Guide to the
Systems Engineering Body of Knowledge (SEBoK) version 0.75

Part 2

Part 2

From .75 Sebokcase

Part 2

- Part 2: Systems
    - Knowledge Area: Systems Thinking
        - Topic: What is a System?
        - Topic: Types of Systems
        - Topic: What is Systems Thinking?
        - Topic: System Concepts
    - Knowledge Area: System Science
        - Topic: History of System Science
        - Topic: System Methodology
        - Topic: Groupings of Systems
        - Topic: Complexity
        - Topic: Emergence
    - Knowledge Area: Representing Systems with Models
        - Topic: What is a Model?
        - Topic: Why Model?
        - Topic: Types of Models
        - Topic: System Modeling Concepts
        - Topic: Modeling Standards
    - Knowledge Area: Systems Approach
        - Topic: Overview of the Systems Approach
        - Topic: Engineered System Context
        - Topic: Identifying and Understanding Problems and Opportunities
        - Topic: Synthesizing Possible Solutions
        - Topic: Analysis and Selection between Alternative Solutions
        - Topic: Implementing and Proving a Solution
        - Topic: Deploying, Using, and Sustaining Systems to Solve Problems
        - Topic: Acquirer/Supplier Responsibility
        - Topic: Applying the Systems Approach
    - Knowledge Area: Systems Challenges
        - Topic: Complex System Challenges
        - Topic: Dynamically Changing Systems

Part 2: Systems

Part 2 is a guide to knowledge associated with systems (glossary), particularly knowledge relevant to systems engineering (glossary) (SE). Part 2 elaborates on the underlying systems ideas upon which the following parts of the SEBoK are based, thus providing a foundation for the remainder of the SEBoK. Part 2 also defines the key principles of a Systems Approach, which will be referred to directly in explaining the practices of SE.

Knowledge Areas in Part 2: Systems

Part 2: Systems, contains the following knowledge areas:

- Systems Thinking
- System Science
- Representing Systems with Models
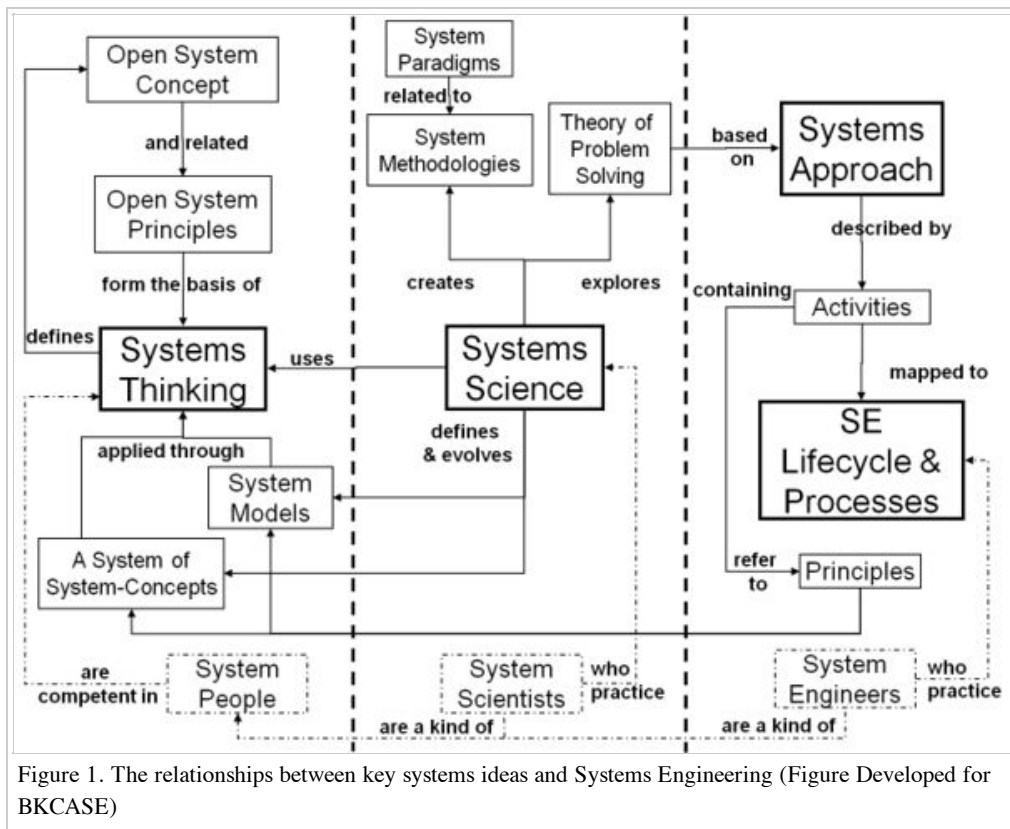- Systems Approach
- Systems Challenges

Scope of Part 2

Part 2 of the SEBoK contains a guide to knowledge about systems, which is relevant to a full understanding of SE. As such, it deals with both abstract concepts (glossary) about systems and practical principles (glossary) which guide the use of these concepts to underpin the understanding, creation, management, and use of (socio-technical) engineered systems (glossary).

- Concepts are sub divisions of knowledge that describe a single idea or property of things.
- Principles are statements which guide the way we might think or act in a given situation as a consequence of one or more concepts.

For example, the concept of *openness* states that some systems must exchange energy, information or material with their environment to exist and function. Principles based on this concept include that an open system (glossary) can only be fully understood in its environment (glossary), or that changes to the environment may change how a system behaves.

The following diagram summarizes the way in which the knowledge in SEBoK Part 2 is organized.

Figure 1. The relationships between key systems ideas and Systems Engineering (Figure Developed for BKCASE)

The aim of this model is to provide a guide to the major aspects of systems knowledge in such a way that it can be useful to SE in 2 ways:

- To define an underlying theory for SE standards and descriptions; and
- To describe a fundamental way of thinking about complex situations as systems, which should guide the way in which people apply SE practices to best effect.

Each part of this diagram is explained in more detail below. The model is divided into four sections, each describing how we have treated systems knowledge in the SEBoK.

1. The System Science Knowledge Area provides an overview of the most influential movements in systems science. This section explores the chronological development of systems knowledge and discusses some of the different approaches taken in applying it to real problems. This is useful background knowledge of general interest to systems engineers, in particular those involved in development of SE standards and descriptions.
2. The Systems Thinking Knowledge Area describes key concepts shared across systems research and practice and organizes them as a system of related ideas. Understanding this way of thinking should be a key competence for anyone undertaking systems research or practicing SE.
3. The Representing Systems with Models Knowledge Area considers the key role that abstract models play in both the development of system theories and the application of system thinking.
4. The Systems Approach Knowledge Area defines a structured problem or opportunity discovery, as well as exploration and resolution

approaches that can be applied to all engineered systems (glossary), which is based on systems thinking. This KA provides principles that map directly to SE practice.

It should be noted that, while the knowledge presented in this part of the SEBoK has been organized into these four areas to facilitate understanding, the intention is to present a rounded picture of research and practice based on system knowledge. These four knowledge areas are tightly coupled and should be seen together as a system of ideas for connecting understanding, research, and practice, based on system knowledge which applies to all types of domains and underpins a wide range of scientific, management, and engineering disciplines.

Systems Thinking

The origins of systems thinking began in attempts to better understand complex situations in biology, organizations, control, etc. From these attempts come a set of fundamental concepts defining the idea of an open system and associated principles such as holism, emergence, etc. which have become the foundations of system thinking.

Figure 2. Systems Thinking and Systems Science (Figure Developed for BKCASE)
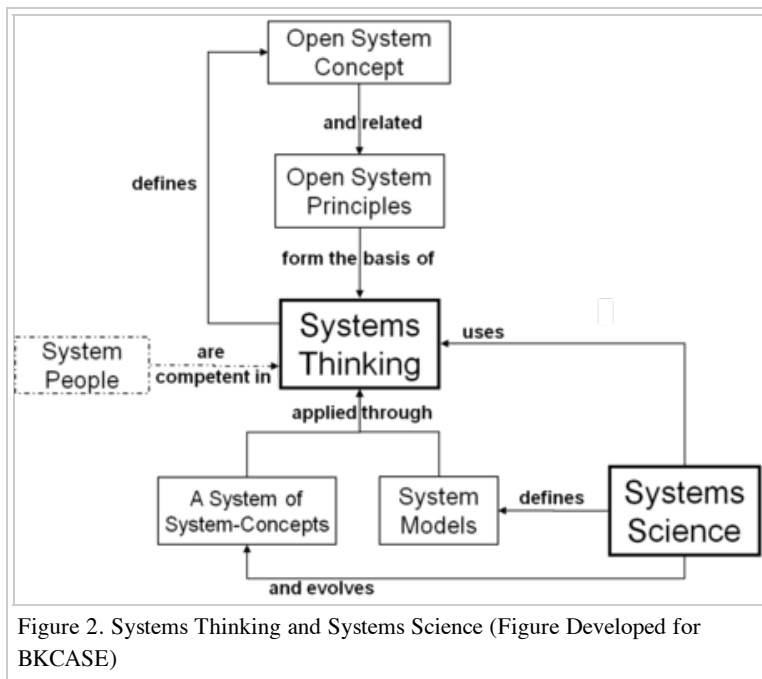
Over time, systems thinking has been extended and refined by the creation of a set of abstract system models and a system of systems-concepts, which apply to all systems, independent of their domain.

System science is a community of research and practice which is based on systems thinking, and which adds to and evolves the systems thinking body of knowledge.

Systems Thinking stands alone as a way of thinking which "Systems People" can use to gain a fuller understanding of any situation and through this, guide a wide range of human activity.

Checkland (1999) discusses the use of systems thinking as a way to both understand and intervene in a problem situation. Lawson (2010) defines three related contexts in which systems thinking can be used:

1. to better understand a current real world situation by defining an abstract Situation System;
2. to describe the Respondent System that might be used to understand, use, manage, sustain or change the situation system;
3. to understand how one or more system assets needs be acquired, modified, or created to achieve the purpose of the respondent system.

Systems Science

Systems Science (glossary) is an interdisciplinary field of science that studies the nature of complex systems in nature, society, and science. It aims to develop interdisciplinary foundations, which are applicable in a variety of areas, such as engineering, biology, medicine and social sciences. This systems science is practiced by a community of system researchers who perform research based on systems thinking.

Figure 3. Systems Science and Systems Thinking (Figure Developed for BKCASE)
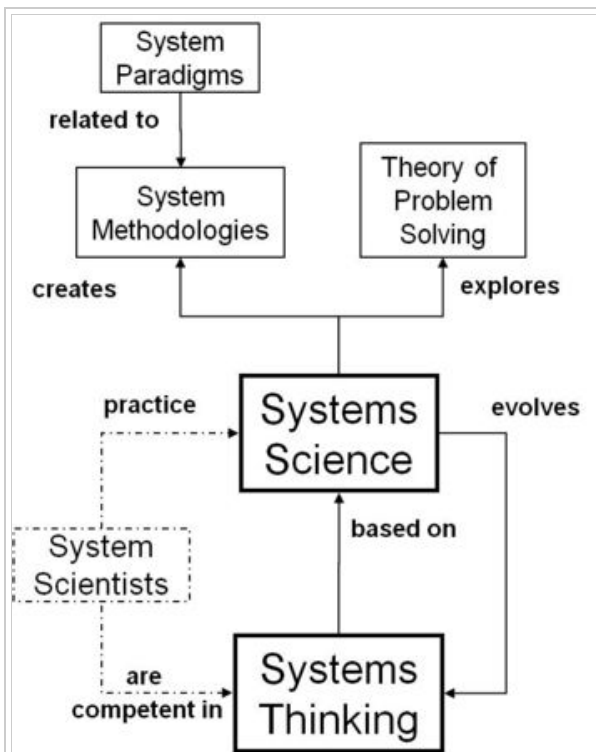
Many systems science practitioners also develop system methodologies that provide a framework of concepts and principles for tackling specific aspects of system problems.

These methodologies are grouped around a set of paradigms which define particular world views or ways of thinking about systems, e.g., hard systems, soft systems, system dynamics, etc.

Another output of system science is the emergence of a theory of problem solving (also referred to as a theory of engineering, design, intervention, etc.). Some of this theory is published, and some is embedded in the methodologies.

The work of system science expands the shared understanding of systems and is used to evolve the body of knowledge of systems thinking, both in expanding the system-concepts, and in creating new models or modeling notations.

Systems science should be conducted by researchers who are themselves competent in systems thinking. As discussed above, this will include understanding the situation system under study, creating a research resolution system, and understanding any needed research system assets.

