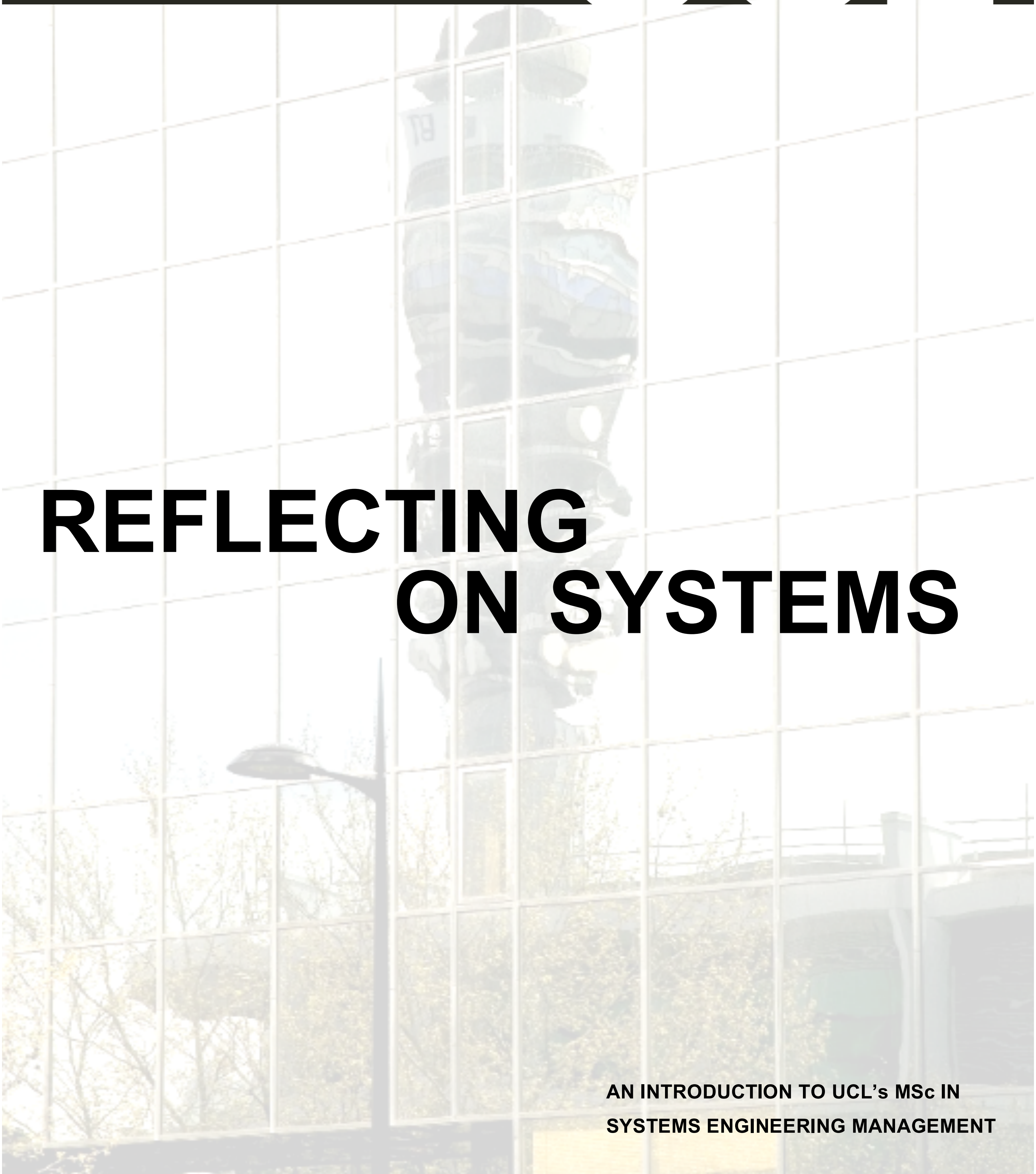
UCL CENTRE FOR SYSTEMS ENGINEERING

UCL



REFLECTING ON SYSTEMS

AN INTRODUCTION TO UCL's MSc IN SYSTEMS ENGINEERING MANAGEMENT

Version details

Blue:Users:UltraMatt:Teach:SE&CAIS:SEM MSc:Intro Paper:T.doc
S: 2, 20/09/2005 20:57:00, 831 kB, 22 p, 4640 w, Matt Whyndham.
P: 10/07/2006 13:57:00, Matt Whyndham.

REFLECTING ON SYSTEMS

A PRIMER FOR UCL's MSc IN SYSTEMS ENGINEERING MANAGEMENT

Matthew Whyndham and Michael Emes

UCL

August 2005

Overview

This is a preparatory text for the UCL MSc in Systems Engineering Management. It is an introduction to the subject that aims to present enough basic knowledge so that those taking the course can proceed quickly and profitably toward the detailed objectives of the first taught modules.

It is assumed that the reader is familiar with the broad aims of the course and the main features of the tuition and examination methods.

This document is mainly intended for students enrolled on the MSc who are shortly to begin formal study. Their managers, supervisors and tutors will also find it useful as a reference.

Objectives

On reading this document, the beginning student will :

- Be introduced to generic concepts in Systems Engineering.
- Appreciate the relationship of the subject to related disciplines, including Project Management, Speciality Engineering and Management in general.
- Be prepared to study the topic in detail.
- Understand the relationship of the Core and Optional modules in UCL's MSc course.

What is Systems Engineering?

"Life was simple before World War II. After that, we had systems."

