

Introduction to Systems Engineering Fundamentals

Introduction to Systems Engineering Fundamentals

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

Lead Author: Rick Adcock, **Contributing Authors:** Brian Wells, Scott Jackson, Janet Singer, Duane Hybertson

This article forms part of the Systems Fundamentals knowledge area (KA). It provides various perspectives on systems, including definitions, scope, and context.

This article provides a guide to some of the basic concepts of systems developed by systems science and discusses how these relate to the definitions to be found in systems engineering (SE) literature. The concept of an engineered system is introduced as the system context of critical relevance to SE.



Contents

Overview

A General View of Systems

System Context

Systems and Systems Engineering

Introduction to Engineered Systems

Life Cycle Definitions

Engineered System Context

References

Works Cited

Primary References

Additional References

Overview

In the System Fundamentals KA we will define some terms and ideas which are foundational to the understanding and practice of Systems Engineering (SE). In particular, a number of views of system are explored; these are summarized below and described in more detail with links to relevant references in the rest of this article.

- A simple definition of **System is any set of related parts for which there is sufficient coherence between the parts to make viewing them as a whole useful**. If we consider more complex situations in which the parts of a system can also be viewed as systems, we can identify useful common systems concepts to aid our understanding. This allows the creation of systems theories, models and approaches useful to anyone trying to understand, create or use collections of related things, independent of what the system is made of or the application domain considering it.
- Many of these common systems ideas relate to complex networks or hierarchies of related system elements. A **System Context is a set of system interrelationships associated with a particular system of interest (Sol) within a real world environment**. One or more views of a context allow us to focus on the Sol but not lose sight of its broader, holistic relationships and influences. Context can be used for many kinds of system but is particularly useful for scoping problems and enabling the creation of solutions which combine people and technology and operate in the natural world. These are referred to as socio-technical system contexts.
- Systems Engineering is one of the disciplines interested in socio-technical systems across their whole life. This includes where problems come from and how they are defined, how we identify and select candidate solutions, how to balance technology and human elements in the wider solution context, how to manage the complex organizational systems needed to develop new solutions, and how developed solutions are used, sustained and disposed of. To support this, we define an **Engineered System as a socio-technical system which is the focus of a**

Systems Engineering life cycle.

- While SE is focused on the delivery of an engineered system of interest, an **SE should consider the full Engineered System Context so that the necessary understanding can be reached and the right systems engineering decisions can be made across each Life Cycle.**

A General View of Systems

The idea of a system whole can be found in both Western and Eastern philosophy. Many philosophers have considered notions of holism, the concept that ideas, people or things must be considered in relation to the things around them to be fully understood (M'Pherson 1974).

One influential systems science definition of a system comes from general system theory (GST):

A System is a set of elements in interaction. (Bertalanffy 1968)

The parts of a system may be conceptual organizations of ideas in symbolic form or real objects. GST considers **abstract systems** to contain only conceptual elements and **concrete systems** to contain at least two elements that are real objects, e.g. people, information, software, and physical artifacts, etc.

Similar ideas of wholeness can be found in systems engineering literature. For example:

We believe that the essence of a system is 'togetherness', the drawing together of various parts and the relationships they form in order to produce a new whole... (Boardman and Sauser 2008).

The cohesive interactions between a set of parts suggest a system boundary and define what membership of the system means. For **closed systems**, all aspects of the system exist within this boundary. This idea is useful for abstract systems and for some theoretical system descriptions.

The boundary of an **open systems** defines elements and relationships which can be considered part of the system and describes how these elements interact across the boundary with related elements in the environment. The relationships among the elements of an open system can be understood as a combination of the systems structure and behavior. The structure of a system describes a set of system elements and the allowable relationships between them. System behavior refers to the effects or outcomes produced when an instance of the system interacts with its environment. An allowable configuration of the relationships among elements is referred to as a system state. A stable system is one which returns to its original, or another stable, state following a disturbance in the environment. *System wholes entities often exhibit emergence, behavior which is meaningful only when attributed to the whole, not to its parts* (Checkland 1999).

The identification of a system and its boundary is ultimately the choice of the observer. This may be through observation and classification of sets of elements as systems, through an abstract conceptualization of one or more possible boundaries and relationships in a given situation, or a mixture of this concrete and conceptual thinking. This underlines the fact that any particular identification of a system is a human construct used to help make better sense of a set of things and to share that understanding with others if needed.

Many natural, social and man made things can be better understood by viewing them as open systems. One of the reasons we find the idea of systems useful is that it is possible to identify shared concepts which apply to many system views. These recurring concepts or isomorphies can give useful insights into many situations, independently of the kinds of elements of which a particular system is composed. The ideas of structure, behavior, emergence and state are examples of such concepts. The identification of these shared system ideas is the basis for systems thinking and their use in developing theories and approaches in a wide range of fields of study the basis for system sciences.

