to handling complexity then leads us to ask if it would not be possible to describe the functionality as a system of smaller, simpler, but interacting functional elements, and make the transition for each one individually. Answering that question is the subject matter of Chapter C3.

References

1. Bohm, D.: In: Nichol, L. (ed.) On Creativity. Routledge, London (1998)
2. Gruber, H.E.: The Evolving Systems Approach to Creative Work. In: Wallace, D.B., Gruber, H.E. (eds.) Creative People at Work. Oxford University Press, Oxford (1989)
3. Ertas, A., Jones, E.C.: The Engineering Design Process. Wiley, New York (1993)
4. Sawyer, R.K.: Explaining Creativity: The Science of Human Innovation. Oxford University Press, Oxford (2006)
5. Holt, K., Innovation: A Challenge to the Engineer. Elsevier, New York (1987)
6. Taylor, D.W.: Thinking and Creativity. Annals of the New York Academy of Science 91, 108–127 (1960)
7. Aslaksen, E.W.: Designing Complex Systems – Foundations of design in the functional domain. CRC Press (2008)
8. Palazzi, W.D., Aslaksen, E.W.: Requirements Definition: Evaluation of a Methodology With Client Feed-Back, internal report. School of Electrical Engineering, University of Technology, Sydney (June 1996); and reproduced in E.W. Aslaksen, The Changing Nature of Engineering. McGraw Hill, Sydney (1996)

B5 Requirements Definition [1]

B5.1 Introduction

In Sec. B4.2 we saw how a set of stakeholder requirements, as contained e.g. in a Requirements Definition Document (RDD), forms the starting point of the design process. The requirements document the interface between the stakeholders and the engineers, and any incompleteness or ambiguity in the requirements is bound to result in a less than satisfactory progression of the project. It is therefore not surprising that the process of defining and controlling requirements is a subject of much interest and activity in the engineering community, and numerous references can be found in [2]. A lively on-line discussion group is the INCOSE Requirements Working Group (see the corresponding web site at www.incose.org/).

A central aspect of the process of requirements definition is the manner in which the requirements are expressed and documented, and a review of the literature over the last fifty years shows a significant shift in emphasis. In the 50ies and 60ies, considerable effort was spent on developing a usage of the natural language, e.g. English, suitable for technical writing, both in educational institutions (a typical textbook for a two-semester subject was [3]), in industry (for example, the Bell Laboratories Graduate Study program, mandatory for all engineers below PhD level), and, in particular, in the US Department of Defence (as evidenced by e.g. MIL-ST-961, MIL-STD-490, and MIL-HDBK-63038-2). But with the rise of software engineering, and with the significant problems experienced in converting requirements into code in a controllable and efficient manner, there has been a gradual shift in emphasis, to the extent that, at least in electrical engineering and systems engineering, requirements specification has almost become synonymous with software requirements specification. In the case of systems engineering this is particularly ironic, because if there is one engineering discipline that is concerned with a holistic view and with engaging a wide segment of society, it is systems engineering, and it needs to use a means of communication that is acceptable to all within this segment (see also [4]).

The process of engineering is a step-wise process of converting user requirements into requirements for the fabrication or construction of the equipment or works that will meet the user requirements. The requirements are written in a language that is appropriate to their use, and therefore, towards the fabrication end of the process, the language becomes specialised to the fabrication process and relies heavily upon graphics (drawings, diagrams, and the like). Mechanical engineering, power engineering, electronics, civil engineering, etc. all have their specialised languages that have developed over a long time, in some cases centuries.

When computers arrived on the scene, a completely new and different situation arose. To design and fabricate the computer, the existing language for electronics was adequate (with appropriate extensions). But in addition, one now had to communicate with this entity and tell it what to do in order to perform its intended function. Until then, all communications in the process of engineering had been between humans; there had never been any question of communicating with a bridge or a diesel engine. The problem now was that the computer understood only an extremely simple language, and so needed specialised people who could translate between English and machine language. For well-known reasons this was a highly unsatisfactory situation, and mitigation took place on two fronts. On the one hand, the computer was enabled to do part of the translation itself, thereby narrowing the gap between its programming language and English. On the other hand, the process of bridging this gap was subdivided into two (or more) steps, with the intermediate result formulated in a language somewhere in between English and the programming language. As in the other engineering disciplines, these intermediate languages often made considerable use of graphics, and a recent example is the Unified Modelling Language (UML).

When a new situation arises, or a new invention or theory is put forward, it is natural to at first become so absorbed in the novel aspects that one completely overlooks the aspects it has in common with existing practice and knowledge. Software engineering has been no exception, and it has been nothing if not amusing to observe the glee (even, at times, pompousness) with which software engineers have rediscovered such concepts as objects and reusable modules; concepts that have been fundamental to engineering for a very long time. Just take a normal machine screw; it is an object belonging to the class "screws" which belongs to the class "fasteners", it has attributes (diameter, length, thread length, head type, material, surface finish, etc.), and is certainly reusable. If we did not have standardised components, but had to design and specially manufacture each one as we needed it, engineering would be hopelessly inefficient, as we shall address in the next Chapter. We are now just waiting for the software engineers to catch up, so that when we need a piece of software to perform a particular function, we go out and buy a couple of hundred standard software modules, link them up in the right order, and that's it [5].

But, back to reality and on a more serious note, the current trend towards applying what has been developed for the special case of software engineering to the general case of engineering, and in particular to the most general case of systems engineering, must be viewed with some scepticism. Graphics can be a very useful illustration of relationships and structure, but to display in a one-page use-case diagram what is much more adequately described in about ten lines of text makes no sense, and the little stick man popping up everywhere is frankly beginning to look a bit ridiculous.

B5.2 The Basics

Let us now return to the basic case, where the human mind is the "computer" that processes data between the input and the output of a step in the engineering process and then transmits the data to the person carrying out the next step, using English as the transmission language. What is the machine language of this computer? What is the nature of its variables, and what is its instruction set? The "final" answers to those questions are still shrouded in mystery, but at a reasonably high level (say, corresponding to a high-level computer language) the variables are words. A word is the smallest unit to which one can give a meaning, and thus words provide the basic elements of semantic interpretation.

The two principal ways in which two humans can communicate using English is shown in Fig. B5.1. They are distinguished by the physical nature of the signal transmitted between the two persons; in the one case it is an acoustic signal, in the other case an optical signal. Aside from this difference there is, however, another one which is much more significant - in the case of the acoustic path the human organism contains the two transducers required. The vocal tract produces sounds that are immediately recognisable by the ear, whereas in the case of the optical path, one needs the intermediate artefacts of pencil and paper (or their equivalent) in order to achieve a reasonably efficient encoding of the signal. (Sign language being so inefficient as to be uninteresting in the present context.) The visual sense has a much higher information acquisition capability than the auditory sense, but it is the presence of matched (in the sense of encoding) receiving and transmitting capabilities that makes the spoken language the primary representation. This may be what makes the human species unique. And, to continue this line of thought a little further, if humans had been provided with integral visual display units that directly put thoughts into pictures, it could well have been the optical channel that would have been the primary one, and transmission speeds could have been orders of magnitude higher. (Of course, considering how some people turn on their speech-apparatus without first switching on their brains, this enhanced capability could have some very unfortunate unintended consequences.)

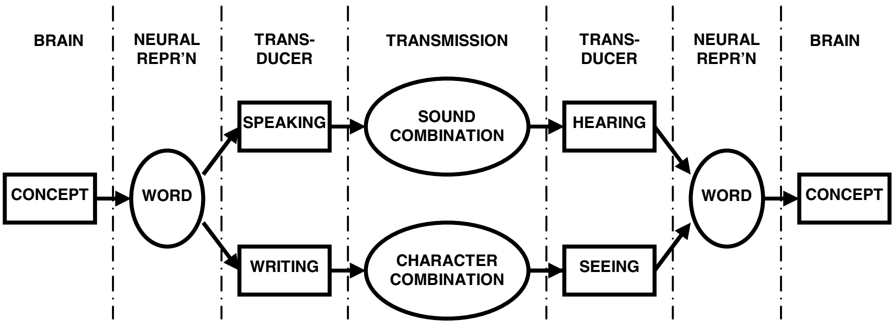


Fig. B5.1 Block diagram of linguistic interaction, showing the two physical channels between two brains. The brain transforms concepts into neural representation of words, this is converted to a physical signal by the transducer, and on the receiving end this process is reversed.

While the encoding of the neural representation into a physical signal and the corresponding decoding in both the acoustic and visual channels is an interesting subject, we need not consider it any further here. It is in the process of putting the concepts, i.e. what we understand, into words and vice versa where the problems associated with writing requirements come in. Whereas computer languages are built up in a hierarchical fashion, with the concepts on one level being defined in terms of the simpler concepts on the level below (e.g. macros in an assembler language being defined by machine language instructions), English has a completely different structure. There are really no “levels” beyond or within the set of words; there is no visible structure that expresses the relation of simple words to complex words (measured by the complexity of their meaning). Except for certain combinations, such as downfall, interrelation, overcompensate, etc., there are no rules for constructing complex word from simpler words (although this varies considerably from one natural language to another, with e.g. German containing a lot more combinations of the type listed above than does English). The length of the word is not even significant; the noun “God” and the verb “to think” both represent highly complex concepts, whereas the noun “tomorrow” represents a simple and immediate concept. Of course, words are combined into sentences, sentences combined into paragraphs, and so on in order to express more detail and provide a greater depth of definition, but this is done by creating “new” relations between the same type of elements - the words - whose meanings are again defined by such relations, and so on. It is a circular process, much subtler than the straightforward logic of computer programming.

There are also some further problems associated with the translation of concepts into words, such as the lack of boundedness in English. There is in principle no limit to the number of words in a lexicon, nor to the length of a sentence. And, perhaps most importantly, the meaning of a sentence, its *semantic interpretation*, is dependent on the wider context in which it is embedded and on the background of the reader. In addition to the *literal meaning*, an inherent property of the sentence that is independent of the context in which it finds itself, there are the *pragmatic implications*, information conveyed by the sentence when it is combined with all other knowledge available to the reader at the time of reading. So, there it is no wonder that writing requirements in English is problematic, but that is in itself no reason to give up on trying to overcome or, at least, reduce these problems.

B5.3 Using Natural Language

Before going on to outline how some of the problems encountered when using the natural language, in our case English, to express requirements might be reduced, we need to be clear about what the advantages are, so that we do not inadvertently negate them.

a. It is the only language (aside from a simple sign language used e.g. for traffic signs) that is understood to a significant degree by the whole population.

b. English has a long history, and there is a great deal of information available about it in the form of dictionaries and thesauri, numerous books and papers on its grammar and semantics, and on its use in various applications (such as technical writing).

c. It is much more powerful than any other language. Its richness in words and in the ways these words can be combined is unparalleled, and its ability to develop along with our expanding knowledge and intellectual capability ensures that it is never outdated.

d. It is universally applicable, used both in specialised applications (e.g. to express technical requirements) and in all areas of everyday life. Therefore, as we all get constant practice, maintaining our proficiency requires relatively little additional effort.

There are probably a number of other advantages that could be put forward, but already the ones listed above suffice to demonstrate what an extraordinary means of communication English constitutes. It should, therefore, not be surprising that, in order to get the full benefit out of it, one needs a considerable amount of instruction and training, but somehow this is not a truth that enjoys much popularity these days. There is a feeling that a natural language should be just that: Children pick it up by listening and practice, and any further formalisation only serves to stifle their individuality and creativeness. This is not the place to discuss educational policy, but this attitude must surely be quite at odds with the way we, as engineers, would approach the matter. If we have a complex piece of equipment - say, a sophisticated, numerically controlled machine tool, or perhaps an aeroplane - would we let somebody operate the equipment just from looking at how a skilled operator does it? Without any understanding of its internal workings, the principles underlying its operation, the designer's intent, its inherent limitations, and so on? Of course not; the exclusively on-the-job or "sit by Jenny" approach to training went out the door long ago. But this is, to a large extent, the situation with English today, and the frustration expressed with the "limitations" and "imprecision" of English within the engineering community is due more to an inability to exploit the full power of English than to any shortcomings of English itself.

The approach to realise these benefits is really just a continuation of the work done in the 50ies and 60ies, but with some guidance from the work done in linguistics in the meantime, not least the work of Chomsky [6]. It can be considered to consist of two parallel paths, a syntactic path and a semantic path, and they are outlined very briefly in the following.

If one is asked to define a semantic element, say X, the answer is usually a sentence of the form "X is a Y which {}". Here Y is the *class* to which X belongs; it is a more general element. To it are related a large number of qualifying or specialising elements, and {} is the appropriate subset of these. Thus, the definition process consists of three subprocesses: First, the *generalisation*, which places the element into its class, then the *analysis*, which determines all the characteristics of this class, and then the *specialisation*, which chooses the appropriate characteristics for the present case. This is illustrated in Fig. B5.2, which also indicates that in order to be able to carry out the specialisation properly, i.e. to make the right choice among the possibly very large set of characteristics, the context in which the element is used may have to be considered.

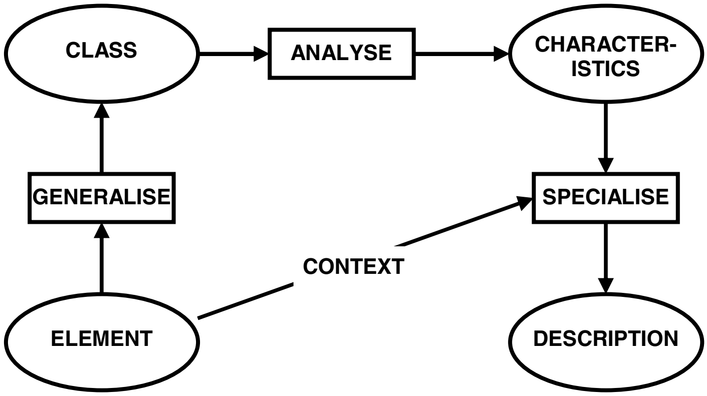


Fig. B5.2 The concept definition process, consisting of three subprocesses and involving four entities.

The following definition provides an example:

> The *transfer function* of a two-port is a complex-valued functional, F, of a complex-valued function, the input amplitude spectrum g(ω), where ω is a positive, real variable called the (angular) frequency, such that a voltage source with amplitude spectrum g(ω) and internal impedance z_0 connected to the input port results in an output voltage spectrum F(g(ω)) across a load impedance z_0.

The generalisation is making the element a functional, the analysis is characterising a functional as a mapping from one set of functions to another, and the specialisation consists of specifying the type of functions and the boundary conditions, i.e. the impedance level. Within the area of electrical engineering, this is a context-free definition.

Thus, a definition, as it appears in a requirements definition, is the result of this definition process, and the subprocesses described above find their expression in different parts of the definition, leading to an ordering or *higher-level syntax*. The two main parts of the definition are:

- The *classification*, which describes the class to which the element to be defined belongs, and
- the *relations*, which relate the element to other, already defined elements.

It should be noted that the relations mentioned here serve a purely semantic purpose - to define the *meaning* of an element, as discussed further below. They must not be confused with the relations that arise as a result of the top-down systems engineering process, and which express the structure of the system being designed. There is no engineering involved in or implied by the definition process at all; it is a secondary process providing a necessary support to the engineering process by defining the concepts with which the latter operates.

The previous process of defining new concepts, and which was reflected in a higher-level syntax, leads one intuitively to a semantic ordering. It would be natural to say that a concept defined in terms of a number of other concepts is in some way more *complex* than the latter. In this manner, concepts become ordered by the number of definition processes they are removed from some initial set of words or concepts. But how is such an initial set determined? An obvious choice would be the set of words used frequently, in everyday speech. That set would differ greatly between a mathematician, a priest, and a longshoreman, but taking the intersection of all such sets for different users, one could come up with a basic set of words.

On the other hand, one could look at the set of words whose semantic components or meanings are, at least to some high degree, independent of the context in which they are used. The intersection of this set with the basic set is a *primitive set*; it is made up of all those words well known to and assigned the same meaning by all persons regarded as having some defined minimum degree of competence in English.

This ordering makes the set of all words appear as an unbounded sphere, with the primitive set as a central core and then layer upon layer of words of increasing complexity. The direction in which one progresses outward represents the particular profession or area of application, and within such a cone there is a further subdivision into the particular application or context. This is illustrated in two dimensions in Fig. B5.3. However, as one progresses outward in any direction, the meanings of the words become more and more context-dependent; the sphere is decomposed into finer and finer cones or fibres, between which there are significant differences in the meaning given to the same word. This is on the one hand a result of increasing specialisation and lack of communication between professions, but on the other hand, and more importantly, it is an inherent function of the increasing complexity, of the increasing richness of the concepts. Even for the specialist it becomes impossible to give an abstract, context-free meaning to concepts far removed from the primitive set. This point was discussed in [7].

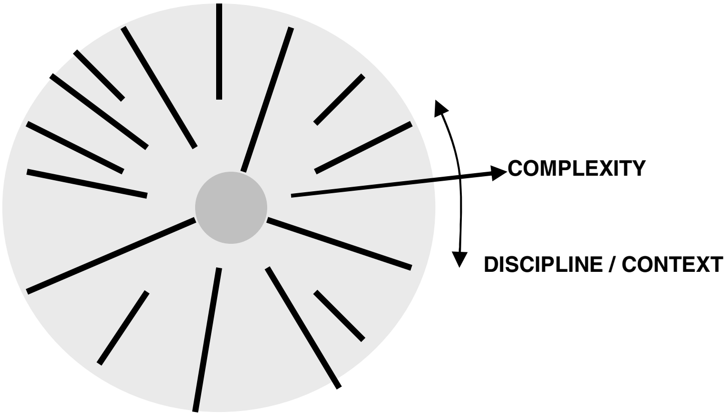


Fig. B5.3 A two-dimensional representation of a lexicon, parameterised by the degree of complexity and discipline (or context). The radial lines separate different disciplines, and the darker grey area in the middle represents the primitive set.

Introducing an arbitrary measure of context dependency, κ, normalised to $0 \leq \kappa \leq 1$, then κ will be a function of the distance, *r*, from the centre of the sphere, as shown in Fig. B5.4. That is, this figure is meant to demonstrate what is the general trend (but not an absolute rule) for complexity and context dependency to be related and increase together. Up to a certain radius, r_0, there is no context dependency; this is the primitive set. (The value of r_0 should be increasing steadily with time due to an increase in the level of general education, but there is some doubt as to whether that is true or not.) From here on the value of κ rises slowly until, at $r = r_k$, it rises rapidly towards 1. As the value of *r* increases in the range $r_0 \leq r \leq r_k$, persons with increasingly specialised knowledge will automatically infer the correct pragmatic implications, thus allowing virtually context-free definitions. In this range it therefore becomes a question of correctly identifying the degree of specialisation of the person who is to read the document. For *r* greater than r_k even the specialist will need to know the context in order to give precise meaning to the concepts; using abstract concepts alone will lead to ambiguity.

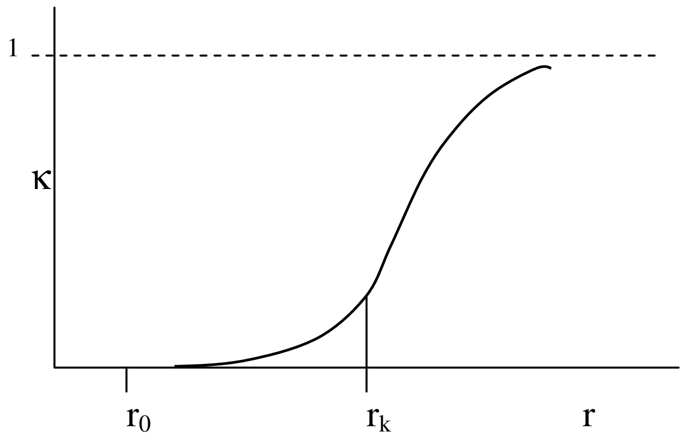


Fig. B5.4 The increase in context dependency, κ, (arbitrarily normalised to unity) as one moves away from the primitive set, with complexity $r < r_0$. For $r < r_k$ context-free definitions are possible within areas of specialisation; beyond r_k the context will need to be specified along with the requirements.

For engineering, which often deals with concepts in the region $r > r_k$, the implications are clear:

a. Before producing a requirements definition document, we need to be clear about the intended readership.

b. Where the meaning of a concept depends on the context, *and* the readership cannot be reasonably assumed to automatically infer and have an understanding of that context, it has to be described as part of the document.

It is the second implication that seems to cause engineers the most frustration; on the one hand they would like to have very minimalist formulations of the requirements, on the other hand they would like their requirements to be understood by and useful to a wide audience, which is appropriate to such a multidisciplinary activity as systems engineering. The complaints about the "fuzziness" of English are often a reflection of the inability (or unwillingness) to write a good description of the context, and the expectation of using brief, formal formulations of requirements is a reflection of confusing humans with computers. We *tell* a computer what to do, but a human needs to *understand* what to do, and that takes a few extra words.

References

1. Much of the material in this Chapter was presented in Aslaksen, E.W., Requirements Definition - A Plea For a Return To English. In: Proc. Thirteenth Int'l Symposium of INCOSE, Arlington, VA, July 1-3 (2003)

2. Some papers that contain substantial lists of references are: van Lamsweerde, A.: Requirements Engineering in the Year 00: A Research Perspective. In: Proc. 22nd Int'l Conference on Software Engineering, Limerick. ACM Press (June 2000); Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: Finkelstein, A. (ed.) ICSE 2000, Future of Software Engineering, Limerick, Ireland, June 4-11. ACM Press (2000); Gambhir, S.G.: An Investigation of Facilitor-Assisted and CONOPS-Based Requirements Elicitation Methods Using a 2x2 Factorial Experimental Design. Systems Engineering 4(4), 272–286 (2001); A more recent book is that by Hood, C., Wiedemann, S., Fichtinger, S., Pautz, U.: Requirements Management: Interface Between Requirements Development and All Other Engineering Processes. Springer, Berlin (2007)
3. Thomas, J.D.: Composition for Technical Students, revised. Charles Scribner's Sons, New York (1957)
4. Alexander, I.: The Glamour of Formalisation - A Warning, `http://easyweb.easynet.co.uk/~iany/consultancy/glamour.html`
5. The issue of software as a special case within engineering is the topic of the article by Aslaksen, E.W., Why Software Is Different, Systems Engineering Society of Australia (SESA) Newsletter No. 48 (July 2009)
6. Introductions to the relevant parts of Chomsky's work can be found in Chomsky, N.: Syntactic Structures. Houton & Co., S'Gravenhage (1957); 2nd printing (1962); Chomsky, N.: Language and Mind, Enlarged Edition, Harcourt Brace Jovanovich, Inc. (1972); and Smith, N., Wilson, D.: Modern Linguistics. Penguin Books Ltd. (1979)
7. Aslaksen, E.W.: Systems Engineering and System Specification. J. Electrical and Electronics Eng. Aust. 7(3), 159–165 (1987)

B6 Standardisation and the Bottom-Up Design Process

B6.1 The Engineering Body of Knowledge

Every profession is based on a Body of Knowledge (BoK), and from our brief survey of the history of engineering in Chapter B1 we understand how the engineering BoK developed, and that it is now both very extensive and rapidly expanding. And it is obvious that such a vast body of knowledge (and data) can only be useful if we have some means of finding the information we need for a particular application, so that the BoK must be *structured* in some way. It is therefore not surprising that there is a strong connection between the structure of the BoK and the manner in which engineering is carried out, in particular, how design is carried out, and this relationship works both ways. On the one hand, we approach a design problem in a manner dictated largely by the structure of the BoK; on the other hand, the structure of the BoK evolves in response to the changing nature of the problems engineering is called upon to solve. So, understanding the structure of the BoK must be an important component of our understanding of engineering as a profession.

The most immediate and obvious feature of the structure is the division into *disciplines*. Traditionally these have been civil, structural, mechanical, electrical, and chemical engineering, but as a result of the growth of the BoK, further *specialisation* is taking place, resulting in such disciplines as electronics, aerospace, naval, mechatronics, and biomedical engineering, just to name a few. While all these disciplines have a common base in natural science and mathematics, the specific engineering BoK within each discipline is related to the types of physical entities with which the discipline is concerned. For example, civil engineers are basically concerned with modifying the shape of the terrain; providing flat areas for buildings, suitable paths and foundations for roads and railways, channeling water to provide drainage or canals, etc. Consequently, their BoK consists of knowledge about the properties of the materials making up the terrain, such as soil properties and soil dynamics, properties of such building materials as concrete and bitumen, techniques for stabilising embankments, the dynamic effects of flowing water on the terrain, and so on. But in addition to this grouping, there is the fact that the environment in which the engineering takes place is different for each discipline. Civil engineering projects take place on work sites, electrical and mechanical engineering projects take place (mainly) in factories, and chemical engineering projects take place in process plants. These differences have led to distinct work methods and what might be called distinct *engineering cultures*, which also form part of the BoK of the disciplines and add a further aspect to the structure of the BoK.

However, there is another dimension to the structure of the BoK, and it is common to all of the disciplines. It arises from the fact that engineering evolved from a craft to a profession when the knowledge of the craftsman was placed in a formal framework based on natural science, with mathematics as a major component of that formalisation, as we saw in Chapter B1. The manner in which that took place, and in which it has taken place ever since, is through what we shall call *construction elements* (although they are called by many other names, such as components, building blocks, unit processes, etc.). Examples of such construction elements are culverts and soil nails in civil engineering; beams, columns, and plates in structural engineering; bolts, shafts, and bearings in mechanical engineering, capacitors, resistors, and transistors in electronics engineering; transformers, generators, and cables in electrical engineering, and distillation, filtration, and reaction in chemical engineering. Through the introduction of these construction elements, engineering separated into two more or less distinct groups of activities; the development of new elements, and the use of these elements in solving design problems. This is what we identified as the development of technology and the application of technology in Section B4.2.

Now, while there is certainly an important design activity involved in creating new construction elements, this only takes place once per element, whereas the element may be used thousands of times in the design of new applications. As a result, engineering design has become largely the art of choosing and combining such construction elements to achieve a required performance, and this is reflected in the education of engineers. Following a grounding in physics, chemistry, and mathematics, this knowledge is used to give the students an understanding of how the construction elements work and how they are designed and manufactured. But the emphasis soon turns to characterising the elements by their external characteristics, the knowledge required to be able to combine them with other elements to form a design solution. For example, most electronics engineers will look upon a transistor as a 3-port with transfer functions characterised by a few parameters, and it is unlikely they have anything but the vaguest recollection of energy band gaps, doping levels, and the like.

With this understanding (and simplification), engineering can be presented in the manner shown in Fig. A6.1, with the construction elements taking on the role of an interface between the two main engineering activities.

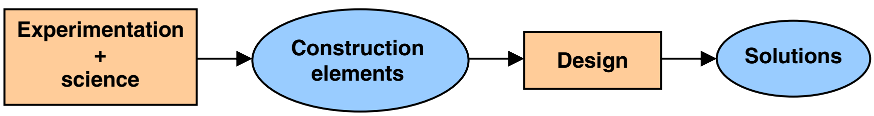


Fig. B6.1 A representation of engineering, consisting of two groups of activities and two types of artefacts.

As a consequence of this view of engineering, the BoK can be subdivided into three main parts:

- Knowledge about the development of construction elements and their inner workings;
- knowledge about the external properties of construction elements; and
- knowledge about how to apply construction elements in order to achieve solutions; i.e. knowledge about the process of design.

Before we go on to examine what this structure means for the process of design, we need to take a step back and comment on a couple of issues that were neglected in the above very brief discussion. The first of these, which has probably occurred to you, is that the focus and the examples were concerned with hardware, whereas today software engineering is a major part of engineering. However, the situation in software engineering is much the same; the construction elements are algorithms, modules, and standard applications, such as sorting algorithms, fast Fourier transform algorithms, database frameworks, Active-X controls, Excel, etc, and the majority of software engineers are involved in designing custom or one-off applications using such algorithms and standard software.

The second issue is that there are obviously relationships between construction elements. For example, while capacitors and inductors are construction elements, so is a band-pass filter consisting of a few capacitors and inductors. And while a transistor is a construction element, so is an integrated circuit, such as e.g. a quad dual input AND gate, consisting of about 40 transistors, and so is a microprocessor, consisting of maybe a million transistors. There is an informal hierarchical ordering of the construction elements, and the boundary between what is a construction element and what is a new construction using these elements is a fluid one, constantly moving towards greater complexity.

B6.2 The Bottom-Up Design Process

Taking the picture presented in Fig. A6.1 at face value, design becomes a matter of picking the right subset of construction elements and combining them in the right way. That is, the design process is one of *synthesizing* a new object out of the set of construction elements, such that this object will provide the required service. There are numerous techniques and heuristics for how to carry out this process, but what they all have in common is that they start from an initial set of elements. This initial set exists due to the fact that the design problem is defined in terms of a particular type of object; for example, it is defined in terms of designing a car, or a boat, or a bridge. The initial set is then the set of elements that were used to construct this type of object in the past.

From that initial set, the design progresses through a step-wise, iterative process, with each step consisting of three basic activities:

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

This iterative process is shown diagrammatically in Fig. A6.2; this figure also indicates where one enters and exits the process. While the first selection of elements is normally from the initial set, the designer is, in principle, free to include any available element in an iteration, and the process ends when the performance is tolerably close to the required performance.

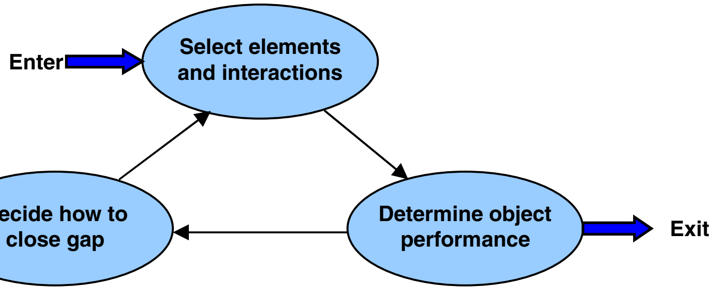


Fig. B6.2 The iterative design process.