Systems Approach

SE lifecycle and process definitions, standards, and guides are underpinned by aspects of systems science and make use of system methodologies, but this is often not done in a rigorous or consistent way. Those conducting SE are often simply following process definitions and are not aware of the fundamentals and relevance of systems thinking.
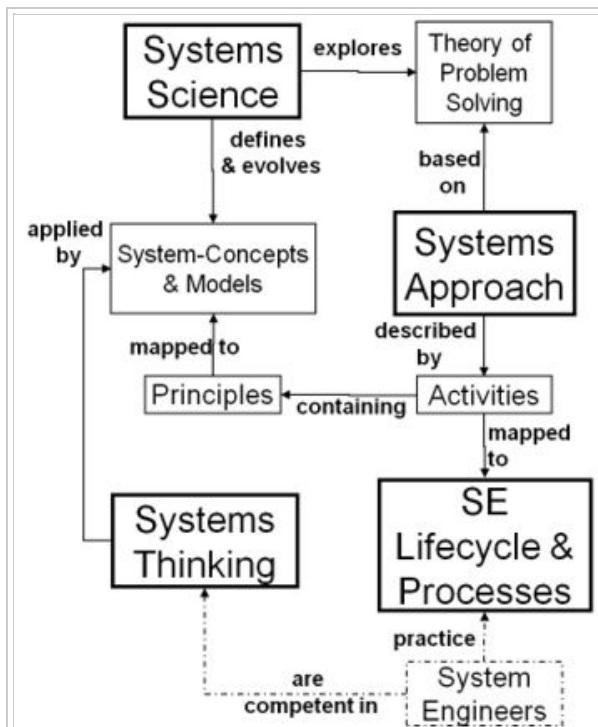
Figure 4. Systems Engineering and the Systems Approach
(Figure Developed for BKCASE)

A Systems Approach can be defined, synthesizing elements of Systems Science to create:

- A framework of activities that can be applied to complex situations requiring engineered system based solutions.
- System principles within each activity that relate back to the systems thinking models and concepts.

The activities and principles of the systems approach can be mapped onto the processes of SE to increase the system science foundations of SE.

This mapping will provide guidance on which system-concepts should be considered when applying a process and which system models can be used to support process activities.

SE should be practiced by those who are themselves competent in systems thinking. As discussed above, this will include understanding the problem or opportunity situation of a system, creation of a respondent system, and the understanding of life cycle management any system products or services assets.

References

Works Cited

Checkland, P B. 1999. *Systems Thinking, Systems Practice*. Chichester, England, UK: John Wiley and Sons.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College, UK.

Primary References

Checkland, P. B. 1999. Systems Thinking, Systems Practice. Chichester, UK: John Wiley & Sons Ltd.

Lawson, H. 2010. A Journey Through the Systems Landscape. London, UK: College Publications, Kings College, UK.

Additional References

MITRE Corporation. 2011. *Systems Engineering Guide: Comprehensive Viewpoint*. Accessed 2/28/2012 at http://www.mitre.org/work/systems_engineering/guide/enterprise_engineering/comprehensive_viewpoint/

MITRE Corporation. 2011. *Systems Engineering Guide: Systems Thinking*. Accessed 2/28/2012 at http://www.mitre.org/work/systems_engineering/guide/enterprise_engineering/comprehensive_viewpoint/systems_thinking.html

Senge, P. M. 1990. *The Fifth Discipline: The Art & Practice of the Learning Organization*. New York, NY: Doubleday Business.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

Like

Add New Comment

Post as …

Showing 0 comments

Sort by   Popular now ▲▼   ✉ Subscribe by email   📶 Subscribe by RSS

blog comments powered by **DISQUS**

Systems Thinking

This Knowledge Area (KA) provides a guide to knowledge about systems (glossary) and systems thinking (glossary), which together form the foundations for all applications of a systems approach (glossary) and applications of systems engineering (glossary) (SE). This knowledge is not specific to SE, but is part of a wider systems body of knowledge. The SEBoK does not capture all of the system knowledge here; rather, it identifies those aspects relevant to the SEBoK.

Topics

The topics contained within this knowledge area include the following:

- What is a System?
- Types of Systems
- What is Systems Thinking?

- System Concepts

Introduction

The word system (glossary) is used in all areas of human activity and at all levels. But what do people mean when they use the word "system" and is there some part of that meaning which is common to all applications? What is a system? Are there different types of systems? These and similar questions, all relating to the use of the word "system" in everyday language, need to be given careful consideration in order to achieve a clear understanding of the underlying concepts of systems thinking before applying it to an engineering (glossary) context.

Systems thinking (glossary) is concerned with understanding or intervening in systems, based on the principles and concepts of systems. This KA offers some basic definitions of systems thinking.

General System Theory (glossary) (GST) (von Bertalanffy, 1968) considers the similarities between systems from different domains as a set of common systems principles and concepts.

- A **principle** is a rule of conduct or behavior. To take this further, a principle is a "basic generalization that is accepted as true and that can be used as a basis for reasoning or conduct" (WordWeb.com). A principle can also be thought of as a "basic truth or law or assumption" (ibid).

- A **concept** is an abstraction, or a general idea inferred or derived from specific instances. For example, by viewing a pet dog, one can infer that there are other dogs of that "type." Hence, from this observation (or perhaps a set of observations) the concept of a dog is developed in one's mind. Concepts are bearers of meaning, as opposed to agents of meaning, and can only be thought about, or designated, by means of a name.

Principles depend on concepts in order to state a "truth." Hence, principles and concepts go hand in hand; principles cannot exist without concepts and concepts are not very useful without principles to help guide the proper way to act (Lawson and Martin 2008).

GST tends to concentrate on the principles and philosophy behind the aforementioned idea. "Despite the importance of system concepts … we do not yet have a unified or integrated set (i.e., a system) of such concepts" (Ackoff, 1971).

Ackoff proposes a **system of "system-concepts"** to bring together the wide variety of concepts which have been proposed. His 30 distinct concepts are grouped under four headings, or principles, "**How Systems are formed**," "**How Systems Change**," "**How Systems Behave**," and "**How Systems Adapt and Learn**."

Lawson describes a system of "system concepts" (Lawson 2010) where systems are categorized according to Fundamental Concepts, Types, Topologies, Focus, Complexity, and Roles. Hitchins defines a similar set of principles which also consider some of the issues of hierarchy and complexity of particular relevance to a system approach (Hitchins 2009).

This KA provides an overview of the most important aspects of this system of systems concepts (glossary).

References

Works Cited

Ackoff, R.L. 1971. "Towards a System of Systems Concepts". *Management Science*. 17(11).

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

Hitchins, D. 2009. "What Are the General Principles Applicable to Systems?" *Insight* 12(4).

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Lawson, H., and J.N. Martin. 2008. "On the Use of Concepts and Principles for Improving Systems Engineering Practice". Proceedings of the 18th Annual International Council on Systems Engineering (INCOSE) International Symposium, 5-19 June 2008, Utrecht, The Netherlands.

Primary References

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

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

Churchman, C. W. 1968. *The Systems Approach and its Enemies*. New York, NY, USA: Dell Publishing.

Flood, R. L. 1999. *Rethinking the Fifth Discipline: Learning Within The Unknowable*. London UK: Routledge.

INCOSE. 2011. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBok 0.5 Wiki

Please note that in version 0.5, this article was titled "Systems Overview".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

What is a System?

This article forms part of the Systems Thinking Knowledge Area. It provides various perspectives on systems, including definitions, scope, and context. The basic definitions in this article are further expanded and discussed in Types of Systems and What is Systems Thinking?.

This article asks the following question: how can the many connotations of the word system (glossary) in everyday use be turned into a theoretical framework capable of supporting the application of systems thinking (glossary) in such a way that it can be usefully applied across all potential application domains and disciplines?

Systems Science View

The most basic ideas of a system whole can be traced back to the thinking of Aristotle and culminate in the works of the philosopher Hegel (M'Pherson 1974). Hegel's view of holism (glossary) states that an individual as part of an organic whole is fully realized only through his relationship to the whole. Humans have used this idea to help explain the relationships between abstract ideas by describing them as a system of related concepts, rules, or ideas. Some examples of such systems are the natural number system and political systems. The SEBoK uses the notion of a "system of ideas" to help present and explain system and systems engineering (glossary) (SE) knowledge.

A system is a combination of elements with some relationship(s) which can be best understood by considering them as a whole. The generally agreed upon systems science (glossary) definition of a system (glossary) considers a set of related elements in an environment (glossary) which contains related systems and conditions:

- A **closed system** (glossary) has no relationship(s) with its environment.
- An **open system** (glossary) shares both inputs (glossary) and outputs (glossary) with its environment across the boundary.

System elements may be conceptual organizations of ideals in symbolic form or real objects; e.g., people, data, physical artifacts, etc.

- **Abstract** systems contain only conceptual elements.
- **Concrete** systems contain at least two elements which are objects.

Many natural systems (glossary) and social systems (glossary) are formed through the inherent cohesion between elements, which leads them to form stable structures and hold those structures when disturbed. We can observe a process in nature where simple systems develop into more complex systems over time (Simons 1962).

Laszlo summarizes the properties which prompt considering a set of related elements as a whole (Laszlo 1972):

1. **Systemic state** (or wholeness): A property of the system elements and how they are related in the system structure which leads them to create a cohesive whole.
2. **Adaptive self-regulation**: All systems will tend to return to a defined steady state in response to external stimulus.
3. **Adaptive self-organization**: Some systems not only return to a previous steady state, but also reorganize to create new steady states which are more resistant to change.
4. **Holon**: Systems displaying characteristics two and three will tend to develop increasing hierarchical structures.

Thus, a collection of elements which tend to group together will form themselves into coherent wholes. 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. *The Architecture of Complexity* (Simon 1962) has shown that systems which evolve via a series of stable "hierarchical intermediate forms" will be more successful and adapt more quickly to environmental change.

Natural and social systems can be understood and managed through an understanding of this wholeness. Engineered systems (glossary) can be deliberately created to take advantage of this holism (glossary).

General System Theory (Bertalanffy 1968) considers the similarities between systems from different domains as a set of common system principles and concepts. The following system concepts, derived from *General System Theory,* relate directly to the definition of a system:

1. **Wholeness**: All systems are formed from groups of related elements into a whole with an observable shared identity in an environment. Many systems will form very stable relationships and tend to maintain their wholeness in the face of external change.
2. **Behavior**: All systems exhibit behaviors resulting from the interaction between elements. This may range from simple stimulus responses to conscious goal seeking actions.
3. **Control**: All systems have regulation and control mechanisms to guide their behaviors. For some systems, control may be a consequence of the relationships between its elements, while others may exhibit control mechanisms in a way guided by an identified purpose.
4. **Hierarchy**: All systems form hierarchical structures and will exhibit additional behaviors that emerge within the hierarchy due to interaction between system elements.
5. **Complexity**: At some levels of a hierarchy, systems are sufficiently complex that the ability of an observer to understand, interact with, or change them may be reduced.

While abstract systems are a useful way of expressing complex ideas and have value in systems science, it is the study of concrete systems which forms the majority of the systems knowledge of direct relevance to SE.

A concrete system contains a collection of elements and relationships which form coherent wholes. Systems exploit this cohesion to sustain themselves in the face of threats or environmental pressures, as well as to produce behaviors not possible by the elements alone. Some systems are created by people for specific reasons and will need to not only exist and survive, but also achieve necessary outcomes. At some point, the nature of the relationships between elements leads to complexity not associated with the elements themselves; this complexity can only be dealt with by considering the system as a whole.

The joining together of the various elements of a system can be related to either the structure of a system or to the behavior of a system. The structure of a system is its static existence, namely its elements and their relationships, whereas system behavior refers to the effect produced when an instance of the system is in operation. The actual behavior produced in operation leads to the fundamental property of emergence: "Whole entities exhibit properties which are meaningful only when attributed to the whole, not to its parts…" (Checkland 1999).

The system context (glossary) describes the system in its environment. The system context includes the boundaries of the system of interest (glossary) and the relationship of that system of interest to the environment in which it exists.

Understanding these system concepts and associated principles forms the basis of systems thinking. An expanded system of these system concepts (glossary) are described in more detail in System Concepts.

System Definitions – a Discussion

How is a system (glossary) defined in the SE literature? Systems engineers generally refer to their system of interest (SoI) (glossary) as "the

system," and their definitions of a "system" tend to characterize engineered systems (glossary). Two examples follow:

- "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 INCOSE Handbook (2011) generalizes this idea of an engineered 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."

However, engineered systems often find that their environment includes natural systems that don't follow the definitions of a "system" above in that they have not been defined to satisfy a requirement or come into being to satisfy a defined objective. These include such systems as the solar system if one's engineered system is an interplanetary spacecraft. This has led to more general definitions of a system following the systems science approach. For example, Aslaksen (2004) says a system consists of the following three related sets:

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

This definition of a system enables people to reason about numerous classes of dynamic systems that involve engineered, social, and natural systems.

Fundamental properties of a system described in the SE literature include togetherness, structure, behavior, and emergence. These properties provide one perspective on what a system is. "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). Hitchins (2009, 59-63) refers to this systems property as cohesion.

Natural systems and social systems often form part of the environment in which engineered systems need to exist. For a natural or social system, simply continuing to exist, and when appropriate, to adapt and grow, is sufficient. Man-made, engineered, and sociotechnical systems are created with a defined purpose (Hitchins 2009). Thus, these systems must not only be able to exist within their environment, but also do what is necessary to achieve their purpose.

System of Interest

As can be seen from the discussion above, most attempts to define the term "system" in SE either include assumptions about the system domain being considered, or are attempts to take a systems science view which risk becoming too abstract to be of practical use. A clear distinction is needed between defining "the system" to which a systems approach is applied and defining "systems" as an abstract idea which can be used to help understand complex situations.

The concept of a system helps make sense of the complexity (glossary) of the real world. This is done either by creating an abstract system to help explain complex situations, such as the real number system, by creating a standardized approach to common problems, such as the Dewey Decimal System, or by agreeing on a model of a new situation to allow further exploration, such as a scientific theory or conceptual system design. People use systems to make sense of complexity in an individual way and when they work together to solve problems.