– Admiral Grace Hopper[1]

Systems Engineering (SE) is not a new discipline; the term has been in use since World War II. But though there have been many definitions of the term over the years (not all of which are consistent), there is little consensus on the scope of Systems Engineering. This is particularly true in relation to other overlapping disciplines such as System Dynamics, Operations Research, Industrial Engineering, Project Management, Soft Systems Methodology, Specialist Engineering and Control Theory, which share many of the origins and techniques of Systems Engineering.

SE has an "international professional society for systems engineers whose mission is to foster the definition, understanding, and practice of world class systems engineering in industry, academia, and government" – namely the International Council on Systems Engineering [INCOSE, 2005]. Known as INCOSE, the organisation defines SE as follows:

> "Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: Operations, Performance, Test, Manufacturing, Cost & Schedule, Training & Support, Disposal. Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs"
>
> [INCOSE, 2004]

[1] Grace Hopper (1906-1992), Rear Admiral in the U.S. Navy, and an early computer scientist. Developer of the first compiler for a computer programming language. Source, Wikiquote: http://en.wikiquote.org/wiki/Grace_Hopper (viewed 1 August 2005)

It is interesting to note that the INCOSE definition includes no description of what is meant by a system and has no reference to engineering; it also makes no assumption that SE is relevant only to machines or technical systems.

In contrast, the Oxford English Dictionary [1989] defines Systems Engineering as:

> "the investigation of complex, man-made systems in relation to the apparatus that is or might be involved in them; so systems engineer".

This definition is more restrictive than the one used by INCOSE, limiting attention to man-made systems, and underlining the importance of

> "the apparatus that … might be involved in them"

– suggesting a focus on physical machines rather than systems in a general sense, "apparatus" being defined as

> "…equipments, material, mechanism, machinery; material appendages or arrangements…".

In fact, a distinct branch of science called 'soft systems methodology' was developed specifically to investigate those 'systems' problems that involved humans rather than or as well as machines – for more detail see Checkland's *Systems Thinking, Systems Practice* [Checkland, 1999]. Today, whether 'soft systems' are considered alongside 'hard systems' within the domain of SE is still debated (see, for example, Emes, Smith and Cowper, 2005).

Overlapping disciplines

Whilst SE emerged during World War II, other similar disciplines were established with similar goals and methods, applying mathematical and scientific rules to real-world problems. These include in particular Operational or Operations Research, which concerns itself with the optimal allocation of resources and Systems Analysis, concerned with applying economics and mathematics to non-engineering problems (although systems analysis now seems increasingly to refer to Information Systems only). Figure 1 shows how some of the most closely related fields to Systems Engineering share a significant heritage.

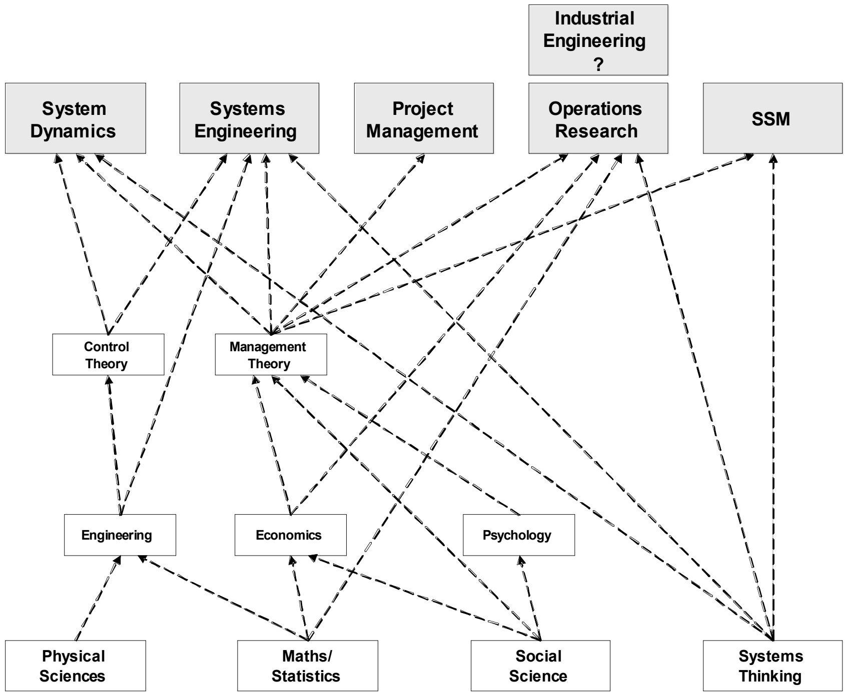


Figure 1: Roots of systems-related disciplines

In addition, SE overlaps significantly with several newer areas, namely Project Management, System Dynamics and Soft Systems Methodology. For a comprehensive history of the development of SE, Operations Research and Project Management, see Johnson [Johnson, 1997].

Even before WWII, the systems idea was gaining momentum, although it wasn't referred to in the same terms as used today. F. W. Taylor, the pioneer of industrial efficiency and specialization of work, noted that "in the past, the man has been first; in the future the system must be first" [Taylor, 1911]. Industrial Engineering was effectively born with the thinking of Taylor as well as Frank and Liliam Gilbreth [Martin-Vega, 2001]. It developed with Ford's assembly lines, Elton Mayo's Hawthorne experiments and later with the motivation theories of Herzberg, Maslow and McGregor into what we would consider modern scientific management [Brown, 1954]. Following WWII, Industrial Engineering, Operations Research and (to a lesser extent) SE began to converge as they attempted to answer similar questions of optimization; now, where one field ends and another begins is particularly cloudy, as the importance of applying 'systems thinking' in a broad range of disciplines is being recognised.