Systems Engineering (SE), and a number of other related disciplines use systems concepts, patterns and models in the creation of useful outcomes or things. The concept of a network of open systems created, sustained and used to achieve a purpose within one or more environments is a powerful model that can be used to understand many complex real world situations and

provide a basis for effective problem solving within them.

System Context

Bertalanffy (1968) divided open systems into nine real world types ranging from static structures and control mechanisms to socio-cultural systems. Other similar classification systems are discussed in the article Types of Systems.

The following is a simple classification of system elements which we find at the heart of many of these classifications:

- Natural system elements, objects or concepts which exist outside of any practical human control.
Examples: the real number system, the solar system, planetary atmosphere circulation systems.
- Social system elements, either abstract human types or social constructs, or concrete individuals or social groups.
- Technological System elements, man-made artifacts or constructs; including physical hardware, software and information.

While the above distinctions can be made as a general abstract classification, in reality there are no hard and fast boundaries between these types of systems: e.g., natural systems are operated by, developed by, and often contain social systems, which depend on technical systems to fully realize their purpose. Systems which contain technical and either human or natural elements are often called socio-technical systems. The behavior of such systems is determined both by the nature of the technical elements and by their ability to integrate with or deal with the variability of the natural and social systems around them.

Many of the original ideas upon which GST and other branches of system study are based come from the study of systems in the natural and social sciences. Many natural and social systems are initially formed as simple structures through the inherent cohesion among a set of elements. Once formed, they will tend to stay in this structure, as well as combine and evolve further into more complex stable states to exploit this cohesion in order to sustain themselves in the face of threats or environmental pressures. Such complex systems may exhibit specialization of elements, with elements taking

on roles which contribute to the system purpose, but losing some or all of their separate identity outside the system. Such roles might include management of resources, defense, self-regulation or problem solving, and control. Natural and social systems can be understood through an understanding of this wholeness, cohesion and specialization. They can also be guided towards the development of behaviors which not only enhance their basic survival, but also fulfill other goals of benefit to them or the systems around them. In *The Architecture of Complexity*, Simon (1962) has shown that natural or social systems which evolve via a series of stable “hierarchical intermediate forms” will be more successful and resilient to environmental change.

Thus, it is often true that the environment in which a particular system sits and the elements of that system can themselves be considered as open systems. It can be useful to consider collections of related elements as both a system and a part of one or more other systems. For example, a “holon” or system element was defined by Koestler as something which exists simultaneously as a whole and as a part (Koestler 1967). At some point, the nature of the relationships between elements within and across boundaries in a hierarchy of systems may lead to complex structures and emergent behaviors which are difficult to understand or predict. Such complexity can often best be dealt with not only by looking for more detail, but also by considering the wider open system relationships.

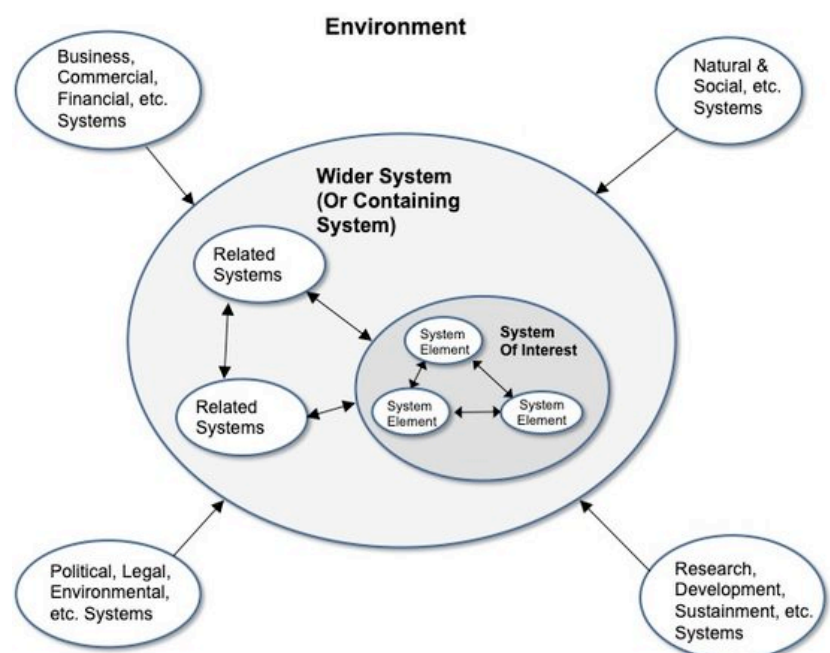


Figure 1: General description of System Context (SEBoK Original)

A system context describes all of the external elements

which interact across the boundary of a particular system of interest (SoI) and a sufficient view of the elements within its boundary, to allow the SoI to be better understood as part of a wider systems whole. To fully understand the context, we also need to identify the environment in which the SoI and wider system sit and the systems in the environment which influence them.