In the systems approach (glossary), a number of relevant systems may be considered to fully explore problems and solutions and a given element may be included in several system views. Thus, it is less important that "the system" can be defined than it is that combinations of systems can be used to help achieve engineering or management tasks.

The idea of a system context (glossary) is used to define a system of interest (SoI) (glossary), and to identify the important relationships between it, the systems it works directly with, and the systems which influence it in some way. This engineered system context relates to the systems science ideas of an open, concrete system, although such a system may include abstract system elements. There are a number of more detailed system concepts (glossary) which the systems approach must also consider, such as static or dynamic, deterministic or non-deterministic, chaotic or homeostatic, complexity and adaptation, feedback and control, and more.

All applications of a systems approach (and hence of SE) are applied to an engineered systems context, and not to an individual system.

References

Works Cited

Alaksen, E. 2004. "A Critical Examination of the Foundations of Systems Engineering Tutorial". Paper presented at 14th Annual

International Council on Systems Engineering (INCOSE) International Symposium, Toulouse, France, 20-24 June 2004.

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.

Hitchins, D. 2009. "What Are the General Principles Applicable to Systems?" INCOSE *Insight*, 12(4): 59-63.

INCOSE. 2011. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.

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.

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

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.

M'Pherson, P. K. 1974. "A perspective on systems science and systems philosophy". *Futures*. 6(3):219-39.

Simon, H. A. 1962. "The Architecture of Complexity." Proceedings of the American Philosophical Society, 106(6) (Dec. 12, 1962): 467-482.

Primary References

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

INCOSE. 2011. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.

Additional References

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

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

Types of Systems

This article forms part of the Systems Thinking Knowledge Area. It provides various perspectives on system classifications and types of system, expanded from the definitions in What is a System?.

The article defines the idea of an engineered system (glossary). Three specific types of engineered system context are generally recognized in systems engineering: product system (glossary), service system (glossary) and enterprise system (glossary).

System Classification

A taxonomy is "a classification into ordered categories" (Dictionary.com 2011). Taxonomies are useful ways of organizing large numbers of individual items so their similarities and differences are apparent. Classification methods for systems (glossary) have been proposed over the past forty years, yet no standard classification system exists.

The early developers of General System Theory (glossary) developed systems classification which has been the starting point for much of the subsequent work. Bertalanffy (Bertalanffy 1968) divided systems into 9 types, including control mechanisms, socio-cultural systems, open systems, and static structures. These classification approaches tend to focus on making sense of the natural and social world around us.

As system science moved away from the theory of systems and began to consider how this theory might be used to provide practical approaches and tools classification approaches began to separate human-designed from non-human-designed systems or natural from man-made systems (Magee and de Weck 2004). While they provide some methods for classifying natural systems, their primary emphasis and value to the practice of systems engineer is in their classification method for human-designed or man-made systems.

A discussion of and guide to the various classification methods proposed by systems scientists is included in History of System Science.

Types of Systems

The modern world has numerous kinds of systems that influence daily life. Some examples include transport systems, solar systems, telephone systems, the Dewey Decimal System, weapons systems, ecological systems, space systems, and so on; indeed it seems there is almost no end to the use of the word "system" in today's society.

The classification methods above tend to classify systems either by the types of elements they contain or by their purpose.

A simple classification of system elements (glossary) is:

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

Peter Checkland (Checkland 1999) proposed a classification of systems based on purpose into five classes: natural systems, designed physical systems, designed abstract systems, human activity systems and transcendental systems.

From this three related system domains are identified as follows and shown in Figure 1:

- A Natural System (glossary) is one whose elements are wholly natural.

- A Social System (glossary) includes only humans as elements.

- An Engineered System (glossary) is a man-made aggregation which may contain physical, informational, human, natural and social elements; normally created for the benefit of people.

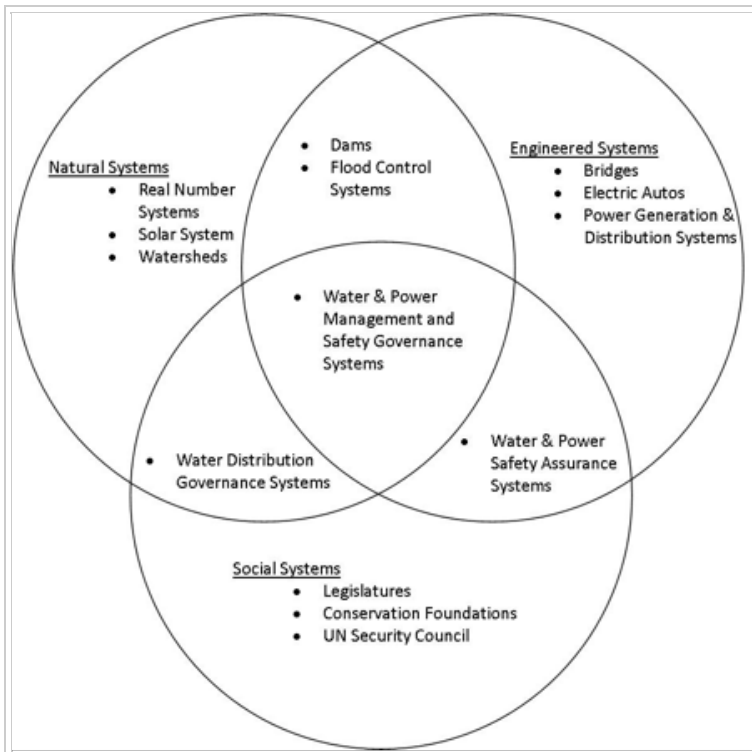These three types overlap to cover the full scope of real-world systems.

Figure 1.System Boundaries of Engineered Systems, Social Systems, and Natural Systems (Figure Developed for BKCASE)

Natural systems (glossary) are real world phenomena to which systems thinking is applied to help better understand what those systems do and how they do it. A truly natural system would be one that can be observed and reasoned about, but over which people cannot exercise control, such as the solar system.

Social systems (glossary) are purely human in nature, such as legislatures, conservation foundations, and the United Nations (UN) Security Council. These systems are human artifacts created to help people gain some kind of control over, or protection from, the natural world.

Note: from the definitions above Natural and Social Systems can contain only natural and human elements respectively. In reality, while it is possible to describe and reason about social systems, many of them rely on some interaction or relationship with engineered systems to fully realize their purpose and thus will form part of one or more engineered systems contexts.

Engineered Systems (glossary) may be purely technical systems, such as bridges, electric autos, and power generation. Engineered systems which contain technical and either human or natural elements, such as water and power management, safety governance systems, dams and flood control systems, water and power safety assurance systems are often called Sociotechnical Systems (glossary). The behavior of such systems is determined both by the nature of the engineered elements, and by their ability to integrate with or deal with the variability of the natural and social systems around them. The ultimate success of any engineered system is thus measured by its ability to contribute to the success of relevant sociotechnical system contexts.

Wherever system elements are combined into an engineered system, the complexity (glossary) of the resulting system will be increased. It is this increase in complexity that creates the need for a systems approach (glossary).

Groups of Systems

Systems can be grouped together to create more complex systems. In some cases it is sufficient to consider these systems as system elements in a higher level system, as part of a system hierarchy.

However, there are cases where the groupings of system produce an entity that must be treated differently from an integrated system. The most common groupings of systems that have characteristics beyond a single integrated system are Systems of Systems (SoS) (glossary) and Federations of Systems (FoS) (glossary).

Maier examined the meaning of System of Systems in detail and used a characterization approach which emphasizes the independent nature of the system element, rather than "the commonly cited characteristics of systems-of-systems (complexity of the component systems and geographic distribution) which are not the appropriate taxonomic classifiers" (Maier 1998, 268).

A more detailed discussion of the different system grouping taxonomies developed by systems science can be found in Groupings of

Systems.

Wherever independent systems are combined into groups the interaction between the systems adds a further complexity (glossary) in particular by constraining how the resulting system can be changed or controlled. This dimension of complexity leads to the management and control aspects of the systems approach (glossary).

Engineered Systems Classifications

Engineered systems:

1. Are created, used and sustained to achieve a purpose, goal or mission that is of interest to an enterprise (glossary), team (glossary), or an individual.
2. Require a commitment of resources for development and support.
3. Are driven by stakeholders (glossary) with multiple views on the use or creation of the system, or with some other stake in the system, its properties or existence.
4. Contain engineered hardware, software, people, services or a combination of these.
5. Exist within an environment that impacts the characteristics, use, sustainment and creation of the system.

Engineered systems typically:

1. Are defined by their purpose, goal or mission.
2. Have a life cycle (glossary) and evolution dynamics.
3. May include human operators (interacting with the systems via processes) and other natural components that must be considered in the design and development of the system.
4. Are part of a system-of-interest hierarchy.

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). A product or service is developed and supported by an individual, team, or enterprise. For example, express package delivery is a service offered worldwide by many enterprises, both public and private, both small and large; these services might use vehicle, communications or software products as needed.

The nature of engineered systems has changed dramatically over the past several decades from systems dominated by hardware (mechanical and electrical) to systems dominated by software. In addition systems that provide services, without delivering hardware or software, have become common as the need to obtain and use information has become greater. Recently organizations have become sufficiently complex that the techniques that were demonstrated to work on hardware and software have been applied to the engineering of enterprises.

Three specific types of engineered system context are generally recognized in systems engineering: product system (glossary), service system (glossary) and enterprise system (glossary).

Products and Product Systems

The word product (glossary) is define as "a thing produced by labour or effort; or anything produced" (Oxford English Dictionary). In a commercial sense a product is anything which is acquired, owned and used by an enterprise (hardware, software, information, personnel, an agreement or contract to provide something, etc.)

Product systems (glossary) are systems in which products are developed and delivered to the acquirer for the use of internal or external user. For product systems the ability to provide the necessary capability (glossary) must be defined in the specifications for the hardware and software, or the integrated system that will be provided to the acquiring enterprise (glossary).

Services and Service Systems

A service (glossary) can be simply defined as an act of help or assistance, or as any outcome required by one or more users which can be defined in terms of outcomes and quality of service without detail to how it is provided. e.g. transport, communications, protection, data processing, etc. Services are processes, performances, or experiences that one person or organization does for the benefit of another – such as custom tailoring a suit, cooking a dinner to order, driving a limousine, mounting a legal defense, setting a broken bone, teaching a class, or running a business' information technology infrastructure and applications. In all cases, service involves deployment of knowledge and skills (competences) that one person or organization has for the benefit of another (Lusch and Vargo 2006), often done as a single, customized job. In all cases, service requires substantial input from the customer or client (Sampson 2001) – for example, how can a steak be customized unless the customer tells the waiter how the customer wants the steak prepared? (business/marketing science definition)

A service system (glossary) is one that provides outcomes for a user without necessarily delivering hardware or software products to the service supplier. The hardware and software systems may be owned by a third party who is not responsible for the service. The use of service systems reduces or eliminates the need for acquirers to obtain capital equipment and software in order to obtain the capabilities needed to satisfy users.

Services have been part of the language of Systems Engineering for many years. The use of the term service system (glossary) in more recent times is often associated with information system, i.e:

> ...unique features that characterize services – namely, services, especially emerging services, are information-driven, customer-centric, e-oriented, and productivity-focused. (Tien and Berg 2003, 13)

A more detailed discussion of the system theory associated with service systems can be found in History of System Science.

Enterprises and Enterprise Systems

An enterprise (glossary) is one or more organizations or individuals sharing a definite mission, goals, and objectives to offer an output such as a product or service.

An enterprise system (glossary) consists of a purposeful combination (e.g., network) of interdependent resources (e.g., people, processes, organizations, supporting technologies, and funding) that interact with 1) each other (e.g., to coordinate functions, share information, allocate funding, create workflows, and make decisions), and 2) their environment(s), to achieve (e.g., business and operational) goals through a complex web of interactions distributed across geography and time (Rebovich and White 2011, 4, 10, 34-35).

Both product and service systems require an enterprise system to create them and an enterprise to use the product system to deliver services either internally to the enterprise or externally to a broader community.

According to Maier's definition, an enterprise would not necessarily be called a system of systems (SoS) (glossary) since the systems within the enterprise do not usually meet the criteria of operational and managerial independence. In fact, the whole purpose of an enterprise is to explicitly establish operational dependence between systems that the enterprise owns and/or operates in order to maximize the efficiency and effectiveness of the enterprise as a whole. Therefore, it is more proper to treat an enterprise system and an SoS as different types of things, with different properties and characteristics (DeRosa 2005).

Enterprise systems are unique, compared to product and service systems, in that they are constantly evolving, they rarely have detailed configuration controlled requirements, they typically have the goal of providing shareholder value and customer satisfaction, which are constantly changing and are difficult to verify, and they exist in a context (or environment) that is ill-defined and constantly changing.

Links to other areas of the SEBoK

SEBoK Part 4 Applications of Systems Engineering explores how systems engineering is applied differently in product, service, and enterprise systems. The notion of enterprises and enterprise systems permeates Part 5 Enabling Systems Engineering.

References

Works Cited

Bertalanffy, L. von. 1968. *General System Theory*. New York, NY, USA: Brazillier.

Checkland, P.B. 1999. *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley & Sons Ltd.

Dictionary.com, s.v. "Taxonomy," accessed September 7, 2011. Available at http://dictionary.reference.com/browse/taxonomy.