Note that soft systems methodology challenges Taylor's 'specialisation of work' - one of the earliest instances of a 'systems approach'. Whereas Taylor and his advocates attempted to find ways of increasing system efficiency by mechanising work, by trying to make man an integral part of a well-designed 'machine', now we find man's inclusion in a system blamed for a breakdown of a mechanistic approach to 'hard' SE.

Figure 2 presents a landscape of disciplines related to Systems Engineering today. Note that there is no intended scale – the difference in shapes and sizes of the loops are merely to allow possible overlaps to be demonstrated. Since many of the fields presented have rather loosely defined scopes, the relationships shown are very subjective and open to debate. It is presented here to illustrate the diversity of views rather than as an item of core knowledge. Note that in the diagram there is a part of the 'Systems Engineering' scope that is independent of other surrounding fields: this represents the competencies unique to SE.

It is unclear how the different disciplines may evolve in the future; perhaps some will merge, others will disappear altogether. What is clear, though, is that the essence of SE – the application of systems thinking, an appreciation of a system's lifecycle, and the issues to do with managing the SE process in an organisation – will remain critical for the successful management of engineering projects.

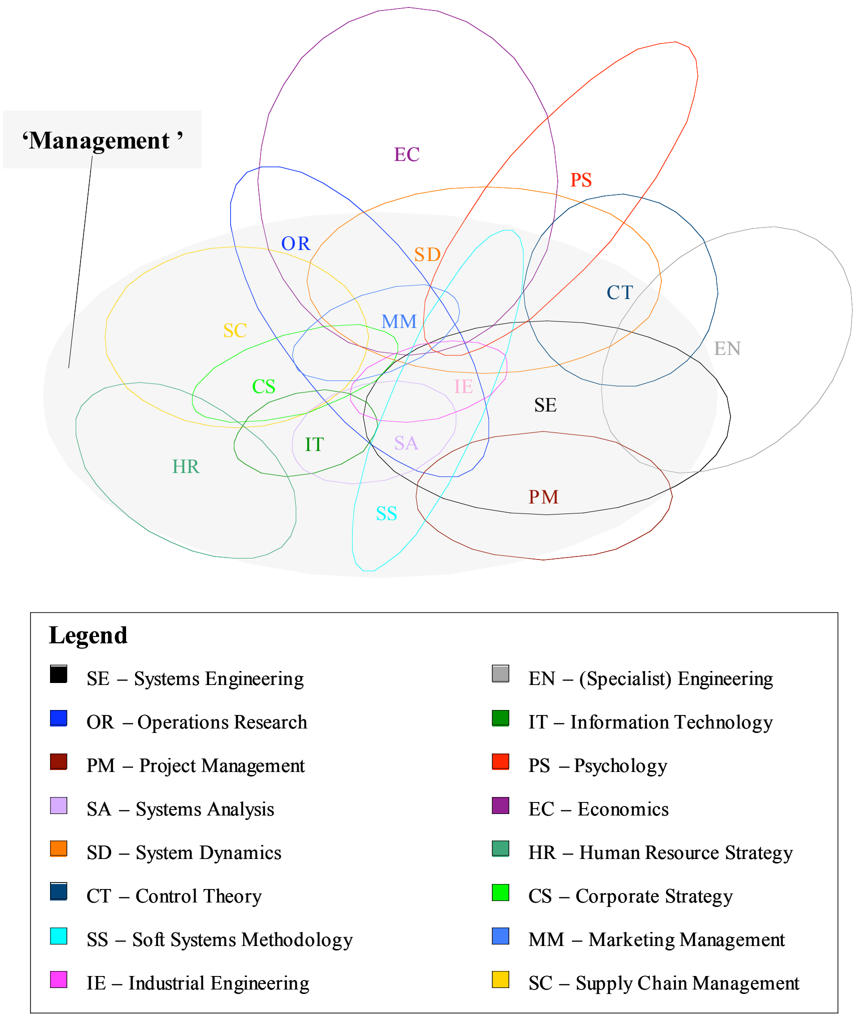


Figure 2: Landscape of disciplines relating to Systems Engineering

Core competencies of Systems Engineering

In order to bring clarity to the debate on the scope of SE, INCOSE's UK Advisory Board (UKAB) has identified three categories of core competencies for Systems Engineering:

Systems Thinking

The underpinning systems concepts and the system/super-system skills including the business and technological environment.

Holistic Lifecycle View

The skills associated with the systems lifecycle from need identification, requirements through to operation and ultimately disposal.

Systems Engineering Management

The skills of choosing the appropriate lifecycle and the planning, monitoring and control of the systems engineering process. Some of the key concepts of SE are outlined in the next section. We introduce a key component, the V-diagram (starting on page 8). This is an extension of the standard project initiation process shown in Figure 3. Whereas the management of all aspects project according to Time Cost and Quality constraints will be under the control of a project manager, the responsibility for integrating the Technology Features of the product will be devolved to the Systems Engineer.