Many man-made systems are designed as networks and hierarchies of related system elements to achieve desirable behaviors and the kinds of the resilience seen in natural systems. While such systems can be deliberately created to take advantage of system properties such as holism and stability, they must also consider system challenges such as complexity and emergence. Considering different views of a SoI and its context over its life can help enable this understanding. Considering systems in context allows us to focus on a SoI while maintaining the necessary wider, holistic systems perspective. This is one of the foundations of the Systems Approach described in SEBoK part 2, which forms a foundation of systems engineering.

Systems and Systems Engineering

Some of the systems ideas discussed above form part of the systems engineering body of knowledge. Systems engineering literature, standards and guides often refer to “the system” to characterize a socio-technical system with a defined purpose as the focus of SE, e.g.

- “A system is a value-delivering object” (Dori 2002).
- “A system is an array of components designed to accomplish a particular objective according to plan” (Johnson, Kast, and Rosenzweig 1963).
- “A system is defined as a set of concepts and/or elements used to satisfy a need or requirement” (Miles 1973).

The International Council on Systems Engineering Handbook (INCOSE 2015) generalizes this idea, defining system as “an interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, people, information, techniques, facilities, services, and other support elements.” While these definitions cover the socio-technical systems created by SE, it is also necessary to consider the natural or social problem situations in which these systems sit, the social systems which developed, sustained and used them, and the commercial or public

enterprises in which these all sit as systems (Martin 2004).

Hence, while many SE authors talk about systems and systems ideas, they are often based on a particular world view which related to engineered artifacts. It would also be useful to take a broader view of the context in which these artifacts sit, and to consider through life relationships as part of that context. To help promote this, the SEBoK will attempt to be more precise with its use of the word system, and distinguish between general systems principles and the specific socio-technical systems created by SE.

The term socio-technical system is used by many in the systems community and may have meanings outside of that relevant to SE. Hence, we will define an engineered system as a socio-technical system which forms the primary focus or system of interest (SoI) for an application of SE. A SE life cycle will consider an engineered system context, from initial problem formulation through to final safe removal from use (INCOSE 2015). A more detailed discussion of engineered system context and how it relates to the foundations of systems engineering practice can be found below.

Introduction to Engineered Systems

An engineered system defines a context containing both technology and social or natural elements, developed for a defined purpose by an engineering life cycle.

Engineered system contexts:

- are created, used and sustained to achieve a purpose, goal or mission that is of interest to an enterprise, team, or an individual.
- require a commitment of resources for development and support.
- are driven by stakeholders with multiple views on the use or creation of the system, or with some other stake in the system, its properties or existence.
- contain engineered hardware, software, people, services, or a combination of these.
- exist within an environment that impacts the characteristics, use, sustainment and creation of the system.

Engineered systems typically:

- are defined by their purpose, goal or mission.
- have a life cycle and evolution dynamics.
- may include human operators (interacting with the systems via processes) as well as other social and natural components that must be considered in the design and development of the system.
- are part of a system-of-interest hierarchy.

Open systems are a useful way to understand many complex situations. Traditional engineering disciplines have become very good at building up detailed models and design practices to deal with the complexity of tightly integrated collections of elements within a technology domain. It is possible to model the seemingly random integration of lots of similar elements using statistical approaches. Systems Engineering makes use of both these aspects of system complexity, as discussed in the Complexity article.

SE also considers the complexity of relatively small numbers of elements taken from a range of design disciplines together with people who may not always be experienced or have detailed training in their use. Such engineered systems may be deployed in uncertain or changing environments and be used to help people achieve a number of loosely defined outcomes. Relatively small changes in the internal working of these engineered systems' elements, or in how those elements are combined, may lead to the emergence of complex or un-expected outcomes. It can be difficult to predict and design for all such outcomes during an engineered system's creation, or to respond to them during its use. Iterative life cycle approaches which explore the complexity and emergence over a number of cycles of development and use are needed to deal with this aspect of complexity. The ways that system engineering deals with these aspects of complexity in the definition of life cycle and life cycle processes applied to an engineered system context is fully explored in Part 3

Life Cycle Definitions

As well as being a kind of system, an engineered system is also the focus of a life cycle and hence part of a commercial transaction. Historically,

| *Economists divide all economic activity*

into two broad categories, goods and services. Goods-producing industries are agriculture, mining, manufacturing, and construction; each of them creates some kind of tangible object. Service industries include everything else: banking, communications, wholesale and retail trade, all professional services such as engineering, computer software development, and medicine, nonprofit economic activity, all consumer services, and all government services, including defense and administration of justice.... (Encyclopedia Britannica 2011).