This seemingly simple process has many complex issues associated with it; in the following we consider these only to the extent relevant to our purpose of examining the use of the system concept in engineering in Part C.

Within the activity of deciding how to close the gap we can identify three basic approaches. One is to modify the interactions within the set of elements, without changing the set of elements. This can mean changing the structure of the system, as for example in changing the organisation of a team. The persons remain the same, but the performance of the team is changed by changing the flow of information between them. It can also mean changing the strength and/or type of the interaction; interactions are characterised by a number of parameters, and we have a choice of their values. For example, communications between persons or organisations can be verbal or written, and can take place with varying frequency. Interactions between electronic modules can be analogue or digital, and with varying bandwidths or bit rates. The interaction between two structural components may be bolted, welded, or glued, and so on.

Another approach is to change the set of elements; making a different choice from the set of existing construction elements. The most immediate action would be to look for other elements within the initial set, but innovative solutions require consideration of elements outside this set. As that is a vast population of elements, some form of methodology or process is needed to search for appropriate elements, and a number of these are available. An example of such a methodology is TRIZ, developed originally by Altshuller [1]. Instead of focusing on the initial set of elements used to construct objects of the same type as the object to be designed, it focuses on the set of problems the object needs to solve, and then searches for cases where the same or closely related problems were solved, but in different contexts. This identifies a new set of elements that can then be assessed for their usefulness in solving the current design problem.

Clearly, any search for suitable existing construction elements can be viewed as an exercise in information processing. This aspect is discussed in [2]; a book that also gives a good account of numerous other approaches to solving design problems, albeit with a strong focus on mechanical design.

The third approach is to identify a new, or ideal element that would close the performance gap, and then develop a new element with a performance that approaches this ideal to the extent that it is economically viable.

An important factor in choosing which one of these approaches to employ is how close the performance of the initial set and interactions is to the required performance. If it is close, we would employ the first approach, and this is the case in most engineering design work; our designs are based largely on previous work and our experience. If it is not close, we would search among the set of existing elements for some replacement or additional elements, and if the performance gap is great, we would identify what element, or set of elements, would ideally be required to provide the sought-for performance, and then go back to the technology developers with a request for a new construction element [3].

The activity of determining the object performance may not be straightforward. A project will, in general, have numerous stakeholders, and while there has, hopefully, been a process of requirements elicitation and definition preceding the start of design, once a particular solution emerges there may be different views of what its required performance is and how to measure it. And the activity may involve a process of reaching consensus rather than using a quantitative evaluation model. Even worse, the definition of the required performance may change with each pass through the process in Fig. A5.2, as the stakeholders get a clearer view of what the solution involves.

The activity of deciding how to close the gap is best illustrated by considering a very simple (and simplified) case; a design involving a resonant circuit. The resonant frequency of the circuit is given by the relation $\omega = 1/(LC)^{1/2}$, and initial values of L and C were chosen (from the set of standard components) to give what was thought to be the required frequency. However, it is then found that the design can be improved by changing the resonant frequency. Given the above relation, that appears a trivial matter, but not so; there are a number of combinations of standard inductors and capacitors that will give the new frequency. Which one is the least costly? Which one takes up less space? Which one has the lowest weight? Depending on the importance of each of these decision criteria, the selection process can be quite involved, and it is easy to see how the effort required increases dramatically with an increasing number of variables.

B6.3 Standardisation

It is easy to overlook the importance of standardisation and standardised construction elements and the role this has played, and is playing, in handling complexity within engineering projects. We are so used to having a wealth of such elements at our disposal that we forget the tremendous amount of work that has gone into bringing this about and, perhaps more significantly, that this is an ongoing process that shapes the nature and content of engineering. The first step in this process is the recognition of *function*; a repeated demand for a certain relationship between physical bodies and/or variables. For example, the function of a resistor is to provide a linear relationship between voltage and current, and the demand for this relationship arises in a great number of different applications. The function of a bolt is to transmit a force between two static bodies and so maintain their relationship; the function of a bearing is to transmit a force between a fixed and a rotating body and so maintain their relationship.

The next step is how to *parametrise* the relationship. In the case of the resistor, resistance, tolerance, and power rating are the three most common ones, but over time a host of other parameters have arisen in response to new and more detailed demands and new technology, including temperature coefficient and operating range, parasitic inductance and capacitance, size, and weight. Add to that the various manufacturing and packaging technologies, ranging from resistors on integrated circuits to liquid motor starting resistors, and the number of different resistors (i.e. differing in at least on parameter value) is staggering.

This leads us to the third and crucial step in the process, the actual standardisation. This is the activity of forming classes of elements that differ in the values of only a few of the parameters; the others being defined by agreement between the majority of users and documented in a publicly available document. This activity is carried out by a variety of bodies; most well-known are the major international standards organisations, such as the International Standards Organisation (ISO), the International Electrotechnical Commission (IEC), and the International Telecommunication Union (ITU), national standards organisations, such as the American National Standards Institute (ANSI), Verband Deutscher Elektrotechniker (VDE), and Deutsches Institut für Normung (DIN), and organisations developing industry-specific standards, such as the Institution of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF).

In addition to these well-know bodies, there are numerous industry- and product-specific organisations that carry out this activity, and the results are documented in a variety of forms, such as data sheets, specifications, and catalogues. What is common to all variants of this activity is that it involves *information hiding* or, as we discussed in Section A3.1, *chunking*. Take just such a simple and frequently used construction element as an M4 bolt; when we use one in an equipment design, we simply put “M4 x 12, cyl head” in the parts list. But behind this sits a mass of information, such as the values of the parameters Proof Load Stress, Tensile Yield Strength, Tensile Ultimate Strength, and Core Hardness, thread shape and tolerances (as set out in ISO standards), and head dimensions and tolerances. If we had to design such a bolt and determine the optimal values of these parameters each time we needed one, design would come to a halt. *The engineering design of physical plant rests on the foundation of standardised construction elements*.

While every engineer is aware of this fact through his or her daily work, it is what might be called a "hidden awareness". In that sense it is a bit like our awareness of gravity; we experience it all the time, and of course we know that the earth is round and that we "down under" stand upside-down compared to people in Europe, but in daily life we just think of it in terms of physical objects having a weight. It is the main and recurring theme of this book that engineering has a long and extremely successful tradition, during which it has accumulated a wealth of knowledge about materials and processes and, above all, what might be considered "meta-knowledge" about how to handle and employ this knowledge in solving a wide range of problems, and that "systems engineering" is nothing but the addition of the system approach to that "meta-knowledge". One element of that knowledge is standardisation; how to do it and how to use it, and we shall argue in Part C that this knowledge remains just as applicable when using the system approach in engineering projects.

References

1. No publication by the inventor of TRIZ, Genrich S. Altshuller, is know to the author, but a translation, by L. Shulyak, of a book by his son, Henry Altshuller, is available as The Art of Inventing (And Suddenly the Inventor Appeared), Technical Innovation Center, Worchester, MA (1994); ISBN 0-96-40740-1-X. The main promoter of TRIZ is Ideation International (`http://www.ideationtriz.com`), and a good introduction is available online at, `http://www.mazur.net/triz`
2. Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H.: Engineering Design. Springer, London (2007)
3. Aslaksen, E.W.: A System Approach to Flash Dryer Design. In: 4th Asia-Pacific Conf. on Systems Engineering, Keelung, Taiwan, October 4-6 (2010)

B7 Managing the Process of Engineering

B7.1 Management Is an Inherent Component of Engineering

Throughout this Part B we have developed a picture of engineering as a process that has as its purpose the creation of products that, through their operation, produce services that meet needs expressed by society, or by groups within society. This process was illustrated in Fig. B2.2, and it is clear that, for all but the simplest products, this is a complex process, consisting of the major activities shown in that figure, and within each of those, numerous work packages. All of these work packages interact in a particular manner, forming a structure, the well-known Work Breakdown Structure, or WBS, and the outcome of the work depends on both the content of the work packages and this structure. It is what we in Sec. A4.1 called an emergent property of the set of work packages, it emerges as a result of letting the work packages interact and form a system. Thus, we realise that the system concept was in use in engineering long before the advent of systems engineering, but it was used in an implicit manner. What we shall explore in Part C are the improvements that can be achieved by explicitly exploiting the properties of the system concept.

However, irrespective of how the system concept is used or viewed, it is clear that determining what work has to be done, what skills are needed to do it, how to apportion the work to the various skill groups, what information needs to flow between these groups and at what points in time, and then how to ensure that what has been determined also actually gets done, must constitute a significant portion of work; it is what we shall call *engineering management*. This management activity is not something that stands outside of or is in addition to engineering; it is an inherent part of engineering. The great engineers, exemplified by Watt and Eiffel in Chapter B1, were also very good managers, and in general it is also true that it is not possible to be an effective and successful engineer without also being a good manager. The reason for this is to be found in the unique nature of engineering, sitting in the middle of the triangle with science, art, and trades as its corners, as illustrated in Fig. B7.1 As we have discussed throughout this Part B, engineering is about creating objects that provide a service of commercial value, and this process involves management activities at every step.

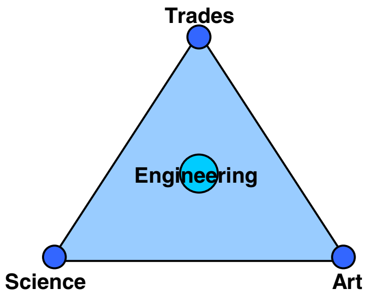


Fig. B7.1 Engineering at the intersection of science, art, and trades.

B7.2 The Industrial Environment

But before considering engineering management in more detail and, in particular, its relationship to other types of management, we have to briefly point out that engineering takes place in two distinct, but closely related environments; the manufacturing industry and the construction industry. The former is based on operating within an environment (the factory or process plant) which has been designed to produce a particular type of product, or perhaps a fixed range of products; typical examples would be an automobile plant, an integrated circuit plant, a power station, an oil refinery, and a software house. In all of these cases there is a high degree of *continuity*, of the product (new models or versions), of the manufacturing process (continuous improvement), and of the organisation, whereas in the construction industry, the engineering takes place within the boundaries of individual projects. That means that the structure and staffing are established anew each time, and while this may take place within a certain project management framework, the focus is on completing each project within its defined budget and timeframe. Each project may have a different owner, and many construction projects are one-off, so that there is less incentive to invest in improvements for future projects.

Of course, there are projects carried out within the manufacturing environment also, so there is no sharp boundary between the two environments. But there are some subtle differences between what is called *line management* and what is called *project management* that need to be recognised, and we shall see that they influence the manner in which the systems concept is applied.

There is also a psychological aspect to the difference between the two industries, and it arises because the outcome of a process to create a new object is a random variable. The more one invests in the design and control of the process, the higher is the probability of a successful outcome. In a manufacturing process that produces thousands or more of identical objects, it is a straight forward matter to determine the optimal investment in the process; it is the investment up to the point where the cost of any further investment exceeds the savings resulting from the resultant lowering of the product rejection rate, and this investment may be many times the cost of a single object. However, there is *uncertainty* attached to the investment decision, and this is best illustrated by a highly simplified example. Consider a facility that is to manufacture 10,000 units of a product, with each unit returning \$125 to the manufacturer. As it stands, the capital cost of the facility (including all engineering) is \$800,000, and each unit will incur \$20 in material and labour costs. The rejection probability has been determined to be $\lambda = 0.05$, and rejected units have to be disposed of at a cost of \$5 per unit. Also, at the end of the production run, the facility has no residual value. An improvement to the product design is now proposed that will lower the rejection probability to $\lambda = 0.01$. How does the cost of the improvement, C, influence the decision to implement it?

The net return (or profit) from the manufacturing operation, y, is the revenue minus the cost, both of which are functions of the number of rejected units, x, and is given by the expression

$$y(x) = 125(10^4 - x) - 10^6 - 5x = 3 \cdot 10^5 - 130x \ .$$

The number of rejected units is a random variable, with a probability density function $p(x)$ given by the binomial distribution,

$$p(x) = \binom{10^4}{x} \lambda^x (1-\lambda)^{(10^4 - x)} \ .$$

Let the number of rejected units without the improvement be denoted by x_a and the number of rejected units with the improvement by x_b, then a particular case of comparing with and without the improvement is defined by the two-dimensional vector (x_a, x_b), and the probability density function for such cases is given by the product $p(x_a) \cdot p(x_b)$. The corresponding benefit of the improvement, $z(x_a, x_b)$, is given by the expression $130 \cdot (x_a - x_b)$, and the resultant probability density function for $z(x_a, x_b)$ is shown in Fig. B7.2.

We see that the decision is fairly straight forward in this case; there is a clear boundary between GO and NOGO at a value of C around \$40,000, and no one would seriously consider going ahead on the basis that the benefit *might* be close to \$60,000.

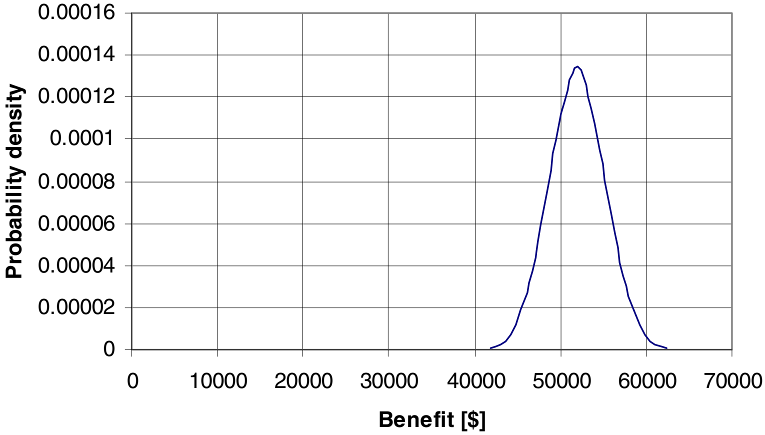


Fig. B7.2 The probability density function of the benefit of the proposed improvement to the product design.

Consider now the case of a construction project, e.g. the design and construction of a process plant, or even a software application, and for the sake of this comparison, let the total cost equal $1,000,000, of which 100.000 is for design. Again, there is a probability that the outcome will be unsuccessful, in which case the whole cost must be written off as a loss, and as this is quite an unprecedented product, the probability might be quite high, say 0.2. However, the return from a successful product is correspondingly high, say, resulting in a Net Present Value of $500,000. It is then proposed that if an additional $100,000 were to be spent on investigations and design development, this probability could be reduced by a factor of five, i.e. to 0.04.

In this case we cannot really speak of "most likely outcome", because the outcome of not accepting the proposal will be an NPV of either $500,000 or -1,000,000, and the outcome if the proposal is accepted is an NPV of either $400,000 or -1,100,000. But if we use the weighted average as the criterion for a rational decision, the weighted NPV is $200,000 if the proposal is not accepted and $440,000 if it is, and on this basis the proposal would be accepted. However, a very likely reaction to the proposal is the following: "What, double the design costs? Those engineers are trying to rip us off again! Four out of five looks like pretty good odds to me; let's just go ahead." The fact is that most entrepreneurs have a strong gambling streak.

B7.3 The Special Features of Engineering Management

At a first, somewhat superficial glance it might appear that management is a discipline in itself, independent of what area of activity or industry it is applied to, as there are a number of processes that are common to management within any area. These include the management of human resources (remuneration, motivation, training, etc.), commercial management, sales, contract management, and governance, and also such processes and tools as planning and controlling. But only in few cases is it possible to separate the performance of these activities from what is being managed; one such case is perhaps health care administration, where e.g. the person in charge of a hospital does not have to be a medical doctor. The hospital administrator provides the infrastructure in which the doctor's work takes place, and it is possible to create an interface between the two which provides the information each one needs in order to carry out their work, without necessarily knowing much about how that information was generated. This obviously requires a good deal of trust.

In most cases, such a separation is not advisable; managers need to understand the activities they are managing and to be able to interact with the personnel they are managing on a professional level. The manager of a bank needs to be a banker, the manager of hotel chain needs to be a hotelier, and so on, and nowhere is this integration of general management activities with professional knowledge and experience more important than in the management of engineering, as already noted. However, that total integration of the management activities within the engineering process demands a particular approach to what might otherwise be considered "normal" management activities. This is well known and has been emphasized by various industry leaders; for example, in the foreword to the 1989 edition of the *Handbook of Engineering Management* [1], F. Tombs, who at the time was Chairman of Rolls Royce plc, stated

> "Formal training as an engineer provides a good background for management, because engineering necessarily involves the exercise of judgement in the absence of complete data; a situation frequently encountered in management problems. But it is difficult for the professional engineer to acquire other skills, including that of management, while practising engineering – itself a very demanding occupation.
>
> A number of attractive MBA courses exist, but tend to concentrate on management as a subject in itself rather than, as this handbook does, as an extension of the engineer's own skills. There is the further important point that engineering management is difficult for non-engineers, because of the way in which the engineering discipline pervades many of the problems to be tackled."

However, if we consider managing the process of engineering as managing all the work involved in an engineering project, then it soon becomes apparent that these management activities fall into two distinct groups, and nowhere is that more clearly demonstrated than in what we might call human resources management. On the one hand, there is the management of the engineers on the project, of which the majority would be engaged in a design activity, be it the design of the object to be produced or of the means of production. On the other hand, there is the engineer as manager of the workforce constructing or manufacturing the product. In some projects this distinction is reflected in the two positions of design manager and construction manager, in other projects a distinction is made between the engineering manager and the project manager.

It is in the activities in the first group that the difference between engineering management and other types of management becomes most evident. Not only must engineering managers have a good understanding of what their engineers are doing, so that they can guide and assist them in their work, but in order to get the best performance out of them they must understand how engineers think and what motivates them. There are almost as many jokes about engineers as there are about blondes, but there is no doubt that, on the average, engineers are different to artists, or doctors, or scientists, or lawyers. Engineers are seen by many as anything from nerds to colourless plodders engaged in uninteresting, not particularly well paid, and largely routine work, and they cannot understand why anyone would go through the hardships of an engineering education for that. Well, "it takes one to know one", and that is why an engineering manager needs to be an engineer. An engineer understands what motivates other engineers, and that understanding is a prerequisite to managing engineers. And it is more than just an intellectual understanding of the motivational elements; it is also the belief that they are appropriate and valuable, and not something to be suppressed by "good" management.

So what is it that characterises and motivates engineers? This cannot be a definite list with absolute rankings, nor apply universally to all engineers, but some of the important motivations are

- a belief in technology as the main underpinning of civilization and the means of solving many of the problems facing us;
- confidence in their ability to develop and apply technology;
- a strong desire to create new solutions and a willingness to face the associated challenges; and
- taking satisfaction in seeing their designs realised and operating, thus receiving confirmation that their work is really useful.

Given these characteristics, engineering management must consider how best to exploit them on each individual project. Some projects will rely heavily on creativeness and the willingness to try new approaches, other projects on the drive and determination to reach the end goal in a timely fashion. Management must create an environment that supports the corresponding characteristics and select the engineers accordingly.

An increasingly important aspect of engineering management is the choice and provision of design tools. The slide rule has disappeared, calculators are almost gone, and pencil and paper is also on the way out; engineering today is completely dominated by electronic data processing in its various forms. Personal computers have resulted in a great increase in productivity and the ability to routinely tackle problems not even attempted fifty years ago, but anyone involved in, and giving some thought to, engineering management will recognize several serious issues associated with this development. And most of these have their root cause in the fact that humans and computers are very different in their natures and abilities, necessitating both a special interface not required for communication between humans, and a subdivision of tasks into those that predominantly require human abilities and those that are most appropriately handled by computers.

The interface issue means that, for every software tool chosen by management as part of the design platform, there is an initial effort required to learn the "language" of that tool. This effort is somewhat reduced by the fact that some tools belong to families of tools, such as MS Office, which all have the same "look and feel", but even so, my experience is that in any industrial engineering organisation less than one-quarter of the engineers are familiar with more than one-quarter of the features and capabilities of these tools, and that considerable time is lost through inefficient use of the tools. This issue is exacerbated by the relatively rapid rate of version upgrades, with often significant changes to the interface from one version to the next.

Reference

1. Tombs, F.: In: Lock, D. (ed.) Handbook of Engineering Management. Heinemann Newness (1989)

PART C

Applying the System Concept to Engineering

C1 Complexity in the Context of Engineering

C1.1 Overview

In the first part of this book we looked at the system concept; its basic nature as a mode of description, its use and meaning as a linguistic item, and its basis in the way the mind works. Above all, you hopefully gained a clear understanding of what we called *the systems approach*; the application of the system concept for the purpose of handling complexity. The second part was concerned with engineering, a subject you would know well, but we wanted to discuss some features of the profession that are sometimes overlooked, such as its long and successful tradition; the very substantial Body of Knowledge, its central objective of creating objects that, through their operation, provide required services to society, and that attaining this objective includes all activities which such creation and operation require.

This understanding of the system concept and of engineering is the foundation on which we now undertake the investigation of the application of the system concept to engineering, the subject of this third part of the book. There is a whole Body of Knowledge called Systems Engineering, and much of that will fit in with the outcome of our investigation, but there is a very subtle, but important conceptual distinction between Systems Engineering and The Application of the System Concept to Engineering. If we consider ISO 15288 to be the authoritative standard for systems engineering, then there is very little in the description of the processes in that standard that relates to a system approach. Already the title, *Systems and software engineering – System life cycle processes*, is an indication of the focus on the object as a system; something *is* a system, and then the engineering processes are tailored to suit it. And, as an aside, systems engineering does not get a mention at all, and design is treated only as architectural design. In contrast, we shall apply the system concept to the engineering processes in order to handle the complexity of the object; the object is then *described* as a system as a result of applying these processes. The description of a complex object as a system of interacting elements is neither new nor particular to engineered objects; it is the application to the process of engineering that is (relatively) new.

The purpose of applying the system concept to engineering is the same as for any other application – to handle the complexity of the subject matter by structuring it in a manner most suitable for processing by the brain. But before we look into how this could be achieved, we need to be clear about the nature of this complexity we want to handle; what are the sources of it, why has it become more of a problem in recent times, and how can we best describe or classify it? Complexity theory and complex systems have become very “in” subjects in the last ten years or so, with a focus on areas such as ecology, biology, and social systems and organisations, and much of the research activity in systems engineering is attaching itself to this bandwagon. But to what extent this is directly relevant to engineering, as we described it in Part B, is not obvious, and can only be ascertained by a critical examination of the problems encountered in engineering.

C1.2 The Nature of Complexity

What do we mean when we say that something is complex? That it has many sides or aspects to it, needs many variables or parameters to describe it, or consists of many parts? Or that it is hard to understand, needs many words to explain, or is difficult to predict? There are many different definitions and views on this concept [1], but usually we mean an unspecified combination of some or all of these and similar definitions, with the emphasis depending on the particular case, and in one way or another, complexity is related to the number of parameters required to describe behavior.

Complexity is a thoroughly human concept. Something is considered complex because it is difficult for us, as humans, to come to grips with and to work with; it has to do with the capabilities of our brain. It makes no sense to say that something is complex in itself, without putting it in the context of whatever entity is going to operate on it; what is complex to a human may be very simple for a computer, and vice versa. The difficulty we have in conceiving of something as a single entity once it has more than about seven parameters [2] is a characteristic of the brain. Indeed, the success of our whole system design methodology will depend on how well it exploits the strengths and avoids the weaknesses of our brains.

System complexity arises in two fundamental forms, as identified by Peter Senge [3]; namely *detail complexity* and *dynamic complexity*. Detail complexity arises from the volume of systems, system elements and defined relationships. This complexity is related to the systems as they are; their static existence. Dynamic complexity, on the other hand, is related to the expected and even unexpected behavior of systems during their operation. These two forms of complexity can synonymously be referred to as *structural complexity* and *behavioral complexity*. The concept of the structure of a system was introduced in Sec. A4.3, and with that description of interactions as links, a simple expression for the structural complexity is [4]

$$\chi = \frac{1}{n}\sum_{i=1}^{n-1} i \cdot \omega_i ,$$

where ω_i is the number of elements supporting i links to other elements. The values of χ for four simple structures are

Structure	Structural Complexity
Linear chain	$\chi = 2(n\text{-}1)/n$
Closed chain (circle)	$\chi = 2$
Central element (e.g. broadcast)	$\chi = 2(n\text{-}1)/n$
All-with-all (maximally connected)	$\chi = n\text{-}1$

However, in addition to the number of elements and relationships, factors such as linearity or non-linearity in relationships, asymmetry of elements and relationships determine the degree of complexity.

Dynamic complexity arises in systems that are either significantly influenced by humans or where humans are actually system elements, or, most commonly, both [5]. However, it is convenient to think of dynamic complexity as arising either externally or internally, because the former is more prevalent during the design of a project, whereas the latter is related to the ability of the system to respond to a changing environment during its operation, a subject that is treated under the heading of adaptive systems [6].

C1.3 Two Sources of Complexity as Drivers of Systems Engineering

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

The first one is that, for most of the last century, the development of new technology through research was seen as an imperative for developed nations, and there was such an appetite for new technology that almost any new development found an application and a market somewhere. Only in the last quarter did we start to notice some serious concerns about where all this technology was leading to, and whether its application was always in the best interest of society as a whole. The question started to shift from “can it be done?” to “should it be done?”, and the increase in knowledge, both through travel and television, of what was happening in the world outside our own local community made us aware of the fact that we are all sharing the same limited resources and influencing a common environment. It is becoming clear that it is not just a matter of having better technology, it is also a matter of knowing how to apply this technology in the most appropriate manner. This requires an understanding of the interrelation of the application with its environment. While this was always within the scope of engineering, the immediate and direct benefits of introducing a new technology were usually so major that other effects appeared relatively insignificant.

Sometimes they were actually insignificant because the scale of the application was initially so small that the side-effects, which are generally dependent on the scale in a very non-linear manner, were also small; at other times they were simply assumed to be small because no methodology existed to handle the increase in complexity involved in a proper assessment. The former reason no longer holds in many cases; for technologies such as the internal combustion engine, irrigation, and power generation the applications have grown to such a scale that what was earlier side-effects have become major effects. The latter reason is no longer acceptable to society, and the legislative framework in which engineering takes place is continually being tightened to ensure that a holistic approach is being taken to determining all the effects of every project over its life cycle. The result is that a whole new dimension of complexity has been added to engineering, creating a strong demand for adopting systems engineering as an intrinsic component of the engineering process.

The second driver is to be found in the relationship between humans and technology, which in recent times has started to develop from a purely physical one to one involving cognitive aspects. This development is made possible by the advances in electronic data processing, and the computer itself is the best illustration of this. In the early sixties, the human-machine interface was via the card reader as input device and the line printer as output device; in between the computer operated autonomously. Twenty years later, the advent of the PC allowed a form of dialogue between the user and the machine, and today the development of the interface is about mutual understanding, or cognition. A simple example of this is the auto-correction function in a word processing program.

For systems, this development has meant that the human is no longer outside the system, as a user, but is increasingly an element of the system, and the behaviour of the human is an essential factor in the functionality of the system. As that behaviour is vastly more complex than that of any man-made component, the complexity of cognitive systems is moving system design into a new realm, one in which the application of the system concept will be the dominant paradigm.

C1.4 A Taxonomy of Complexity in Engineering Projects

We start this examination by recalling the view of engineering introduced in Sec. B2.3. Engineering activity takes place in the form of *projects*, and each project has a *purpose*; it is intended to achieve something, and the degree to which it achieves it is the measure of success of the project. Here is already a significant distinction between engineering projects and e.g. biological or ecological systems; the latter have no known purpose. They are very complex systems, and through research we are unravelling this complexity and so gain a better and better understanding of them; how they propagate, how they survive, their internal processes, their interactions with other species or parts of Nature, and so on, but this does not lead to any identification of a purpose.

The purpose of an engineering project falls into one of two groups, which we might characterise as internal or external to engineering, depending on whether the project is concerned with developing technology or applying technology, as discussed in Sec. B4.2. And again, as in that section and for much the same reasons, we shall focus on projects that apply technology in order to achieve an external purpose; a purpose defined mainly by people outside the engineering body that is to carry out the project. That purpose is generally stated as providing a *service*; while the *product* of the project that is to provide the service is the engineer's solution to the problem of providing the service.

The use of the words "service" and "product" introduced here needs to be clearly understood; these words, just as the word "system", have different meanings when used in different contexts. Throughout the remainder of the book, the *purpose of a project* is to fulfil a *need*, the *service provided by a project* is that which fulfils the need, and the *product of a project* is the engineered object that provides the service. In some cases, the service may be a service in the narrower sense, such as public transport, a financial service, health, or education; in other cases the concept must be broadened to include providing a product, such as providing a raw material. In this latter case, the product of the project is the object or facility that provides the raw material, such as a mine.

With this understanding, let us proceed by developing a taxonomy of the complexity encountered in engineering projects by considering *where* in the project the complexity arises. To this end it is useful to view a project as having four components,

- the requirements on the service to be provided;
- the environment in which the project is to be executed;
- the two bases, knowledge base and resource base (technology, manpower, facilities, etc.) needed to create and maintain the product that will provide the service; and
- the engineering process,

as shown in Fig. C1.1.

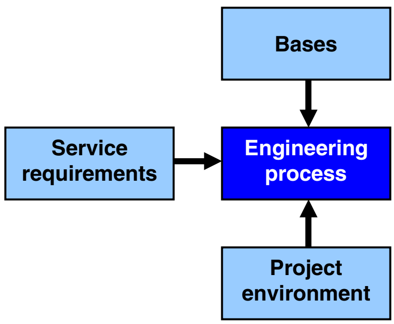


Fig. C1.1 A view of an engineering project as consisting of four components.