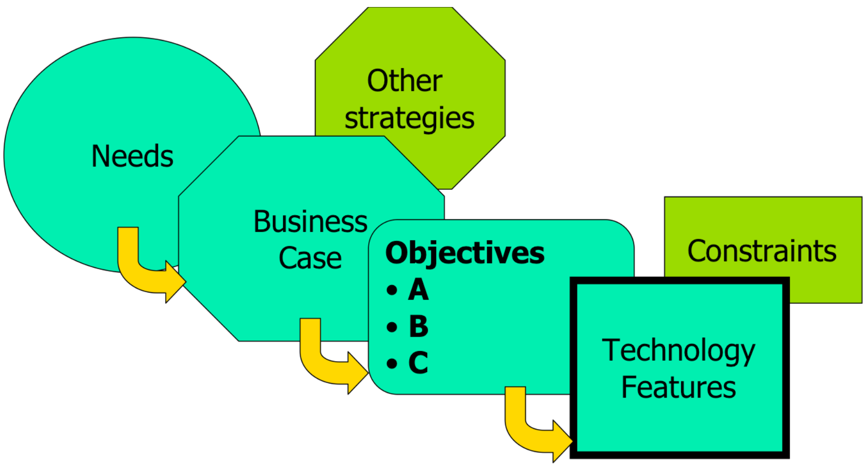


Figure 3. Needs, Technology and Constraints combine to delineate the possible Requirements (or Objectives) for a project

Working with organisations

So far we have focussed on the hierarchical organisation of the system product. Now we consider some wider aspects of management, in particular the relationship with the discipline of Project Management.

It should be remembered that Businesses are themselves systems. These are often organised in a hierarchical manner, and require to be managed carefully in order to deliver the right products to the customer. Most frequently this will be done by considering the work as a Project.

Projects are formal methods that address the matter of designing and developing the systems and products already considered. A project is an assemblage of people and equipment, normally managed by a Project Manager (PM), working toward satisfying the set of goals set forth by a customer. The success of the system is dependant on the skills of the people on a project and how well they are able to work together. It is often the PM that will have the responsibility of driving this complex situation forward, but the role of the Systems Engineer in handling the technological aspect of the work is also vital.

Systems Thinking

A systems engineer will consider the holistic nature of the problem and of the system being developed. Although the product may rely on specialist technology or skills, these will not drive the solution in isolation.

Emergent properties

A system of complex parts organised in a complex way will often exhibit *emergent properties*, and these may well be the prime focus of the system engineering efforts. An emergent property is something interesting about the system as a whole that is not a characteristic of any of its parts. Very often the collective properties of a system are driven by the way in which the sub-systems interact, i.e. the architectural design, rather than by their individual performance characteristics. Systems Engineering is very concerned with emergent properties, in promoting useful ones and suppressing harmful ones.

Global optimisation

With a focus on emergent properties, it is possible to optimize a whole system, rather than just individual parts (called “system elements” in general). Formal methods for evaluating such tradeoffs exist. Design budgets, monitoring and testing are focussed strongly toward whole-system and emergent properties rather than those of individual system elements. There are also techniques for evaluating complex product quality, and various types of risk as well the usual project management tools surrounding schedule and cost.

Interfaces

The characteristics of systems that users are interested in are those to do with how the system works, i.e. the functional characteristics. Users in general are not interested in the internal details and interfaces. Engineers, whose job it is to create such systems, *are* interested in such details, and will do so by attending to the functional characteristics of each sub-system. Whilst specialist engineers tend to concentrate on the performance of individual sub-systems in isolation, systems engineers focus development work on the interfaces between sub-systems.

Process to guide problem solving

A rational process to guide an organisation (or team of organisations) through a long series of decisions is needed. There is no single process that will satisfy all contexts, but successful systems engineering processes have several themes in common.

Most project managers will be familiar with the pair of curves in Figure 4. One aim of the systems engineering approach is to promote earlier action and decision-making, thereby increasing the effectiveness of the system. This is done specifically in relation to the activities on the left side of the V model – introduced below – and is often referred to as *left-shift*.

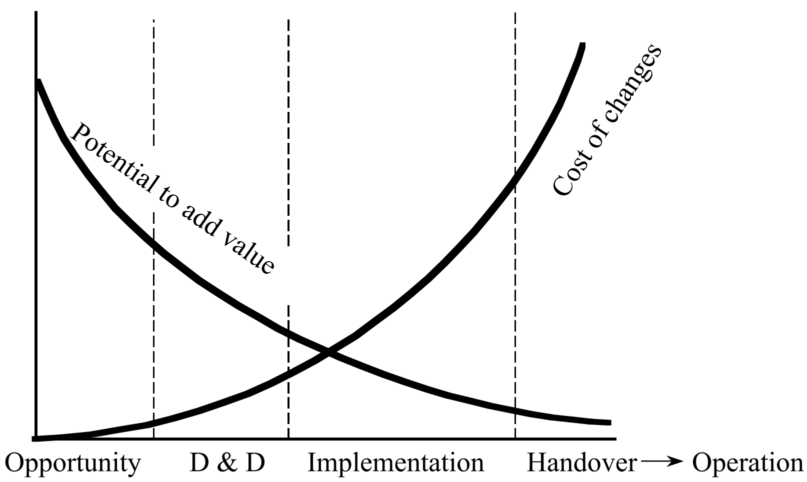


Figure 4. Costs of changes and impact of change over the life of a project

When the design focus is drawn to activities further downstream in the lifecycle, such as manufacturing, test or maintenance, so that we may have design-for-test as an additional factor in the design process, then this is termed *concurrency*.

This approach, which can be contrasted with an "over-the-wall", i.e. serial, paradigm, will require more than merely high-level attention to the whole problem. It will require suitable technical tools (such as modelling and cross-disciplinary performance budgeting) as well as involvement and commitment across the engineering disciplines involved.

Heads Up

It remains important to keep the business[3] and end-users' contexts in mind. Because of the hierarchical nature of most systems, and contractual practices in many engineering industries, it can be difficult to do so effectively. A function of systems engineering is to provide a continuous focus on relevant stakeholder needs when evaluating any design or change in the project.

[3] Here, "business" stands for the customer organisation of the product, and could in be a commercial, academic, military, medical or any other type of application domain.