Encyclopedia Britannica, s.v. "Service Industry," accessed September 7, 2011. Available at http://www.britannica.com/EBchecked/topic/535980/service-industry.

DeRosa, J. K. 2005. "Enterprise Systems Engineering." Air Force Association, Industry Day, Day 1, 4 August 2005, Danvers, MA. Available at: https://www.paulrevereafa.org/IndustryDay/05/presentations/index.asp.

Lusch, R.F. and S. L. Vargo (Eds). 2006. *The service-dominant logic of marketing: Dialog, debate, and directions*. Armonk, NY: ME Sharpe Inc.

Magee, C.L. and O.L. de Weck. 2004. "Complex System Classification". Proceedings of the Fourteenth Annual International Symposium of

the International Council on Systems Engineering. June 20-24, 2004, Toulouse, France.

Maier, M., and E. Rechtin. 2009. *The Art of Systems Architecting*, 3rd Ed.. Boca Raton, FL, USA: CRC Press.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press.

Sampson, S.E. 2001. *Understanding Service Businesses*. New York, NY, USA: John Wiley.

Tien, J.M. and D. Berg. 2003. "A Case for Service Systems Engineering". Journal of Systems Science and Systems Engineering. 12(1): 13-38.

Primary References

Chang, C.M., 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley and Sons.

Checkland, P. B. 1999. *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley & Sons.

Magee, C. L., O.L. de Weck. 2004. "Complex System Classification." Proceedings of the 14th Annual International Council on Systems Engineering International Symposium, 20-24 June 2004, Toulouse, France.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press.

Tien, J.M. and D. Berg. 2003. "A Case for Service Systems Engineering". Journal of Systems Science and Systems Engineering. 12(1): 13-38.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

Comments from SEBoK 0.5 Wiki

This article was updated to include materials from 3 other 0.5 articles: The Product View of Engineered Systems, The Service View of Engineered Systems, and The Enterprise View of Engineered Systems. These articles have been folded into the current article and therefore are not included in 0.75. However, the review comments from 0.5 for the earlier articles can be seen below.

The Product View of Engineered Systems

The Service View of Engineered Systems

The Enterprise View of Engineered Systems

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

What is Systems Thinking?

This article forms part of the Systems Thinking Knowledge Area. The basis of systems thinking is to use the notion of system holism (glossary) as defined in What is a System? to better understand all types of system, in particular the engineered system (glossary) contexts described in Types of Systems.

The article considers the different ways in which systems thinking (glossary) has been defined and how it relates to the perspective of those defining it. The scope of systems thinking is the foundation for dealing with real world situations using a set of related system concepts.

System Theory

Many attribute the notion of systems thinking to the work of Aristotle due to his efforts in examining multiple discipline related aspects in

what is termed metaphysics. In modern times, the field of systems thinking has been evolving since the 1920's when the Austrian biologist von Bertalanffy introduced the idea of using biological analogues to systems in general (von Bertalanffy 1968). The work of system scientists such as von Bertalanffy has been the foundation for the creation of applied methodologies to deal with real world system problems and the development of these ideas has in turn influenced the scope of systems thinking. These approaches have been categorized as *hard* and *soft* approaches and are defined as follows:

- *Hard* thinking considers problems as "a difficult matter requiring solution, something hard to understand, accomplish or deal with" (Oxford English Dictionary).
- *Soft* thinking considers problems as "arising from everyday events and ideas, and may be perceived differently by different people. Such problems are not constructed by the investigator as are laboratory problems" (Checkland, 1981).
- *Hybrid* thinking considers combinations of hard and soft thinking as appropriate the the situation (Jackson 1989).

Churchman (1968) and others have also considered broader ethical, political and social questions related to management science with regards to the relative power and responsibility of the participants in system interventions.

A more detailed discussion of the most important movements in system theory can be found in the History of System Science article.

Definitions of Systems Thinking

Senge (1990, 6-7) defines systems thinking in his seminal work on learning organizations: "Systems thinking is a process of discovery and diagnosis – an inquiry into the governing processes underlying the problems we face and the opportunities we have." Senge further describes systems thinking as follows:

> *Systems thinking is a discipline for seeing wholes. It is a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static snapshots. It is a set of general principles- distilled over the course of the twentieth century, spanning fields as diverse as the physical and social sciences, engineering, and management... During the last thirty years, these tools have been applied to understand a wide range of corporate, urban, regional, economic, political, ecological, and even psychological systems. And systems thinking is a sensibility - for the subtle interconnectedness that gives living systems their unique character.* (Senge 2006, 68-69)

According to Senge and his colleagues (Senge 1994), a good systems thinker, particularly in an organizational setting, is someone who can see four levels operating simultaneously: events, patterns of behavior, systems, and mental models.

More recent chaos and complexity theories have also impacted the development of systems thinking, including the treatment of such concepts as emergence. According to Gharajedaghi:

> *Systems thinking is the art of simplifying complexity. It is about seeing through chaos, managing interdependency, and understanding choice. We see the world as increasingly more complex and chaotic because we use inadequate concepts to explain it. When we understand something, we no longer see it as chaotic or complex.* (Gharajedaghi 1999, 283)

The definition of systems thinking has evolved over time as advances have been made in systems theory. Some additional examples of systems thinking definitions are as follows:

- "Systems thinking requires the consciousness of the fact that we deal with models of our reality and not with the reality itself." (Ossimitz 1997, 1)
- "…what is often called "systemic thinking" …is …a bundle of capabilities, and at the heart of it is the ability to apply our normal thought processes, our common sense, to the circumstances of a given situation." (Dörner 1996, 199);
- "Systems thinking provides a powerful way of taking account of causal connections that are distant in time and space." (Stacey 2000, 9)

A broader perspective considers systems thinking to be one element in a wider system of holistic thinking. Kasser defines holistic thinking as follows: "...the combination of analysis [in the form of elaboration], systems thinking and critical thinking." (Kasser 2010)

Churchman believes that an important component of system knowledge comes from "others" or "enemies" outside the system: The systems approach begins when first you see the world through the eyes of another. Churchman (1968)

In this famous phrase, Churchman suggests that people can step outside a system they are in and mentally try to consider it through the lenses of other people's values. Churchman (1979), identified four main enemies of the systems approach (SA) namely: politics, morality, religion and aesthetics. In doing so he pioneered the use of the epistemological inquiry an approach which can be used to enlarge the scope of available knowledge and to increase our understanding of the underlying concepts of these issues.

To Churchman, the "enemies" of the SA provide a powerful way of learning about the systems approach, precisely because they enable the rational thinker to step outside the boundary of a system and to look at it. It means that systems thinkers are not necessarily involved within a system but are essentially involved in the "outside" of systems rationality. In Churchman's systems approach, decision makers or system

designers admit the freedom of individual thinking and our world view is very restricted and that it is possible to develop an alternative 'mode of thinking'.

For several years, Gene Bellinger has provided insight into the field of systems thinking via his popular web-site www.systems-thinking.org. He initiated a LinkedIn discussion group entitled *Systems Thinking World* and the wiki site www.systemswiki.org. Bellinger makes the following highly relevant observation concerning the field of systems thinking and its benefits on his website:

> *As I have continued to ponder the meaning of Systems Thinking over the years in conjunction with reading and many conversations it would seem that the understanding has evolved, thankfully. There was a time when I thought Systems Thinking was just a not very grown up version of System Dynamics though I have come to understand it is really far more encompassing. While the meaning continues to evolve my foundational belief remains solid. Systems Thinking will enable you to better understand the world around you and enable you to have more control over your life than any other subject you may undertake to study. For situations that concern you Systems Thinking will enable you to create approaches for dealing with these situations that are highly likely to produce the desired results while minimizing unexpected consequences.* (Bellinger 2011)

From these definitions, one can see systems thinking as both a set of founding ideas for the development of systems theories and practices and also as a pervasive way of thinking need by those developing and applying them. These ideas are further developed in the System Science and Systems Approach knowledge areas.

The Systems Thinking Paradox

In order to examine a particular group of elements in more detail, to understand, use or change them in some way, people are faced with an apparent "systems thinking paradox". One can only truly understand a system by considering all of its possible relationships and interactions, inside and outside of its boundary and in all possible future situations (of both system creation and life), but this makes it apparently impossible for people to understand a system or to predict all of the consequences of changes to it.

If this means that all possible system relationships and environmental conditions must be considered to fully understand the consequences of creating or changing a system, what useful work can be done?

In many ways this is the essence of all human endeavors, whether they be technical, managerial, social or political, the so called *known knowns* and *unknown unknowns*. The systems approach (glossary) is a way of tackling real world problems and making use of the tools of systems science to enable systems to be engineered and used. The key principle of the systems approach is that some of the detail of complex situations must be hidden to allow focus on changes to a system element. The impact must be considered of any changes that might be made across sufficient related system components to fit within acceptable commercial and social risks must be considered. Engineering and management disciplines deal with this by gathering as much knowledge as necessary to proceed at a risk level acceptable to the required need. The assessment of what is enough and how much risk to take can, to some extent, be codified with rules and regulations, and managed through processes and procedures; however, it is ultimately a combination of the skill and judgement of the individuals performing the work.

System of Interest

When humans observe or interact with a system, they allocate boundaries and names to parts of the system. This naming may follow the natural hierarchy of the system, but will also reflect the needs and experience of the observer to associate elements with common attributes of purposes relevant to their own. Thus, a number of systems of interest (SoI) (glossary) (Flood and Carson 1993) must be identified and they must be both relevant and include a set of elements which represent a system whole. This way of observing systems wherein the complex system relationships are focused around a particular system boundary is called **systemic resolution**.

A system context (glossary) is used to define a SoI and to capture and agree on the important relationships between it, such as the systems it works directly with and the systems which influence it in some way. When this approach is used to focus on part of a larger system, a balance of reductionism (glossary) and holism (glossary) is applied. This balance sits at the heart of a systems approach. A systems context provides the tool for applying this balance, and is thus an essential part of any systems approach and hence, of SE as well. Approaches for describing the context of the different types of engineered systems (glossary) are discussed in Engineered System Context within the Systems Approach knowledge area.

References

Works Cited

Bellinger, G. 2011. "Systems Thinking Definitions", Retrieved September 7, 2011 from http://www.systemswiki.org/index.php?title=Systems_Thinking_Definitions.

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

Churchman, C. W. 1968. *The Systems Approach*. Delacorte Press.

Churchman, C. West. 1979. "The Systems Approach and Its Enemies". New York: Basic Books.

Checkland, P. 1981. Systems Thinking, Systems Practice. New York, NY, USA: Wiley.

Dorner, H., and A. Karpati. 2008. "Mentored innovation in teacher training using two virtual collaborative learning environments." In *Beyond knowledge: The legacy of competence--meaningful computer-based learning environments*., eds. J. Zumbach, N. Schwartz, T. Seufert and L. Kester. Vol. VIII. New York, NY: Springer.

Flood, R. L. and E.R. Carson. 1993. Dealing with Complexity: An Introduction to the Theory and Application of Systems Science, 2nd ed. New York, NY, USA: Plenum Press.

Gharajedaghi, J. 1999. *Systems Thinking: Managing Chaos and Complexity: A platform for designing*. 1st ed. Woburn, MA: Butterworth-Heinemann.

Jackson, M. 1989. "Which Systems Methodology When? Initial Results from a Research Program." In: R Flood, M Jackson and P Keys (eds). *Systems Prospects: the Next Ten Years of Systems Research*. New York, NY, USA: Plenum.

Kasser, J. 2010. "Holistic thinking and how it can produce innovative solutions to difficult problems." Paper presented at 7th Bi-annual European Systems Engineering Conference (EuSEC), 24-27 May 2010, Stockholm, Sweden.

Ossimitz, G. The development of systems thinking skills using system dynamics modeling tools. in Universitat Klagenfurt [database online]. Klagenfurt, Austria, 1997 [cited November 12 2007]. Available from http://wwwu.uni-klu.ac.at/gossimit/sdyn/gdm_eng.htm.

Senge, P. M. 1990, 2006. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY, USA: Doubleday Currency.

Senge, P. M., A. Klieiner, C. Roberts, R. B. Ross, and B. J. Smith. 1994. *The Fifth Discipline Fieldbook: Strategies and tools for building a learning organization*. New York, NY: Crown Business.

Stacey, R. D., D. Griffin, and P. Shaw. 2000. *Complexity and management: Fad or radical challenge to systems thinking?*. London, U.K.: Routledge.

Primary References

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

Churchman, C. West. 1979. "The Systems Approach and its Enemies". New York: Basic Books.

Gharajedaghi, J. 1999. *Systems Thinking: Managing Chaos and Complexity: A platform for designing*. 1st ed. Woburn, MA: Butterworth-Heinemann.

Senge, P. M. 1990, 2006. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY, USA: Doubleday Currency.

Additional References

Jackson, M. 2003. *Systems Thinking: Creating Holisms for Managers*. Wiley; Chichester

Edson, R. 2008. *Systems Thinking. Applied. A Primer*. In: ASYST Institute (ed.). Arlington, VA: Analytic Services.

Lawson, H. 2010. *A Journey Through the Systems Landscape*, London, Kings College, UK.

< Previous Article | Parent Article | Next Article >

Comments from SEBok 0.5 Wiki

Please note that in version 0.5, this article was titled "Systems Thinking". This should not be confused with the new Systems Thinking article, which is the first knowledge area in Part 2.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

System Concepts

This article forms part of the Systems Thinking Knowledge Area.