Due to the differences between these four components, it is obvious that when describing "where" complexity is found, it must be understood in a very generalised sense; the item that contains the complexity (i.e. the "location") may be a set of requirements, a specification, a body of work, or a physical object. And we might think of locations within the first three components as being "external" to the engineering process, whereas complexities within the engineering process are characterised as "internal", as indicated in Fig. C1.1 by means of the shading; this reflects the degree of (or lack of) control the project has over the sources.

The division into "internal" and "external" components is also useful in viewing the complexities in the engineering process as being of two kinds; those determined or induced by the external sources, and those that arise from the engineering process itself. The latter will be discussed in Sec. C1.3, and the view of two kinds of complexities will be very important in developing our approach to handling complexity within the engineering process in Ch. C2.

C1.5 Complexity in the External Project Locations

C1.5.1 The Service Requirements

The service requirements are generally formulated in the context of a *business case*; that is, the service is to be provided to a *market*, and the provider, or *principal*, will receive a revenue that makes the project worth his while. This immediately identifies two groups of service requirements; those determined by the market, and those determined by the principal. The former are the result of a number of factors, including fashion, culture, standard of living, and climate, and are therefore expressed in a variety of forms; sometimes quantitatively, but very often qualitatively in the form of preferences and desires. They all contribute to what the market perceives to be the *value* of the service. We will return to the concept of the value of a service many times in the following; for the time being, we might think of it simply as what the market is willing to pay for the service. This value then becomes a function of the parameters that describe the factors influencing it. As a concrete example, consider the project of providing a service well known to us all, the ability to access IT services on a mobile platform. A factor common to all realisations of this ability is the capacity of the battery to maintain the service between recharges. How does this factor influence the value of the service? If the time is less than, say, thirty minutes, the service would probably be considered to have little value, and once the time goes beyond, say, eight hours, the value does not increase much more, and we obtain the well-known S-curve shown in Fig. C1.2.

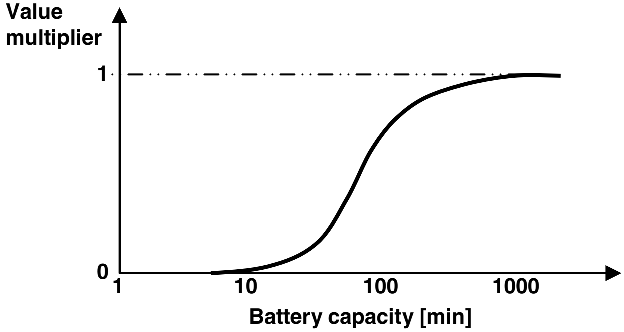


Fig. C1.2 The value of battery capacity in a mobile communications device, as an illustration of the S-curve.

However, there are also a large number of other factors influencing the value, including memory capacity, processing speed, the ability and ease of accessing content, weight, size, etc, and they are not independent, so that what we end up with is a value function made up of a complex set of interacting functions. Added to this is the fact that there is some uncertainty attached to most of these functions; firstly and inescapably, because they refer to the future; secondly, because there are numerous other factors, not directly related to the service itself, that influence people's perception of the value of this particular service, such as competing services and changing social attitudes; and thirdly, because of the rapid increase in the cost of obtaining the market information as a function of the accuracy of the information. So, not only is there a complex set of interacting functions, but the parameters defining these functions are themselves probability functions.

In its simplest form, a business case evaluates the viability of introducing an additional amount of an existing service into an established market; three examples are

- the provision of additional coal through the development of a new coal mine;
- the provision of additional energy through the development of a new wind farm; and
- the provision of additional transport capacity through the construction of a new tollway.

In all three examples the requirements of the market on the nature of the service are well defined; the factors of significance in the value function are all external to the service itself. In particular, government legislation regarding greenhouse gas emissions and subsidies for green power, advances in coal gasification and CO_2 capture and sequestration, and public awareness and attitudes regarding environmental protection and sustainability are important for the first two examples. The perceived value of the time saved through better transport infrastructure, and its relative position in the ranking of the demands on personal finance, is a complex issue in the third example, and three recent tollway projects in Australia got the patronage estimates completely wrong [7].

In addition to the service requirements of the market, the Owner will have a set of requirements on the service related to ensuring the viability of the project or, conversely, related to reducing the *risk* and enhancing the *opportunities* arising from the relationship of the service to the principal's existing business and capabilities. This includes such features as the ability to easily modify aspects of the product in response to expected changes in the market (e.g. increasing disposable income) and the use of existing distribution arrangements.

In many cases there may be a complicating factor that is not related to the nature or type of the service, but to its timeliness; the market presents a *window of opportunity*. While this does not make the service in itself more complex, it results in a significant increase in the complexity of the engineering process, as we will discuss in Ch. C2.

In all the project locations, and perhaps particularly here in the service requirements, the complexity has two sides to it. On the one hand, there is the complexity at any given moment in time, as evidenced by the number of requirements and the number of relations between them. This is what we most immediately recognise as a complex situation. But, on the other hand, both the requirements and their relationships may change over the duration of the project, and this *dynamic complexity* can be much more difficult to recognise and to handle adequately. Due to the relationships between requirements, a change to one requirement may propagate throughout the set of requirements, and a structured and careful approach is needed in order to determine and document all the implications. What we are faced with here are two opposing timescales: the timescale for externally introduced changes, and the timescale for the process of determining and executing the response to the changes. If the latter becomes too long compared to the former, the project will not progress, but consist only of processing changes. Examples of this can be found in the defence area, where the service (or capability) requirements can change over a relatively short period due to changes in the threat assessment and also due to technological advances, whereas the time required to process a set of requirements through the bureaucratic sequence of RFP, tendering, tender evaluation, and contract negotiations can be equally long or even longer, and so the process starts all over again, with a new set of requirements.

C1.5.2 The Project Environment

An engineering project is executed within a certain *environment*; that is, all those non-technological factors that influence how the product is created and how it is operated to provide the service. These factors include:

- Legislation and regulations regarding how work is performed, such as OH&S regulations, environmental protection legislation and consent conditions, and contracting conditions.
- Government policies, reflected in such factors as subsidies and tariffs, land use (zoning), and taxation rulings.

- Community concerns regarding noise impact, visual impact, health risks (e.g. high voltage transmission lines) etc, generally known as NIMBY (not in my back yard), but also concerns about the environment and endangered species (e.g. resistance to mines and dams).
- Special interest groups (representing industry sectors, such as the building industry, or sectors of the workforce, in the form of unions).

These factors increase the complexity of a project not only by their existence and interactions, but also because they need to be managed, in the form of such activities as lobbying, public relations, and community consultation. In infrastructure projects it is not unusual for this management effort to amount to several percent of the total engineering effort.

C1.5.3 The Resource Base

Every engineering project consumes resources in the form of finances, labour, energy, and materials. The totality of the sources or pools of these resources that are available to a project and on which it might potentially draw is its resource base. From this base the engineers will then have to make a choice of which resources they actually employ in the project, and it is the existence of a great (and increasing) number of possible choices that provides a further dimension to the complexity of engineering projects. Some of the main aspects of this resource base are

- The changing technological resource base, including new materials, construction elements, and processes, and the retirement of existing items.
- Economic factors, including labour availability and cost, material cost (e.g. the fluctuating price of steel), transport cost, and the cost of funds (credit availability and interest rates).
- A variety of possible contracting strategies, as discussed in Sec. B3.3, and illustrated in [2] by a couple of examples from the power generating industry.

C1.5.4 The Knowledge Base

In Sec. B2.4 we introduced a knowledge base as one of the properties of a project, and defined it as the base from which the knowledge required in order to be able to apply the resources is drawn. It will now be useful to consider that knowledge base to consist of two parts; a *domain knowledge base* and a *technology knowledge base*. By the domain knowledge base we shall understand the knowledge required to understand and analyse the stakeholder requirements; the technology knowledge base is the knowledge required to develop a solution.

Even though the two knowledge bases may overlap to a great extent, this conceptual distinction is very important, and the reason was touched upon at the end of Sec. B3.1, where we mentioned the work of John Warfield [8] and the idea of a “problematique”, an extended view of the service required that encompasses the context in which the need is expressed. The knowledge required to fully develop and understand the "problematique" will often be considerably different to the knowledge base the engineer would utilise in developing a solution. From our point of view, in our current discussion of complexity in engineering projects, the distinction is important in that the sources of complexity contained within the two knowledge bases are also different.

In the case of the domain knowledge base, the complexity arises from the fact that the "problematique" may have numerous aspects, each involving different knowledge areas, from politics and government policies to individual beliefs and value judgements, and these aspects may interact in subtle ways. As a result, understanding the "problematique" may be a complex process, requiring both a structured approach (e.g. as advocated by Warfield) and the involvement of various specialists outside of engineering.

In the case of the technology knowledge base, the complexity arises mainly from the extent of the base, and the fact that the increase in knowledge leads unavoidably to greater specialisation. The result of this specialisation is that there are barriers to the information flow between disciplines, and we now have the situation that, due to the increase in knowledge, we have increasing specialisation and therefore more barriers, at the same time that engineering projects are becoming increasingly multidisciplinary. Achieving optimal outcomes means balancing performance parameters and costs across all disciplines (in addition to all the non-engineering aspects), and rather than as an issue of understanding, the complexity manifests itself in the difficulty of *selecting* a solution. And again, a structured approach is required in order to arrive at a solution reasonably close to the optimal one in an efficient manner.

C1.5.5 Quantifying the External Complexity

Quantifying the complexity introduced into a project through its relationship with its environment is a difficult and largely unsolved task. There are many approaches discussed in the literature, often under the heading of risk assessment [9], but in practice the uncertainty and subjectivity makes the results largely qualitative. In a general fashion, the complexity can be thought of as arising as a result of the relative *distance* between the object required to meet the stakeholder requirements and the totality of existing objects. The space in which this distance is measured is a multi-dimensional one, and while the number of dimensions and the definition of the individual coordinates are project-specific, the following coordinates will normally be present:

1. The extent of the domain knowledge base required to address the requirements, relative to the existing knowledge base.
2. The number of technologies (or disciplines) required to address the requirements.
3. The extent of each technology required to address the requirements, relative to its existing state. This is also called *technology maturity*, and is an area where quantitative methods are relatively well established [10].

4. The extent to which the same or similar objects have been realised (with an allowance for the success of the realisations). This is particularly important when it comes to the host of issues related to community acceptance.

The same coordinates are involved in assessing project risk, but complexity and risk are by no means identical. If risk is defined as the probability of failing to meet the performance requirements within the given time and budget constraints, multiplied by a measure of the consequences of that failure, then risk is clearly dependent on those two constraints (as is easily seen by the fact that if the budget and timeframe both go to infinity, the risk goes to zero, no matter how complex the project is). But furthermore, the risk is also dependent on the particular manner in which it is proposed to carry out the project, i.e. on the Project Plan, as will be discussed in the next section.

By "existing" in items 1 and 3 above we should understand "available to the project". Both domain knowledge and technology may or may not be available within the initial project organisation, but if it exists and can be made available, it is simply a matter of the cost (and possibly the time frame) involved, which brings us to the complexity within the project, i.e. to the complexity of the work.

C1.6 Complexity within the Project

Having assessed the complexity of the project in terms of the external influences, and having developed a good understanding of what work has to be undertaken in order to handle this complexity, there now remains to determine how to carry out that work. In general it is true that the complexity of the external influences is reflected onto the project itself, i.e. onto the object that will satisfy the requirements and onto the body of work required to create that object. It was to address this internal complexity that systems engineering was initially developed. This approach, which effectively puts a barrier around the project in its earliest phase, was appropriate to defence projects in the Cold War, where commercial aspects and community influence were relatively unimportant. And this approach is still quite apparent in many of the processes that make up systems engineering, as will be discussed in the following chapter.

However, with the wider application of systems engineering to areas outside defence and aerospace, and also somewhat of a changing view of the military role, the system approach to the external complexity is taking on increasing importance, and is being integrated into many of the systems engineering processes. Consequently, although the characterisation of a source of complexity as external may still be useful in *understanding* its nature; when it comes to *handling* complexity in the process of engineering there is little benefit in making this distinction. An example that is probably well known to most readers is that of change management; it is required to handle change whether it arises from the dynamic nature of the development process (internal) or from changes to the requirements (external). The dynamic nature of the development process is again a result of the complexity of the service requirements, and so on. Our approach to handling complexity in engineering projects focuses on handling the manifestations of complexity, such as the number of disciplines involved, the number of requirements, their interdependencies, their dynamic nature, etc; trying to reduce the sources through such ideologies as a return to Nature *à la* Rousseau is outside the scope of engineering.

References

1. Weaver, W.: There is a large literature on the subject of complexity, for example. Science and Complexity 36, 536–544 (1948); Flood, R. L., Carson, E. R.: Dealing with complexity: An introduction to the theory and application of systems science, 2nd edn. Plenum Press, New York (1993); Sheard, S.A., Mostashari, A.: A Complexity Typology for Systems Engineering. In: Proc. 20th Int'l Symposium of INCOSE, Chicago, Illinois (2010)
2. Miller, G.A.: The Magical Number Seven, Plus of Minus Two: Some limits on Our Capacity for Processing Information. The Psychological Review 63, 81–97 (1956); available online at, http://www.well.com/user/smalin/miller.html References to subsequent papers can be found at, http://citeseer.nj.nec.com/context
3. Senge, P.M.: The Fifth Discipline: The Art & Practice of The Learning Organization. Currency Doubleday, New York (1990)
4. Aslaksen, E.W.: System Thermodynamics: A Model Illustrating Complexity Emerging from Simplicity. Systems Engineering 7(3) (2004)
5. The central role of humans in creating complexity is discussed in Kline, S., Foundations of Multidisciplinary Thinking. Stanford University Press (1995)
6. There is a great deal of activity taking place in the area of adaptive systems and, perhaps in particular, in what is termed complex adaptive systems or CAS (although there is no definite boundary between the two). Besides the ubiquitous Wikipedia, a starting point might be the UCLA Adaptive Systems Laboratory, http://asl.ucla.edu/, or the Adaptive Systems Research Group at the university of Hertfordshire, http://adapsys.feis.herts.ac.uk/. A textbook that addresses computational models of social systems (which are a focal point of CAS) is Miller, J.H., Page, S.E.: Complex Adaptive Systems. Princeton University Press (2007)
7. Three road tunnel projects – Cross City Tunnel and Lane Cove Tunnel in Sydney, and the Clem 7 tunnel in Brisbane – have had initial patronage of only 30-40 % of that estimated, http://www.bitre.gov.au/?publications/Oz/Files/BITRE_literature_review.pdf
8. Warfield, J.N.: An Introduction to Systems Science. World Scientific Publishing (2006)

9. Complexity, risk, and uncertainty are bound together like the three corners of a triangle, and this triangle is itself an inseparable aspect of any human endeavour, with the relative importance shifting around in the triangle depending on the particular situation. Giving references to such a wide and divers topic is not very useful (Google "Complexity and Risk" and get about 40 million hits). ISO 31000:2009. Risk management – Principles and guidelines, is the international standard, and most engineering companies would have their own approaches to this topic. Quite a good little Excel-based tool for engineering projects is that developed by Public Works and Government Services Canada, PWGSC Project Complexity and Risk Assessment (PCRA) Tool and Manual, `http://www.tpsgc-ppwgsc.gc.ca/biens-property/sngp-npms/pcra-ecrp-outil-tool-eng.html`
10. Technology maturity, also called technology readiness level, has been particularly important to the military and to the aerospace industry, as major users of advanced technology, and both the US DoD and NASA have well-developed and documented approaches to this issue (as do other defence departments)

C2 The Systems Engineering Approach to Handling Complex Engineering Projects

C2.1 Overview

In the preceding chapter we identified the main sources of complexity in engineering projects, and in Sec. B2.3 we subdivided projects into a number of stages. We can now relate the two, in the sense of determining which sources determine the complexity in the various stages and thereby determine the complexity of the objects created within each stage. These objects are, as we know from Sec. B2.4, processes and the artefacts and descriptions resulting from them, and while our approach to handling them has a common basis, as was already foreshadowed in Ch. A5 and will be developed in more detail in the next section, there are also significant differences, as we shall discuss later in this chapter.

The matrix below is an attempt to indicate in which stages the various sources *introduce* complexity. The emphasis on "introduce" is important, a stage may well *reflect* the complexity of a source, but it was introduced in an earlier stage. And, with the red colour signifying a very significant contribution and the blue colour signifying significant contribution, we recognise the imperative of addressing the complexity of engineering projects in the concept stage. If complexity is not addressed and handled as soon as it is introduced, the result is likely to be, at best, cost and schedule overruns and, at worst, useless or abandoned projects.

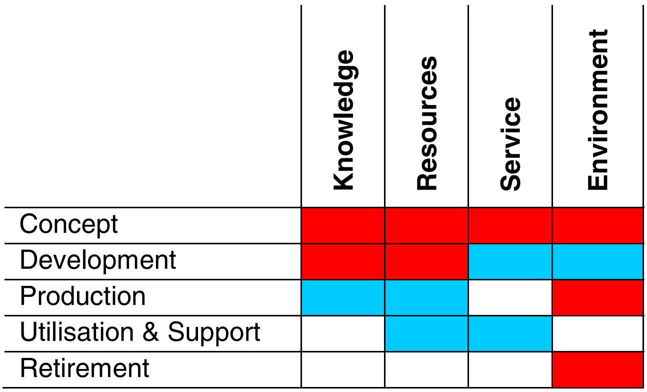


Fig. C2.1 Where and from what sources complexity is being introduced into projects, with the red colour indication very significant, and blue colour indicating significant, levels of complexity.

In order to interpret the above matrix, it is necessary to think of the engineering activity in each stage as a three-step process of problem definition, option identification, and solution selection and documentation. Under this perspective, the extent and, above all, the rapid expansion of the knowledge base (i.e. a large number of options), introduces a high level of complexity into the concept and development stages, but once it is addressed in these stages, only a modest further amount of complexity is introduced in the production stage, and in the following stages the expansion in the knowledge base will, if anything, reduce the reflected complexity (i.e. the work associated with carrying out the planned activities).

The complexity of the service requirements needs to be taken into account in the very earliest part of the project; defined, analysed, and transitioned into requirements on the functionality of the plant. The further development of the design through to data for production or construction can be a complex process, but much of the groundwork for handling this complexity, in particular the structuring (or architecting), should already have been carried out in the concept stage.

The complexity introduced by the environmental requirements also needs to be addressed in the concept stage, but many environmental requirements are focused on the manufacturing and construction processes. And, because the retirement stage comes tens of years after the plant was designed and constructed, the requirements on decommissioning will often have changed (increased) considerably (just think of the requirements on demolishing plant containing asbestos or nuclear material).

Following on from where we left off in the previous Chapter, the matrix in Fig. C1.1 can also be interpreted as showing the relationship between the "external" complexity, represented by the sources, and the "internal" complexity, represented by the stages, and this leads us to the view of the "internal" complexity as belonging to something that is *designed* in *response* to the complexity imposed by the "external" sources. That is, while the concept of complexity remains the same, the characterisation as "internal" or "external" has to do with the degree of control we have. It is important to *understand* the "external" sources of complexity, but when we now turn to the design and management of the engineering process, it is the "internal" complexity we have to be concerned with.

We also discussed, briefly, the difference between complexity and risk, and noted that, while they are coupled in the sense that increasing complexity leads to an increasing number of hazards (i.e. things that can go wrong), risk involves both the probability and the consequence of such a hazard. Both of those parameters are not intrinsic to the objectives of a project and their complexity, but determined mainly by how we carry out a project. The key to handling the risks associated with complexity is therefore careful and detailed *planning*. It is my experience that many of the causes of unsatisfactory project outcomes can be traced back to inadequate planning. Issues become problems only because no thought have been given to how to resolve them and what resources will be needed, and then, more often than not, a "quick fix" is the result. Some contractors even put their ability to "get stuck right into it" and do projects "on the run" forward as a desirable quality. The approaches to handling complexity put forward in the following sections all assume that they will be applied within a rigorous planning framework; anything else will reduce their effectiveness greatly and often render them purely cosmetic.

C2.2 Plant and Work–Two Complex Entities

It is beneficial to view an engineering project as containing two complex entities; the plant to be created and operated in order to satisfy the stakeholder requirements, and the body of work required to create and operate it. They are often called "the system" and "the project", or also "the Works" and "the Work"; we shall stay with our previous nomenclature of plant and work. These two entities are obviously tightly coupled, but they are of very different natures. The relationship is somewhat analogous to that of a glove and a hand; the size and shape of the glove is determined completely by the hand, but they are otherwise very different.

The approach to handling any complex entity is always the same: describe it as a system of less complex, interacting elements. However, this is not generally as straight-forward as it sounds, as there is usually a choice of elements and of the way in which they interact, so that we are faced with three problems: Identifying sets of elements that, when interacting, can represent the complex entity; for each set determining which set of interactions (i.e. what structure) provides the best representation; and, above all, formulating an operational definition of what "best" means. The obvious solution to the last problem is to define the best architecture as the one that provides the greatest reduction in complexity, and while this is, indeed, the underlying criterion in most cases, it is too general to be an operational criterion, and it masks practical aspects that depend on the nature of the complex entity and what we want to do with it. In particular, we need to have a very clear understanding of the difference between the plant and the work, and we can start out by recalling the development of an engineering ontology in Sec. B2.4. In terms of the top level categories in Fig. B2.4, the plant is an object and the work is a process; that is, the plant is a continuant and the work is an occurrent. Work is defined completely in terms of what it does; it has no substance, nothing that exists when the work is not being carried out. The plant, on the other hand, has both an enduring physical presence and a functionality; it *is* something and has the ability to *do* something; it has a dual nature. With a bit of simplification, we might say that the work exists in time, whereas the plant exists in space. This difference is reflected in the manner in which the system approach is used to handle complexity in the two cases. Let us first look at what it means for the work.

The description of the work as a set of elements is captured in the Work Breakdown Structure (WBS), where the elements are Work Packages (WPs), defined in terms of their results or outputs, the activities required to produce these outputs, and the resources or inputs required to support the activities and complete them in a certain time period. However, it is often overlooked that the word "structure" has two quite different meanings here. The one directly linked to "breakdown" arises because the work is broken into WPs in a step-wise fashion; first the whole body of work is broken down into a few main, very large WPs. Then, in the next step, each of these are broken down into smaller WPs, and then, in the next step, each of these are again broken down into smaller WPs, and so on. This is often represented graphically as a tree structure, but this is not the structure involved in representing the work as a system. To describe the work as a system, we choose a complete set of elements (i.e. that cover the whole body of work, but not necessarily all from the same level of the breakdown) and define the interfaces between them; there are a number of well known ways to do this, such as PERT diagrams, N^2 matrices, etc. There is a choice of elements and structure within the constraints imposed by the overall outputs required as a result of the work, the inputs present at the beginning of the work, the overall timeframe, and the resources available. In the next section we shall look at some rules for developing the "best" WBS.

Describing the plant as a system is also generally a step-wise process, but due to its dual nature mentioned above, it is best viewed as two parallel and closely coupled processes. One is the development of the description of the plant in terms of its components. Depending on the industry and the type of project, the resulting artefact has various names, such as Bill of Materials, Schedule of Works, or Works Definition Document (WDD), but in any case it should fulfil two essential requirements:

a. As it is developed in steps, from the description as a single component, the plant, to descriptions in terms of more detailed components, each level must be a *complete* description of the plant. When a component at one level is broken down into a set of smaller components at the next lower level, that set of components is equivalent in all respects to the original component.
b. The document must reflect this step-wise process. It must not be just a simple list of all the components on the lowest level, but define the components at each level and display their hierarchical ordering, i.e. the Plant Breakdown Structure (PBS). This is analogous to the WBS.

The other process is the development of the set of requirements on these components. It is contained in documents with various names, such as Technical Requirements, Requirements Definition Document (RDD), or Technical Specification. This document (or set of documents) is developed in parallel with the PBS, and as with the PBS, there is a requirement for completeness at each level. However, this "completeness" is partly explicit and partly implicit, with the fraction of requirements that are explicit increasing with each step in the development until, at some level, all the requirements at the top level, i.e. the stakeholder requirements, are explicitly satisfied by the requirements on the components at this level.

In Sec. C2.4 we look at some issues involved in developing the "best" PBS.

In the application of the systems approach to both the plant and the work there is a step-wise "un-hiding" of requirements as the partitioning into more detailed elements progresses. This is one side of the manner in which the systems approach handles the complexity of engineering projects. The other side arises from the fact that at each level of the top-down process, the components are not a collection of unrelated components, but interact in a particular way to form a *system* with a particular structure; this system can have properties, the *emergent properties* we discussed in Sec. A4.1, that are not present in any of the components. As the overall complexity is a combination of the complexity of the elements and the complexity of their interactions, the art of system design is to reduce this combined complexity, not just the one or the other.

However, it must again be emphasized that there are two very significant differences between the plant and the work that have greatly influenced the extent of development and approach to their execution. They will also determine the further course of our development of the application of the system concept to engineering. The first one is that while plants can differ substantially in almost all aspects, from the nature of their product to the technology employed and the size and structure of the physical realisation, the work is basically always quite similar. Not in the details and extent of the work, but it always consists of the same type of activities, has the same structure, uses the same management processes, and so on. As a result, the handling of the complexity of the work has been developed into a well-established set of processes; this is the application of the system concept to the management of engineering projects, as we have alluded to earlier, and any inadequacy or inefficiency in the execution of the work is more often due to not following these processes than to anything else. The process of designing the plant is much less well developed; while there are a number of design methodologies, few, if any, apply the system concept in a consistent manner.

The second difference is that in the case of plants, we can consider what a plant must do, i.e. its functionality, without considering a particular physical realisation; a plant has an "image" in both the functional and the physical domain. That is not the case with work; as we mentioned above, work is defined completely in terms of what it does; it has only this one "image".

Because of these two differences, the subsequent chapters will focus on exploiting the existence of the functional domain and the consistent application of the system concept in that domain as a means of handling the complexity in designing a plant to meet a demanding set of stakeholder requirements.

C2.3 Work Breakdown Structure and Contracting Strategy

As we saw earlier, the Work Breakdown Structure (WBS) for a project is a hierarchical ordering of Work Packages (WPs), with the whole body of work as a single WP at the top level, and as a set of WPs at whatever level is deemed appropriate for each part of the work. The WPs at this lowest level of breakdown are sometimes called *tasks*. However, connected to what might look like a simple (but structured) tabulation of WPs is a number of central features of the project.

This central position of the WBS in the project explains why it acts as the *key* to the data base that contains the data defining the various types of entities constituting the project.

Firstly, with each WP is associated an *effort*, measured e.g. in the number of person-hours (estimated or actually) required to perform the work (this may be subdivided into disciplines). This provides a relationship between resourcing and *duration* of the WP. Secondly, besides manpower, many WPs will require other *resources*, such as computer time, manufacturing facilities, construction equipment, test facilities, and so on. Thirdly, with each WP we can associate a *risk profile* in terms of the maturity of the technology employed, the experience of the allocated personnel, etc, and there are numerous other fields that can be added to this project data base. Finally, as each WP is defined in terms of its outputs and the inputs required to deliver these outputs, this defines the *interfaces* between the WPs (and the interaction of the project with its environment, in the form of project inputs and deliverables). This then provides the description of the work as a system, i.e. as a set of elements with particular interactions and thereby a particular structure. This, and not the hierarchical structure of the WBS, is the structure of the work as a system, and the temporal aspect of this structure is what is usually called the project *program*.

Now, to address the question of what is the "best" representation of the work as a system, it is useful to think of the work as consisting of two types or categories of work: the work in the narrowest sense; i.e. the work directly involved in creating the plant, such as the actual design and construction activities, and the work involved in the management of these activities. This separation is also often reflected in the program, in that the WPs containing activities of the first type have definitive durations, whereas many of the WPs containing the management activities are tied to the duration of the phases or stages of the work. The point of this separation is that the totality of the work in the first category is, at least to a first order, independent of how it is subdivided into WPs, whereas the management work is highly dependent on the particular representation of the work as a system, and so an obvious initial choice for the criterion for "best" system is the one that minimises the management effort, which leads directly to a number of simple rules:

1. The breakdown of a WP should be according to a single criterion only. The only exception to this rule is at the first level, where management is normally separated out from the rest of the work (the work being managed).
2. Minimise the number of interfaces and the number of parameters involved in defining each interface. This can also be phrased as requiring the WPs to be as *self-contained* as possible.
3. Ensure that the management responsibilities (i.e. completion on time and within budget) are mapped onto the structure of the technical responsibilities (i.e. meeting the technical requirements) to the greatest extent possible. This means that, at the lowest level of the WBS, i.e. at the task level, there should be no distinction between management and technical responsibility; one person is responsible for both.

4. A corollary to this last point is that tasks should be discipline based.
5. Group the low risk activities into the same area of the WBS and the program, thereby allowing the management effort to be more focused and the change management effort to be minimised.
6. Examine the sequential vs. parallel character of the structure to ensure that it reflects how it is intended to carry out the work.
7. The size of the tasks (WPs on the lowest level), i.e. the extent of the breakdown, should reflect the desired progress monitoring accuracy. If a task is properly defined, it is easy to determine its completion; estimating progress within a task is always somewhat subjective.

Minimising the management effort is not the only criterion that can apply; depending on the project, such criteria as minimising the completion time or maximising local content may play a dominant role in determining the work structure. In general all such criteria will have to be considered and given an appropriate weighting.