The following diagram defines some terms related to an engineered system life cycle and the development of goods (products) and services.

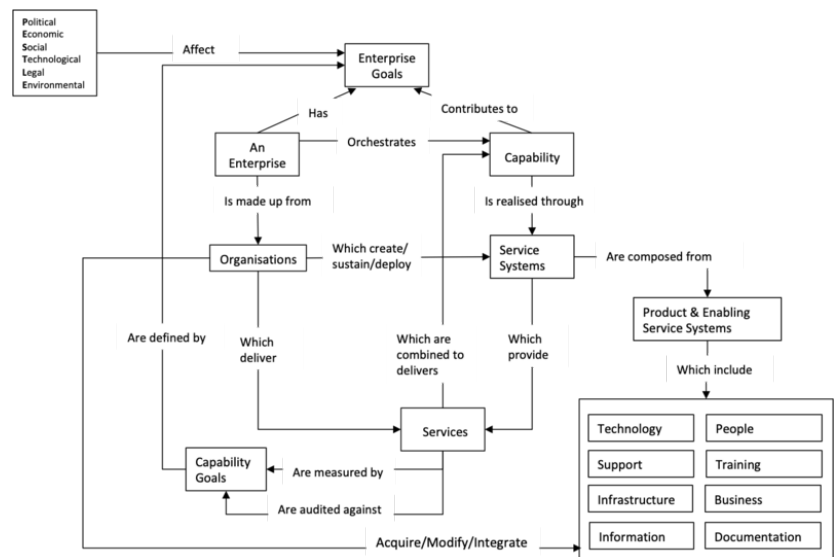


Figure 2: Life Cycle Terminology (Modified from Capability Engineering - an Analysis of Perspectives (modified from (Henshaw et al, 2011), used with permission))

In the above figure the capability needed to enable an enterprise to achieve its goals is delivered by the synchronized use of services. Those services are provided by a service system, which is created, sustained and deployed by one or more organizations. A service system is composed of people, technology, information, and access to related services and other necessary resources. Some of these resources are provided by enabling services and the technological elements may be developed and supplied as product systems. An enterprise system describes a collection of related capabilities and associated services which together enable the achievement of the overall purpose of an enterprise as a government, business or societal entity.

Measurement and review of enterprise goals may define needs for change which require an organization to acquire or modify, and integrate the elements needed to evolve its service systems. The general terminology above is described briefly in the associated glossary definitions and expanded in related articles in Part 4: Applications of Systems Engineering.

Engineered System Context

Engineered systems are developed as combinations of products and services within a life cycle. The figure below gives a general view of the full context for any potential application of a SE life cycle.

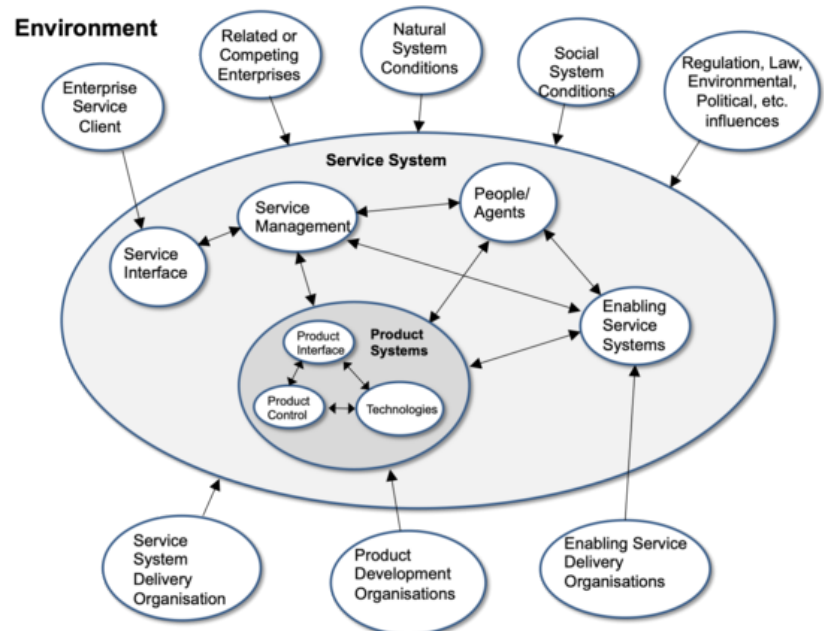


Figure 3: General Engineered System Context (SEBoK original)