The V-diagram

One of the most important tools for a systems engineer is the V-diagram, which is a conceptual model of a system development lifecycle. Most Systems Engineering processes include or incorporate at least one iteration of a V. The particular diagram in Figure 5 is an orientation diagram showing the main activities of systems engineering. Other V-diagrams can be drawn to show individual processes in greater detail, or to illustrate related activities.

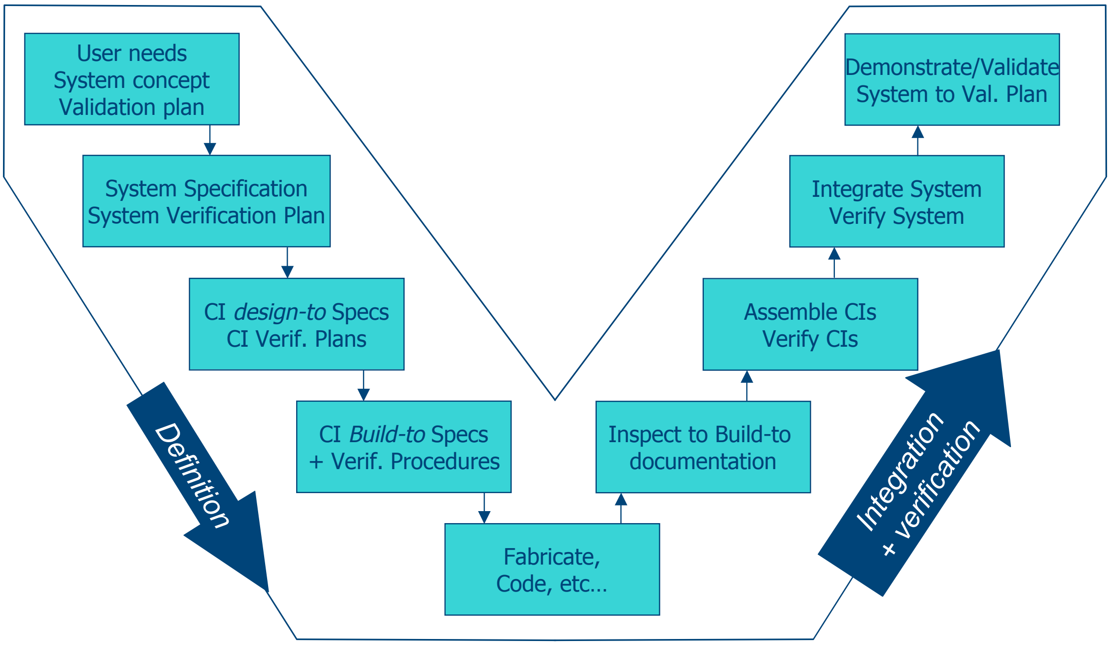


Figure 5: The V-diagram model of system development [Forsberg, Mooz and Cotterman, 2000]

All V-diagrams share the same principles of organisation. Time flows from left to right, and the process begins at top left with a consideration of the user needs, and ends at top right with the acceptance of the final system.

The vertical dimension in a V-diagram is the level of decomposition and definition of detail:

> Decomposition: The hierarchical functional and physical partitioning of any system into hardware assemblies, software components, and operator activities.
>
> Definition: The "design-to", "build-to" and "code-to" documentation that defines the functional and physical context of each entity.

Items further down the diagram are subordinate in the hierarchy of the system (in the sense of Figure 7). So, we can see that work on the lowest level components is deferred, in the formal sense, until the specifications of the higher-level sub-systems are finalised.

This is not to say that early work on the component level is impossible or discouraged. Indeed, early work, preparatory technological studies, for example, is often necessary to provide confidence of project feasibility. However such work could be rendered pointless by a change at a higher level before the finalisation of the upper layers. Work not in the main flow of the V is regarded as "off-core" [Forsberg, Mooz and Cotterman, 2000], and therefore at risk of being invalidated. Knowing where any work stands in relation to the V is a key management perspective.

The V is also useful in identifying the opportunities for testing the system products. In Figure 6, we make explicit the connection between the making of specifications at one time and the Verification and/or validation of those specifications at a later time. *Verification* confirms that an item meets its specifications ("building the system right"), whereas *Validation* confirms that its performance is appropriate to the context in which it is to be put ("building the right system").

This process map can be read in the following sequence:

1. Understand customer's requirements, develop the system concept and make the validation plan.

2. Develop the system specification and a system verification plan.

3. The system specifications are expanded into specifications and verification plans for the lower level components. These are called *Configuration Items* or simply CIs.

4. The specifications we have made for the Configuration Items (CIs) are "design-to" specifications – these describe the 'Whats'. They do not say how the components are actually to be made. They are now turned into "build-to" documentation and verification procedures – the 'Hows'. It is at this stage that the majority of creative work, technology development and engineering is done.

5. Now we are at the bottom of the V. In some versions of the diagram, this is identified with the piece-part level of the system, whereas others identify this as the level at which the specifications exist for a lower-level entity. In that case, the whole system is produced by managing a hierarchy of nested V-processes. Ultimately, the individual components are fabricated (or in the case of software, coded) in accordance with higher-level specifications that have been flowed down in a controlled manner.

6. The components (CIs) are then inspected against the build-to documentation, using the verification procedures. This is the first stage of verification.

7. The components or assemblies of components are then verified against the design-to documentation and verification procedures.

8. The entire system is then assembled from the individual CIs, and verified against the system performance specification.

To validate the system, the performance is compared, using the validation plan, to the user's requirements. Typically this will involve a demonstration of the product.