However, overlaid on all of these considerations are a number of commercial considerations that impose a structure on the work in the form of a *contracting strategy*. Engineering projects are carried out within a contractual framework between the various bodies involved in the project, such as the Owner, the Engineer, the Constructor (or Contractor), various Suppliers, the Operator, the Maintainer, etc. The contracts and the interfaces between them form the top level representation of the work as a system, and for many large projects, such as major infrastructure projects, this system can already be complex enough to warrant a formal system design approach [1]. The rules given above will still apply to a certain extent, but there are also some additional considerations:

1. The financing structure. If the Owners are financing the project off their balance sheet, they have a free hand in choosing the contracting strategy, but if there are debt providers involved, they will often impose certain requirements. Alliance Contracts may have very individual contract structures, Public Private Partnerships (PPPs) have other requirements, and so on.
2. The availability of an adequate number of contractors with all the required competencies in order to ensure competitive bidding. There is no sense in asking firms to bid, of whom it is known that they are already fully (or over-) committed.
3. The ability of the Owner or Proponent to manage contracts.
4. The need for highly specialised abilities. This may make it unavoidable to engage a number of smaller firms.

The "best" contracting strategy is the one that results in the lowest cost when all factors are considered, including the Owners' total risk exposure. And when one considers the performance record of complex projects, such as large software projects [2] or large infrastructure projects [3], it becomes obvious that minimising the risk exposure may be at least as important as minimising the cost. Through their interactions or interdependencies, the contracts form a complex system, and the Owners' risk exposure is an emergent property of this system. As with any system property, it is determined by both the properties of the individual elements, in this case the likelihood of non-performance of the individual contractors, and the interfaces between them, and for a particular system implementation, a performance model can be developed in a manner quite similar to that used in the well-known FMECA [4]. Various possible implementations can then be considered in order to determine the "best" one.

C2.4 Developing the Plant Breakdown Structure

As already discussed, the Plant Breakdown Structure (PBS) is a structured description of the physical (or spatial) characteristics of the elements making up the plant, without specifying their performance characteristics; it may be viewed as a structured, high level Bill of Materials. As such, it provides a check list against which the Owners can check that they are receiving everything that was agreed; that the *extent* of the plant is correct at handover. The nature of the PBS is different to that of many other project artefacts, in that it may be viewed as a "book-keeping" artefact, completely void of any engineering content. Of course, that is a highly simplified view, in that the Owners would often be heavily influenced by advice from the Engineer in what they ask for, but in the end, the resulting PBS is the Owners' "shopping list".

However, in addition to its importance as a structured check list, both in contractual terms and, as we shall discuss in a moment, as a dimension of the project data base, it is the progressive development of the PBS that is such a valuable process. Firstly, depending on the starting point of the project, the Owners' perception of what they want to end up with at the completion of the project will vary greatly in both justification and level of detail. Consequently, there is a process of questioning and probing to be completed before a clear picture emerges of what is required to meet the Owners' initial requirements.

Secondly, once an initial version of the PBS has been agreed and the project gets underway, both the Owners and the Engineer recognise omissions and opportunities for improvement and, depending on the stage of the project, the PBS is developed in greater detail as the design progresses. In any case, the PBS is a living document and provides an up-to-date record of the agreed extent of the plant, and it becomes a check list for the updating of any activities and artefacts that are linked to it. In particular, one or more levels of the WBS would normally be structured on items in the PBS, so that, for example, if it is agreed to add another production unit or another access road, corresponding tasks, such as the design, construction, and assurance of these items must be added to the WBS, and from there the changes propagate into budget, resource allocation, program, etc.

Just as was the case for the WBS, there are some useful rules for developing the PBS, and they are (for obvious reasons) quite similar to those for the WBS:

1. At some level, the breakdown should reflect the contracting strategy. For example, if the plant consists of a number of production modules, each involving building services, and a contract is to be let for building services plant wide, then building services should be a separate physical element, subdivided into production modules. However, if a contract is to be let for the production modules, then the modules will be a separate entity, subdivided into such parts as building services.
2. Within a contract, the partitioning into elements should reflect well-understood areas of competence and responsibility within the organisation.
3. The partitioning should be chosen so that the interfaces are minimised and, where possible, in accordance with industry interface standards. This applies to information exchange interfaces as well as physical interfaces, such as voltage levels, flange sizes, standard packaging (container) sizes, etc.
4. Examine the use of Commercial-Off-The-Shelf (COTS) elements at the first opportunity in the development of the PBS, as there would have to be a robust cost-effectiveness argument (taking development risk into account) for choosing anything else.

C2.5 The Project Database

The project database is the most direct manifestation of the description of the project as a system; it is the structured collection of all project-specific data, such as drawings, reports, specifications, plans, calculations, and software programs, and each item is an electronic *file*. The files fall into two categories: those arising out of, or primarily associated with, a management activity, which we shall call *management files*, and the rest, which we shall call *production files*; this separation is due to the very different structures of the management activities and of the other tasks, as mentioned earlier. And furthermore, while files in both categories are identified by the associated WBS number, the production files have a second identifier – their associated PBS numbers. These two identifiers can be thought of as two coordinate axes spanning a surface, and each file represented by a point of this surface, but the two coordinates are not completely orthogonal because, as mentioned, it is common for the partitioning criterion on one or more levels of the WBS to be identical to the partitioning on one or more levels of the PBS. For example, if the project involves the design of a plant consisting of a number of modules, then a file arising out of the design of a module (say, a specification) will be identified by the WBS number of the WP for the design of the module and by the PBS number of the module; any other PBS number would be inadmissible in combination with this WBS number.

C2.6 Standards and Current Practice

C2.6.1 A Note on The Influence of Software [5]

Software science and software engineering have been the originators of much of what we call systems engineering. This occurred, on the one hand, because software programs soon became so complex that their development became natural applications for the systems approach and, on the other hand, because the abstract nature of software made it relatively easy to apply the abstraction that is central to the system approach. However, if we take systems engineering to literally mean the "engineering" of systems, then we need to recognise that software development is very different from engineering and, as a consequence, carefully consider to what extent some of the software-oriented developments in systems engineering are applicable to, or appropriate for, the broader scope of engineering.

Engineering can be viewed as consisting of two distinct, but closely coupled sets of activities; the development of technology based on natural science, and the application of this technology to meet the needs of society. That is, the end objective of engineering is to create objects that provide services, and a century ago that definition would not have raised any questions. The objects were machines, boats, bridges, substances, etc., and the services they produced were clearly identifiable, even if they were embedded in a greater collection of objects providing the ultimate service. As an example, consider a newspaper. Its service has many aspects, such as bringing information to its readers, allowing businesses to reach their customer base through advertising, and so on. Producing this service requires the interplay of many elements, such as the paper, the printing machine, the journalists, and so on, but there would be no doubt about characterising the printing machine as an engineered object and a story in the paper as not. Nobody would call journalism engineering or characterise the writing of a story as text engineering.

With the advent of computers, a new dimension was introduced, in that the engineered object now consisted of two distinct parts, the hardware and the software, and they exist in a sort of symbiotic relationship, in that each part is useless by itself; only together do they provide a service. However, the nature of this relationship depends on the application; in a simple application, such as the interlocking or automation of a piece of machinery, the instructions that make up the software are identical to the hard-wired connections in a corresponding relay logic, and we would not have any hesitation in calling the development of these instructions an engineering task. But the fact that the development of the instructions or the wiring connections is carried out using a formalism, Boolean algebra, that is part of mathematics, does not make mathematics a part of engineering. We do not speak of "mathematics engineering"; engineering uses mathematics, just as we use the laws of physics, and developing the instructions using Boolean algebra is no different to using calculus to determine the strength of a shell.

If we now move to more complex applications, as the development of an accounting program or a program for calculating the strength of a shell, the relationship between hardware and software has almost disappeared. It is certainly still true that without a computer to run on the software is useless, but the developer does not have to give much, if any, consideration to the characteristics of the hardware. As long as he obeys the rules of the programming language, his program can be compiled to run on any computer. And he does not have to have any domain-specific knowledge; in the case of the accounting program, accountants will specify what the program must do and the rules to be obeyed, in the case of the shell program, engineers will specify what it must do and provide all the equations etc. The situation is quite analogous to that of the journalist writing a story; he does not have to consider what paper it will be printed on or the characteristics of the printing machine, his skill lies in describing the observed facts in a manner that will meet the readers' needs, while adhering to the rules of the language. So, why do we call the activity of the software author software *engineering*, when we do not associate any engineering with the journalist's activity?

This question is not semantic nit-picking nor an attempt at demarcation; it arises out of a concern over the current direction and focus of systems engineering, as exemplified by the recent revision of the ISO standard, ISO 15288 (2002 vs. 2008), and the adaptation of a software modelling language, UML, to systems, as SysML. To me, this focus seems to ignore two fundamental differences between systems engineering and software development; the level of abstraction in the subject matter itself, and the level of creativity. Engineering is about successful outcomes, and where success is measured not by one's peers, but by the stakeholders. Both the problems in executing the projects and the measures of success are often largely of a non-technical nature, and the complexity that systems engineering is called upon to handle is only partly due to the advanced technology required or the multidisciplinary nature of the project. Factors such as market forces, financial backing, politics, union pressures, environmental interest groups, personal preferences, etc. are always significant and often dominating, and these are not factors that are effectively handled by introducing any formalism or abstraction. They require extensive communication with the diverse bodies involved, and natural language is the only realistic option for this. Furthermore, the engineering of hardware is about the engineering of real, physical objects, and that activity has its own elements of abstraction and associated formalisms and languages, in the form of engineering drawings and diagrams, architectures, and ontologies (to use a very *in* word, even if a bastardisation of the philosophical meaning). Software, on the other hand, is already an abstraction; software development can be thought of as a process for expressing given dependencies between actions (or variables) in a language that can be executed by computers, and as such software development lies somewhere within the triangle spanned by mathematics, logic, and linguistics.

Not that this process does not require great skill, but it is a skill analogous to that of the journalist; of choosing the right words and constructing sentences and paragraphs that produces the correct understanding in the reader's mind of the situation being described. It also involves creativity, but it is creativity on a completely different level to that involved in engineering. Whereas creativity in engineering is concerned with creating possible solutions and then selecting the best one of these, the creativity in software development is concerned with giving the computer the most correct and efficient "explanation" of what it has to do as its part of the solution. It involves no interaction with stakeholders at all.

Both engineering and software development can be highly complex processes, and the system approach is equally applicable in both cases. In the case of engineering, the application is called systems engineering; in software development it is called something like structured programming, and while some of the basic features are the same, as is to be expected, given their common origin, there are also many features that are specific to the different natures of the two domains, as discussed above. In particular, within a given engineering project, they are located in quite different parts of the process, as illustrated in Fig. C2.2.

It is also worth while noting that the relative effort expended on engineering and on software development can vary greatly from project to project. On a 2 billion dollar freeway project the software development component may account for 5 million dollars and the engineering 100 million, whereas on a project to develop a new ERP module, there may be no engineering at all. In the latter case, the requirements are developed by business analysts, and the hardware platform is existing. This variability is a further indication of the relative independence of the two activities, and dispels the notion that they are two sides of the same coin and therefore need to have a common approach.

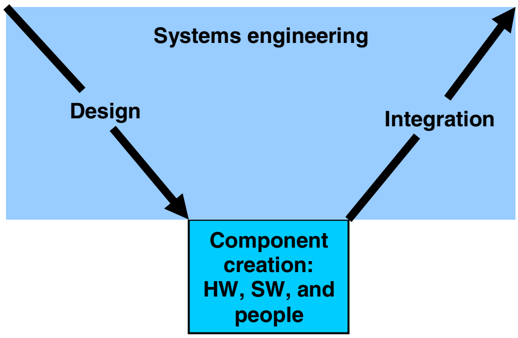


Fig. C2.2 Systems engineering and software development are located in different areas of an engineering project.

In summary, then, the significant differences between engineering and software development in the nature of both their products and their activities give rise to doubts about the current trend in systems engineering of adopting methodologies, representations, and tools from software development without a critical examination of their applicability. You should keep this in mind when assessing the applicability of the standards and current practice to your projects.

C2.6.2 Standards

Over the last fifty year or so, a number of standards and guides have addressed systems engineering, or aspects of systems engineering. Some of these have evolved and are still in use today; others have been superseded and are mainly of historical interest. The early development of systems engineering, at Bell Labs, initially for telecommunications and subsequently for some major defence projects, was supported by various company-internal and project-specific documentation, but the first publicly available standard was MIL-STD-499, *Systems Engineering Management*, released in 1969. The title is interesting, as it reflects the initial priority of the systems engineering effort: that of handling the complexity involved in *managing* these large projects and responding to the imperatives of the Cold War arms race. The title was maintained in the first revision, MIL-STD-499A, released in 1974, but in 1992 a draft version of the second revision, MIL-STD-499B, was released for comment with the title *Systems Engineering*, reflecting the importance placed on doing systems *engineering*, rather than just managing it. However, this version was never released, and the US DoD changed its policy to rely on commercial standards where applicable standards were available. Two such standards appeared in draft form at that time, EIA 632, *Processes for Engineering a System*, and IEEE Std 1220, *Standard for the Application and Management of the Systems Engineering Process*. (For completeness, it should be noted that the US Air Force Space Command issued version C of the standard in 1995 for its own use in supporting materiel acquisitions.)

It is also appropriate to mention the significant contribution of the US Defense Systems Management College, in particular through the publication of the *Systems Engineering Management Guide* (1990), and the impact of the book *Systems Engineering and Analysis*, by B.S. Blanchard and W.J. Fabrycky, first published by Prentice-Hall in 1981and now in its fifth edition.

Numerous other organisations in the defence and aerospace industry, such as NASA and the European Cooperation for Space Standardization, have published their own systems engineering standards, and the Australian DoD issued its *Capability Systems Life Cycle Management Manual* in 2002.

In 2002 the International Standards Organisation issued ISO 15288, *Systems engineering – System life cycle processes*. It was updated in 2008, with its title change to *Systems and software engineering – system life cycle processes*, and it has now become the central standard for systems engineering. As the title indicates, it presents systems engineering as a collection of processes, which results in a heavy emphasis on management aspects. Through the intent of the 2008 revision to harmonise it with ISO 12207, the standard's direct applicability to non-software-intensive industries has perhaps been somewhat reduced. Nevertheless, with judicious tailoring, the standard provides the framework for the application of systems engineering in most industries.

However, when applying ISO 15288 to any organisation, be it a permanent organisation, such as a company or a government body, or a temporary project organisation, there are a couple of issues that need to be considered. The first is that, as with many other ISO standards, in particular ISO 9001 *Quality Systems*, ISO 15 288 defines general features of the processes that must be met in order for an organisation to be able to claim conformance with any one of those processes; it does not say anything about the extent of the processes. Secondly, it says nothing about the procedures and tools that would be required to support an efficient execution of the processes.

C2.6.3 Current Practice

The heading of this section, "Current Practice", is a convenient short-hand for the headings of a number of more detailed, but related topics, and perhaps the most immediate one could be phrased as "What do most of the people who consider themselves to be systems engineers actually do?". The answer to this is found by considering the evolution of systems engineering. We touched briefly on this in Chapter B1, where we noted that a formative influence on systems engineering was its role as an enabler in the arms and space race of the Cold War. As such, considerable resources in the aerospace and defence industries and their government clients were invested in developing a framework of processes and procedures that would significantly increase the probability of completing complex projects to specification and within tight timeframes, and it resulted in a Body of Knowledge (BoK) embodied in numerous specifications, Data Item Descriptions (DIDs) [6], text books, and guides, as well as numerous courses provided by universities and other training institutions. It is the practice of this BoK, which we might call *classical* systems engineering, that still today constitutes what the majority of systems engineers do, and the reason is that the aerospace and defence industries are the only industries where a systems engineering framework is mandated, and therefore where the majority of professional systems engineers are found.

There is no single, authoritative guide to this classical BoK, although such products as the previously referenced *Systems Engineering Handbook*, published by the International Council on Systems Engineering (INCOSE), and the book by Blanchard and Fabrycky contain references to a significant part of the literature.

Notes and References

1. The complex contracting arrangements in a number of large infrastructure projects are discussed in section 6.7 of Managing Large Infrastructure Projects, ref. 3 below
2. A well-known assessment of the success of software project is the CHAOS Manifest of the Standish Group, `http://standishgroup.com`, but it has also been criticised by a number of sources, many of which can be found by simply Google on "standish report". However, there seems to be no doubt about the fact that large software projects have had a relatively poor rate of completion on time, within budget, and to agreed performance criteria, whatever may be valid reasons for this

3. The book Managing Large Infrastructure Projects, published by A.T. Osborne BV, 2008, is a most interesting and valuable documentation of lessons learned on 15 major infrastructure products in Europe, with the findings clearly organised into eight groups. The study specifically addresses Project Management, but because these are large and very complex projects, many of the problems encountered are those that arise in complex systems in general
4. Failure Modes, Effects, and Criticality Analysis (FMECA) is a well-established process, documented in numerous textbooks and articles, and supported by different many tools. The best introduction is to look it up in Wikipedia, `http://en.wikipedia.org/wiki/Failure_mode,_effects,_and_criticality_analysis`
5. This text appeared in the Newsletter of the Systems Engineering Society of Australia (SESA), No. 48 (July 2009); under the title Why Software is Different
6. Data Item Descriptions (DIDs) were originally defined as part of MIL-STD-498, which consisted of two parts, Overview and Tailoring Guidebook and Application and Reference Guidebook, and aimed at software development. However, the concept has proved to be of enduring value and applicability, and current defence contracts often require compliance with a large number of DIDs

C3 Architecting and Functional Analysis

C3.1 Introduction

In Sec. B4.3 we introduced the concept of functionality and the concept of the functional domain. In particular, we discussed the existence of a transition from the functional domain into the physical domain; that is, a transition from an abstract domain in which there are only requirements on services (essentially activities) to a domain in which these requirements are reflected onto an architecture, i.e. a set of interacting physical elements.

This transition is always present, and it can be seen as making a choice; a choice between the many physical architectures that can meet the requirements to a greater or lesser extent. There are a large number of methodologies for carrying out this choice, i.e. for performing the activity of *architecting* [1], but they all make the choice in the physical domain, effectively synthesizing a solution by trying out various combinations of subsystems, suppressing any explicit mention of a transition. However, as the requirements and the synthesized solution become more and more complex, it becomes increasingly unlikely to get anywhere close to the right performance on the first try and, even worse, progressively 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.

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. A small example of this is how a few capacitors and inductors can give rise to a band-pass filter. So, if we had initially required the properties of a (passive) band-pass filter, how would we have deduced that the elements we needed should have the properties of capacitors and inductors? This problem is the converse to the problem, discussed in Sec. A4.1, of predicting the emergent properties from the properties of the elements.

This immediate focus on the physical domain is also evident if we look at some of the defining documentation on systems engineering: 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 it is my experience that, more often than not, an unsatisfactory outcome of a project (i.e. a dissatisfied client) is due to a lack of attention to this transition rather than to any inadequacy of the engineering in the physical domain.

The solution to this problem lies in not making the transition directly, but via an intermediary step, in which the engineer finds the answer to the question: "What functions must a plant have if it is to satisfy the stakeholder requirements?" For example, if the service is to prevent ships from colliding at sea, the functions would include determining the locations of ships that are within a certain distance of each other, obtaining their current course and speed, determining the potential for a collision, having the means of influencing their trajectories, etc., and there would be requirements on each of these functions regarding such parameters as accuracy and latency. The set of functions for a particular plant is its *functionality*.

And so we are led to ask ourselves if there would be some way in which we could express the satisfaction of the stakeholder requirements as a set of interacting functions, each of lesser complexity, and then carry out the conversion into the physical domain for each function. In other words, would there be any benefit in introducing the system concept into the functional domain, and if so, how should we approach it? These are the questions we shall try to answer in this chapter, and in doing so, we are confronted with two central issues: how to partition a large set of functional requirements into a system of smaller, simpler *functional elements*, and how to measure the success of that process, as a prerequisite for optimisation. It will turn out that both issues can be resolved by introducing the concept of Return on Investment, already discussed briefly in Sec.B2.4, into the process, but before addressing these two issues, we need to look more closely at the concept of a functional element.

C3.2 Functional Elements

While the functions defining a plant's functionality are described by physical parameters, there is no mention of any physical plant used to realise them; in this sense the functions are *abstract* [2], and such abstract descriptions of functionality consist of *functional elements*. Functional elements are discussed in considerable detail in [3] and will be treated formally in the next chapter; briefly, a functional element is a description of one or more aspects of the functionality of a plant, and consists of a set of variables and a set of relationships between them, as well as any values required of the elements of these two sets. The use of "aspect" allows a functional element to describe not only immediate, physical functions of the plant, but also functions of interest to the wider stakeholder group, such as, for example, to provide an opportunity to build expertise (technology transfer), support political stability, etc. and, above all, to provide a return to the investors.

It is important not to lose sight of the fact that the functionality of a plant is what the plant does *in the context of a particular project*. It is not just its intrinsic capability to provide a defined service, but its ability to actually meet the stakeholder requirements throughout the project's timeframe. It is only under this perspective that considering the ROI as the ultimate purpose makes sense.

Because a functional element represents an activity, it is natural to consider its set of variables as containing two subsets: the *functional parameters* that describe the output or result of the activity, and the variables describing interactions required in order to provide the output, which we may call *dependencies*. The latter are related to the functional parameters through the set of *relationships* that forms part of the definition of a functional element and describes the behaviour of the element. But there is a third subset, consisting of those variables that, while they do not describe intended interactions, describe necessary external interactions. Typical examples might be ambient conditions and interest rate. We shall call these variables *influences*. Consequently, a symbolic representation of a functional element takes the form shown in Fig. C3.1.

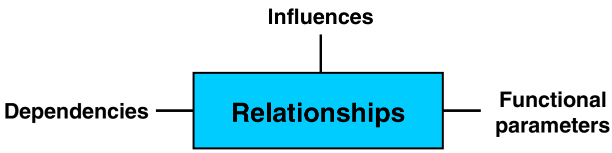


Fig. C3.1 Symbolic representation of a functional element.

At this point it is appropriate to acknowledge an earlier (and vastly more successful) approach to modelling and to the associated ontology development: the IDEF family of modelling languages for software engineering [4]. Some of the similarities are the requirement for a top-level function (called the A-0 diagram in IDEF0 [5]), the use of a standard building block, and the top-down (parent-child) development of models in terms of such building blocks. However, there are also some very significant differences. Firstly, functional element models are models of the project, not of the system. Secondly, they are not aimed at software engineering. Thirdly, they provide an interface between engineering and business, rather than between a government agency and its suppliers, and, fourthly, a functional element is an extension of an IDEF0 box in that it can represent *aspects* of a project, such as cost, revenue, and reliability. Finally, and perhaps most importantly, in a functional element model the top element is the same for all projects, whereas the A-0 diagram is unique to each system.

The totality of all possible functional elements make up the functional domain. With reference to Fig. B2.3, the functional domain is a domain of process universals, but once a particular physical realisation is chosen, the set of functions become part of the *functionality* of that plant. It instantiates the process universals and describes the ability of that particular plant to perform the processes required to produce the service. This difference between functionality and the functional domain is reflected in the fact that, in Fig. B2.3, there is a link from substance particulars to individual processes, but no link from substance universals to process universals. However, there is an "instantiate" link from individual processes to process universals. That is the implication of the link "participate in" between substance particulars and individual processes in Fig. B2.3; the instantiation of process universals requires linking the process to the particular plant performing it.

The *state* of a functional element is a set of parameter values, one for each functional parameter of the element. The space spanned by the functional parameters is the *state space* of the element. As an element goes through its timeframe, the state will pass along a trajectory in state space; the possible trajectories are limited by *constraints* in the form of the stakeholder requirements and the functions defining the element.

C3.3 The Top-Down Approach to Functional Analysis

In this book, *functional analysis* is the process of determining an architecture in the functional domain; in other areas, e.g. in mathematics [6], it has quite different meanings. That is, the outcome of the functional analysis is a set of functions that will provide the required service, without (at least at first) any consideration as to how these functions might be realised. These functions form a system; firstly, because the functions may interact in the sense that the output of one function is the input to another function, and secondly, because they relate to the same service. They interact in the sense that, together, they provide the service; if one function is removed, the reduced set is no longer able to provide the service. Recalling the discussion of emergence in Sec. A4.1, we could say that the service is an emergent property of the system description of the functionality, and we will call that description the *functional system*, to distinguish it from the physical system (which, once we make the transition into the physical domain, is most often just referred to as the system).

The first central issue in functional analysis is then: Starting out with a complex requirements definition document (RDD) (i.e. with numerous and interacting requirements on the functionality), how do we select a set of functional elements and their interactions such that the resulting functional system satisfies the RDD? We could try to pick a subset from a set of previously used elements and let them interact in a particular fashion. But we are faced with exactly the same problem as we encountered with the bottom-up approach to design in the physical domain, as discussed in Sec. B5.2. It is unlikely that the emergent behaviour of the functional system will meet all the requirements in the RDD. It will require a number of iterations to achieve that, and even then we have no way of knowing and demonstrating that we have found the best solution (the second central issue in functional analysis, to be discussed in the next section).

The approach taken in applying the system concept to functional analysis is the same as is used in applying the system concept to design in the top-down system design process mentioned at the end of Sec. B4.2. It is based on the fundamental process of *abstraction*: The process of moving from the particular to the general. It is the process which generates language: Words such as tree and house are abstractions of particular objects perceived by us; they are concepts generated by means of our understanding, as discussed briefly in Sec. A2.3. It is a process that defines *classes* of objects by what is common to all members of the set making up the class.

At this point it is important to recognise, if you have not already done so, that we have been using the word "abstract" to mean two different things. In Sec. C3.2, we characterised functional elements and the functional domain as abstract, meaning non-physical; now we are also using the word to mean generalised, or high level, or lacking in detail, i.e. the opposite of particular. But we can apply abstraction, in the sense of generalisation, in both the physical and functional (abstract) domain, and a functional element can be highly detailed (particular). In the next chapter we shall remove this ambiguity, at least as far as the functional domain is concerned; for the time being the meaning will have to be determined by the context.

In Sec. A3.1 we introduced the process of *chunking*, and, at a first glance, it might appear that it is a form of abstraction. That is not the case, because while chunking hides information, abstraction discards information. A typical chunk of information in engineering is a standardised construction element, such as an M4 bolt; going to the standard we can find all the information implied by the identification "M4 bolt". But going to the dictionary and looking up "tree", we will not find any definition of a specific tree. By using the concept of a tree, we lose most of the information about any specific tree, but by using the concept of an M4 bolt we have not lost any information about the object, we have only hidden it.

Consider now the universe of the functionalities of all engineering projects. The definitions of functionality will vary greatly, from a few requirements to hundreds of pages of requirements. But if we compare these definitions, there are groups of projects that have a number of requirements in common. We find these groups by discarding detailed requirements, and the groups become larger as more and more detailed requirements are discarded. Finally, we arrive at a number of very large groups of projects that each have a small number of very general functional requirements; these requirements define what we shall call the *purpose* of the projects in a group, or what Hitchins calls Prime Directive [7]. This purpose, defined in the form of a functional element (see Sec. B4.5), becomes the point of departure for the top-down design of the functional systems for the projects in the group. But before we describe that design process in the functional domain, there is one question we need to consider: Can we carry the process of abstraction to the point where all engineering projects have the same purpose? That would mean a significant simplification in the application of the top-down process in the functional domain, because for any new project, the determination of the purpose and obtaining stakeholder agreement to the definition of the purpose would otherwise be fraught with all the problems that make projects complex, as we saw in Ch. C1.

The answer to the question is, in my opinion, yes. In Sec. B2.4 we put forward the following argument: Every engineering project involves the expenditure of resources in some form; as labour, as materials, as natural resources, and so on. As a result, there is a *cost* associated with every project. It may not always be explicitly accounted for, e.g. if it is in the form of voluntary labour or a gift, but it is a cost attributable to the project all the same, in the sense of opportunity cost. But nobody would expend resources without the expectation of some form of *revenue*. Again, this may not always be accounted for directly in monetary terms;

it may be in the form of lives saved, personal satisfaction, disasters averted, and so on, but one way or another, the stakeholders value this ahead of other uses of the same resources, and this makes it possible to put a value on it.

The cost precedes the revenue, at least to some extent, and so it is in the nature of an *investment*, and the quantity of interest is the revenue relative to the investment, or the *Return on Investment* (ROI). The exact definition of this relationship in accounting terms may vary somewhat between various application domains and, in particular, due to the different compositions of the stakeholder groups. This will be discussed in Sec. C4.5, but for the time being the simplest relationship, such as revenue minus cost, divided by cost, is an adequate definition of ROI. The function of generating a return on investment then becomes the universal top element in developing a functional system; the starting point of design in the functional domain, and what we shall call the *irreducible element*.

C3.4 Optimisation and the Concept of Value

The second central issue now confronting us is that there is generally not a single functional system that will provide the service. There will be different sets of functional elements and different ways of combining these into systems that all will satisfy the functional requirements. A simple (or even simplistic) example of this is the function of laundering fabrics (clothing, bed-linen, towels, etc.). One approach to this is to have a washing machine in each household and have members of the household provide the manpower; this also requires the functions of producing, distributing, and maintaining these washing machines. Another approach is to have a centralised laundry with dedicated staff, but this requires the additional function of collecting the soiled items and delivering the laundered ones. The basic (high level) function is the same in both cases, but the representation of this high level element as a system of more detailed elements is different in the two cases, as are the elements that result from the transition into the physical domain. In particular, we note that the structures of the two systems are quite different: centralised versus decentralised.

Another example, this time from the mining industry, is the service of presenting the mined ore, called Run of Mine ore, to the concentrator plant. Analysing the service requirements (spatial separation between extraction point and concentrator location, and the concentrator's requirements on ore size), one immediately finds that there are two functions involved, transportation and size reduction. But even at this high level of analysis there are two basic options, either do the size reduction first and then the transport, or *vice versa*, and if we go to a further (lower, more detailed) level of analysis, we can distinguish the functions of vertical and horizontal transportation, and primary size reduction and secondary size reduction, and these functions can again be combined in various ways to form a number of different functional systems.