This article defines a system of Systems Concepts (glossary), which describe specific knowledge that can be used to better understand any system, whatever its domain or technology, as an extension of the basic ideas of Systems Thinking.

These concepts have been synthesized from the work of a number of System Science researchers. Ackoff (1971) proposes a *System of "System Concepts"* to bring together the wide variety of concepts which have been proposed. Ackoff's concepts are written from a systems research perspective and can be a little abstract and hard to relate to practice. (Skyttner 2001) describes the main General Systems Thinking (GST) concepts proposed by a number of authors; (Flood and Carlson 1993) give a description of concepts as an overview of systems thinking; (Hitchins 2007) relates the concepts to systems engineering (glossary) practice.

Wholeness

The definition of system (glossary) includes the fundamental concepts of a set of elements which exhibit sufficient cohesion (glossary) (Hitchins 2007) or "togetherness" (Boardman and Sauser 2008) to form a "bounded" whole.

A system exists in an environment (glossary) which contains related systems and conditions:

- A Closed System (glossary) has no relationships with the environment.
- An Open System (glossary) shares inputs and outputs with its environment across the boundary.

System elements may be either conceptual organizations of ideals in symbolic form or real objects, e.g. people, data, physical artifacts, etc.

- Abstract system elements are conceptual.
- Concrete systems contain at least two elements which are objects.

Unless otherwise stated, the remaining concepts below apply to open, concrete systems.

Behavior

State

Any quality or property of a system element is called an Attribute (glossary). The State (glossary) of a system is a set of system attributes at a given time. A "system Event" describes any change to attributes of a system (or environment) and hence its state:

- Static - a single state exists with no events.
- Dynamic - multiple possible stable states exist. A stable state is one in which a system will remain until another event occurs.
- Homeostatic - system is static but its elements are dynamic. The system maintains its state by internal adjustments.

State can be monitored using "State Variables"; i.e., values of attributes which indicate the system state. The set of possible values of state variables over time is called the "state space". State variables are generally continuous, but can be modeled using a "Finite State Model" (or "State Machine").

- **Deterministic** systems have a one-to-one mapping of state variables to state space, allowing future states to be predicted from past states.
- **Non-Deterministic** systems have a many-to-many mapping of state variables; future state cannot be reliably predicted. This may be due to random changes in state, or because their structure is sufficiently complex that while they may be deterministic, it may take up different states due to very small (below our ability to measure) differences in starting state.

The latter case is one definition of a Chaotic (glossary) system; e.g., the stock market or weather are examples whose past states can be explained using deterministic reasoning, but whose future states cannot be predicted with certainty.

System Events

(Ackoff 1971) considers "change" to be how a system is affected by events, and "behavior" as the effect a system has upon its environment. Three kinds of "system change" are described; e.g., a system can "react" to a request by turning on a light, or "respond" to darkness by deciding to turn on the light or "act" to turn on the lights at a fixed time, randomly or with discernible reasoning.

System Behavior (glossary) is a change which leads to events in itself or other systems. Thus, action, reaction, or response may constitute behavior in some cases. Systems have varying levels of behavior.

Survival Behavior

Systems often act to continue to exist, behaving to sustain themselves in one or more alternative viable states. Many natural or social systems have this goal, either consciously or as a "self organizing" system, arising from the interaction between elements.

Entropy (glossary) is the tendency of systems to move towards disorder or disorganization. In physics, entropy is used to describe how "organized" heat energy is "lost" into the "random" background energy of the surrounding environment; e.g. 2nd Law of Thermodynamics.

A similar effect can be seen in engineered systems. What happens to a building or garden, which is left unused for any time? Entropy can be used as a metaphor for aging, skill fade, obsolescence, misuse, boredom, etc.

"Negentropy" describes the forces working in a system to hold off entropy. Homeostasis (glossary) is the biological equivalent of this, describing behavior which maintains a "steady state" or "dynamic equilibrium". Examples of this process in nature include human cells, which maintain the same function while replacing their physical content at regular intervals. Again this can be used as a metaphor for the fight against entropy, e.g. training, discipline, maintenance, etc.

(Hitchins 2007), describes the relationship between the viability of a system and the number of connections between its elements. In cybernetics, variety (glossary) is used to describe the number of different ways elements can be controlled, dependent on the different ways in which then can be combined. Hitchins' concept of "connected variety" states that stability of a system increases with its connectivity (both internally and with its environment).

Goal Seeking Behavior

Goals and Objectives

Engineered systems generally have reasons for existence beyond simple survival.

- A "goal" is a specific outcome which a system can achieve in a specified time
- An "objective" is a longer term outcome which can be achieved through a series of goals.
- An "ideal" is an objective which cannot be achieved with any certainty, but for which progress towards the objective has value.

Systems may be single goal seeking (perform set tasks), multi-goal seeking (perform related tasks) or reflective (set goals to tackle objectives or ideas). There are two types of goal seeking systems:

- Purposive (glossary) systems have multiple goals, with some shared outcome. Such a system can be asked/used to provide pre-determined outcomes, within an agreed time period. Such a system may have some freedom to choose how to achieve the goal. If it has memory it may develop processes describing the behaviors needed for defined goals. Most machines or software systems are purposive.

- Purposeful (glossary) systems are free to determine the goals needed to achieve an outcome. Such a system can be tasked to pursue objectives or Ideals over a longer time through a series of goals. Humans and sufficiently complex machines are purposeful.

Function

Ackoff defines Function (glossary) as outcomes which contribute to goals or objectives. To have a function, a system must be able to provide the outcome in two or more different ways (this is called **Equifinity**).

This view of function and behavior is common in systems science. In this paradigm all system elements have behavior of some kind, but to be capable of functioning in certain ways requires a certain richness of behaviors.

In most hard systems approaches (Flood and Carson 1993) a set of functions are described from the problem statement, and then associated with one or more alternative element structures. This process may be repeated until a system Component (glossary) (implementable combinations of function and structure) has been defined (Martin 1997). Here "function" is defined as a task or activity that must be performed to achieve a desired outcome; or as a "transformation" of inputs to outputs. This transformation may be:

- **Synchronous**, a regular interaction with a closely related system.

- **Asynchronous**, an irregular response to a demand from another system, often triggering a set response.

The behavior of the resulting system is then assessed. In this case behavior is seen as an external property of the system as a whole, and often described as analogous to human or organic behavior (Hitchins 2009).

Control

Cybernetics (glossary), the science of control, defines two basic control mechanisms:

- "Negative feedback", maintaining system state against a set objectives or levels.

- "Positive feedback", forced growth or contraction to new levels.

One of the main concerns of cybernetics is the balance between stability and speed of response. Cybernetics considers systems in three ways. A black-box system (glossary) view looks at the whole system. Control can only be achieved by carefully balancing inputs with outputs which reduces speed of response. A white-box system (glossary) view considers the system elements and their relationships; here control mechanisms can be imbedded into this structure giving more responsive control and associated risks to stability. A "grey-box" view sits between these two, with control exerted at the major sub-system level. Another useful control concept is that of a "meta-system", which sits over the system and is responsible for controlling its functions, either as a black-box or white-box. In this case, behavior arises from the combination of system and meta-system.

Systems behavior is influenced by variety (glossary), in particular in its control functions (Hitchins 2009). The law of requisite variety (Ashby 1956) states that a control system must have at least as much variety as the system it is controlling. The effect of variety on system behavior can often be seen in the relationship between:

- *Specialization*, the focus of system behavior to exploit particular features of its environment.

- Flexibility (glossary), the ability of a system to adapt quickly to environmental change.

Effectiveness, Adaptation and Learning

Systems Effectiveness (glossary) is a measure of the system's ability to perform the functions necessary to achieve goals or objectives. Ackoff (1971) defines this as the product of the number of combinations of behavior to reach a function and the efficiency of each combination.

Hitchins (2007) describes effectiveness as a combination of **performance** (how well a function is done in ideal conditions), **availability** (how often the function is there when needed) and **survivability** (how likely is it that the system will be able to use the function fully).

An Adaptive (glossary) System is one that is able to change itself or its environment if its effectiveness is insufficient to achieve its current or future goals or objectives. Ackoff defines four types of adaptation, changing the environment or the system, in response to internal or external factors.

A system may also "learn", improving its effectiveness over time, without any change in state or goal.

Hierarchy, Emergence and Complexity

System behavior is related to combinations of element behaviors. Most systems exhibit "increasing variety"; i.e., they have behavior resulting from the combination of element behaviors. The term "synergy", or weak emergence, is used to describe the idea that "the whole is greater than the sum of the parts". While this is generally true, it is also possible to get **reducing variety** in which the whole function is less than the

sum of the parts.

Open systems tend to form hierarchies (glossary) of coherent system elements, or sub-systems. A natural system hierarchy is a consequence of wholeness, with strongly cohesive elements grouping together forming structures which reduce complexity and increase robustness (Simons 1962). Socio-technical systems form "control hierarchies", with systems at a higher level having some ownership of control over those at lower levels. Hitchins (2009) describes how systems form "preferred patterns" which can be used to the enhanced stability of interacting systems hierarchies.

Systems can be viewed using "systemic resolution". A system is characterized by its behavior in a wider system or environment and considered in detail as a set of sub-system structures and functions. This system description is focused at a particular level of resolution. The level of resolution is changed by focusing on the wider system or on one of the sub-systems. While this allows a focus on a given system-of-interest, the holistic view of the wider system and environment must not be lost.

Looking across a hierarchy of systems generally reveals increasing complexity (glossary) at the higher level, relating to both the structure of the system and how it is used. The terms emergence (glossary) and emergent properties (glossary) are generally used to describe behaviors emerging across a complex system hierarchy. These last two ideas are fundamental to engineered systems and the Systems Approach.

Practical Consideration

How do people know how to apply system concepts to an Engineered System?

All SE texts, for example (INCOSE 2011), describe processes and activities based upon the application of systems thinking (glossary) to an engineered system (glossary) context. Often the link between SE and systems thinking is embedded in the details and not clear to those applying the processes. Systems Approach and its links to the rest of the SEBoK provide a guide to this linkage.

Hitchins (2007) proposes a set of necessary and sufficient questions to help ensure all systemic issues have been considered when assessing an existing or proposed system description. These questions attempt to relate system concepts to high level concerns more relevant to Systems Engineers.

Hitchins "Generic Reference Model" asks questions under six heading based on these concepts, related to its function (what it does) and form (what the system is). Each question expanded into a number of more details questions related to system concepts:

1. **Function: Mission Management**. How does the system deal with setting of objectives and plans; control and behavior; relationships with other systems?
2. **Function: Viability Management**. How does the system deal with state, survival, maintenance and repair?
3. **Function: Resource Management**. How does the system deal with exchange of information, energy, people, material, finance, etc. with its environment?
4. **Form: Structure**. Are system boundaries, sub-elements, connections and relationships understood?
5. **Form: Influence**. Are the systems' wider relationships and influences with its environment understood?
6. **Form: Potential**. Has the structure to achieve all objectives been considered, how often and well those objectives must be achieved, and how faults or failures are addressed?

References

Works Cited

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

Ackoff, R.L. 1971. "Towards a System of Systems Concepts". *Management Science*. 17(11).

Ashby, W R. 1956. "Chapter 11". *Introduction to Cybernetics*. London, UK: Wiley.

Flood, R.L. and E.R. Carson. 1993. *Dealing With Complexity: An Introduction to the Theory and Application of Systems Science*. New York, NY, USA: Plenum Press.

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology*. Hoboken, NJ, USA: John Wiley and Sons.

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?" INCOSE *Insight*. 12(4): 59-63.

Martin J. N. 1997. *Systems Engineering Guidebook*. Boca Raton, FL, USA: CRC Press.

Skyttner, L. 2001. *General Systems Theory: Ideas and Applications*. Singapore: World Scientific Publishing Co. p. 53-69.

Simon, H. A. 1962. "The Architecture of Complexity." *Proceedings of the American Philosophical Society*. 106(6) (Dec. 12, 1962): 467-482.

Primary References

Ackoff, R.L. 1971. "Towards a System of Systems Concepts." *Management Science*. 17(11).

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?" INCOSE *Insight*. 12(4): 59-63.

Additional References

Edson, Robert. 2008. *Systems Thinking. Applied. A Primer*. Arlington, VA, USA: Applied Systems Thinking Institute (ASysT), Analytic Services Inc.

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology*. Hoboken, NJ, USA: John Wiley & Sons.

Jackson, S., D. Hitchins, and H. Eisner. 2010. "What is the Systems Approach?" INCOSE *Insight*. 13(1) (April 2010): 41-43.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Waring, A. 1996. "Chapter 1." *Practical Systems Thinking*. London, UK: International Thomson Business Press.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "Overview of System Concepts".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

System Science

This Knowledge Area (KA) provides a guide to the major developments in Systems Science (glossary) which is an interdisciplinary field of science that studies the nature of complex systems in nature, society, and science. Grounded in systems thinking, and based on theory and practice, it aims to develop interdisciplinary foundations applicable in a variety of areas, such as engineering (glossary), biology, medicine and social sciences. This knowledge is not specific to systems engineering (glossary), but is part of a wider systems body of knowledge. Not all system knowledge is captured here - only those aspects relevant to the SEBoK.