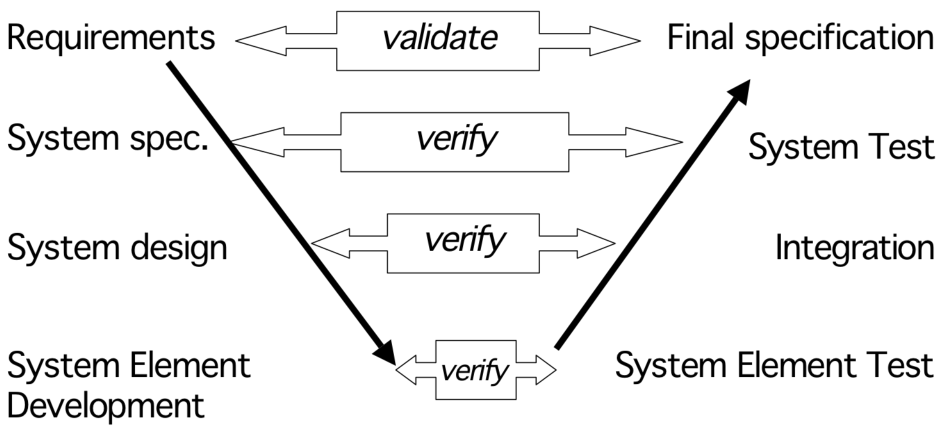


Figure 6: Verification and Validation

System Terminology

This is a guide to some of the standard terminology in relation to systems, extracted from the INCOSE SE Handbook [INCOSE, 2004].

The Hierarchy of System Elements

One of the Systems Engineer's first jobs on a project is to establish nomenclature and terminology that support clear, unambiguous communication and definition of the system, its functions, components, operations, and associated processes.

It is essential … that common definitions and understandings be established regarding general methods and terminology. … Toward that end, the following definitions of succeeding levels of the system hierarchy are useful.

System	An integrated set of elements, segments and/or subsystems to accomplish a defined objective, such as an air transportation system. The phrase "system of systems" is increasingly being used to denote collections of systems that, despite having significant functionality in their own right, are managed together as a complete entity for a certain application. A deployment of military forces which aggregates ships, aircraft and land assets is one example of this. Participating systems (and organisations!) are required to be engineered to be ready for such aggregations.
Element or Segment	A major product, service, or facility of the system, e.g., the aircraft element of an air transportation system (commonly used, but subsystems can be used instead of element/segments).
Subsystem	An integrated set of assemblies which performs a cleanly and clearly separated function, such as communications, electronics, structures, or controls; involving similar technical skills, or possibly a separate supplier. Examples are an aircraft on-board communications subsystem or an airport control tower as a subsystem of the air transportation system.
Assembly	An integrated set of components and/or subassemblies that comprise a defined part of a subsystem, e.g., the pilot's radar display console or the fuel injection assembly of the propulsion subsystem.
Subassembly	An integrated set of components and/or parts that comprise a well-defined portion of an assembly, e.g., a video display with its related integrated circuitry or a pilot's radio headset.
Component	Comprised of multiple parts; a cleanly identified item, e.g., a cathode ray tube or the ear-piece of the pilot's radio headset.
Part	The lowest level of separately identifiable items, e.g., a bolt to hold a console in place.

An example of a common hierarchy scheme is shown in Figure 7. This illustrates how the terms relate to each other on a particular system project.

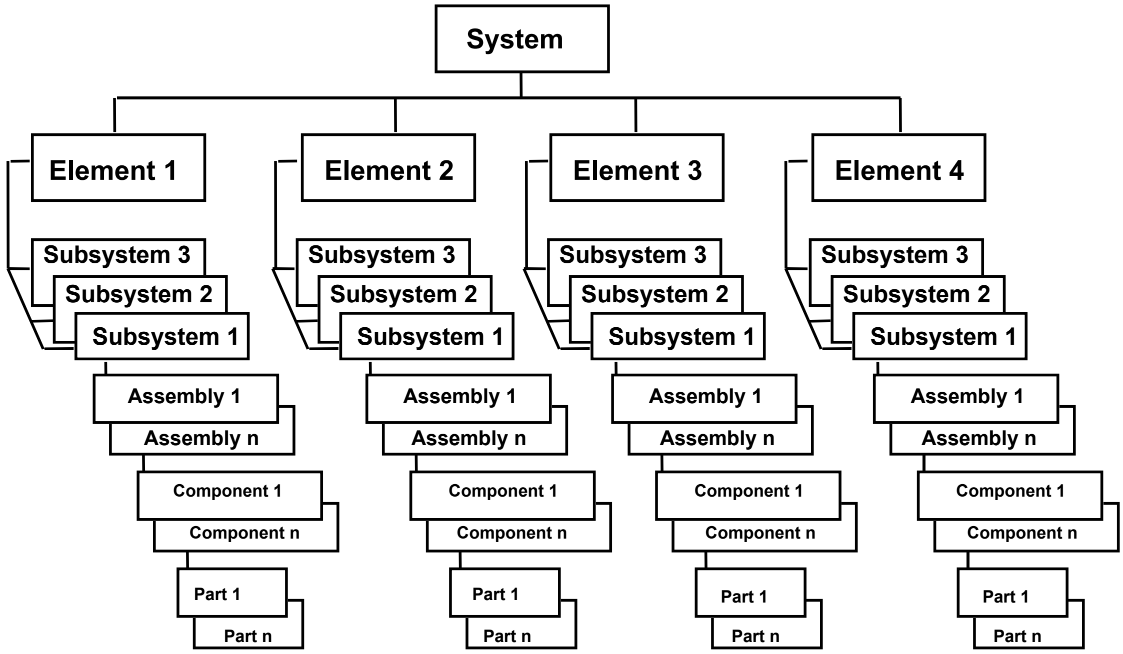


Figure 7. Hierarchy of System Elements

The depth of the hierarchy can be adjusted to fit the complexity of the system. For example, in the complex Apollo program, NASA added a "Module Level" in the hierarchy to breakout the Command Module, Lunar Module, etc. of the Space Vehicle Element. Simple systems may have fewer levels in the hierarchy than complex systems. Some examples of the hierarchy of system terminology, this time with names of actual system parts, are shown in Figure 8. It should be understood that these are examples only, to illustrate the variety of nomenclatures in use.