So, there will be more than one possible functional system, and we must make a *choice*. As we saw in Sec. B5.2, the process of identifying options and then making a choice is the essence of design, and the combination of identifying possible functional systems and then choosing the best one constitutes what we will call *design in the functional domain* [3]. But choosing the best functional system requires us to have a decision criterion; a definition of what we mean by "best". If all the systems provide the required service, providing the service would obviously be a useless decision criterion. However, in all but the simplest projects, it is often the case that certain aspects (i.e. parameters) of the service are fulfilled to a greater or lesser degree by different functions, which introduces the concept of the *value* placed on having a particular aspect fulfilled, as already discussed in sec. C1.5.1. The functions of value vs. degree of fulfilment are often S-shaped, as shown in Fig. C1.2, but the values of the parameters defining the functions will depend on the *judgement* of the persons (or groups of persons) providing the definitions, which we in Sec. B2.4 identified as the user group.

There are numerous approaches to unifying diverging value judgements, but a problem common to them all, at least to some extent, is the handling of different *units of measure*. With the value of some functions measured in dollars, of others on a scale of 1-100 %, of others again on a no value/desirable/essential scale, and so on, it is effectively impossible to reach an objective measure of overall value of a proposed functional realisation. The most obvious way out of this difficulty is to require all values to be measured in monetary units, e.g. dollars or euros, but there are commonly a number of objections being put forward to this approach. One, it is unethical to put a price on everything, there are human values, such as beauty, human life, biodiversity, and generally the quality of our environment, that cannot be measured in monetary terms. But is it really any more "ethical" to hide behind such phrases as "zero harm", "highest quality", and "all possible care"?

Two, and related to this, is the opinion that certain things, in particular human life are simply beyond any measure; they are just infinitely valuable. This only has the effect of making any rational allocation of resources impossible, as was the case in a meeting with the engineering group of a major mining company, when it was stated that it was company policy that the probability of a failure that could lead to a fatality had to be zero.

Three, a monetary measure is too precise, and may often give a false impression of the actual state of knowledge about the value of a function. This objection is easily overcome by using ranges of monetary values, expressed either as actual ranges, such as \$20 - \$50, or as a value with a tolerance, such as \$35 ±40 %, or even as an order of magnitude.

Four, and perhaps the most common objection to putting a monetary value on a function is the reluctance to be pinned down and to put *any* value on it; in this way one cannot be held to account later on. This attitude is particularly prevalent where the decision-making is open to public scrutiny, such as in politics. Decisions are made in these areas also, it is just that the reasons for the decisions are sometimes quite confused and irrational, and the decisions are therefore wrong, as far as achieving the stated goal goes.

However, the greatest benefit of assigning a monetary value (or value function) to every function is that it allows us to define a common decision criterion for all projects. This criterion, which expresses the purpose of any project when we abstract from all project-specific features, becomes the point of departure for a step-wise, top-down development of a system description of the functionality.

The argument for this criterion is very simple: Who would not want to conduct a project so as to get the greatest revenue for the same cost? Or, conversely, who would not want to choose the least cost approach to obtaining the same revenue? Consequently, *maximising the return on investment (ROI) is the common purpose of all engineering projects*, and it becomes the universal optimisation criterion.

Or course, this presupposes that both cost and revenue have been defined to reflect the judgement and values of the stakeholder group, and so the introduction of the ROI does not change the fact that one needs to develop and agree a consensus view of the project. But it does have two very significant effects: Firstly, by providing a common measure, it puts the discussion about the value of different aspects of the project on a rational basis; it forces all stakeholders to recognise that fulfilling any particular requirement will have a cost, and that this cost must be balanced by generating a revenue. Secondly, once the ROI has been defined for a particular project, it provides a separation between the stakeholders and the engineer with regard to optimising the design; the engineer can progress the design and make choices between options without any further consultation with the stakeholders (at least in principle).

Value and optimisation are so central to the application of the system concept to engineering presented in this book that the whole of the last chapter is dedicated to them.

C3.5 Architecture Descriptions

Consider a completed, operating plant, and imagine that you are given the following task: "Describe the architecture of this plant". How would you go about it? As a most obvious first step, you might identify the main hardware elements (subsystems, equipment, etc.) and how they are interconnected (e.g. through material flow); this would be one architecture description. Then, you might be shown how the operation of the plant is controlled by software, and see that this software consists of several elements (applications, packages, modules). Again, these elements interact (by passing data between them) and thereby form a structure, and so you have a second architecture description. Then you talk to the production manager running the plant, and he gives you a description in terms of what he does, the products the plant produces, and the various processes involved in going from raw materials to products. These processes (materials handling, storage, mechanical and chemical processes, etc.) are linked in a definite structure, and so form a third architecture description.

You come to realise that "the architecture" of the plant is an abstract concept, instantiated by numerous different types of descriptions. When you were asked to describe the architecture, you should have been told what the description was to be used for and who was going to use it, as the appropriate descriptions depend on the *view* one takes of the plant.

In some industries or jurisdictions, agreement has been reached on which views are required for their purposes and which descriptions are required to document each view; such an agreed set of views and descriptions is called a *framework*. Perhaps the best known one is the US Department of Defence Architecture

Framework, or DoDAF [8]. This framework, which defines a way of representing an enterprise architecture that enables stakeholders to focus on specific areas of interests in the enterprise, while retaining sight of the big picture, has a number of views:

- The All Viewpoint describes the overarching aspects of architecture context that relate to all viewpoints.
- The Capability Viewpoint articulates the capability requirements, the delivery timing, and the deployed capability.
- The Data and Information Viewpoint articulates the data relationships and alignment structures in the architecture content for the capability and operational requirements, system engineering processes, and systems and services.
- The Operational Viewpoint includes the operational scenarios, activities, and requirements that support capabilities.
- The Project Viewpoint describes the relationships between operational and capability requirements and the various projects being implemented. The Project Viewpoint also details dependencies among capability and operational requirements, system engineering processes, systems design, and services design within the Defense Acquisition System process. An example is the Vcharts in Chapter 4 of the Defense Acquisition Guide.
- The Services Viewpoint is the design for solutions articulating the Performers, Activities, Services, and their Exchanges, providing for or supporting operational and capability functions.
- The Standards Viewpoint articulates the applicable operational, business, technical, and industry policies, standards, guidance, constraints, and forecasts that apply to capability and operational requirements, system engineering processes, and systems and services.
- The Systems Viewpoint, for Legacy support, is the design for solutions articulating the systems, their composition, interconnectivity, and context providing for or supporting operational and capability functions.

The DoDAF documentation emphasizes that DoDAF is fundamentally about creating a coherent model of the enterprise to enable effective decision-making; however, as is evident even from the snapshot above, this framework is very defence-specific, and the decision-making process supported is embedded in the DoD’s acquisition framework. In Chapter C6 we shall develop a model that is more directly aimed at commercial enterprises.

C3.6 Addition to the Engineering Ontology

In the above development of functional analysis, we have implicitly defined a number of new concepts related to the functional domain. These need to be added to the ontology we started developing in Sec. B2.4, and their definitions are contained in Table C3.1.

Table C3.1 Definitions of concepts used in functional analysis.

Name	Definition	Synonyms/Examples
Functionality	The capability to perform the functions required to provide a service	ability; performance
Functional element	A description of one or more aspects of functionality	
Functional parameters	The parameters describing the result of the activity performed by a functional element (i.e. its output)	output parameters
Dependencies	The parameters describing the interactions of a functional element with its environment, required in order to perform its function	input parameters
Influences	The parameters describing the intrinsic (i.e. non-function specific) interactions of a functional element with its environment	environmental parameters
Functional domain	The set of all functional elements	solution space
State (of an element)	A particular set of values of the functional parameters	activity level
State space (of an element)	The set of all possible states.	allowable parameter values
Constraints	The limitations on allowable parameter values	boundary values, ranges

Some of the relations between the concepts in Table C3.1 are illustrated in Fig. C3.2.

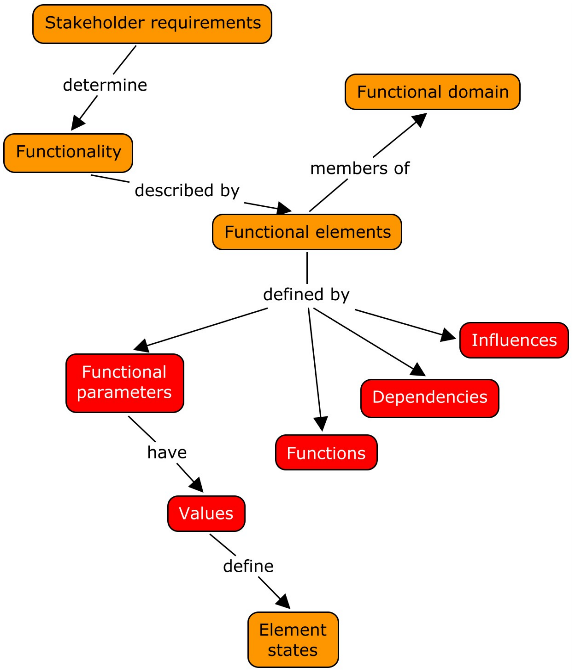


Fig. C3.2 Relationships between the concepts defining functionality.

References

1. A central figure in the development of systems architecting was Eberhardt Rechtin, and his development of the subject matter is contained in the two books. Systems Architecting: Creating and Building Complex Systems. Prentice-Hall (1991); The Art of Systems Architecting, with M.W. Maier, 2nd edn. CRC Press (2000)
2. A discussion and illustration of this abstraction is contained in E.W. Aslaksen. The System Concept and Its Application to Engineering. Presented at COGIS (2009), `http://www.gumbooya.com`
3. Aslaksen, E.W.: Designing Complex Systems – Foundations of design in the functional domain. CRC Press (2008)
4. `http://en.wikipedia.org/wiki/IDEF`
5. Integration Definition for Function Modeling (IDEF0), Federal Information Processing Standards (FIPS PUBS) 183, issued by the National Institute of Standards and Technology, December 21 (1993)
6. Riesz, F., Sz.-Nagy, B.: A standard, but old textbook on functional analysis is Functional Analysis. Frederick Ungar Publishing Co., New York (1965); Otherwise, Wikipedia contains a list of textbooks and lecture notes on the subject
7. Hitchins, D.R.: Systems Engineering: A 21st Century Systems Methodology. Wiley (2007)
8. The current version, 2.02, can be downloaded from, `http://dodcio.defense.gov/sites/dodaf20/`

C4 The Functional Domain

C4.1 Foundations

The functional domain was introduced in Sec. B4.3, and there is nothing about this concept that limits it to systems engineering; functionality is a feature of all engineered objects, and the abstraction away from any particular physical object has been utilised by engineers in their search for better solutions since the very beginning of engineering. This is exactly what James Watt did when he realised the function of creating a vacuum by condensation in the form of a separate condenser, instead of it being integral to the cylinder. However, as this example demonstrates, the search for solutions was always carried out in the physical domain; the functional domain was never considered as a domain in which one could actually perform engineering. This was, and still is, quite appropriate and efficient for simple functions. But as the functions become more complex, the process becomes inefficient, as we already mentioned in Sec. B4.4, and we want to improve the efficiency by carrying out some of the design in the functional domain before making the transition into the physical domain.

To do that, we need to develop a much more detailed understanding of the functional domain. That is the purpose of this chapter, and to that end we provide a rigorous foundation by formalising some of our earlier concepts. As a starting point, we provide a formal definition of a functional element,

Definition C4.1 A *functional element* is a description of one or more aspects of the interaction of a plant with the outside world. It consists of a set of variables and a set of relations between them, as well as any values required of the members of these two sets.

Note that this definition does not limit the interactions to intended interactions. So, while the functionality of a plant is described by a functional element, there may be other functional elements associated with a plant, representing unintended interactions. As a result, we extend the definition of the functional domain:

Definition C4.2: The *functional domain* is the set of all functional elements.

The types of variables associated with a functional element were discussed briefly in Sec. C3.2; this can be formalised by the following definitions:

Definition C4.3: The *functional parameters* associated with a functional element are the variables that describe a subset of the stakeholder requirements.

Definition C4.4: The *dependencies* associated with a functional element are the variables describing the interactions with the outside world or other functional elements in order to meet the element's subset of the stakeholder requirements.

We recall that the stakeholder requirements are defined by a set of parameters *and* a set of values on some or all of these parameters. That is why a functional element also needs to encompass both functional parameters and their required values.

Definition C4.5: The *influences* associated with a functional element are the variables describing the inherent interactions with the outside world.

The interactions described by the influences are those that are not part of the interactions intended in order to fulfil stakeholder requirements, but inherent in the physical nature of the plant and the fact that it exists within a physical environment. Typical examples of such variables range from ambient temperature and pressure to interest rate and technology maturity.

In addition to the variables, a functional element is defined by the *relationships* that relate the dependencies to the functional parameters. However, these relationships, and thereby the dependencies, are particular to a given plant, i.e. to a given solution to meeting the stakeholder requirements. And this is, of course, equally true of the influences. But our stated aim, at least in the first instance, is to develop the foundations of design in the functional domain; that is, partitioning the stakeholder requirements into subsets of requirements and then representing these subsets as a system of functional elements *prior* to making the transition into the physical domain. Consequently, our focus in this chapter will be on properties of functional elements defined entirely by the functional parameters; this is *abstraction* in one of the two senses of the word discussed in Sec. C3.3, going from the physical to the functional view of a functional element.

The transitions between the physical and functional domains are of fundamental importance in engineering. They are a reflection of the human intellectual capability of transitioning between the observed and the conceptual, as discussed in Sec. A2.3. In order to *operationalise* this capability, that is, to turn it into a methodology or tool that can be used effectively in design, we have introduced functional elements. A functional element has, so to speak, one leg in the physical domain and one in the functional, and for all that follows, we need to have an absolutely clear understanding of what this dual nature of a functional element entails. So, at the risk of being slightly repetitive, let us go through the features of a functional element again:

- According to Def. C4.1, a functional element is a description of (one or more aspects of) the *functionality* of a plant.
- According to Def. B4.1, functionality of a plant is its *intended* capability for interacting with its operating environment.
- The intention is that of the engineer; the engineer designs the plant to have a particular functionality with the intention of meeting the stakeholder requirements. As there may be several ways of achieving this, there may be several different systems of functional elements associated with the same set of stakeholder requirements. In particular, we understand that functional elements are *not* parts of stakeholder requirements; they only arise through the engineering effort to meet those requirements.
- A functional element is tied to physical plant, but it describes only what the plant is intended to *do*, not what it *is*; so, even though the functionality may be described in great detail, the functional element *represents* a (potentially large) *class* of individual plants; all those plants that are intended to do the same things, but accomplish this by using different construction elements, different plant-internal interactions, or different material properties (e.g. steel instead of concrete).

So, what we have, so far, are three related entities: Stakeholder requirements, functional elements (in the functional domain), and plants (in the physical domain). To a given set of stakeholder requirements there corresponds one or more functional elements, and to each of these functional elements there corresponds one or more (usually a large number) of plants (i.e. physical realisations), as illustrated in Fig. C4.1.

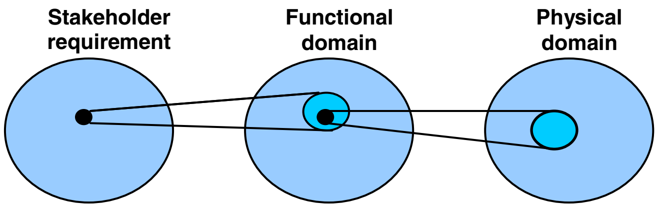


Fig. C4.1 Illustrating how a functional element is related to a set of stakeholder requirements and to a set of plants.

We see that a functional element can be viewed under two perspectives: as providing the functions that will satisfy a set of stakeholder requirements, or as describing the interactions of a plant with its operating environment. In this sense, a functional element provides a link between a set of stakeholder requirements and possible means of meeting them, and it is, of course, this property we want to exploit in our design methodology. We first find a representation of the stakeholder requirements in terms of functional elements and then address the issue of the physical realisation of the elements. However, referring back to our discussion of engineering activities in Secs. B2.4 and B4.2, a functional element may be viewed as completely independent of any stakeholder requirements, in the sense that it represents the capability of a plant (equipment, device, component) to do something, without any reference to *why* this needs to be done, i.e. without embedding it in any project. It represents an item of technology; an item in our engineering resource base, and as such is a reflection of standardisation, as discussed in Sec. B6.3.

To finalise this discussion of the dual nature of functional elements, we need to consider the extension we made of the functional domain to include not only the intended interactions of plants, but also the *unintended* interactions with the outside world. The probabilities of these interactions occurring during the life time of the plant may be so small that we ignore them, or small enough that we accept them without any further design effort, but their consideration is becoming an increasingly important component of the design process, as we shall see later. For the moment, the importance of this extension is that it allows us to make the following definition:

Definition C4.6: The element describing all possible interactions of a plant with the outside world is the *maximal element* associated with that plant.

A maximal element is not something we would be able to realise or use in a design activity; it is a limiting case, much as probability is a limiting case of letting the number of trials go to infinity. But it has the distinction of being completely independent of any stakeholder group or any intention of a designer; it is a characteristic of the plant and exists wholly within the physical domain.

Quite the opposite is the case with a complete element:

Definition C4.7: A functional element is *complete* if it is adequate for expressing (or defining) all the functional requirements in a set of stakeholder requirements.

A complete element is defined solely in relation to a set of stakeholder requirements; it does not need to refer to any physical realisation, nor make any assumptions about, or mention of, dependencies or influences. In particular, it follows that the functional parameters must span the same space as the parameters used to express the stakeholder requirements. And the relationship depicted between the stakeholder and functional domains in Fig. C4.1 can now be more precisely defined in terms of complete elements; the set of functional elements corresponding to a given set of stakeholder requirements is its set of complete elements.

Maximality and completeness represent two very different views of a plant's interactions with the outside world, but because we postulated that every plant reflects an engineer's intent to meet a set of stakeholder requirements, there is one and only one maximal element and at least one complete element associated with every plant. And, in particular, it follows that every maximal element is complete. (If the plant then actually performs as the engineer intended is a different matter.) However, to a given complete functional element there will correspond numerous realisations in the form of plants, and therefore possibly numerous maximal elements (i.e. differing in their unintended interactions).

Many of the properties of functional elements discussed in this chapter were developed in earlier publications, in particular in [1]. However, for various reasons, the emphasis there was on functional elements as representations of the functionality of plants, and consequently the development was based on maximal elements. Here, we are focused on the initial stage of design based on the stakeholder requirements – design in the functional domain – and so we are primarily concerned with functional elements as representations of these requirements. This is reflected in the next two sections.

C4.2 Some Properties of Complete Functional Elements

Because our main interest in functional elements is in using them to partition sets of stakeholder requirements, we need to understand what is involved in representing a complete functional element as a system of "smaller" elements. Consider the situation where we have been given a set of stakeholder requirements and have, through a process of analysis of these requirements, developed a complete functional element. That is, we have determined a set of functions such that, if we created a plant that had these functions (i.e. had this functionality), it would meet all the stakeholder requirements. Then, the stakeholders add a further requirement; for example, in addition to the capability of transferring iron ore from train to stockpile to ship, the service shall now include the ability to blend ores from different mines. We can, again, develop a functional element that is complete with respect to this new set of requirements. But while adding requirements in the stakeholder domain, it is not clear what we are doing in the functional domain. We are going from one complete element to another by adding something, but what is this "something" we are adding? It is a function the plant must have, so it can be represented by a functional element. But is it a complete functional element? Well, in principle there could be a project solely to provide the service of blending iron ore. It would have a whole set of detailed requirements on this capability, such as throughput, number of different ores, how the ore is provided to the process and how it is to be delivered from the process, and so on. There would be a value placed on this service, and the project would have a return on investment. This all could then be represented by a complete functional element, but a complex element in the sense of having a large number of variables and functions relating them. In practice, and in the case we are considering, the service is to be provided as an addition to an existing service. Most of the detailed requirements are already present, so that the functional element that represents the blending function *as an element in a system of functional elements* representing the required overall functionality is relatively simple.

There is clearly an issue here regarding the relationship of complete functional element to functional elements in general, i.e. without any reference to a project, that will have to be resolved. That is the subject of the next section. But first let us return to our question about the operation of adding a functional element to an existing complete element in order to create another complete element. How is that element related to the additional stakeholder requirement? The short answer is: There is no direct relationship. To see why this is so, recall how complete functional elements are created in a top-down fashion, starting with the irreducible element representing ROI, as first introduced in Sec. B2.3 and then discussed further in Sec. C3.3. This process introduces a linking between the elements belonging to a complete element, and this linking has two aspects to it. One, it is the relationship of increasing detail, such as splitting cost into acquisition cost, operating cost, and maintenance cost; i.e. a parent-child relationship, and we shall call this process *decomposition*. The converse process, which we shall call *condensation*, then expresses a *self-consistency* requirement by the requirement that to any one parameter there corresponds one and only one set of parameters that will allow a condensation; in the case of operating cost, it is the set of acquisition cost and maintenance cost that will allow a condensation to cost. Two, the decomposition process does not only create new variables, but, above all, new, more detailed functions. These functions introduce new, more detailed relationships between the variables. These two aspects are illustrated in Fig. C4.2, where the rings represent successive levels of detail. And the fact that they are rings rather than disjoint rays radiating from the centre is supposed to show that the variables within one level of detail are all related.

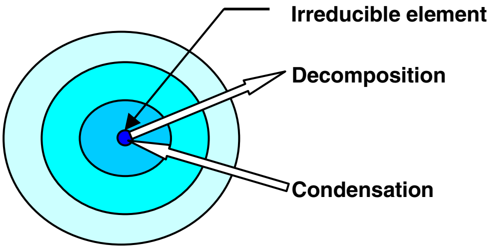


Fig. C4.2 Levels of detail and relationships between functional variables (functional parameters and dependencies). Variables in one ring are at the same level of detail and are related through the functions at that level; variables in different rings have a parent/child relationship.

The important point here is that, as we add functional elements in the decomposition process, the resulting elements are complete with respect to a larger and larger subset of the stakeholder requirements, until we arrive at an element that is complete with regard to the whole set of stakeholder requirements. But what element we add in order to cover one or more specific stakeholder requirements is up to us; there will in general be several different functionalities that will satisfy the same set of requirements. And the effect of adding a particular element (such as blending) depends on its *interaction* with the existing elements. The aim is to cover the stakeholder requirements with as simple a system of functional elements as possible, making the satisfaction of the stakeholder requirements *emerge* as a result of the interaction of (relatively) simple elements; this is the art of system architecting.

With this we can define a further concept that will be useful in the next section:

Definition C4.8: The *included set* of a functional element consists of all those functional elements that are generated through the process of condensation.

That is, whatever aspects of the functionality of a plant the functional element describes, the included set is the set of elements that describe those same aspects, but at higher levels (i.e. in less detail).

The following theorem follows directly from our assertion that maximising the ROI captures all the stakeholder requirements of any plant at the top level:

Theorem C4.1: The included set of a complete functional element contains the irreducible element.

In keeping with our focus on stakeholder requirements, it is convenient to define the complexity of a functional element as follows:

Definition C4.9: The *complexity* of a functional element is equal to its number of functional parameters.

This is a very simple definition, as it does not explicitly take into account how the functional element satisfies the stakeholder requirements by means of its functions and dependencies. But if we recall what we said about our ability to handle a set of variables as representing a single object in Sec. A3.1, then this definition does reflect our ability to conceptualise and manipulate a functional element. Or, in other words, it reflects the usefulness of a functional element as an element in our design process.

C4.3 Types of Functional Elements

We recognise that a functional element can be a description of doing something, completely independently of why we are doing it and of any project and stakeholder requirements. In the case of the function of blending, introduced in the last section, it is the same whether we are blending iron ore or the ingredients for making bread, and is characterised by a small set of variables, such as the number of components, the range and accuracy of the composition, and the throughput. There is no question of completeness; the completeness of a functional element only arises once we relate it to a particular set of stakeholder requirements. The ore handling project will have a very large number of stakeholder requirements, one of which is the requirement for a consistent quality of the ore delivered to each customer, and our analysis of this requirement has resulted in requiring the plant to have the function of blending. The system of functional elements covering all the stakeholder requirements is again a functional element, and it is this element that has the distinction of being complete.

So, there would seem to be two types of functional elements: those that can express the functionality of a plant to meet stakeholder requirements at some level of detail, and those that can not. But how can we define these two types within the functional domain; that is, without making reference to stakeholder requirements? This is where the concept of the included set comes into play:

Definition C4.10: A functional element is a *real* functional element if and only if its included set contains the irreducible element.

Definition C4.11: A functional element is an *imaginary* functional element if and only if its included set does not contain the irreducible element.

Relating this to the process of condensation, and keeping in mind that a functional element is a representation of an aspect (or aspects) of a plant's functionality, we obtain the following theorem:

Theorem C4.2: A functional element is a real functional element if and only if it is a member of the included set of a complete element.

Proof: The "if" part of the theorem follows immediately from the Def. C4.8 and Theorem C4.1. The "only if" part follows from the fact that any functional element that condenses to the irreducible element, as real elements do, must include within it the definition of the purpose of a project, to some level of detail. Consequently, there exists a complete element representing the stakeholder requirements of that project, and by Def. C4.8, the real functional element is a member of the included set of that complete element.

A functional element is either a real or an imaginary element, so that the functional domain consists of two disjoint parts. However, real elements are systems of imaginary elements. This is illustrated in Fig. C4.3, where the functional elements are depicted as triangles in a process of decomposition/ condensation, with the real elements condensing from the stakeholder requirements to the irreducible element.

To make Fig. C4.3 less abstract, consider the specific case where the stakeholder requirements include a requirement on the maximum value of the penalties incurred per year due to failure to provide the specified service. So, although the stakeholder requirements do not include any requirement on reliability of the plant, the decomposition will, at some level, need to introduce service reliability as a parameter. To what extent this aspect is then further decomposed depends on the nature of the plant and the level of detail in the stakeholder requirements.. But the point is that condensing the element representing the aspect of reliability will never reach the irreducible element.

Being reliable can, in itself, never result in a return on investment; it can only be an aspect of a plant that provides a service and thereby generates a return on investment.

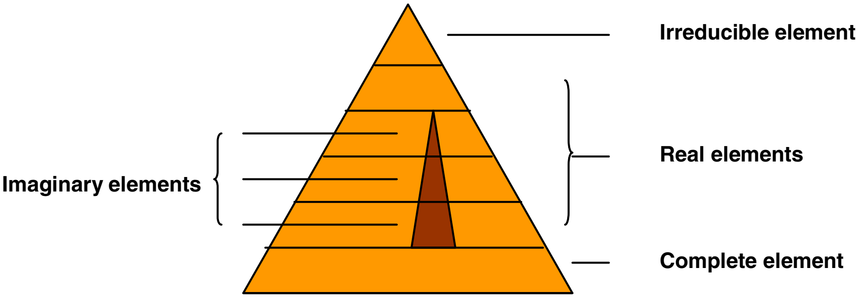


Fig. C4.3 As an illustration of decomposition, the figure illustrates the progress from irreducible element to the complete element that represents all aspects of the stakeholder requirements. As an illustration of condensation, the figure illustrates how the irreducible element represents the purpose common to all projects. The lighter coloured "slices" show the progression of the real elements making up the included set of the complete element, the "slices" of the darker coloured triangle show the progression of the imaginary elements making up the included set of a particular aspect.

Naming these two types of elements real and imaginary is, of course, a somewhat arbitrary choice, but it is motivated by the fact that the functionality of a real plant is always described by a real element; the functionality described by an imaginary element is one where we have removed some of the aspects that are found in a real plant. That is, we imagine an abstract "plant" that has only the limited functionality we are focusing on at the time.

We understand that an imaginary functional element expresses one or more aspects of what a plant does in the context of a project. But it will probably already have occurred to you that the property of being reliable is very different to a function such as generating power, providing a telecommunications service or transporting ore. The latter is *what* a particular plant does, the *service* provided; the former is a characteristic of operating any plant. Allowing functional elements to describe both kinds of functions was the reason for introducing the word "aspect" into the definition of a functional element. We shall therefore want to subdivide the set of imaginary elements into two further types: those that describe an action, which we shall call *primary elements*, and those that describe properties or consequences of actions, which we shall call *secondary elements*.

An important example of the distinction between primary and secondary elements is provided by two elements we are quite familiar with by now: the Cost and Revenue elements that describe these two dependencies of the irreducible element. The Revenue element is a primary functional element: generating a revenue is a measure of what a plant does, of its purpose. The Cost, on the other hand, is a secondary element: the purpose of a plant is not to generate a cost; the cost is a consequence of having to create and operate a plant in order to provide the service. This distinction is reflected in the further execution of the top-down process; the elements that result from describing the Cost in greater and greater detail will all be secondary elements. They will describe costs arising from the development of the purpose in greater detail. That is, the cost elements will be related to elements describing functions or properties of functions, but the latter two types of elements are primary and secondary elements, respectively.

C4.4 Primary Functional Elements and Linguistics

A primary 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 classifying primary elements according 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 elements. On the other hand any element 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 something” 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 primary element 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, such as “doing A, then B, and then C to something, while at the same time doing D to something else”. In other words, a primary element can be a system of (smaller) primary elements.

Finally, a real FE will 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, reliability, and cost-effectiveness, and then their required values (or ranges of values). This is what we alluded to earlier; a real FE is a system of imaginary elements, each one describing an aspect of the functionality. What we call the “basic” function of a real functional element is the “bare” primary element, stripped of its associated secondary elements.