Topics

The topics contained within this knowledge area include:

- History of System Science
- System Methodology
- Groupings of Systems
- Complexity
- Emergence

Machine vs System Age Thinking

Many attribute the notion of thinking about the whole to the Greek philosophers, exemplified by the work of Aristotle in examining multiple discipline related aspects in what is termed metaphysics. The explosion of knowledge in the natural and physical sciences during the Enlightenment of the 18th and 19th centuries made the move away from this natural philosophy approach to the creation of specialist disciplines inevitable. The only way for science to advance was for scientists to become expert in a narrow field of study. As disciplines emerged they created their own models and views of reality, which become increasingly specialized and associated with a field of study. The creation of educational structures to pass on this knowledge to the next generation of specialists perpetuates the fragmentation of knowledge into the present day (M'Pherson 1973).

Along with this increasing specialization of knowledge and education, the majority of western scientific study in the 19th century was based upon Descartes' notion of reductionism (glossary) and closed system (glossary), sometimes call **Machine Age**, thinking (Flood 1999). This approach forms models based on the study of things in isolation and the establishment of rules on how they relate to each other. Unfortunately, this also led to a rational science movement, popularized in (Popper 1972), which rejects any phenomena which do not fit with this rational view as not worthy of study.

While these ideas of specialist knowledge and rational analysis have provided a useful model through which a vast amount of scientific knowledge has been gained, they can also be a barrier to our ability to gain knowledge across disciplines and outside of the closed system view. The systems movement has its roots in two areas of science: the biological-social sciences, and a mathematical-managerial base stemming first from cybernetics and later from organizational theory. Both of which have developed around an open system (glossary) and systemic thinking approach.

Open system theory considers an organism as a complex entity composed of many parts with an overall integrity, co-existing in an environment. In an open system the organism's structure is maintained, or adapts, through a continual exchange of energy and information with its environment.

Development of System Theory

Systems Thinking (glossary) is an approach to understanding or intervening in systems, based on the principles and concepts of systems. The Systems Thinking KA offers definitions of systems thinking and the systems theory which supports it.

The development of these theoretical ideas to a point where they can be considered part of the cannon of systems thinking is like any other branch of science - not a straight forward or linear process. General System Theory (glossary) (GST) (von Bertalanffy, 1968) enables comparisons between systems that rely on different technologies, judging the goodness or completeness of a system, and developing domain-independent systems approaches which can form the basis of disciplines such as Systems Engineering. While many researchers and practitioners have created GST concepts, these tend to be a stepping stone to theories and approaches. This situation is made worse by the variety of domains and disciplines in which systems research is conducted and reported.

While the System of System Concepts presented in the Systems Thinking Knowledge Area is a powerful set of ideas for better understanding all kinds of systems, it is neither rigorous nor complete.

This Knowledge area describes the most important movements in Systems Science and presents a guide to the overlapping and sometimes contradictory theories it has created and used.

Three specific areas of systems research have been highlighted and discussed in more detail as of particular interest to the systems engineering community:

- Grouping of systems and the idea of a system of systems
- The different aspects of system complexity (glossary) and how they are affect a systems approach (glossary)
- The different views of emergence (glossary) and how they affect a systems approach (glossary)

References

Works Cited

Ackoff, R.L. 1971. "Towards a System of Systems Concepts". *Management Science*. 17(11).

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

Flood, R.L. 1999. *Rethinking the Fifth Discipline: Learning within the Unknowable*. London, UK: Routledge.

M'Pherson, P, K. 1974. "A Perspective on Systems Science and Systems Philosophy." *Futures* 6(3) (June 1974): 219-239.

Popper, K. R. 1979. *Objective Knowledge*, 2nd edition. Oxford, UK: Oxford University Press.

Primary References

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

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology*. Hoboken, NJ, USA: John Wiley & Sons.

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?" *Insight*. 12(4).

Page, S.E. 2009. *Understanding Complexity*. The Great Courses. Chantilly, VA, USA: The Teaching Company.

Sheard, S. A. and A. Mostashari. 2008. "Principles of Complex Systems for Systems Engineering." *Systems Engineering*. 12(1): 295-311.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "System Concepts".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

History of System Science

This article is part of the System Science Knowledge Area. It gives some of the history and detail of the development of a number of different apsects of systems research and practice. A second article describing System Methodology in more detail accompanies this overview. Some of these ideas form the basic theory and methods that are used to define Systems Thinking.

Engineering can be defined as "*the application of scientific principles to practical ends*" (Oxford English Dictionary). We would expect engineering disciplines which take a Systems Approach (glossary) (such as systems engineering) to be based upon a Systems Science (glossary). The term science implies a well defined branch of knowledge, with a clearly recorded and coherent historical development. This is yet not the case for systems science, which has a fragmented history. For instance, some fundamental concepts now used in systems science have been present in other disciplines for many centuries, while equally fundamental concepts have independently emerged as recently as 40

or so years ago (Flood and Carson 1993).

The Development of Systems Research

The following overview of the evolution of systems science is broadly chronological, but also follows the evolution of different paradigms in system theory.

Open Systems and General Systems Theory

Ludwig von Bertalanffy developed a research approach based on **Open System Theory** (Bertalanffy 1950). He was one of a number of natural scientists who realized that the **reductionist closed system** approach could not be used to explain the behavior of an organism in its environment.

General System Theory (glossary) (GST), attempts to formulate principles relevant to all open systems (Bertalanffy 1968). GST is based on the idea that correspondence relationships (**homologies**) exist between systems from different disciplines. Thus, knowledge about one system should allow us to reason about other systems. Many of the generic system concepts come from the investigation of GST.

GST also implies a scientific approach, with identified laws and generalized theory to unify all science. Bertalanffy was co-founder, along with **Kenneth Boulding** (economist), **Ralph Gerard** (physiologist) and **Anatol Rapoport** (mathematician), of the Society for **General Systems Research** in 1957. This group is considered by many to be the founders of **System Age Thinking** (Flood 1999).

Cybernetics

Cybernetics (glossary) was defined by **Wiener**, **Ashby** and others as the study and modeling of communication, regulation, and control in systems (Ashby 1956; Wiener 1948). Cybernetics studies the flow of information through a system and how information is used by the system to control itself through feedback mechanisms. Early work in cybernetics in the 1940s was applied to electronic and mechanical networks, and was one of the disciplines used in the formation of early systems theory. It has since been used as a set of founding principles for all of the significant system disciplines.

Operations Research and Organizational Cybernetics

Operations Research (glossary) (OR) considers the use of technology by an organization. It is based on mathematical modeling and statistical analysis to optimize decisions on the deployment of the resources under an organization's control. It arises from military planning techniques developed during World War II.

Operations Research and Management Science (ORMS) was formalized in 1950 by **Ackoff** and **Churchman** applying the ideas and techniques of OR to organizations and organizational decisions (Churchman et al 1950).

Stafford Beer was one of the first to take a cybernetics approach to organizations (Beer 1959). For Beer the techniques of ORMS are best applied in the context of an understanding of the whole system. Beer also developed a **Viable Systems Model** (Beer 1979), which encapsulates the effective organization needed for a system to be Viable (glossary) (to survive and adapt in its environment).

Work in cybernetics and ORMS consider the mechanism for communication and control in complex systems, and particularly in organizations and management sciences. They provide useful approaches for dealing with operational and tactical problems within a system, but do not allow consideration of more strategic organizational problems (Flood 1999).

Hard and Soft Systems Thinking

Action Research is an approach first described by Kurt Lewin as a reflective process of progressive problem solving in which reflection on action leads to a deeper understanding of what is going on and to further investigation (Lewin 1958).

Peter Checkland's action research program in the 1980's led to an Interpretative-based Systemic Theory which seeks to understand organizations by not only observing the actions of people, but by building understandings of the cultural context, intentions and perceptions of the individuals involved. Checkland, himself starting from a systems engineering perspective, successively observed the problems in applying a systems engineering approach to the more fuzzy, ill-defined problems found in the social and political arenas (Checkland 1975). Thus he introduced a distinction between hard systems and soft systems:

Hard systems (glossary) views of the world are characterized by the ability to define purpose, goals, and missions that can be addressed via engineering methodologies in attempting to, in some sense, "optimize" a solution.

In Hard System approaches the problems may be complex and difficult, but they are known and can fully expressed by the investigator. Such problems can be solved by selecting from the best available solutions (possibly with some modification or integration to create an optimum

solution). In this context, the term "systems" is used to describe real world things, a solution system is selected, created and then deployed to solve the problem.

Soft systems (glossary) views of the world are characterized by extremely complex, problematical, and often mysterious phenomena for which concrete goals cannot be established and which require learning in order to make improvement. Such systems are not limited to the social and political arenas and also exist within and amongst enterprises where complex, often ill-defined patterns of behavior are observed that are limiting the enterprise's ability to improve.

Soft System approaches reject the idea of a single problem and consider **problematic** situations in which different people will perceive different issues depending upon their own viewpoint and experience. These problematic situations are not solved, but managed through interventions which seek to reduce "discomfort" among the participants. The term system is used to describe systems of ideas, conceptual systems which guide our understanding of the situation or help in the selection of intervention strategies.

These three ideas of "problem vs. problematic situation"; "solution vs. discomfort reduction" and "the system vs. systems understanding" encapsulate the differences between hard and soft approaches (Flood and Carson, 1993).

Hard and Soft thinking became the dominant paradigms for Systems Science investigations in the 1980's and early 90'2, leading to the creation of a number of System Methodologies

Critical Systems Thinking

The development of a range of hard and soft methods naturally leads to the question of which method to apply when (Jackson 1989). Critical Systems Thinking (glossary) (CST) or **Critical Management Science** Jackson (Jackson 1985) attempts to deal with this question.

The word **critical** is used in two ways. Firstly, critical thinking considers the limits of knowledge and investigates the limits and assumptions of hard and soft systems, as discussed in the above sections. From this comes frameworks and meta-methodology for when to apply different methods and **Multi-Methodology** approaches which recognize the value of combining techniques from several hard or soft methods as needed (Mingers and Gill 1997).

Churchman (Churchman, 1979) and others have also considered broader ethics political and social questions related to management science, with regards to the relative power and responsibility of the participants in system interventions. The second aspect of **critical** thinking considers the ethical, political and coercive dimension in Jackson's SOSM framework (Jackson 2003) and the role of system thinking in society.

Classification of Systems

Many systems researchers have attempted to produce a useful classification taxonomy, giving some logic to the mapping of Systems Thinking to the real world.

Kenneth Boulding, one of the founding fathers of General System Theory (glossary), developed a systems classification which has been the starting point for much of the subsequent work. (Boulding 1956). He classifies systems into 9 types:

1. Structures (Bridges)
2. Clock works (Solar system);
3. Controls (Thermostat);
4. Open (Biological cells);
5. Lower organisms (Plants)
6. Animals (Birds)
7. Man (Humans)
8. Social (Families)
9. Transcendental (God).

Bertalanffy (Bertalanffy 1968) divided systems into 9 types, including control mechanisms, socio-cultural systems, open systems, and static structures. Miller (Miller 1986) offered cells, organization, and society among his 7 system types. Other similar categorizations of system types can be found in (Aslaksen 1996), (Blanchard 2005) and (Giachetti 2009).

These approaches also highlights some of the subsequent issues of classification. Boulding implies that physical structures are closed and natural or social ones are open. He also separates humans from animals. (Hitchins 2007).

In addition to making the distinction between Hard and Soft systems, Peter Checkland divided systems into five classes: **natural systems, designed physical systems, designed abstract systems, human activity systems** and **transcendental systems**. The first two classes are self explanatory.

- Designed abstract systems – These systems do not contain any physical artifacts but are designed by humans to serve some explanatory purpose.

- Human activity systems (glossary) – These systems are observable in the world of innumerable sets of human activities that are more or less consciously ordered in wholes as a result of some underlying purpose or mission. At one extreme is a system consisting of a human wielding a hammer. At the other extreme lies international political systems.

- Transcendental systems – These systems that go beyond the aforementioned four systems classes, systems beyond knowledge. Checkland refers to these five systems as comprising a "systems map of the universe". (Checkland 1999, p.111)

Magee and de Weck (Magee and de Weck 2004) examine many possible methods that include: degree of complexity, branch of the economy that produced the system, realm of existence (physical or in thought), boundary, origin, time dependence, system states, human involvement / system control, human wants, ownership and functional type, such as (Maier 2009),(Paul 1998) and (Wasson 2006). They conclude by proposing a functional classification method that sorts systems by their **process**: transform, transport, store, exchange, or control and by the **entity that they operate on**: matter, energy, information and value.

Development of Service Science and Service Systems Engineering

The world economies have transitioned over the past few decades from manufacturing economies that provide goods -- to service based economies. Along with this transition there has been a new application of systems thinking, The disciplines of service science and Service engineering, built on principles of systems thinking but applied to the development and delivery of service systems (glossary) have developed to support this expansion.

One of the first articles to define the field of service science and service engineering was published in 2006 in the *Communications of the ACM*, titled "Service Systems, Service Scientists, SSME, and Innovation" (Maglio, et al. 2006). This article was written to "establish a new academic discipline and new profession" (Maglio, et al. 2006, 8).

Published in 2008, *Service Science, Concepts, Technology, Management*, by Harry Katzan (2008) claims to be the first book to define the newly emerging field of service science: "Service science is defined as the application of scientific, engineering, and management competencies that a service-provider organization performs that creates value for the benefit of the client of customer" (Katzan 2008, vii).