In this view a service system related directly to a capability need sets the overall boundary. This need establishes the problem situation or opportunity which encapsulates the starting point of any life cycle. Within this service system are the related services, products and people (or intelligent software agents) needed to fully deliver a solution to that need. The environment includes any people, organizations, rules or conditions which influence or constrain the service system or the things within it. The SoI for a particular SE life cycle may be defined at any level of this general context. While the focus of the context will vary for each life cycle it is important that some version of this general context is considered for all SE life cycles, to help maintain a

holistic view of problem and solution. This is discussed in Types of Systems.

An engineered system context describes the context for a SoI so that the necessary understanding can be reached and the right systems engineering decisions can be made across the life of that SoI. This will require a number of different views of the context across a SE life cycle, both to identify all external influence on the SoI and to guide and constraint the systems engineering of the elements of the SoI. A full engineered systems context will include the problem situation from which a need for a SoI is identified, one or more socio technical solutions, the organizations needed to create and sustain new solutions and the operational environment within which those solutions must be integrated, used and eventually disposed. The kinds of views which can be used to represent a SoI context over its life and how those views can be combined into models is discussed in the Representing Systems with Models KA in Part 2. The activities which use those models are described conceptually in the Systems Approach Applied to Engineered Systems KA in part 2 and related to more formal SE life cycle processes in Part 3.

References

Works Cited

Bertalanffy, L. von. 1968. *General System Theory: Foundations, Development, Applications*, rev. ed. New York: Braziller.

Boardman, J. and B. Sauser. 2008. *Systems Thinking: Coping with 21st Century Problems*. Boca Raton, FL, USA: Taylor & Francis.

Checkland, P. 1999. *Systems Thinking, Systems Practice*. New York, NY, USA: Wiley and Sons, Inc.

Dori, D. 2002. *Object-Process Methodology - A Holistic Systems Paradigm*. New York, NY, USA: Springer.

Henshaw, M., D. Kemp, P. Lister, A. Daw, A. Harding, A. Farncombe, and M. Touchin. 2011. "Capability engineering - An analysis of perspectives." Presented at International Council on Systems Engineering (INCOSE) 21st International Symposium, Denver, CO, USA, June 20-23, 2011.

Hitchins, D. 2009. "What are the general principles

applicable to systems?" *INCOSE Insight*, vol. 12, no. 4, pp. 59-63.

INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

Johnson, R.A., F.W. Kast, and J.E. Rosenzweig. 1963. *The Theory and Management of Systems*. New York, NY, USA: McGraw-Hill Book Company.

Koestler, A. 1990. *The Ghost in the Machine*, 1990 reprint ed. Sturgis, Michigan, USA: Penguin Group.

Martin, J, 2004. "The seven samurai of systems engineering: Dealing with the complexity of 7 interrelated systems." Proceedings of the 14th Annual International Council on Systems Engineering International Symposium, 20-24 June, 2004, Toulouse, France, 20-24 June, 2004.

Miles, R.F. (ed). 1973. *System Concepts*. New York, NY, USA: Wiley and Sons, Inc.

M'Pherson, P.K. 1974. "A perspective on systems science and systems philosophy." *Futures*. Vol. 6, no. 3, pp. 219-39.

Simon, H.A. 1962. "The architecture of complexity." *Proceedings of the American Philosophical Society*. Vol. 106, no. 6 (Dec. 12, 1962), pp. 467-482.

Primary References

Bertalanffy, L., von. 1968. *General System Theory: Foundations, Development, Applications*, rev. ed. New York, NY, USA: Braziller.

INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

Additional References

Hybertson, Duane. 2009. *Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boca Raton, FL, USA:

CRC Press.

Hubka, Vladimir, and W. E. Eder. 1988. *Theory of Technical Systems: A Total Concept Theory for Engineering Design*. Berlin: Springer-Verlag.

Laszlo, E., ed. 1972. *The Relevance of General Systems Theory: Papers Presented to Ludwig von Bertalanffy on His Seventieth Birthday*. New York, NY, USA: George Brazillier.

< [Previous Article](#) | [Parent Article](#) | [Next Article](#) >

SEBoK v. 2.9, released 20 November 2023

Retrieved from

"https://sebokwiki.org/w/index.php?title=Introduction_to_Systems_Engineering_Fundamentals&oldid=69708"

This page was last edited on 18 November 2023, at 22:38.