As noted above, a primary functional element has the basic 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. C4.4:

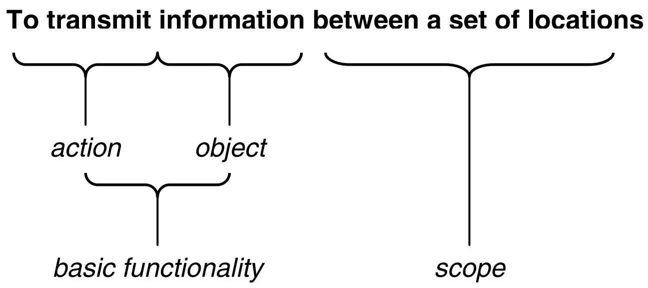


Fig. C4.4 The components of a primary element.

From this example we deduce that a primary element consists of three components that 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). This basic functionality defines a *class* of primary elements.

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 action. 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. C4.5. 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.

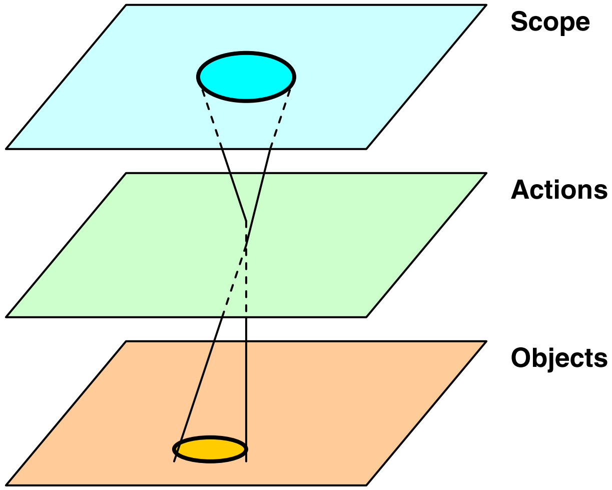


Fig. C4.5 The relationships between the components of a primary element.

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 functional elements, and it is therefore reasonable to first investigate whether there is a natural grouping of verbs.

In approaching this investigation, we recognise that there is a number of different perspectives on verb classification. One of these is the linguistic perspective: a very simple classification based on syntax is into transitive and intransitive verbs. Another well known example is the direct semantic grouping, with verbs within a group being synonymous, as in a thesaurus. A more sophisticated grouping, based on components of meaning and identified by various aspects of syntactic behaviour, in particular the so-called diathesis alternations - alternations in the expression of the arguments of the verbs, sometimes accompanied by changes in meaning - and a seminal work here is that of Levin [2], with more recent work referenced in [3].

Other perspectives are those that are related to particular areas of activity, of which engineering is one. The services that can be satisfied by an engineered object are those that can be measured in terms of physical parameters, such as weight, colour, etc. Therefore, the corresponding actions can be grouped according to their physical effect, i.e. into the following three groups:

1. Those that constitute a *translation in space* (transport)
2. Those that constitute a *translation in time* (store)
3. Those that constitute a *transformation* or change (process)

We note that other areas of activity may have other or additional groups. For example, commerce will have actions related to change of ownership, such as *sell* and *barter*, which are not found in engineering.

Within each of the three groups we can further group the verbs according to the objects they can take. The following grouping suggests itself:

a. Electromagnetic energy, i.e. pure energy without any rest mass
b. Information, i.e. the pure non-material content (disregarding the material carrier or energy required to contain the information)
c. Goods, i.e. items with size, weight, colour, etc.

Within each of these two groupings, a verb can belong not only to a single group, but to two or even all three groups. Thus, if we classify a verb by a tag of the form (1, 2, 3, a, b, c), i.e. as a six-component vector, where each component can take on one of the two values 0 or 1, then this classification contains $2^6 = 64$ groups. (Note that if we leave out the commas between the component values, as we shall do, the tag looks like a six digit binary number, but it has, of course, none of the properties of a binary number.) A few examples will illustrate this:

convert (001111): To convert the frequency of the power from 50 Hz to 400 Hz. - To convert the format of the files from .doc to .pdf. - To convert the iron to steel.

distribute (100111): To distribute power from the substations to individual households. - To distribute the information throughout the organisation. - To distribute the newspaper to all subscribers.

mine (101011): To mine the seabed diamonds. (As far as the diamonds are concerned, this is a pure transport activity, but it is also a process.) - To mine the coal. (Both a transformation and a transport activity.) - To mine the data base for relevant information. (A process.)

process (001011): To process the data in order to determine the position of the satellite. - To process the raw materials into masonry products.

wash (001001): To wash the coal.

C4.5 The Irreducible Element

The above introduction of real and imaginary functional elements leads us to the question of the nature of the irreducible element itself: is it a real or an imaginary element? On the one hand, it is a characteristic of any project (i.e. of the operation of any plant over its lifetime) and would therefore seem to be an imaginary element. On the other hand, according to Def. C4.11 it is clearly a real element. The answer is that the irreducible element is a real element. The apparent contradiction only arises because we are so used to thinking of ROI in accounting terms, e.g. as the monetary return to the equity providers. The irreducible element encompasses all aspects of the functionality of a plant *in the context of a particular project*. It is only because we chose to measure the effects of those aspects in monetary terms and classify them as either cost or revenue for the purpose of design optimisation that the ROI element looks somewhat like its much more restricted accounting namesake.

Because the irreducible element is the same for all projects, we can define it in more detail as long as we do not introduce any variables or functions that would be project-specific. In accordance with the format of functional element introduced in Sec. C3.2, the functional parameter is the return on investment, which we had already decided to identify by **U** (Sec. B2.4). It is defined in terms of the two dependencies *cost*, **C**, and *revenue*, **R**, as follows:

$$\mathbf{U} = \mathbf{R}/\mathbf{C} - 1.$$

Both **C** and **R** are defined as the Present Value (PV) of all costs and revenues attributable to the project over its time frame or *life*, L, measured in *accounting periods* (months, quarters, years, etc.). Let C(n) and R(n) be the cost and revenue in accounting period n, then

$$\mathbf{C} = \sum_{n \in L} \frac{C(n)}{(1+p)^n}; \quad \mathbf{R} = \sum_{n \in L} \frac{R(n)}{(1+p)^n};$$

where p is the *discount rate* per accounting period.

With this, the symbolic representation of the irreducible element is as follows:

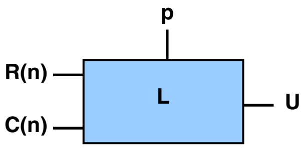


Fig. C4.6 Graphical representation of the irreducible element.

C4.6 Structure of the Functional Domain

With the understanding of the characteristics of functional elements we developed in the previous sections, we can see that the functional domain is not just a collection of such elements, but that these characteristics introduce a structure into the domain. We have already encountered one aspect of this structure: The functional domain consists of two disjoint parts, the real and the imaginary functional domain. We recall that a real element can be thought of as a system of imaginary elements that could completely satisfy the set of stakeholder requirements for some project at some level of detail. That is, for any particular stakeholder requirement, there must be one or more functional parameters and their values that represent the satisfaction of that requirement. So, intuitively, we would say that real functional elements that differ only by one or a few functional parameters must be *close* in the sense that the projects whose stakeholder requirements they represent must be quite similar. We can formalise this by the following definition of the distance between real functional elements:

Definition C4.12: Let x and y be two real elements, and let X and Y be the corresponding set of functional parameters. Then the *distance* between x and y, d(x,y), shall be defined by

$$d(x,y) = c(X \cup Y) - c(X \cap Y),$$

where c(X) is the cardinality of X (i.e. the number of parameters in X).

In a very simplistic manner, we can visualise this as illustrated in Fig. C4.7. The two sets of parameters, X and Y, have the parameters of the irreducible element in common, and if the level of detail is reduced (i.e. X and Y become smaller), x and y will move towards the irreducible element. It is therefore clear that with the above definition, the distance between two real elements depends no only on the difference in functionality but also on the level of detail of the description. If we, for a moment, interpret Fig. C4.7 *in the context of a particular project*, then there will be one or more real elements that are complete, and we can visualise them as lying on the boundary of the real part of the functional domain, with the elements between them and the irreducible element being the included sets.

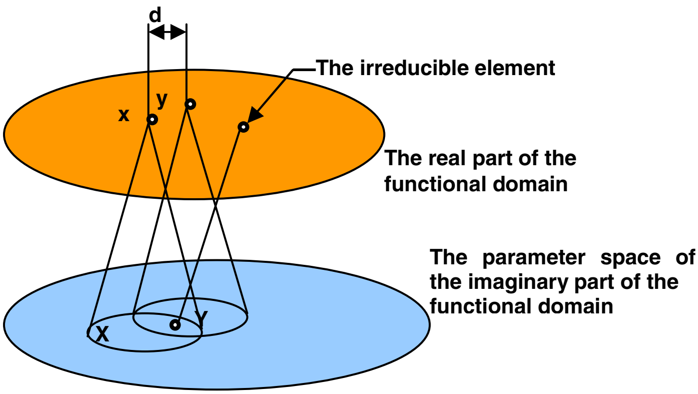


Fig. C4.7 A visualisation of the concept of distance in the real part of the functional domain.

The definition of distance between real functional elements, as illustrated by Fig. C4.7, reminds us that the parameters of a system are completely defined in terms of the parameters of its elements. It is true that a system consists of a set of elements and the interactions between them, and that as a result of these interactions the system can have properties not found in any of the elements, but the interactions have no independent existence, and they do not introduce any new parameters. The ability to interact must already be inherent in the elements that are interacting; the system structure just describes which ones are active. It is this fact that allows us to associate a unique set in the parameter space of the imaginary part of the functional domain with a given system, as indicated in Fig. C4.7. However, more than one system may have the same parameter set; they have different interactions between the same set of elements.

When we use the cumbersome identification “parameter space of the imaginary part of the functional domain” above, it is because there will often be a different set of parameters associated with real elements, as we discussed in Sec. A4.1. These parameters directly describe the behaviour of the system, without recourse to how these parameters are determined by element parameters; i.e. without explicitly considering the behaviour they describe to be emergent. For example, we would not normally think of entropy as an emergent property, nor temperature to be an emergent parameter, nor do we think of the equation PV = RT as describing the emergent behaviour of a gas. As we are only considering the functionality of plants, not their physical characteristics, such as size, weight, and colour, we shall, in the rest of this book, call the parameters of imaginary functional elements “element parameters” and those of real functional elements “system parameters”. Once we define a set of imaginary elements, the set of element parameters, and the element parameter space, are fixed; the system parameter space will depend on the particular project; we are always free to define new system parameters to characterise properties that are of significance to the project.

Imaginary functional element are elements that represent such aspects of a project as reliability, availability, maintainability, safety, environmental impact, societal impact, etc.; parameters that may be viewed as additional requirements on *how* the project fulfils its purpose or primary directive. Each one of these aspects can be defined in more or less detail. And again, a detailed description (i.e. a complex element) may be represented by a system of less complex elements. The processes of decomposition and condensation moves us between levels of detail. But contrary to these processes in the real domain, they never take us outside the aspect we are considering, no matter what level of detail we go to. A well known example of this is that of reliability block diagrams; as we descend from system through subsystem, equipment, and module, down to component level, the elements are always just reliability blocks. The elements in the imaginary part of the functional domain are segregated into disjoint subsets, one to each aspect. For any particular project, *the linking of different aspects takes place by virtue of the corresponding imaginary elements being included in real elements.*

A further feature of the structure of the functional domain arises from the central role of the irreducible element in the description of functionality and the definition of that element, as depicted in Fig. C4.4. Due to this view of an engineering project as an optimisation of the balance between cost and revenue, the secondary elements fall into two completely separate categories: the elements describing aspects of the cost, and the elements describing aspects of the performance, as already mentioned at the end of Sec. C4.3 and illustrated in Fig. C4.6.

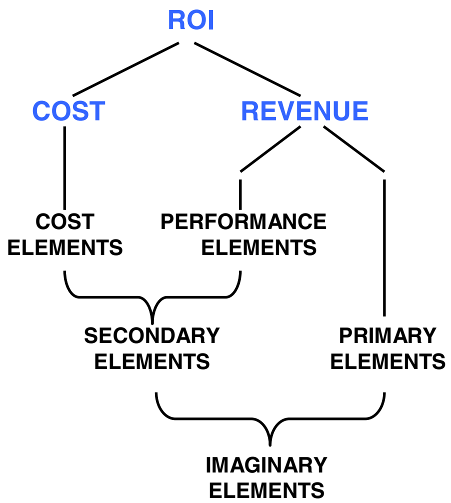


Fig. C4.8 The upper level taxonomy of imaginary functional elements.

C4.7 Structure of Real Functional Elements

Following on from the structure of the functional domain, real functional elements, as a system of imaginary elements, will also have certain structural features. To see how this comes about, consider a particular project and develop, at first, the plant's capability, i.e. what it must do, as a system of primary functional elements. The elements and structure of that system will, of course, depend both on the required capability and on our choice of functions to achieve it. However, once that system has been created, and we turn to describing the cost of providing the functions and the additional requirements on such aspects of the performance as reliability, maintainability, and safety, the corresponding elements *refer to* the primary elements. That is, the primary elements form what we might think of as a skeleton structure, and the secondary elements "dress up" this skeleton to form a real element, thereby inheriting some of the same structure.

When we say "some of", it is because the extent to which the secondary elements inherit the structure of the primary elements depends on the type of element. The cost elements will inherit the subdivision into functions, but will not reflect the interactions between the primary elements; they simply inherit the hierarchical structure or ordering. But the elements describing reliability will be related in accordance with the structure of the primary elements, as in a functional reliability block diagram.

C4.8 The Functional Parameter Space

Stakeholder requirements will not only define capabilities, such as "being able to transmit information", in terms of the parameters associated with the capabilities, such as bit rate, error rate, latency, etc. but will define restrictions on the values these parameters are allowed to have, and the same is true of the parameters associated with the various aspects of performance, such as reliability, maintainability, and safety. These restrictions result in restrictions on the values of the functional parameters of the corresponding functional element If we visualise the space spanned by all the functional parameters, the *functional parameter space*, then these restrictions define a volume within that space, and the solution is *constrained* to lie within that volume. The size and dimensionality of that volume depends on the nature of the restrictions, as the simple case of two parameters, x and y, in Fig. C4.9 demonstrates.

However, in addition to restrictions of the type illustrated in Fig. C4.9, we have the universal requirement of maximising the return on investment, **U**. That quantity is a function of the functional parameters, and at some point in state space it will take on its maximal value. In Chapter C6 we shall show that, as a result of the definition of **U**, the maximum lies within the allowed volume.

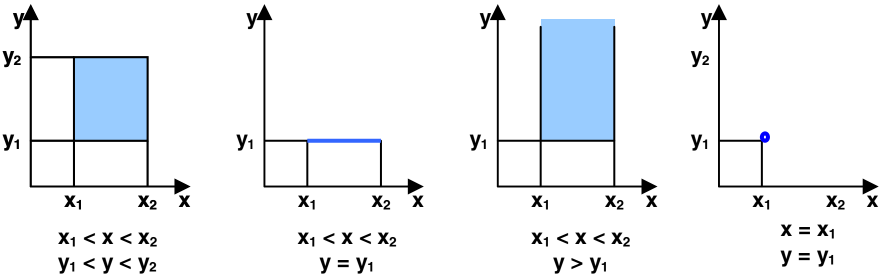


Fig. C4.9 Demonstrating how restrictions on the functional parameter values in a two-dimensional parameter space changes the size and dimensionality of the "volume" within which the solution is constrained to lie.

When the functional model is realised as an operating plant, the parameters will all take on values somewhere within the allowable volume. That is, at any one point in time, the state of the plant will be a point in the allowable volume and, due to both internal (e.g. random failures) and external (e.g. temperature, interest rate, etc.) factors, this point will change during the operational lifetime, describing a trajectory in state space. In general this trajectory cannot be predicted, but it is often possible to give a probabilistic account of it by means of a distribution function that gives the probability of the operating point being in a volume dV around the point **x**. This will be the basis, in Chapter C6, of our approach to optimising the solution in the functional domain.

References

1. Aslaksen, E.W.: Designing Complex Systems – Foundations of design in the functional domain. CRC Press (2008)
2. Levin, B.: English Verb Classes and Alternation: A Preliminary Investigation. The University of Chicago Press (1993)
3. Classification of verbs is an important component of Natural Language Processing and Cognitive Comuter Science, and the most developed classificatioon is VerbNet, developed originally at the Department of Computer and Information Sciences at the University of Pennsylvania with Karin Kipper-Schuler, now at the University of Colorado, `http://verbs.colorado.edu/~kipper/` as the main author. – Another group is located at the Computer Laboratory at University of Cambridge with Anna Korhonen, `http://www.cl.cam.ac.uk/~alk23/` as the main proponent. – Finally, the work on Controlled Natural Language at the Centre for Language Technology at Macquarie University, with Rolf Schwitter, `http://web.science.mq.edu.au/~rolfs/` as the main proponent, is also relevant to functional elements

C5 Systems in the Functional Domain

C5.1 Interactions between Functional Elements

From the development in Sec. C4.3, we understand that the real functional elements associated with a plant describe the functionality of the plant in varying detail; they are *related* in a hierarchical fashion, but there is no *interaction* between real functional elements associated with a given plant. We also saw that, in order to demonstrate that the functionality of a plant satisfies the stakeholder requirements, we need two types of imaginary functional elements; primary and secondary elements, and describe the real functional elements as systems of imaginary elements. So, as far as a given plant is concerned, interactions in the functional domain are between imaginary elements only, and this immediately raises a few questions: Are interactions between secondary elements identical to interactions between primary elements? What is the nature of interactions between primary and secondary elements? Do the primary and secondary elements form discernable subsystems? And if so, is there any benefit, in terms of better insight and understanding, in viewing them as subsystems?

Before going further, I should make a brief comment regarding the use above of the qualifier "associated with a given plant". It reflects the meaning of the term "system of interest" [1] and the fact that the system concept implies a boundary; a rule that divides the universe of elements into those that belong to the system and those that do not. If we combine two systems, A and B, to form a "system of systems", then, within the context of this new "system of interest", the real elements representing the systems A and B in the functional domain disappear, and the new system is represented by a real element describing the functionality of that system. That real element will be a particular system of some or all of the imaginary elements making up the two systems A and B. In other words, real elements are defined relative to "systems of interest", and at any one time there is only one "system of interest".

As a first step towards answering the above and related questions, we must define what we mean by "interaction" between functional elements. A functional element is defined by four sets; the set of functional parameters, the set of dependencies, the set of influences, and the set of functions between the elements of the three first sets. Any or all of these sets could be involved in an interaction, but as the main purpose of a functional element is to represent what an object does, in particular as seen from the users' point of view, and the purpose of decomposing a functional element into a set of interacting elements is to partition the users' functional requirements and thereby reduce the complexity of the design task, it would be preferable to define interactions in terms of functional parameters. A starting point would therefore be the following definition:

Definition C5.1: Two functional elements *interact* if the values of one or more functional parameters in one element depend on the values of one or more functional parameters in the other element.

Note that the definition does not introduce any direction of the interaction; functional interactions are always *between* two elements, even though the physical quantity involved in the interaction will flow from one to the other. That is, for any two elements, A and B, there is a relationship between functional parameters of A and functional parameters of B, but which ones are the independent ones and which ones the dependent ones may depend on the particular application in which the two elements find themselves. This is a reflection of the difference between a functional element and its physical realisation, and a couple of small examples may help to clarify this. Take the case where the desired service is to dispose of garbage, and the corresponding functional element is described by a single functional parameter, the weekly capacity to accept garbage. So here the functional parameter describes a physical *input*, whereas the dependencies, describing the results of the garbage processing, e.g. methane, mulch, and some landfill, are physical *outputs*. Another case is that of electricity production; the functionality may be parametrised by two parameters, say, rated output (in MW) and availability, and in this case they characterise a physical output, but one of the dependencies, the wast heat, is also a physical output.

With this in mind, we see that there are four different combinations of the two types of variables, dependencies and functional parameters, for each of two elements A and B, as shown in Fig. C5.1. But in each case, the variable(s) of one element must be identical to the variable(s) in the other element. That is, the possibility of the interaction must already be inherent in the two elements; a system is created when one or more of these possibilities are realised in a particular case by setting the values of the common variables equal to each other, and this equality relationship is symmetrical.

An understanding of the distinction between what we normally think of as the relationship between inputs and outputs to processes in the physical domain and the relationship between functional parameters and dependencies in the functional domain further requires us to realise that in the functional domain this may also be determined by the *context* in which the functional element finds itself; in particular, how it is included in a system of functional elements. As an example, consider a function (e.g. a chemical process) that converts a raw material into a product with the help of thermal energy (heat). As a stand-alone function, the functional parameter “production rate” may be determined by the demand for the product, and as such is an input, whereas the energy consumption, a dependency, is an output. But if this functional element is used in a system with the electricity producing element, with the interaction being that the waste heat from the electricity generation is absorbed by the chemical process, then the production rate becomes a system output, driven by the electricity demand, and the energy consumption becomes an input to the chemical function.

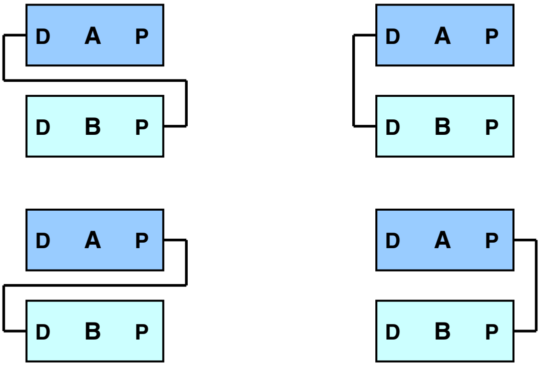


Fig. C5.1 The four possible types of interactions between two functional elements, A and B. Here P denotes a functional parameter, and D a dependency. Variables not involved in the interaction are not shown.

C5.2 Element States and System States

At the end of Sec. C3.2, we introduced the *state* of an element as the set of values of its functional parameters, and from the discussion above, we see that the definition of interaction could equally well be formulated by saying that two elements interact if the states of the two elements are related.

From Definition C5.1 it follows that the functional parameters involved in the description of the service provided by the system must be a subset of the functional parameters of the elements making up the system.. In particular, if x and y are two elements with sets of functional parameters X and Y, respectively, then the set of functional parameters, Z, belonging to the system z resulting from the interaction of x and y satisfies the relation $Z \subseteq X \cup Y$. This is an important observation, because it means that any *emergent properties*, i.e. properties of the system which were not present in the individual elements, can be completely described using the functional parameters of the elements; no new parameters are required. It may be much more *convenient* to introduce new functional parameters which directly characterize the emergent properties of the system, but these will always be related to, or expressable in terms of, the functional parameters of the elements. A little example from an important type of interaction, correlation or coherence of identical elements, is given by considering a set of identical radiators being combined to provide the service of illuminating a small, remote spot with radiation. From the users' point of view, a most useful functional parameter is beamwidth, and they might not even be aware of the relationship between beamwidth and the parameters of the individual radiators (relative position and phase angle). - Another example, from a different type of interaction - elastic collision between mass points - is the characterization of a gas in terms of pressure and temperature and, if the "service" of the gas is its ability to absorb and release heat, the functional parameter heat capacity; all of which are related (albeit in a statistical manner) to the parameters describing the individual mass points. And the ultimate example is, of course, the characterisation of *any* engineering project in terms of its ROI; a sort of Unified Theory of Engineering Projects.

We can formalise our understanding of system states by two definitions:

Definition C5.2: A *system state,* ϕ, is a particular combination of (system) parameter values. The set of all system states, $\Phi = \{\phi\}$, is called *state space.*

The relationship between system states and element states can be developed further if we introduce the concept of basic system states:

Definition C5.3: Let a system consist of *n* elements, each with a set of states E_i, with $i = 1, ..., n$. Then the set of *basic system states*, Φ_0, is defined as the Cartesian product of the sets of element states

$$\Phi_0 = E_1 \times E_2 \times ... \times E_n \,.$$

The sought-after relationship is expressed by the following theorem:

Theorem 5.1: There exists a one-to-one correspondence between the set of system states and a partitioning of Φ_0 into mutually disjoint subsets.

The proof of this theorem is given in [2], and that reference also gives the following simple example: Consider a system consisting of *n* equal elements, each contributing an amount Q to the service being produced by the system, and with each element being characterized by a single parameter which can take on only one of two values, 1 if the element is operating, and 0 if the element has failed. Consequently, there are *n* element parameters, and 2^n basic system states (i.e. Φ_0 is a discrete set with 2^n members). Assume now that the users are only interested in whether the service level equals or exceeds *m*Q or not, with $m < n$, so that there is only one system parameter, *s*, which also takes on only two values, 0 and 1. The subset of Φ_0 which corresponds to the state $s = 0$ then has a number of members, depending on *m*, which equals

$$\sum_{x=0}^{m-1} \frac{n!}{x!(n-x)!} \,,$$

and we could say that these basic states have *condensed* into the one system state, $s = 0$.

C5.3 Primary Element Behaviour

From our immediate experience with most physical engineered objects, i.e. with what we have called plants, it is easy to think of a system element as something with a fixed behaviour. That is, the relationships between "inputs" and "outputs" are time invariant, as e.g. in a regulated dc power supply, where a given change in the input voltage will always produce the same change in the output voltage, and a given change in the load will always produce the same change in both input and output variables (voltage and current). There could also be external influences, such as the ambient temperature, that would affect the input/output relationships, but we tend to consider the behaviour to be basically the same and regard such influences as perturbations on an otherwise fixed behaviour. For many primary functional elements a similar view of their behaviours, i.e. of the relationships between the functional parameters and dependencies (and influences), as something characteristic of the element, may be appropriate, but increasingly we find elements whose behaviours are not always basically the same, but that depend substantially both on how they are embedded in a system and on the environment in which the system operates.

This is a difficult subject matter, and a small example may help to get a better view of it. Consider an element whose function is to decide the best option out of a number of options presented to it. It has a built-in function that provides the decision criterion, and that function depends on one or more variables. The element has the ability to accept values for these variables, but as a stand-alone element it uses default values. If the element is part of a system, and one or more of the other elements in the system are able to provide values for these variables, the behaviour of the element (i.e. its decisions) can be very different to its stand-alone behaviour, and as a consequence, the system has a behaviour that cannot be found in any one of the elements on its own.

It would appear, then, that what we call emergent properties of a system have two quite different sources. One is the interaction between elements that, each one, behave as they do in isolation; this is e.g. the case with the antenna system consisting of identical radiating elements. The other is the activation of inputs that are inactive in a stand-alone application when the elements are embedded in a system environment, causing the individual elements to have a different behaviour and thus providing a contribution to the system behaviour that is different from what would be expected from the stand-alone behaviour. A well-known example of this is the different behaviour of people as individuals and in a crowd.

The second source results in two different cases of emergent behaviour. One is where the additional inputs were designed into an element, and their activation is a decision by the system designer to exploit their influence on the element behaviour. The second case is where the inputs are unintentional, and their activation in the system environment leads to unintended consequences. An example of this is noise pulses entering into embedded control systems and causing unintended behaviour.

C5.4 Systems of Primary Elements

At this point, we should recall our description of stakeholder requirements in Sec. B4.2. There we identified a subset of the stakeholder requirements, the *functional requirements*, as those that define the service; that is, what the stakeholders want to receive (and for which they are willing to pay and/or invest) and the conditions under which it is to be provided, without any reference to any physical entity, or plant, that provides it. A further subset, the *performance requirements*, placed requirements on the performance of the plant, such as its reliability, safety, and levels of emissions.

Through the activities of architecting and functional analysis, described in the previous chapter, we identify the functions that the plant must have in order to meet the functional requirements, and we express this in the form of a system of functional elements. These are the primary elements. However, we now need to realise that the functions we, as engineers, determine will be required to provide the service will generally include a number of functions that are not *directly* involved in providing the service, and in conformance with normal practice we subdivide the system of primary functional elements into three subsystems,

- the operations system;
- the maintenance system; and
- the support system.

The reasons for this subdivision are that the skills and knowledge of the people working in these three areas of a plant are different, and it is important that already at this early stage of a project this is acknowledged and that the best available knowledge is brought to bear on the design.

The operations system has been, and will continue to be, the main focus of our design in the functional domain, because it is the system whose functional parameters must include the parameters defining the service (and which we generally call "service parameters" without differentiating between what they belong to); it is this system that generates the revenue, and it is therefore this system we optimise initially, as we shall see in the next chapter. As such, it is the system to which the other two must relate, and the structure of this system forms what we already referred to, in Sec. C4.6, as the skeleton of the functionality.

As the primary system is developed through the top-down process, the structure of the system will change as the functionality is described in more and more detail by means of an increasing number of more detailed functional elements, until the level of detail is adequate to cover all the service requirements. At that point, the structure will, of course, depend on the particular project and its service requirements, but it will also depend on the engineer's system design solution; there will generally be more than one possible choice on interacting functions that will meet the requirements (although, in theory, only one is optimal).

C5.5 The Secondary Elements

The secondary elements describe aspects of the functionality that are additional to the actions described by the primary elements, and they arise out of two very different groups of requirements. Some secondary elements are required in direct response to corresponding requirements in the stakeholder requirements; a typical example is a stakeholder requirement for dependability of the service, which then needs to be reflected in a requirement on the reliability of the system of primary elements. Other secondary elements are required as inputs to the primary elements, and again, reliability (or availability) might be required as input to a revenue element in order to be able to optimise the ROI, even if there is no explicit stakeholder requirement for reliability. Another example of this is cost; there may not be any direct requirement on the cost, but a requirement for optimising the ROI will demand that cost is included in the description of the functionality.

Each aspect described by secondary elements will form an increasing set of elements as the aspect is described in increasing detail, and the elements are generally associated with (or identified by) the primary elements, although this is not necessarily a one-to-one association. However, secondary elements do not interact and form systems in the conventional sense; there is nothing passing from one element to another. When we show a reliability block diagram of a function that consists of three interacting sub-functions as three reliability elements connected in series, there is nothing passing from one element to the other, and the elements cannot influence one another; the diagram only defines how the reliability parameters (e.g. MTBF or failure rate) are combined to give the value of the same parameter for the function.