SYSTEM	AIR LOGISTICS	AIRCRAFT	INFORMATION	ELECTRIC CAR
ELEMENTS	AIRCRAFT PKG. PROCESSING SUPT. EQUIP. AIR & GRND. CREWS HUB, BASE, FACILITY		COMPUTERS NETWORK PRINTERS DATA STORAGE PERSONNEL	
SUB-SYSTEMS	PROPULSION STRUCTURE CONTROLS	PROPULSION STRUCTURE CONTROLS	DATA PROCESSOR OPERATING SYS. SOFTWARE	PWR. TRAIN BODY CHASSIS
COM-PONENTS	INTAKES COMPRESSOR INJECTORS CONTROLS	INTAKES COMPRESSOR INJECTORS CONTROLS	I/O CPU RAM ROM	BATTERY MOTOR(S) GENERATOR CONTROLLER

Figure 8. Examples of System Hierarchy

Decomposition and Definition

Once the customer's needs have been understood fully, the work to be done can be decomposed and defined in detail. This process involves the development or use of the following:

- **System Requirements Definition** – documenting the needs and translating these into measurable requirements. A complementary System Validation Plan will describe how a completed solution may be shown later to be compliant with the Requirements.
- **System Architecture** – describing the top-level view of the system, mainly its principal component parts (sub-systems) and how they interact (through its interfaces) to produce the required behaviour.
- **System Specification** – a description of the characteristics (functional and constituent properties) of the desired system, in a way that allows it to be produced. This specification will also refer to the required and desired quality parameters of the system.
- **Sub-system Requirements**. The above documents refer to the system as a whole, whereas each lower-order system element in the system hierarchy must also be described, in terms of functional (behaviour), constituent (makeup) and quality Requirements. Specifications can be flowed down from each of the Requirements, so that all elements in the hierarchy of the complete system design can be shown to have properties that support the overall System Requirement, and hence the user's need.

Although outlined here in a top-down manner, practical systems engineering integrates all levels of product design. If there are technological or other valid constraints at any level, these must be accounted for in higher-order activities, such as in the development of the architecture. The design that is sought is one that satisfies and optimises requirements and constraints from all quarters.

In traditional project management terms, the decomposition and definition activities are developed directly from the deliverables, and related to the agreed project objectives (including time and cost constraints). However, the tools of Systems Engineering are necessary to first find an acceptable system solution from which a formal set of deliverables may be defined. The tools of project management are as follows. These are used in conjunction with the Systems Engineering processes.

- **Product Breakdown Structure (PBS)** – Identification of the deliverables and breaking them down into manageable elements.
- **Tasks and Workpackages** – considering how to do things, from simple processes to complex sequences.
- **Work Breakdown Structure (WBS)** – Documenting all the work needed to produce the PBS deliverables.

Integration and Testing

Following Stevens [1998] we now consider what happens when the finished components are ready for assembly into assemblies and sub-systems; this being known as *Integration*.

A key element in the management and control of a systems development is the knowledge of the *baseline* configuration at any point within the process. The baseline refers to the configuration items (CI) that were referred to in Figure 5. *Configuration management* is a process that is concerned with identification, control and traceability of these baselines.

Effective configuration management is used to ensure that the status of each item is fully understood. A series of tests, variously called *verification, validation and/or acceptance* tests, are carried out at numerous levels.

Testing of the product, or of individual parts of it, can reveal faults that have to be corrected in a controlled way, and the configuration management process assists in this as well. The content and purpose of these tests will have been defined beforehand in the earlier planning stages. Part of this planning will determine which aspects of the product are to be tested in what way – the *test matrix*. There is a balance to be found between the thoroughness (and therefore cost) of the testing activities and the desired quality of the product.

The Integrity module of the course will cover these processes and many others in some detail.

Useful background will be found in chapter 5 of Stevens or Chapter 14 of Reilly [1993].

Conclusion and further reading

We have seen how a Systems-aware perspective encompasses the entire product development life cycle and aims to integrate a range of engineering disciplines. We have seen how holistic life cycle models, such as the V-diagram, can assist in developing an optimal management strategy in the face of the many challenges that will undoubtedly occur.

This short introduction cannot claim to introduce anything but a small subset of the techniques involved.

The reader should be aware of the INCOSE *Systems Engineering Handbook* and the emerging standard ISO/IEC 15288 *System Lifecycle Processes* [1999]. Whilst neither of these is particularly readable at an introductory level, they indicate the scope of what is regarded as standard knowledge and terminology in this field. Project management is also the subject of many international standards, but general project management textbooks will probably be found to be more useful. One text that covers the ground with a systems engineering awareness is Forsberg's *Visualizing Project Management* [2000]. Other suitable project management texts can be found on the course reading list.

For further orientation toward the core Systems Engineering processes, the textbooks by Stevens [1998] or Reilly [1993] are strongly recommended.

UCL's MSc in Systems Engineering Management

UCL's MSc in Systems Engineering Management aims to provide the knowledge and skills needed to practice as a Systems Engineering Manager. The way in which the modules relate to the V-diagram lifecycle described earlier is shown in Figure 9.