Within the past few years a number of other books have appeared that address service science and service systems engineering based on the foundation laid by the earlier IBM work on service science, management, engineering and design, also known as SSMED:

> *Service science has grown into a global initiative involving hundreds of organizations and thousands of people who have begun to create service innovation roadmaps and to invest in expanding the body of knowledge about service systems and networks.* (Spohrer and Maglio 2010, 3)

Service systems E]engineering is described more fully in the Service Systems Engineering knowledge area (KA) in Part 4 of the SEBoK.

Systems Methodologies

Much of the work of the Systems Science community has been around the creation of System Methodologies. These describe structured approaches to problem understanding and/or resolution making use of some of the concepts of Systems Thinking. These methodologies are generally associated with a particular System paradigm, or way of thinking, about how Systems Thinking should be applied.

These paradigms arise from the different system movements discussed above. The most widely used groups of methodologies are as follows:

1. Hard System (glossary) methodologies, (Checkland 1978), set out to select an efficient means to achieve a predefined and agreed end.
2. Soft System (glossary) methodologies, (Checkland 1999), are interactive and participatory approaches to assist groups of diverse participants to alleviate a complex, problematic situation of common interest.
3. Critical Systems Thinking (glossary) methodologies,(Jackson 1985), attempts to provide a framework in which appropriate hard and soft methods can be applied as appropriate to the situation under investigation.

More detailed discussion of some of the methods most often used in Systems Engineering can be found in the associated System Methodology topic.

References

Works Cited

Ashby, W. R. 1956. *Introduction to Cybernetics*. London, UK: Methuen.

Aslaksen, E.W. 1996. *The Changing Nature of Engineering*. New York, NY, USA: McGraw-Hill.

Beer, S. 1959. *Cybernetics and Management*. London, UK: English Universities; New York: Wiley and Sons.

Beer, S. 1979. *The Heart of the Enterprise*. Chichester, UK: Wiley.

Bertalanffy, L. von. 1950. "The Theory of Open Systems in Physics and Biology". *Science*, New Series, 111(2872) (Jan 13): 23-29

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

Blanchard, B.S., and W.J. Fabrycky. 2005. *Systems Engineering and Analysis,* 4th ed. Prentice-Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice-Hall.

Boulding, K. 1956 "General Systems Theory: Management Science, 2, 3 (Apr. 1956) pp.197-208; reprinted in General Systems, Yearbook of the Society for General Systems Research, vol. 1, 1956.

Checkland, P. 1975. "The Origins and Nature of "Hard" Systems Thinking." *Journal of Applied Systems Analysis*, 5(2): 99-110.

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

Churchman, C.W. 1968. *The Systems Approach*. New York, NY, USA: Dell Publishing.

Churchman, C.W., R.L. Ackoff. and E.L. Arnoff. 1950. *Introduction to Operations Research*. New York, NY, USA: Wiley and Sons.

Flood, R.L. 1999. *Rethinking the Fifth Discipline: Learning within the Unknowable*. London, UK: Routledge.

Flood, R.L. and E.R. Carson. 1993. *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science,* 2nd ed. New York, NY, USA: Plenum Press.

Giachetti, R.E. 2009. *Design of Enterprise Systems: Theory, Architectures, and Methods*. Boca Raton, FL, USA: CRC Press.

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology*. Hoboken, NJ, USA: Wiley.

Jackson, M. 1985. "Social Systems Theory and Practice: the Need for a Critical Approach." *International Journal of General Systems*. 10: 135-151.

Jackson, M. 1989. "Which Systems Methodology When? Initial Results from a Research Program." In: R Flood, M Jackson and P Keys (eds). *Systems Prospects: the Next Ten Years of Systems Research*. New York, NY, USA: Plenum.

Jackson, M. 2003. *Systems Thinking: Creating Holisms for Managers*. Chichester, UK: Wiley.

Katzan, H. 2008. *Service Science*. Bloomington, IN, USA: iUniverse Books.

Lewin, K. 1958. *Group Decision and Social Change*. New York, NY, USA: Holt, Rinehart and Winston. p. 201.

Maier, M. W. 1998. "Architecting Principles for Systems-of-Systems". *Systems Engineering*, 1(4): 267-84.

Magee, C. L., O.L. de Weck. 2004. "Complex System Classification." Proceedings of the 14th Annual International Council on Systems Engineering International Symposium, 20-24 June 2004, Toulouse, France.

Miller, J.G. 1986. "Can Systems Theory Generate Testable Hypothesis?: From Talcott Parsons to Living Systems Theory." *Systems Research*. 3: 73-84.

Mingers, J. and A. Gill. 1997. Multimethodology: Theory and Practice of Combining Management Science Methodologies. Chichester, UK: Wiley.

Maglio P., S. Srinivasan, J.T. Kreulen, and J. Spohrer. 2006. "Service Systems, Service Scientists, SSME, and Innovation." *Communications of the ACM*. 49(7) (July).

Paul, A. S. 1998. "Classifying Systems." Proceedings of The Eighth Annual International Council on Systems Engineering International Symposium, 26-30 July, 1998, Vancouver, BC, Canada.

Spohrer, J. and P. Maglio. 2010. "Service Science: Toward a Smarter Planet." In *Introduction to Service Engineering*. Ed. G Salvendy and W Karwowski. 3-30. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Wasson, C. S. 2006. *System Analysis, Design and Development*. Hoboken, NJ, USA: John Wiley and Sons.

Wiener, N. 1948. *Cybernetics or Control and Communication in the Animal and the Machine*. Paris, France: Hermann & Cie Editeurs; Cambridge, MA, USA: The Technology Press; New York, NY, USA: John Wiley & Sons Inc.

Primary References

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

Chang, C.M., 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley and Sons.

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

Flood, R. L. 1999. *Rethinking the Fifth Discipline: Learning within the Unknowable*. London, UK: Routledge.

Jackson, M. 1985. "Social Systems Theory and Practice: the Need for a Critical Approach." *International Journal of General Systems* 10: 135-151.

Magee, C. L., O.L. de Weck. 2004. "Complex System Classification." Proceedings of the 14th Annual International Council on Systems Engineering International Symposium, 20-24 June 2004, Toulouse, France.

Additional References

Ackoff, R.L. 1981. *Creating the Corporate Future*. New York, NY, USA: Wiley and Sons.

Bowler, D.T. 1981. *General Systems Thinking: Its Scope and Applicability*. Amsterdam: The Netherlands: Elsevier.

Boulding, K.E. 1996. *The World as a Total System*. Beverly Hills, CA, USA: Sage Publications.

Laszlo, E. (ed). 1972. *The Relevance of General Systems Theory*. New York, NY, USA: George Brazillier.

Skyttner, L. 1996. *General Systems Theory - An Introduction*. Basingstoke, UK: Macmillan Press.

Warfield, J.N. 2006. *An Introduction to Systems Science*. Singapore: World Scientific Publishing Co. Pte Ltd. Chang, C.M., 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley and Sons.

Chesbrough, H. 2004. "A failing grade for the innovation academy." *Financial Times* (September 24):1, cited by Spohrer, J. and P. Maglio. 2010. "Service Science: Toward a Smarter Planet." In *Introduction to Service Engineering*. Ed. G Salvendy and W Karwowski. 3-30. Hoboken, NJ, USA: John Wiley & Sons, Inc.

IBM. 2005. IBM Research: Services Science: A New Academic Discipline? www.almaden.ibm.com/asr/resources/facsummit.pdf (accessed September 12, 2011), cited by Spohrer, J. and P. Maglio. 2010. "Service Science: Toward a Smarter Planet." In *Introduction to Service Engineering*. Ed. G Salvendy and W Karwowski. 3-30. Hoboken, NJ, USA: John Wiley & Sons, Inc.

IfM and IBM. 2008. "Succeeding through service innovation: A service perspective for education, research, business and government." University of Cambridge Institute for Manufacturing. Cambridge, UK, cited by Spohrer, J. and P. Maglio. 2010. "Service Science: Toward a Smarter Planet." In *Introduction to Service Engineering*. Ed. G Salvendy and W Karwowski. 3-30. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Lusch, R.F. and S. L. Vargo (Eds). 2006. *The service-dominant logic of marketing: Dialog, debate, and directions*. Armonk, NY: ME Sharpe Inc.

Maglio P., S. Srinivasan, J.T. Kreulen, and J. Spohrer. 2006. "Service Systems, Service Scientists, SSME, and Innovation." *Communications of the ACM*. 49(7) (July).

Salvendy, G. and W. Karwowski (eds.). 2010. *Introduction to Service Engineering*. Hoboken, NJ, USA: John Wiley and Sons.

Sampson, S.E. 2001. *Understanding Service Businesses*. New York, NY, USA: John Wiley.

Spohrer, J. 2008. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline-Outline & References." *International Journal of Information Systems in the Service Sector*. 1(3) (May).

Spohrer, J. and P. P. Maglio. 2008. "The emergence of service science: Toward systematic service innovations to accelerate co-creation of value." *Production and Operations Management* 17 (3): 238-246, cited by Spohrer, J. and P. Maglio. 2010. "Service Science: Toward a

Smarter Planet." In *Introduction to Service Engineering*. Ed. G Salvendy and W Karwowski. 3-30. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Spohrer, J. and P. Maglio. 2010. "Service Science: Toward a Smarter Planet." In *Introduction to Service Engineering*. Ed. G Salvendy and W Karwowski. 3-30. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Tien, J.M. and D. Berg. 2003. "A Case for Service Systems Engineering". *Journal of Systems Science and Systems Engineering*. 12(1): 13-38.

Vargo, S.L. and R.F. Lusch. 2004. "Evolving to a new dominant logic for marketing." *Journal of Marketing* 68 (January 2004): 1-17.

Zeithaml, V. A., M. J. Bitner, and D. D. Gremler. 2005. *Services Marketing: Integrating Customer Focus Across the Firm*. 4th ed. New York, NY, USA: McGraw-Hill, quoted in Chang, C.M., 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley and Sons.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "System Science".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

System Methodology

This article is part of the System Science Knowledge Area. It gives some of the history and detail of the development of a number of different Systems Methodologies by the system science community. Some of these ideas form the basic theory and methods are used to define Systems Thinking.

Introduction to Methodology

A paradigm is distinct from a theory. A paradigm is a theoretical framework. Kuhn (1962) first popularized the term.

A methodology is a body of tools, procedures, and methods for solving a problem; in this context, a systems engineering problem. Such a set is ideally derived from a theoretical framework.

A method is a particular process or approach to solving a problem.

Systems dynamics

Systems Dynamics (SD) uses some of the ideas of cybernetics to consider the behavior of systems as a whole in their environment. SD was developed by Jay Forrester in the 1960's (Forrester 1961). He was interested in modeling the dynamic behavior of systems such as populations in cities, industrial supply chains.

SD is also used by Senge (Senge 1990) in his influential book The Fifth Discipline. This book advocates a system thinking approach to organization, and makes extensive use of SD notions of feedback and control.

Hard Systems Methodologies

Checkland (Checkland 1975) classifies hard system (glossary) methodologies, which set out to select an efficient means to achieve a predefined end, under the following headings:

1. System Analysis, the systematic appraisal of the costs and other implications of meeting a defined requirement in various ways.
2. Systems Engineering, the set of activities that together lead to the creation of a complex man-made entity and/or the procedures and information flows associated with its operation.

Operational Research is also considered a hard system approach, closely related to the Systems Analysis approach developed by the Rand Corporation, in which solutions are known but the best combinations of these solutions must be found. There is some debate as to whether System Dynamics is a hard approach, which is used to assess the objective behavior of real situations. Many application of SD have focused on the system, however it can and has also be used as part of a soft approach including the modelling of subjective perceptions (Lane 2000).

Systems Engineering allows for the creation of new solution systems, based upon available technologies. This hard view of systems engineering as a solution focused approach applied to large, complex and technology focused solutions, is exemplified by (Jenkins 1969; Hall 1962) and early defense and aerospace standards.

NOTE: Historically, the systems engineering discipline was primarily aimed at developing, modifying or supporting hard systems. More recent developments in systems engineering have incorporated problem focused thinking and agile solution approaches. It is this view of SE that is described in this SEBoK.

All of these hard approaches use systems thinking to ensure complete and viable solutions are created and/or as part of the solution optimization process. These approaches are appropriate to Unitary (glossary) problems, but not when the problem situation or solution technologies are unclear.

Soft Systems and Problem Structured Methods

Problem Structuring Methods (PSM) are interactive and participatory approaches to assist groups of diverse participants to alleviate a complex, problematic situation of common interest. Typically the hardest element of the situation is framing the issues which constitute the problem (Minger and Resenhead 2004).

PSM use systems and systems thinking as an abstract framework for investigation, rather than a structure for creating solutions. Systems descriptions are used to understand the current situation and describe an idealized future. Interventions directly in the current organization to move towards the idea recognize that the assumptions and mental models of the participants are an important obstruction to change and that these differing views cannot be dismissed but must form part of the intervention approach.

Peter Checkland's action research program, see System Science, in the 1980's forms the basis of work by Checkland, Wilson and others in the development of Soft Systems Methodology (glossary) (SSM) (Checkland 1999; Wilson 2001). SSM formalizes the idea of a soft approach using systemic thinking to expose the issues in a problem situation and guide interventions to reduce them. SSM provides a framework of ideas and models to help guide participants through this systemic thinking.