What we have here is an illustration of the many possible meanings of the word "system", as we discussed it in Part A; a "system of secondary elements" sits below the meaning as we have used it so far, but above the meaning in an ordering, as in "the periodic system" or "the Dewey decimal system", if we rank the meanings by the extent of the relationships between the elements. In addition to an ordering or classification, the secondary functional systems contain a *rule* of how to combine elements in order to go from one level of detail up to the next, or, with reference to Sec. C4.2, a rule of how to *condense* a description. It is this rule that is described in the diagrams of various aspects, and this then leads to such characterisations as "the structure of the system". Again, as this use of the word "system" is so entrenched, but also so well understood by its context, its use should not cause any problems.

References

1. ISO/IEC 15288:2008(E), p.9
2. Aslaksen, E.W.: Designing Complex Systems – Foundations of design in the functional domain, p. 92. CRC Press (2008)

C6 Value and Optimisation

C6.1 Overview

The purpose of applying the system concept to engineering is to better handle the complexity in engineering projects. In this last chapter we return to what we have identified as one of the core issues in that regard: The transition from the stakeholder requirements to a physical solution that will meet those requirements in a manner that is efficient while still providing a reasonable assurance of having found the optimal solution. Most of what we need has already been developed in earlier chapters; we now draw that together and develop it into a coherent and detailed, but still generally applicable, methodology.

The point of departure of the methodology is, as we introduced already in Chapter B4, to split the transition; first a transition from the stakeholder requirements into the functional domain, and then a transition from the functional domain into the physical domain. And of these two, it is the first one that is relatively novel and needs to be documented; the second transition is already quite extensively developed in the current state of systems engineering, as we saw in Chapter C2. The system concept is then applied in the functional domain. By describing the functionality of the plant as a system of functional elements, we expect to be able to handle the complexity in a more efficient and effective manner.

At this point we should recall that by describing something as a system we do not change its inherent complexity, as caused e.g. by a large number of stakeholders with different requirements, by many different technologies, a dynamic environment, etc. What we do is to make it easier for us, as humans, to understand it and apply our well-proven engineering processes and techniques to it. The system approach is transforming the description of an object into a format that is suited to our cognitive abilities.

But why go via the functional domain? Why not apply the system concept directly to the physical domain? The reason is, as we have discussed, that design in the physical domain is essentially a bottom-up process of synthesizing a system from familiar components; a process that becomes rapidly less efficient as the number of requirements on the outcome increases. Another way of looking at this same issue is provided by Fig, C4.1; the number of plants corresponding to a given set of stakeholder requirements is very large, and by going via the functional domain, the number can be reduced significantly. However, this still leaves open the question whether the reduction in design effort in the physical domain is greater than the additional design effort in the functional domain. The key to answering that question in the affirmative lies in using a top-down design methodology with reusable functional elements [1].

C6.2 The Top-Down Design Process

Let us start the development of our functional design methodology by considering the end product; i.e. what we want to end up with before making the transition into the physical domain. We would like a description of the functionality of a plant that has the following characteristics:

a. It is a system of functional elements.
b. The functionality of each element is clearly defined: Easy to understand, using a language common to all involved in the design process, and not open to varying interpretations.
c. As many of the primary elements as possible have at least one known physical realisation (ideally as a COTS item).
d. Most, and preferably all, of the secondary elements represent characteristics and system parameters that are industry standard or widely accepted.
e. The development of the description provides an assurance that the result is reasonably close to the optimum, as defined by a criterion accepted by all stakeholders.

Such a description is a *model* of the plant, consisting of real and imaginary functional elements, but it is a very special model. On the one hand, it represents the outcome of *design* in the functional domain. It is not a model of the stakeholder requirements, but a model of the functionality of a possible solution to the problem of meeting those requirements. The architecture of the primary functional system (which we called the “skeleton” in Chapter C4) is identical to the functional view of the plant architecture. On the other hand, it provides a measure of the degree to which the plant will satisfy the stakeholder requirements *within the context of the project*, thereby allowing the design to be optimised.

The process of design in the functional domain, i.e. the development of such a model, is a *top-down* process. By that we understand a step-wise analysis and partitioning, starting from a description of the purpose of the plant as a single element. And we have provided an argument for how it is possible to define, at the highest level of abstraction, a purpose that is common to all engineering projects – the maximisation of the Return on Investment (ROI). The corresponding functional element is the irreducible element, to which we shall return later in this chapter. The next step in the process is to find two elements that describe the two dependencies of the irreducible element: Revenue and Cost. We now have to ask: Will these elements be completely different for every project, or will they have features that are common to all projects (or, at least to the overwhelming majority of projects)? This is the question we have to ask at every step of the process, because only by identifying features common to classes of projects and creating corresponding reusable elements will we be able to reduce the amount of work involved in creating functional models and make design in the functional domain a cost-effective methodology. Let us start with the Revenue element.

C6.3 The Revenue Element

C6.3.1 The Value Function

Our view is that the Revenue arises as a result of the plant providing a *service* to the users, for which they provide the Revenue according to the *value* they place on the service. Service and value will play a central role in what follows, so we need to have a very clear understanding of what is meant by these two measures. In Sec. B2.4 we introduced the service by describing the process of engineering as the creation of a plant that, during its operational lifetime, provides a service that meets a need, and we noted that the service can be almost anything: providing a commodity or a product, a financial service, a professional service, transport, security, etc. The service is generally documented by the engineer and characterised by parameters chosen by the engineer, and just as the nature of the service can vary widely, so can the parameters. However, because the starting point of a project is a set of stakeholder requirements, and it is an intrinsic feature of the design process that the parameters characterising the service (at least) cover the corresponding stakeholder requirements (this is the requirement for traceability within the design process), the service parameters will also be adequate for the users to characterise the service they are receiving.

Let the set of service parameters be denoted by **s**: $\{s_i\}$, and the *value function* by W(**s**); it has the dimension of monetary units per unit time. The revenue in an accounting period, T, is then given by

$$R(T) = \int_T W(\mathbf{s}(t))dt, \qquad \text{(C6.1)}$$

Of course, revenue is restricted to that part of the life cycle in which the plant is operating.

The top-down process could now proceed by developing (or selecting) functional elements that have the $s_i(t)$ as functional parameters, then describing these as systems of smaller elements, and so on, until the elements are small (simple) enough so that the transition into the physical domain is relatively easy. However, the time-dependencies of the s_i are most often not known at this early stage of the design, and in any case many of the parameters will be of a stochastic nature. So in the next subsection we shall examine some of the issues involved in transforming the integration over time into an averaging over probability distributions.

C6.3.2 Fluctuations, Averages, and Non-linearity

The service provided by any modestly complex plant is not constant; it fluctuates due to variations in the numerous components involved, such as random failures of hardware, bugs in software, varying human performance, and random external influences. Examples we are all familiar with are the great variation in download speed over the Internet and the variations in travel time on motorways. If we look at the spectrum of these time-dependent variations, we find that they are often separated into two distinct regions; low frequency components (long-term variations) and higher frequency fluctuations, with the separation between the two regions being such that the typical periods of the fluctuations are much shorter than a normal accounting period (month, quarter, or year), whereas the characteristic times of the long-term variations are of the order of or greater than the accounting period. For service parameters that display such short term, random fluctuations, a common approach is then, in the first instance, to create new (or derived) parameters as averages over the short-term fluctuations. These parameters are well defined (i.e. not random) functions of time, and so is the service, defined in terms of them.

This approach is well known to us from a different part of the engineering knowledge base: thermodynamics. For example, the pressure, P, of a volume, V, of gas is the average of the forces exerted by the individual molecules on the boundary of the volume, and the macroscopic behaviour of the gas is described by the simple equation PV = RT. This relationship between the three parameters P, V, and T holds as they change in time as part of some process, but the change must be slow compared to the timescale on which the microscopic collision processes take place, so that at any point in time, these microscopic processes experience essentially a static or *equilibrium* environment.

The assumption of equilibrium as the justification for working with averages is a powerful and useful assumption in many cases; however, there are also many cases in which the fluctuations are the main drivers of the performance aspect we are interested in. We shall return to this in more detail below; at this point a couple of well known examples will illustrate this and point us towards the underlying issue. The first is the survival of a species. Over any "short" period in time, the characteristics of the species are constant, and the increase or decline of the species is determined by its environment, such as the availability of food and competition from and attack by other species. But over a "long" period of time, fluctuations in the characteristics, i.e. mutations, may allow the species to adapt to and survive in a changing environment through the processes of selection and propagation, and so the fluctuations are the driving force of evolution. Neglecting them and considering only the average characteristics would make evolution inexplicable.

The second example is a system of buses providing a transport service. An important service parameter is the arrival time of buses in relation to the scheduled time at any one location. If the buses are often late and often early, then, on the average they may be just on schedule, but that is clearly not what is of interest; it is exactly the extent of the fluctuations that is a measure of the service.

In both of these cases the underlying issue is that of non-linearity. In the second case, the service is not a linear function of the arrival time, but a function of the absolute value of the difference between the arrival time and the scheduled time, which is a non-linear function of the arrival time. In the first case, the dynamics of the species (increase or decline) is not a linear function of the fluctuation in a characteristic; the function involves feedback from the environment through the selection and propagation processes, and is highly non-linear. To put this slightly more precisely, consider the case where the value function, W(*s*), is a function of a single stochastic variable, *s*, only, and *s* is characterised by the probability distribution φ(*s*). The average value of s, S, is given by

$$S = \int_s s\varphi(s)ds \ ,$$

and the value of the service, using this average parameter, is W(S). But we could also determine the value by calculating the average of W(*s*) directly,

$$\overline{W}(s) = \int_s W(s)\varphi(s)ds \ ,$$

and the two expressions are equal, $W(S) = \overline{W}(s)$, if and only if W(*s*) is a linear function of s.

The situation is that, while linearisation in a limited range around an operating point, as we already mentioned in Sec. C1.5, can be a valid approach to a study of stability or sensitivity, the operating point itself will most often be determined by a non-linearity. If the fluctuations have amplitudes that exceed the range in which linearisation is a good approximation, they will play a part in determining the operating point. The non-linearity does not necessarily have to be related to the value function; it can be through a saturation at either the input or the output of the service production process (i.e. the supply-and-demand relationship), a decrease in the production efficiency at high or low volumes and so on, but there are many cases where non-linearities in the value function play a significant or even the major role in determining the operating point. This arises because the dependence of the value on any single parameter is often in the form of an S-shaped function, with the service having practically no value if the parameter is below a certain value, and the value not increasing significantly once the parameter reaches its nominal value, as indicated in Fig. C6.1.

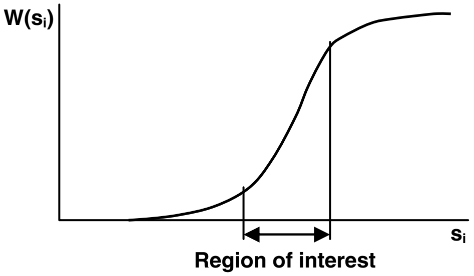


Fig. C6.1 Typical dependence of the value function on a service parameter.

Now, if we are neglecting fluctuations in *s*, and depending on how the cost increases with increasing value of *s*, the nominal operating point (i.e. of a fully intact system) will often lie at the upper end of the region of interest indicated in Fig. C6.1, at the beginning of the region of "diminishing returns", and it is clearly determined by the non-linear nature of the curve. But if we want to, for example, study the effect of failures on the performance (i.e. only values of *s* less than the nominal operating point), it may be perfectly acceptable to linearise the value function in the region of interest. However, if *s* is defined by a probability distribution, then using the average of *s* to determine the operating point will give a different result to using the average of W, as illustrated (very roughly) in Fig. C6.2, where the shift in the value (which will result in a shift in the optimal operating point) is shown as ΔW.

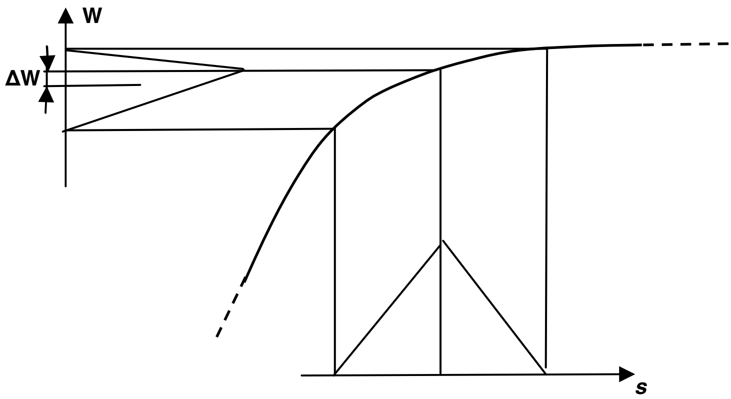


Fig. C6.2 The triangular distribution of the fluctuations in *s* results in a distribution of W with an average value shifted by an amount ΔW from what would be obtained by using the average value of *s*.

To be able to handle the effect of fluctuations in the service parameters due to non-linear value functions, while at the same time retaining much of the simplicity of the linear case, the following model of the value function can be useful.

C6.3.3 A Simple Value Model

Here we consider a general case, where the service has several features, each described by a parameter, s_i. The (yet to be designed) plant providing the service will consist of many different parts, each of which is composed of a number of components that have a variable performance. As a result, the service parameters are stochastic variables, each with a probability distribution $\varphi_i(s_i)$. The value of the service, W, is therefore also a stochastic variable, as illustrated in Fig. C6.3.

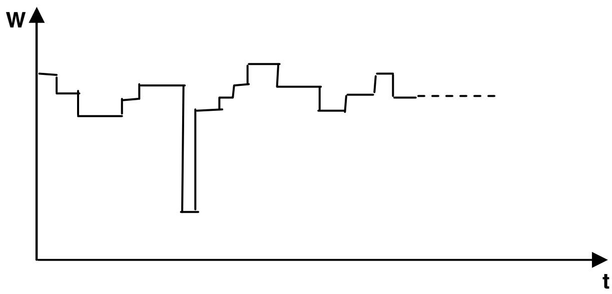


Fig. C6.3 A typical record of the value of the service provided by a complex plant.

Let the value function be a function of n parameters, $W(s_i)$, $i = 1,\ldots,n$, and assign to each parameter a function $w_i(s_i)$,

$$w_i - \begin{cases} 0, & s_i < a_i \\ c_i\,(s_i - a_i)/(b_i - a_i), & a_i \le s_i \le b_i \\ 1, & s_i > b_i \end{cases} \qquad \text{(C6.2)}$$

This type of function is illustrated in Fig. C6.4.

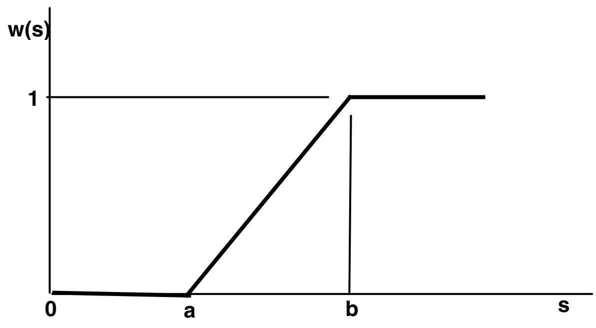


Fig. C6.4 The type of a value function component, w(s), associated with a service parameter, s.

The proposed simple value function is then defined by

$$W(\mathbf{s}) = \prod_k c_k w_k(s_k) \left[1 + \sum_l c_l w_l(s_l) \right], \tag{C6.3}$$

where the c_i are the *weights* of the components, and with k = 1, …, m, and l = m+1, …, n. In this expression, the parameters with index value less than or equal to m are what we might think of as describing the essential requirements, in the sense that if the value of any of these parameters falls below its associated a-value, and w becomes zero, the service, as a whole, is of no value. The parameters with index value greater than m describe features that add to the value of the service, so that if one of these features is missing, the service, as a whole, still has a value.

The form of the w-functions assumes that the value of a feature of the service always increases with increasing value of the associated parameter. If this is not the case, as e.g. in the case of failure rates, it can always be achieved by redefining the parameter (e.g. taking the inverse value).

Now assign to each parameter a probability distribution, $\varphi_i(s_i)$, and its associated cumulative probability function, $q_i(s_i)$, as shown in Fig. C6.5 (without indecies).

We note that the mean of this triangular distribution, μ, is given by

$$\boldsymbol{\mu} = o + \frac{1}{3(v-u)}\left[(v-o)^2 - (o-u)^2\right];$$

and the variance, σ^2, is given by

$$\boldsymbol{\sigma}^2 = \frac{(v-u)^2}{72} + \frac{\left(o - \frac{v+u}{2}\right)^2}{6}.$$

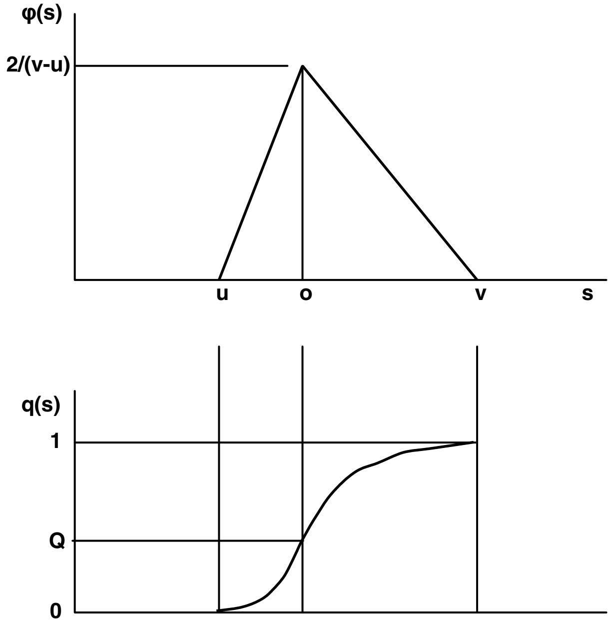


Fig. C6.5 The probability distribution $\varphi_i(s_i)$ and its associated cumulative probability function, $q_i(s_i)$ (indices have been omitted).

The expectation value of a component of the value function is then given by the integral

$$\overline{w_i} = \int_0^\infty w_i(s;a,b)\boldsymbol{\varphi}_i(s;u,v,o)ds\,, \tag{C6.4}$$

and the value in an accounting period, W, is then given by Eq. C6.3, but with $\overline{w_i}$ instead of $w_i(s_i)$. As this value model already averages over an accounting period, the Revenue is simply equal to the value, **R** = W. If the probability distributions $\varphi_i(s_i)$ remain unchanged throughout the life of the project, the revenue is the same in every accounting period z, **R**(z) = W.

The symbolic representation of this simple Revenue element is as shown in Fig. C6.6.

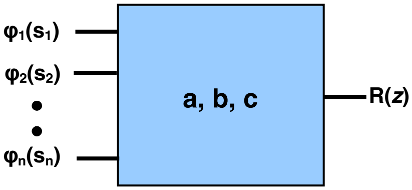


Fig. C6.6 Symbolic representation of the simple Revenue element, with the three vectors **a, b,** and **c** characterising the value the users place on the service.

We generally assume that the user evaluation of the service, represented by the three vectors **a**, **b**, and **c** in the present case, remains unchanged throughout the life of the plant, However, if that is not the case, our models can handle this also.

C6.3.4 Evaluation Methods

The integral in Eq. C6.4 can be evaluated in closed form as one of the following eight cases (in addition to the two trivial cases; v < a, when the integral is zero, and u > b, when the integral is 1):

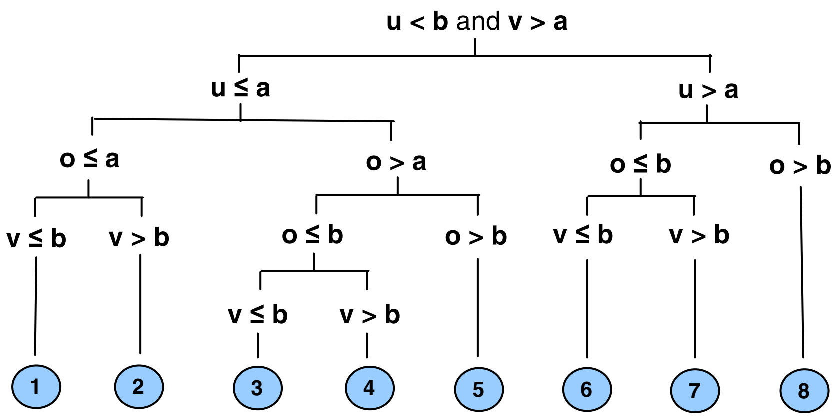


Fig. C6.7 The eight cases for the evaluation of the expectation value of a component of the value function.

This evaluation of the function $\overline{w}(u.v.o;a,b)$ can be carried out by a small Visual Basic routine, downloadable from www.gumbooya.com. However, there are cases where this closed form calculation of the expectation value is not possible or appropriate. One case is where the value functions are more complex that the simple one shown in Fig. C6.3; as a result a closed calculation may not be possible. Another case is where one wants to find not only the expectation value, but the actual distribution of W-values, or at least the variance. In both of these cases, a Monte Carlo calculation may be the only approach, and the calculation process is indicated in Fig. C6.8. In order to determine the s-values, we note that the inflection point, Q, in Fig. C6.5 is given by Q = (o-u)/(v-u), and the relationship between s and q(s) is given by

$$s = \sqrt{\frac{r}{D}} + u \text{ ; for } r < \text{Q, and with D = 1/[(v-u)(o-v)];}$$

$$s = v - \sqrt{\frac{1-r}{E}} \text{ , for } r \geq \text{Q, and with E = 1/[(v-u)(v-o)],}$$

where, as usual, r is a random number between 0 and 1.

C6.3.5 Two Illustrative Examples

To get a better feel for the applicability and limitations of this simple value model, let us look at two (simplified) cases from an important class of systems for which the service is, at least to a first order, parameterised by a single parameter, the output (or throughput), and where the fluctuations in s are due to numerous stochastic processes within the system. Within that class is the subclass of systems that provide mainly a transport service, with only minor modifications to the product being transported. Prominent within this class are the ore transport systems in the mining industry, from which these two cases are taken.

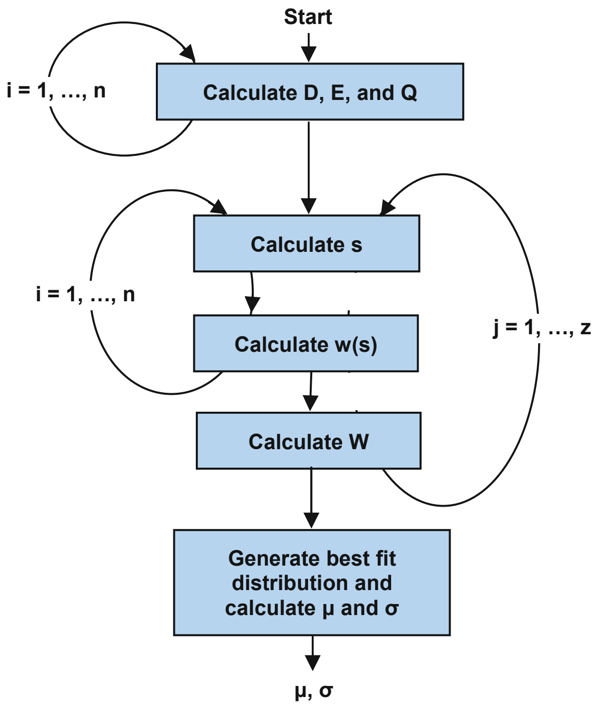


Fig. C6.8 Outline of the Monte Carlo process for determining the distribution of W and for calculating its mean and variance. Here n is the number of service parameters, and z is the chosen number of data points (i.e. the sample size).

In both cases, the user or consumer of the ore, which is typically a concentrator, has a fixed upper limit to its throughput, q, so that there is no further value in the transport system providing ore above this limit. But if the ore supply falls below this limit, the value of the supply falls off rapidly, due to the high capital and fixed operating costs of the concentrator, so that the value function in Fig. C6.3 is a reasonable approximation, and if we denote the upper limit by q_0, i.e. $b = q_0$, a typical value for a might be $0.6q_0$.

In the first case, the supply of ore provided by the transport system is determined by a number of different elements, each with a performance that varies from shift to shift, so that the daily throughput is a stochastic variable that can be characterised by a symmetric triangular probability distribution (i.e. $q = o = (u+v)/2$ in Fig. C6.5). It is then convenient to introduce the variable $d = (u-v)/2$, so that the performance of the transport system is characterised by only two parameters, q and d. Consequently, so is the value W(q,d), and this function, normalised to its value W(q_0,0), is shown in Fig. C6.9.

The curves in Fig. C6.9 reflect the relative sensitivity of the value to a change in the nominal capacity, q, *vs* a change in the variability, d, and, as an example, the changes required to go from an initial operating point (1.00, 0.12) with a value of 0.950 to a value of 0.975 are shown. Increasing the value of q essentially means designing and constructing a larger transport system; decreasing the value of d is largely a matter of improving the operation and maintenance, so once the relative costs of these changes are known, one can determine the optimal operating point.

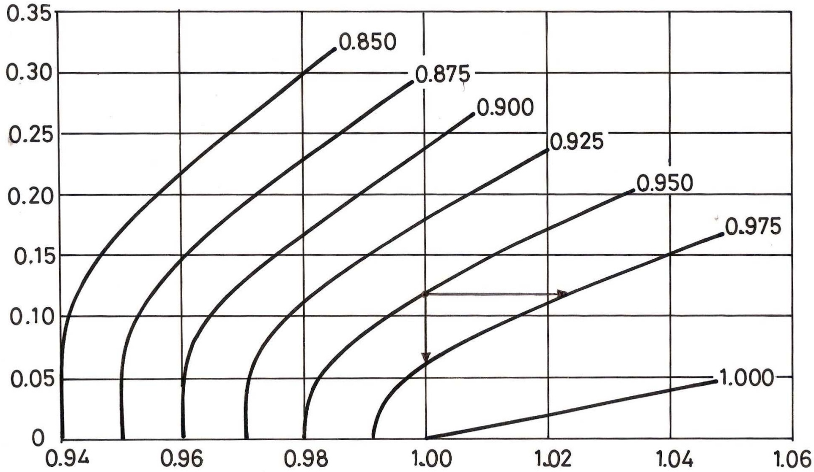


Fig. C6.9 The normalised expectation of the value, W, of the ore transport system as a function of the nominal system capacity, q, on the horizontal axis and the variability, d, on the vertical axis.

In the second case, the nature of the transport system is such that it is either operating at its nominal capacity or not operating at all. So our simple model, with its triangular performance probability distributions, is not directly applicable. However, another simple approach to modelling the stochastic behaviour of a complex system that was presented in [2] can be used. The approach taken there was to characterise the service provided by the system by a single parameter, s, the Quality of Service (QoS), and to describe the stochastic behaviour of the service by means of a *service density function*, φ(s).

In the present case the service parameter *s* takes on two values only, either 0 or 1, and the service density function, φ(*s*), is characterised by a single parameter, the unavailability, α, as follows:

$$\varphi(s) = \delta(s)\alpha + \delta(s-1)(1-\alpha) \ .$$

The unavailability is normally defined in terms of a Mean Time Between Failures (MTBF) and a Mean Time To Repair (MTTR) as

$$\alpha = \frac{MTBF}{MTBF + MTTR} \ .$$

Assuming that the fluctuations of s are rapid as compared with the accounting period, the expectation value of W in an accounting period is equal to the average of *s*, which in this case is (1-α). So, with the value function form Fig. C6.3 (but with *b* = 1). the value is now a function of the parameter α, and given by

$$W(\alpha;a) = \begin{cases} W_0\left(1-\dfrac{\alpha}{1-a}\right); \ \alpha < (1-a) \\ 0; \ \alpha \geq (1-a) \end{cases}$$

where W_0 is the value for an intact system (i.e. no unavailability).

With a transport system of this nature, the consumer will usually have a certain storage capacity (e.g. a stockpile), so that only failures with a duration exceeding Δ will result in s = 0; and to relate the value to system parameters, we need not only the failure rate, but the distribution of failure durations, f(τ), in order to determine the average duration of failures exceeding Δ, κ. Here we can again use a triangular distribution as a first approximation, as shown in Fig. C6.10.

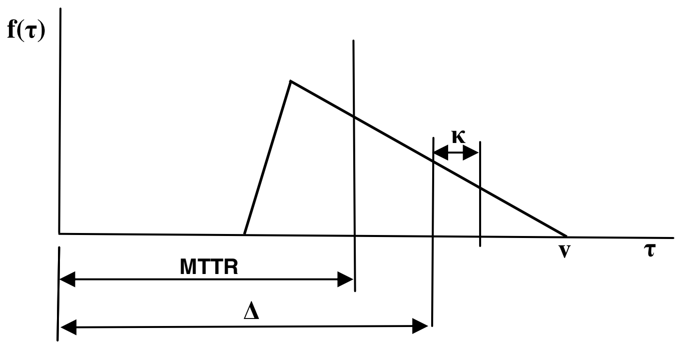


Fig. C6.10 For an ore transport system, with failure duration distribution f(τ), the average value by which the duration of failures exceed Δ is shown as κ.

With this,

$$\boldsymbol{\kappa} = \int_{\Delta}^{\infty} (\boldsymbol{\tau} - \Delta) \cdot f(\boldsymbol{\tau}) d\boldsymbol{\tau} = (v - \Delta)/3 \text{; and}$$

$$\boldsymbol{\varepsilon} = \frac{(v-\Delta)^2}{(v-w)(v-u)} \ ,$$

and as the mean time between failures is now (MTBF+MTTR)/ε, we have

$$\boldsymbol{\alpha} = \frac{\boldsymbol{\kappa} \cdot \boldsymbol{\varepsilon}}{MTBF + MTTR} \ .$$

This simple value model can now be used to optimise the design by trading off between the various parameters. For example, if failure distribution is given, e.g. by u=3 h, v=21 h, and w=6 h, and, as before, a = 0.6, then the value becomes a function of the two variables Δ and MTBF, as shown in Fig. C6.11.