The course also seeks to provide delegates with the knowledge required by the core competencies of SE as outlined by INCOSE's UK Advisory Board (UKAB), together with some of the *supporting techniques* (such as modelling, risk management and technology planning) that are necessary parts of a systems engineering capability. In addition, delegates will develop *basic skills and behaviours* such as communication, lateral thinking and team-working skills.

Each of the MSc modules contributes differently to the core competencies. The three core modules – Systems Engineering Management, Systems Lifecycle and Systems Integrity – together provide a solid framework for the course, covering the majority of the core competencies. The remaining optional and applications modules allow delegates to explore more peripheral areas in more detail. INCOSE UKAB's core competencies work distinguishes between *Awareness* and *Supervised Practitioner* levels. Broadly speaking, the three core modules together seek to take delegates to *Supervised Practitioner* levels in each of the three core competence categories. Within these categories, there are a number of sub-headings; some of these sub-headings (such as 'Modelling'), will only be taught to *Awareness* level in the core modules and will require a more specific optional module (in this case, the 'Modelling' module), to reach the level of *Supervised Practitioner*. More detail on the precise scope of each of the modules is provided in the accompanying course information booklet.

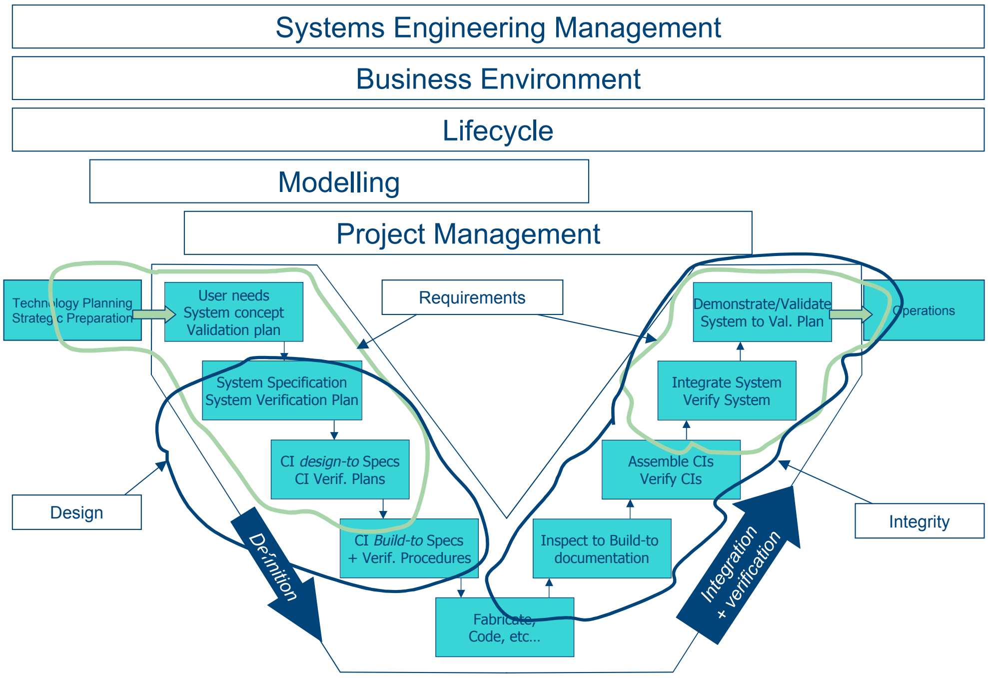


Figure 9: UCL Systems Engineering Management MSc Modules, with reference to the V-diagram§

Systems Thinking
- Systems concepts
- Super-system capability issues
- Business and technology environment

Holistic Lifecycle view
- Determine and manage stakeholder requirements
- *System Design:*
  - Architectural design
  - Concept generation
  - Design for …
  - Functional analysis
  - Interface Management
  - Maintaining Design Integrity
  - Modelling and Simulation
  - Select Preferred Solution
  - System Robustness
- Integration & Verification
- Validation
- Transition to Operation

Systems Engineering Management
- Concurrent engineering
- Enterprise Integration
- Integration of specialisms
- Lifecycle process definition
- Planning, monitoring and controlling

Figure 10: Core Competances of Systems Engineering (INCOSE).

References

The oxford english dictionary, Clarendon, Oxford, 1989.

About iec/iso 15288, http://www.15288.com/about_15288.htm, 1999 [cited 5 August 2005].

J. A. C. Brown, The social psychology of industry, Penguin, Harmandsworth, 1954.

P. Checkland, Systems thinking, systems practice, Wiley, Chichester, 1999.

K. Forsberg, H. Mooz and H. Cotterman, Visualizing project management, Wiley, New York/Chichester, 2000.

INCOSE, Systems engineering handbook, INCOSE, 2004, 288.

---, What is systems engineering? http://www.incose.org/practice/whatissystemeng.aspx, 2004 [cited July 2005].

---, About incose, http://www.incose.org/about/index.aspx, 2005 [cited July 2005].

S. B. Johnson, Three approaches to big technology, Technology and Culture 38(4) (October 1997 1997), 891-919.

L. A. Martin-Vega, The purpose and evolution of industrial engineering, McGraw-Hill, New York, 2001.

N. B. Reilly, Successful systems engineering for engineers and managers, Chapman and Hall, 1993.

R. Stevens, P. Brook, K. Jackson and S. Arnold, Systems engineering: Coping with complexity, Prentice Hall, London, 1998.

F. W. Taylor, Principles of scientific management, Harper, New York, 1911.