Again, the curves in Fig. C6.11 reflect the relative sensitivity of the value to a change in the storage time, Δ, *vs* a change in the MTBF. As an example, the changes required to go from an initial operating point (5, 500) with a value of 0.950 to a value of 0.975 are shown. Increasing the value of the MTBF essentially means designing and constructing a transport system with more redundancy and/or higher cost elements (but can also involve improving the maintenance strategy); increasing the value of Δ means increasing the size of the stockpile, so once the relative costs of these changes are known, one can determine the optimal operating point.

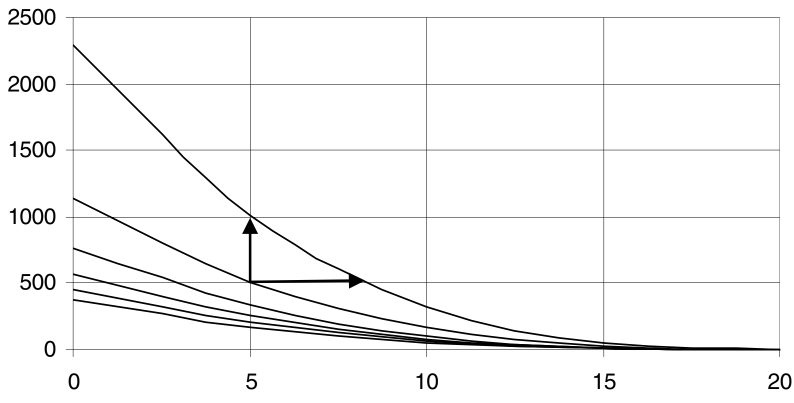


Fig. C6.11 Curves of constant value of the function W(Δ, MTBF), normalised to W_0, with W = 0.975, 0.950, 0.925, 0.900, and 0.875, from top to bottom curve. The horizontal axis shows the value of the storage time, Δ, in hours, and on the vertical axis is the MTBF in hours. The arrows illustrate the changes required to go from W = 0.950 to W = 0.975, starting from an operating point of Δ = 5 h and MTBF = 500 h.

C6.3.6 A Related Model [3]

As a further example of a high-level revenue model, the following model applies to a particular issue that arises in connection with a significant class of complex systems, generally falling under the heading of infrastructure systems. They are part of what might be thought of as the enablers of our society, and include power supply, water supply (including for irrigation), waste water handling, flood mitigation, and transport (both public and private). Each of these systems represents a very significant investment in fixed assets, and they are all subject to increasing demands for their services, with regard to both quantity and quality. Consequently, there is great incentive to operate the assets in such a manner that their intrinsic capabilities are utilised to the fullest, and this leads to the demand for sophisticated control systems.

Due to both the difference in technology and implementation and the difference in durability, there is most often a clear separation and interface between what might be called the *Production System*, consisting of the elements directly involved in providing the service, and the *Control System*. The Production System will often have a durability requirement of 100 years, whereas the Control System will typically have a durability requirement of 20 years, so that the Control System is replaced and updated several times during the life time of the Production System. As a result, projects to *change* (i.e. upgrade or replace) control systems in order to improve overall system performance take the point of view that the Control System must be designed to get the best possible performance out of an *existing* Production System. In accordance with our general approach, "best" means optimising the Return on Investment of any such change, as shown in Fig. C6.12.

Because the Revenue function is now restricted to a particular class of projects, the upgrading of the control systems within infrastructure systems, it can be more specific and detailed. The following should be seen as an example of how such a Revenue element is developed.

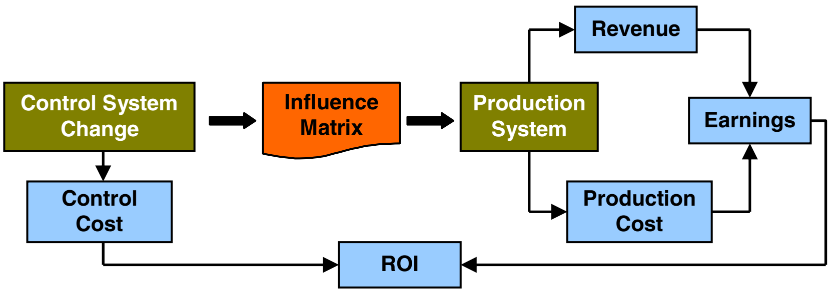


Fig. C6.12 A high-level view of the entities involved in determining the Return on Investment (ROI) of a change to the Control System. The Influence Matrix relates changes in the parameters that determine the cost and revenue of the Production System to changes of the features of the Control System (i.e. new or improved control functions).

A Production System providing a service will generally have a maximal (or rated) capacity, K_0, and a planned capacity, K_1, with the difference between K_0 and K_1 being due to planned shutdown for preventive maintenance. However, the actual capacity, k, is less than K_1, due to random failures of individual production elements, so that k is a stochastic variable.

The subscribers (or customers) will have a varying demand, w, which is also a stochastic variable, and the flow of product from the Production System to the subscribers is accompanied by a flow of payments from the subscribers to the Production System; this is the revenue generated by the Production System. To make the situation as simple as possible, let us assume that the accounting (billing) period, T, is long enough to justify employing statistics within a single period. (That is, the fluctuations in the stochastic variables are rapid compared to T.)

At the end of each period, the subscribers pay for the service. It is quite common for this revenue, **R**, to be made up of two components

$$\mathbf{R} = q_0 D_0 + q_1 D_1 T, \qquad \text{(C6.5)}$$

where D_0 is a measure of the maximum demand in the period T, and D_1 is the average actual demand. It is a characteristic feature of infrastructure systems that the revenue is determined by the quantity of the product provided. The "quality", to the extent that one can even define this, is largely inherent in the design of the Production System or in the type of product (such as electric power).

Because both k and w are stochastic variables, it may happen that the Production System is not able to meet the demand. This can be due an accumulation of system failures, an unusually high demand, or a combination of both. In that case, the amount of service actually delivered during a period T will be less than the demand by an amount, **Z**, which we shall call the *overlap function*, for reasons that will become clear shortly. That is, if we define a function, $q(t) = w(t) - k(t)$ if $w(t) > k(t)$, $= 0$ otherwise, the overlap function is defined by

$$Z = \frac{1}{T}\int_0^T q(t)dt$$

There may, in some cases, also be a penalty, P, levied for not meeting the demand (as long as it is less then D_0). The penalty may be proportional to the value of **Z** or to some other measure related to the failure to provide the required service.

The amount of the service supplied at any point in time cannot exceed the demand or the capacity, whichever is the lower figure. The transition from demand to capacity as the limiting factor takes place through some form of control mechanism. In an electricity grid, it is called load shedding, in a freeway network it is called ramp metering, in a health care system it is represented by waiting lists, and so on. We therefore need to agree that by demand, w, we understand the demand that wants to be satisfied. It would be satisfied if the capacity were sufficient, not the actual, capacity-restrained amount of service supplied, which we might call the *supply*. Also, we shall need to take into consideration the fact that the transition from demand to capacity as the limiting factor varies from one infrastructure system to the other. For some, such as the electricity grid, the service would collapse if no action was taken to limit the performance to the capacity, whereas for others, such as a highway network, the performance becomes demand dependent and decreases with increasing demand, as illustrated in Fig. C6.13.

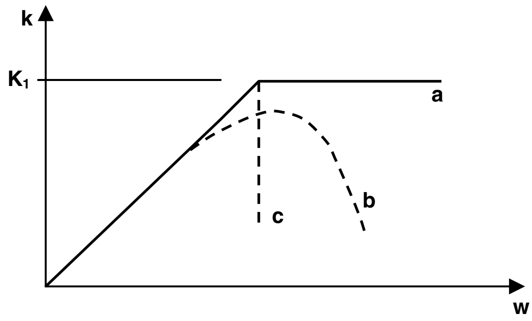


Fig. C6.13 Two different system behaviours as the demand, *w*, approaches and exceeds the capacity limit, K_1 (assuming a fully intact system). Curve a illustrates perfect load shedding, curve b illustrates a choking behaviour, and curve c illustrates total collapse, as in a black-out.

We would now like to be able to relate the revenue, **Q**, (including P if there is a penalty), to factors that can be influenced by the Control System. To that end we will develop a simple revenue model. This model considers two different aspects of the performance of the Production System. One is the effect of random failures and their repair on the capacity of the Production System, the other is the effect of saturation in part or all of the system, resulting in the type of supply reduction illustrated by the curve b in Fig. C6.12.

Let the capacity, k, be characterised by a probability distribution $\varphi(k)$, such that the probability of the system having a capacity between k and $k+\mathrm{d}k$ equals $\varphi(k)\mathrm{d}k$. If failures occur at a rate denoted by λ, and the restoration of planned capacity occurs at a rate denoted by ρ, then the proportion of time the system has its planned capacity is given by

$$\chi = \frac{\rho}{\rho + \lambda}$$

and $\varphi(k)$ is therefore of the form $\varphi^*(k) + \chi \cdot \delta(k - K_1)$. The nature of the stochastic function $k(t)$, as equipment fails and is brought back into service again, is illustrated in Fig. C6.14. It indicates that large capacity reductions are less probable than small ones.

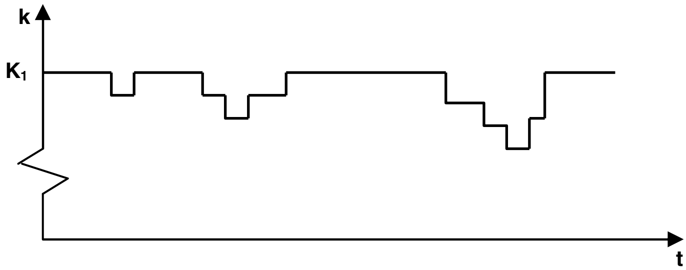


Fig. C6.14 Simplified form of the stochastic function $k(t)$.

If we now further denote the average reduction in capacity when the system is in a failed state by Δ, then, for our purposes, the function φ*(k) can be approximated by a triangular function,

$$\phi^*(k) = (1-\chi)\frac{2}{3\Delta}\,(k - K_1 + 3\Delta) \text{ for } K_1 - 3\Delta \le k \le K_1; \text{ 0 otherwise}$$

as shown in Fig. C6.15.

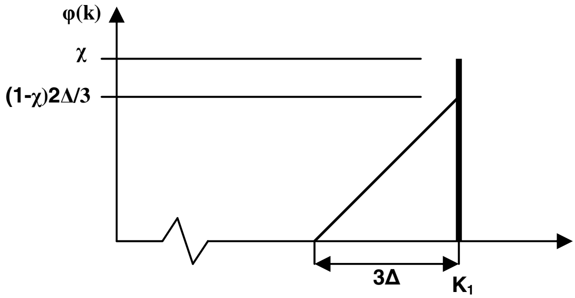


Fig. C6.15 The approximation of the probability distribution φ(k) by a delta function (the thick line at k = K_1) and a triangular function, such that the average severity of a failure (decrease in capacity) is equal to Δ.

Let the demand, w, be characterised by a probability distribution, ψ(w), with $0 \le w \le W_0$, where W_0 is the total installed load. The average (over T) demand is given by

$$W_1 = \int_0^{W_0} w\,\psi(w)\,dw$$

The simplest form of this distribution is a rectangular one,

$$\psi(w) = \frac{1}{2(W_0 - W_1)} \text{ for } 2W_1 - W_0 \le w \le W_0\text{; 0 otherwise}.$$

The overlap function, **Z**, is now a function of the five parameters K_1, χ, Δ, W_0, W_1; for its evaluation the reader is referred to the original publication [3].

In the first instance, only χ and Δ can be influenced by the Control System. The two demand parameters, W_0 and W_1, are characteristics of the users of the service, i.e. the market. The value of the capacity limit, K_1, is determined by the design of the Production System (i.e. choice of components) and the maintenance policy. The parameter χ is determined by the two system parameters λ and ρ. As they can be individually influenced by the Control System, we have three Production System parameters that can be influenced by the Control System.

However, indirectly the Control System can also influence the value of K_1, in that features of the Control System (e.g. condition monitoring) may allow a new maintenance policy, such as reliability-centred maintenance, and thereby reduce the difference between K_0 and K_1. To take this into account, we introduce the *preventive maintenance factor*, γ, with $K_1 = \gamma K_0$.

As mentioned earlier, in relation to Fig. C6.13, some infrastructure systems display a saturation behaviour. That is, if at any one point in time the Production System has a capacity *k*, the amount of product provided, i.e. the supply, will not reach this level as the demand increases, and may even decline when the demand increases further. This is illustrated again in Fig. C6.16, where we have indicated that we shall simplify the effect of saturation by considering it to result in a reduction of the capacity by a factor κ, the *saturation factor*.

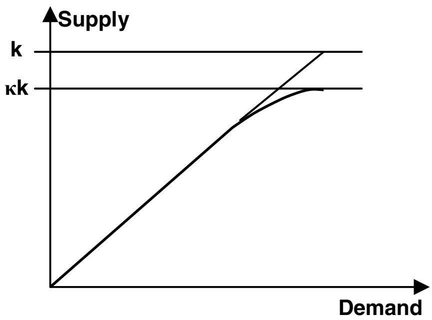


Fig. C6.16 The effect of saturation, reducing the capacity, k, by a factor denoted by κ.

Without any further details of how the saturation occurs, there is no reason to introduce any coupling between the failure/repair behaviour and saturation, and so we will introduce the reduction factor, κ, into the development in the previous section by reducing K_1 accordingly.

The model situation is now the following: A particular Infrastructure System is connected to a set of subscribers with a demand, w, characterised by the probability distribution $\psi(w)$, and is generating a revenue **R**, as defined by Eq. C6.5. A change to the Control System is being contemplated, which will influence the Production System through changing the values of the five Influence Factors λ, ρ, Δ, κ, and γ. What is the expected corresponding change in the revenue?

In our model, the change in revenue results from a change in the value of the overlap function, **Z**. The average amount of the service produced is given by

$$D_1 = W_1 - \mathbf{Z}\,.$$

The maximum demand, D_0 in Eq. C6.5, can either be the actual highest value recorded in a period T or the maximum of some short-term average, or even a more complex function of the demand. But for our purposes it is sufficient to set

$$D_0 = q_2 W_0\,,$$

and, if required, reflect any complexity of the off-take contract, including any take-or-pay clause, in the coefficient q_0.

So far, the revenue function is straight forward; a measure of complexity is only added when we now consider the issue of a penalty, P, for not meeting the demand. This penalty may take a number of different forms, ranging from a cash refund to the subscribers to dismissal of the infrastructure management, but in any case one needs to assign a monetary value to it, and the simplest assumption is to make it proportional to the value of **Z**,

$$P = q_3 \mathbf{Z}\,.$$

The net revenue, as a function of the subscriber base, W_1, then reaches a maximum for a particular value of W_1, W^*, as illustrated in Fig. C6.17.

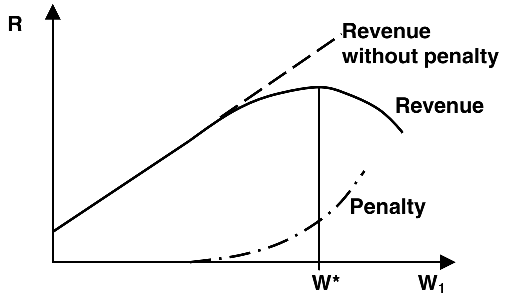


Fig. C6.17 Net revenue, revenue minus penalty, as a function of average demand, W_1.

This behaviour is characteristic of any infrastructure system; it is uneconomical to provide a system that can satisfy any fluctuation in demand. Or, conversely, satisfying the extreme fluctuations while providing a reasonable return on investment will result in a very high charge for the average demand, a relation not always appreciated by the general public.

In order to determine the real change in revenue as a result of a change in the Control System, we need to determine the optimal operating point, W*. To incorporate this feature into our model in a straight forward manner, we need to make the further assumption that the ratio $W_0/W_1 = \alpha$ remains constant as W_1 is varied, i.e. the statistics of the demand, $\psi(w)$, is a characteristic of the subscriber population under consideration. With that, we have

$$\mathbf{R} = q_0 q_2 \alpha W_1 + q_1(W_1 - Z) - q_3 Z\,,$$

and setting $\partial Q/\partial W_1 = 0$ results in the condition

$$\frac{\partial Z}{\partial W_1} = \frac{q_0 q_2 \boldsymbol{\alpha}}{q_1 + q_3}\,.$$

The symbolic representation of this Revenue model is shown in Fig. C6.18.

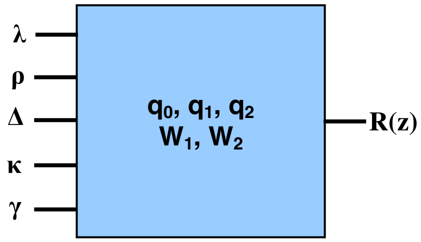


Fig. C6.18 Symbolic representation of the Revenue element for the Control System influence model.

C6.3.7 Discussion

At this point, you might ask: Why are we spending so much time on documenting these revenue models? Do they have any special significance? The answer is that their significance here is only as illustrations; there would be an almost infinite number of possible, different models. These particular models were included for the sole purpose of demonstrating four important features of our methodology:

1. The process of developing a Revenue element is relatively straight forward: Define the service to be provided; identify the parameter(s) describing the main features of this service, including its/their value(s); allocate it/them to feature(s) of the plant; and express this as an executable model of the revenue generation.

2. By choosing the level of abstraction (information hiding) appropriately, an element can be made to apply to an industry sector or a class of projects.
3. The accuracy of the models need only to reflect the high level of the Revenue element in the top-down process, but even so, the models will usually allow us to make some in-principle decisions about the way forward in the design process.
4. The third model (in C6.3.6) demonstrated the flexibility of the methodology. In this case, the plant under consideration is the control system, but in itself this system does not generate any revenue; the revenue is only generated through the influence the control system has on the production system. The revenue element describes the revenue-generation of the production system, but the dependencies of the element (i.e. the inputs to the model) are not functions of the production system, but of the control system (via a dependency matrix).

The underlying reason for dwelling on this issue of the Revenue element is, of course, that our application of the system concept to engineering is based on a top-down approach; a step-wise development of the system (as a description), starting with the *purpose* of the plant. The irreducible element is the top element in this development process; at the next level down, the Revenue element starts to reflect the specific purpose of a class of plant. The description of that purpose and the revenue generated by it sets the direction for all of the following development of the functionality of the plant; i.e. for the process of design in the functional domain.

C6.4 The Cost Element

C6.4.1 The Nature of Cost and Cost Elements

Cost accounting is a well-developed discipline, with numerous methods and standards. And normal life cycle costing, as it is employed in traditional systems engineering, is described in various publications [4]. The life cycle aspect is reflected in our approach, as both the Cost and the Revenue are measured per accounting period over the life of the project. But as noted at the end of Sec. C4.3, when working in the functional domain, cost is seen as a consequence of requiring the plant to have a particular functionality. Consequently, the primary requirement on a cost accounting appropriate to the functional domain is that the cost elements correspond to primary elements. The additional properties of the primary elements, such as reliability and durability, as may be expressed by secondary elements, influence these cost elements.

Definition C6.1: In the functional domain, a *cost element* is a secondary element representing the total cost of providing a function, as defined by the associated primary element.

The allocation of cost to functions can be seen as a version of standard cost accounting or, perhaps even more accurately, as a version of activity based costing [5]. That is, all costs throughout the life of the project are attributed to functions; there are no additional overhead costs. Investigation, design, construction, operating, maintenance, decommissioning, and management costs all have to be allocated to primary elements. Therefore, the primary structure of the development of the cost into a set of elements is determined by the primary elements.

In principle, if a function is described in more detail as a number of interacting sub-functions, the cost element related to the function would be described by a corresponding number of cost sub-elements related to the sub-functions. But here the difference between primary and secondary elements, as described in Sec. C5.5, becomes apparent; the cost sub-elements do not *interact* to form the cost element, their cost values are simply *summed* in the cost element. Consequently, if the value of a cost element at a certain level of development of the functional description is already known, there is no benefit in developing the cost aspect of this function in any further detail. This is the reason why there is not necessarily a one-to-one relationship between primary elements and cost elements.

In addition to the structure induced by the subdivision of functions, we can also, in accordance with the three main classes of primary elements defined in Sec. C5.4, distinguish three main classes of cost elements:

- operations element costs;
- maintenance element costs; and
- support element costs.

The word "element" is included in these class names, as it would otherwise be easy to confuse these functional domain costs with the operating, maintenance, and support costs of the plant (once it exists). The benefit of introducing this additional structure into the cost database is the same as for the primary elements; the people involved in providing and maintaining the cost data have different skills and experience and often belong to different organisational units. This subdivision assists in obtaining the best match between knowledge and the data in the database.

C6.4.2 The Cost Element

The Cost element is the functional element that provides the Cost per accounting period, C(z), as an input to the irreducible element, but due to the difference between primary and secondary elements, it is very different in format to the Revenue element. It simply provides a summing function, and perhaps the simplest format of the Cost element is as an Excel spreadsheet. As a simple illustration, consider a case where the description of the functionality has been developed to a maximum of three levels of detail below the Revenue element. The corresponding cost structure is shown in Fig. C6.19. Only the elements with the lighter shading are individual elements for which the cost is required; the elements with the darker shading are just summing functions, and are included in the Cost element. The corresponding realisation of the Cost element as a spreadsheet is shown in Fig. C6.20, and the graphical representation of this Cost element is shown in Fig. C6.21.

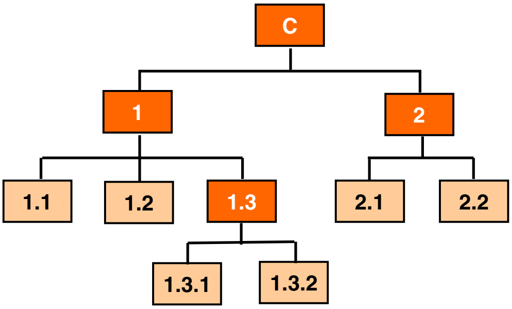


Fig. C6.19 An illustrative cost structure, reflecting the breakdown of the functionality into primary elements.

Levels below C				Accounting periods							
	1	2	3	1	2	3	4	5	6	7	8
C				75	88	122	35	35	35	32	29
	1			65	66	70	22	22	22	19	16
		1.1		25	30	30	8	8	8	5	2
		1.2		20	25	25	9	7	7	7	7
		1.3		20	11	15	5	7	7	7	7
			1.3.1	8	5	7	3	4	4	4	4
			1.3.2	12	6	8	2	3	3	3	3
	2			10	22	52	13	13	13	13	13
		2.1		5	10	22	5	5	5	5	5
		2.2		5	12	30	8	8	8	8	8

Fig. C6.20 The implementation of the Cost element in Fig. C6.16 as a spreadsheet. Only the unshaded rows of the matrix are input values, the shaded rows are summing functions within the spreadsheet.

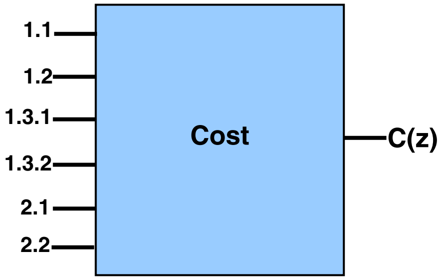


Fig. C6.21 Graphical representation of the Cost element defined in Figs. C6.19/20.

C6.4.3 Cost Estimating and the Cost Database

How can a functional element, that has no physical substance, have a cost attributed to it? The allocation of a cost to a function is an estimate, and as all estimates it is based on past experience. That experience will be partly in an industry-specific database or in a database maintained by the company carrying out the design; it could also be in a database belonging to a supplier. There is in principle no difference between documenting the cost of a function and documenting the cost of a physical object; it is just a matter of identifying and parametrising the information in the appropriate format. For example, the cost of providing a variable message service on a motorway, parametrised e.g. by the size (max. number of characters displayed) and density of signs and given in dollars per kilometre, would be quite well known.

An issue that arises in the connection with the costs of functions is that they are sometimes not independent. That is, the cost of providing a certain function may depend on whether another function is already present or not. A typical example is the function of scheduling the replacement of a component subject to wear and with a well-established mean time to failure. The function involves the acquisition of data regarding the operating time of the equipment and then some processing of this data to bring up an item on the maintenance schedule. However, another function, the allocation of cost to the product provided by the equipment, requires access to the same data. In this case, we would be better off to describe the acquisition of the data as a separate function, and then have two functions for the processing of the data, one for each of the two different applications.

This little example also illustrates the principle that there must be a *reason* for subdividing functions into smaller sub-functions; either because that is how they are realised (when we make the transition into the physical domain), or because that is how our cost database is structured, or because it improves the accuracy of the cost estimate (by differentiating between possible options for realising a function).

C6.5 The Irreducible Element

A consequence of letting a cost element represent all the various types of costs allocated to a primary element is that the cost will be a function of the accounting period, because the various types of cost, such as design cost and fabrication cost, will occur at different times in the life cycle. In particular, there will be costs occurring prior to there being any revenue, which is why every engineering project involves an investment. The manner in which this investment is made, how it is paid back, and how this process is handled by the financial accounting system can be highly complex and can vary from project to project. However, we need to keep in mind that the purpose of our descriptions (or models) is to act as decision criteria during design in the functional domain, i.e. at the very front end of projects. So what we need is a simple, transparent, and consistent way of accounting for the financing costs. One approach is to include them within the irreducible element, as part of the definition of the ROI, rather than including them within the total cost of the individual cost elements.

The simple ROI model is based on the view of the project illustrated in Fig. C6.22, where the curve is the cumulation, Q(z), of the cost, C(z), minus the revenue, R(z), in each accounting period z over the life of the project, shown as z3 accounting periods. The model makes the following assumptions:

a. The amount of equity, Q1, equals 20 % of the total amount of capital required, Q2.
b. The equity partners will forego any return on their investment until the start of operations, but will then withdraw the return as a fixed percentage of Q1 each accounting period until the end of operations. This percentage is the ROI, which we had identified as **U**.
c. The equity, Q1, is paid back to the investors at the end of operations, z3, but after first deducting the decommissioning costs, which are assumed to be 10 % of Q2. At this point, the project capital is reduced to zero.
d. Interest on outstanding debt is charged to the project at the rate of *p* per cent per accounting period, but repayment is by means of a fixed sum per accounting period, starting at the start of operations, z2.

Once the two sets of monetary values, C(z) and R(z), have been generated for a particular functional design, the value of **U** can be determined by a small Visual Basic module. Any change to the design, e.g. when evaluating a number of options, will result in a corresponding change in **U**, thus allowing the functional design to be optimised.

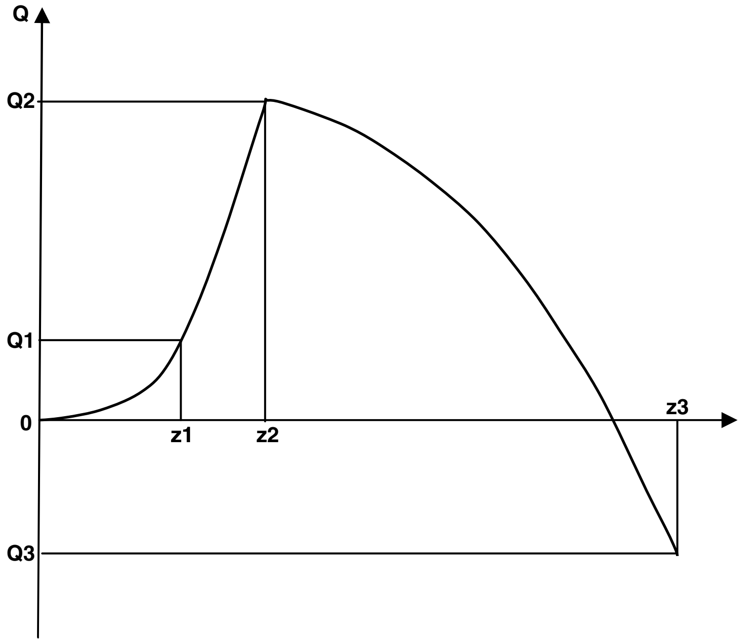


Fig. C6.22 A generalised view of a project's financial liability, Q(z), as the cumulative difference between Cost and Revenue over its lifetime, measured in accounting periods, z. Here Q1 is equal to the equity, Q2 is the total amount of capital required up until the start of operations, and Q3 are the funds available at the end of operations (prior to decommissioning).

References

1. Aslaksen, E.W.: A Leadership Role for NCOSE. In: Proc. 4th Int'l Symposium, National Council of Systems Engineering, San Jose (August 1994)
2. Aslaksen, E.W.: Designing Complex Systems. CRC Press (2009)
3. Aslaksen, E.W.: The detailed model is contained in the paper Infrastructure Control Modelling Language (ICML). Presented at the INCOSE Internat'l Symposium on Systems Engineering, Rome, July 9-12 (2012)
4. Life Cycle Costing (LCC) is detailed in numerous publications; a good introduction is, http://www.treasury.nsw.gov.au/__data/assets/pdf_file/0005/5099/life_cycle_costings.pdf
5. Kaplan, R.S., Bruns, W.: There seems to be no textbook dedicated to activity based costing. In: The first description might have been in Accounting and Management: A Field Study Perspective. Harvard Business School Press (1987); It is also discussed in Management challenges of the 21st Century, by P.F. Drucker. Harper Business Publications, New York (1999)

Index

P

Q

R

T

U

V

W