Other PSM approaches include Interactive Planning Approach (Ackoff 1981); Social Systems Design (Churchman 1968), and Strategic Assumptions Surfacing and Testing (Mason and Mitroff 1981).

SSM and other soft approaches use systems thinking to ensure problem situations are fully explored and resolved. These approaches are appropriate to pluralist (glossary problems. Critics of SSM suggest that it does not consider the process of intervention, and in particular how differences in power between individuals and social groups impacts the effectiveness of interventions.

Critical systems thinking and Multimethodology

The development of a range of hard and soft methods naturally leads to the question of which method to apply when (Jackson 1989). Critical Systems Thinking (glossary) (CST) or Critical Management Science Jackson (Jackson 1985) attempts to deal with this question.

The word critical is used in two ways. Firstly, critical thinking considers the limits of knowledge and investigates the limits and assumptions

of hard and soft systems, as discussed in the above sections. From this comes frameworks and meta-methodology for when to apply different methods such as Total Systems Intervention (TSI) (Flood and Jackson 1991).

The Multi-Methodology approach takes this aspect of critical thinking one stage further to recognize the value of combining techniques from several hard or soft methods as needed (Mingers and Gill 1997).

The second aspect of critical thinking considers the ethical, political and coercive dimension and the role of system thinking in society, see below.

Selecting Systems Methodologies

Jackson proposes a frame for considering which approach should be applied, please see: Jackson's Framework (http://www.systemswiki.org/index.php?title=System_of_Systems_Methodologies_(SOSM)) . In Jackson's framework the following definitions apply to the participants involved in solving the problem:

- Unitary (glossary): A problem situation in which participants "have similar values, beliefs and interests. They share common purposes and are all involved, in one way or another, in decision-making about how to realize their agreed objectives." (Jackson 2003, p. 19)
- Pluralist (glossary): A problem situation involving participants in which "although their basic interests are compatible, they do not share the same values and beliefs. Space needs to be made available within which debate, disagreement, even conflict, can take place. If this is done, and all feel they have been involved in decision-making, then accommodations and compromises can be found. Participants will come to agree, at least temporarily, on productive ways forward and will act accordingly." (Jackson 2003, p. 19)
- Coercive (glossary): A problem situation in which the participants "have few interests in common and, if free to express them, would hold conficting values and beliefs. Compromise is not possible and so no agreed objectives direct action. Decisions are taken on the basis of who has most power and various forms of coercion employed to ensure adherence to commands." (Jackson 2003, p. 19)

Jackson's framework suggests that for simple and complex systems with unitary participants, hard and dynamic systems thinking applies, respectively. For simple and complex systems with pluralist participants, soft systems thinking applies. For simple and complex systems with coercive participants, **emancipatory** and Postmodernist (glossary) system thinking applies, respectively. These thinking approaches consider all attempts to look for system solutions to be temporary and ineffective in situations where the power of individuals and groups of people dominate any system structures we create. They advocate an approach which encourages diversity, free thinking and creativity of individuals and in the organization's structures. Thus, modern system thinking has the breadth needed to deal with a broad range of complex problems and solutions.

While these ideas sit at the extreme of system thinking as a tool for problem solving, Jackson (Jackson 2003) identifies the work of some authors who have included these ideas into their systems approach.

References

Works Cited

Ackoff, R.L. 1981. Creating the Corporate Future. New York, NY, USA: Wiley and Sons.

Checkland, P. 1975. "The Origins and Nature of "Hard" Systems Thinking." Journal of Applied Systems Analysis, 5(2): 99-110.

Checkland, P. 1999. Systems Thinking, Systems Practice, New York, NY, USA: John Wiley & Sons.

Churchman, C.W. 1968. The Systems Approach. New York, NY, USA: Dell Publishing.

Flood, R. and M. Jackson. 1991. Creative Problem Solving: Total Systems Intervention. London, UK: Wiley.

Forrester, J. 1961. Industrial Dynamics. Cambridge, MA, USA: MIT Press.

Hall, A.D. 1962. A Methodology for Systems Engineering. New York, NY, USA: Van Nostrand Reinhold.

Jackson, M. 1985. "Social Systems Theory and Practice: the Need for a Critical Approach." International Journal of General Systems. 10: 135-151.

Jackson, M. 1989. "Which Systems Methodology When? Initial Results from a Research Program." In: R Flood, M Jackson and P Keys (eds). Systems Prospects: the Next Ten Years of Systems Research. New York, NY, USA: Plenum.

Jackson, M. 2003. Systems Thinking: Creating Holisms for Managers. Chichester, UK: Wiley.

Jenkins, G.M. 1969. The Systems Approach. In Beishon, J. and G. Peters (eds.), Systems Behavior, 2nd ed. New York, NY, USA: Harper

and Row.

Kuhn, T.S. *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press, 1962

Lane, D. 2000 "Should System Dynamics be Described as a `Hard' or `Deterministic' Systems Approach?" Systems Research and Behavioral Science 17, 3–22 (2000) John Wiley & Sons, Ltd.

Mason, R.O. and I.I. Mitroff. 1981. Challenging Strategic Planning Assumptions: Theory, Case and Techniques. New York, NY, USA: Wiley and Sons.

Mingers, J. and A. Gill. 1997. Multimethodology: Theory and Practice of Combining Management Science Methodologies. Chichester, UK: Wiley.

Mingers, J. and J. Rosenhead. 2004. "Problem Structuring Methods in Action." European Journal of Operations Research. 152(3) (Feb. 2004): 530-554.

Senge, P. 1990. The Fifth Discipline: the Art and Practice of the Learning Organisation. London, UK: Century Books.

Wilson, B. 2001. Soft Systems Methodology—Conceptual Model Building and Its Contribution. New York, NY, USA: J.H.Wiley.

Primary References

Checkland, P. 1999. Systems Thinking, Systems Practice, New York, NY, USA: John Wiley & Sons.

Forrester, J. 1961. Industrial Dynamics. Cambridge, MA, USA: MIT Press.

Jackson, M. 1985. Social Systems Theory and Practice: the Need for a Critical Approach. International Journal of General Systems 10: 135-151.

Additional References

Jackson, M.C. and Keys, P. 1984. "Towards a System of Systems Methodologies." The Journal of the Operational Research Society. 35(6) (Jun. 1984): 473-486.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Groupings of Systems

Systems can be grouped together to create more complex systems. In some cases considering systems as system elements in a system hierarchy is sufficient. However, there are cases where the groupings of system produce an entity that must be treated differently from a single integrated system. This article explores the different descriptions of how and why system groupings might be considered.

System of Systems (SoS)

The phrase "system of systems" is commonly used, but there is no widespread agreement on its exact meaning, or on how it can be distinguished from a conventional system. An extensive history of SoS is provided in "System-of-Systems Engineering Management: A Review of Modern History and a Path Forward" (Gorod, et. al. 2008). This paper provides a historical perspective for systems engineering

from (Brill 1998). The authors then provide a chronological history for System-of-Systems (SoS) engineering from 1990 to 2008. Their history provides an extensive set of references to all of the significant papers and textbooks on SoS. Gorod et. al. cite Maier as one of the most influential contributors to the study of SoS.

Maier examined the meaning of SoS in detail and used a characterization approach to create a definition (Maier 1998, 267-284). His definition has been adopted by many working in the field (AFSAB 2005). Maier provides this definition:

> *A system-of-systems is an assemblage of components which individually may be regarded as systems, and which possess two additional properties:*
>
> 1. *Operational Independence of the Components: If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.*
> 2. *Managerial Independence of the Components: The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems.* (Maier 1998, 271)

Maier goes on further saying that "the commonly cited characteristics of systems-of-systems (complexity of the component systems and geographic distribution) are not the appropriate taxonomic classifiers" (Maier 1998, 268). According to the Defense Acquisition Guide: "A SoS is defined as a set or arrangement of systems that results from independent systems integrated into a larger system that delivers unique capabilities" (DAU 2010, 4.1.4. System of Systems (SoS) Engineering). For further details on SoS, see the *Systems Engineering Guide for SoS* developed by the US Department of Defense (DoD) (DUS(AT) 2008). Four kinds of SoS have been defined (Maier 1998; Dahmann and Baldwin 2008; DUS(AT) 2008; Dahmann, Lane, and Rebovich 2008):

> - ***Virtual***. *Virtual SoS lack a central management authority and a centrally agreed upon purpose for the system-of-systems. Large-scale behavior emerges—and may be desirable—but this type of SoS must rely upon relatively invisible mechanisms to maintain it.*
> - ***Collaborative***. *In collaborative SoS the component systems interact more or less voluntarily to fulfill agreed upon central purposes. The Internet is a collaborative system. The Internet Engineering Task Force works out standards but has no power to enforce them. The central players collectively decide how to provide or deny service, thereby providing some means of enforcing and maintaining standards.*
> - ***Acknowledged***. *Acknowledged SoS have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system.*
> - ***Directed***. *Directed SoS are those in which the integrated system-of-systems is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.*(DUS(AT) 2008, 4-5; and, Dahmann, Lane, and Rebovich 2008, 4; in reference to (Maier 1998; Dahmann and Baldwin 2008))

The terms Emergence (glossary) and emergent behavior are increasingly being used in SoS contexts. While the concept of emergence and its derivative terms has a long history in science and technology, to this day there is no single, universal definition of emergence. In SoS contexts, the recent interest in emergence has been fueled, in part, by the movement to apply systems science and complexity theory to problems of large-scale, heterogeneous information technology based systems. In this context, a working definition of emergent behavior of a system is behavior which is unexpected or cannot be predicted by knowledge of the system's constituent parts. One of the leading authors in the area of SoS is Mo Jamshidi who is the editor of a leading textbook (Jamshidi 2009) and articles, such as "System of Systems Engineering – New Challenges for the 21st Century" (Jamshidi 2008). His article, that addresses the challenges, also provides numerous references to papers that have examined the definition of SoS. The author selects six of the many potential definitions. His lead definition is

> *Systems of systems exist when there is a presence of a majority of the following five characteristics: operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development* (Jamshidi 2008, 5; adapted from Sage and Cuppan 2001, 326).

Federation of Systems (FOS)

Different from the SoS concept, but related to it in several ways, is the concept called "federation of systems" or FOS. This concept might apply when there is a very limited amount of centralized control and authority (Sage and Cuppan 2001). Each system in an FOS is very strongly in control of its own destiny, but "chooses" to participate in the FOS for its own good and the good of the "country," so to speak. It is a coalition of the willing. An FOS is generally characterized by significant autonomy, heterogeneity, and geographic distribution or dispersion (Krygiel 1999). Krygiel (1999) defined a taxonomy of systems showing the relationships among conventional systems, SoSs, and FOSs. This taxonomy has three dimensions: autonomy, heterogeneity, and dispersion. An FOS would have a larger value on each of these three dimensions than a non-federated SoS. An Enterprise System (glossary) as described in Types of Systems, could be considered to be an FOS if it rates highly on these three dimensions. However, it is possible for an enterprise to have components that are not highly autonomous, that are relatively homogenous, and are geographically close together. Therefore, it would be a mistake to say that an enterprise is necessarily

the same as an FOS.

(Handy 1992) describes a federalist approach called "New Federalism" which identifies the need for structuring of loosely coupled organizations to help them adapt to the rapid changes inherent in the Information Age. This leads to the need for virtual organizations where alliances can be quickly formed to handle the challenges of newly identified threats and a rapidly changing marketplace (Handy 1995). Handy sets out to define a number of federalist political principles that could be applicable to an FOS. Handy's principles have been tailored to the domain of systems engineering and management by (Sage and Cuppan 2001).

Families of Systems

The Defense Acquisition University (DAU 2010, 4.1.4. System of Systems (SoS) Engineering) defines families of systems as:

> *A family of systems is a grouping of systems having some common characteristic(s). For example, each system in a family of systems may belong to a domain or product line (e.g., a family of missiles, aircraft, or situation awareness systems). In general, a family of systems is not considered to be a system per se because it does not necessarily create capability beyond the additive sum of the individual capabilities of its member systems. A family of systems lacks the synergy of a SoS. The family of systems does not acquire qualitatively new properties as a result of the grouping. In fact, the member systems may not be connected into a whole.* (DAU 2010)

Very few papers have been written that address families of systems or compare them to systems of systems.

James Clark (2008) provides a view that a family of systems is equivalent to a product line:

> *By family, we mean a product-line or domain, wherein some assets are re-used un-modified; some assets are modified, used, and re-used later; and some assets are developed new, used, and re-used later. Product-lines are the result.* (Clark 2008)

References

Citations

AFSAB. 2005. *Report on Domain Integration*. Washington, D.C.: U.S. Air Force Scientific Advisory Board/U.S. Air Force. SAB-TR-05-03.

Brill, J. H. 1998. "Systems Engineering – A Retrospective View." *Systems Engineering*. 1(4): 258-266.

Clark, J. 2008. "System of Systems Engineering and Family of Systems Engineering From a Standards, V-Model, and Dual-V Model Perspective." Proceedings of the 18th Annual International Council on Systems Engineering International Symposium 15-19 June, 2008, Utrecht, The Netherlands.

Dahmann, J.S., J.A. Lane, and G. Rebovich. 2008. "Systems Engineering for Capabilities." *CROSSTALK: The Journal of Defense Software Engineering*. (November 2008): 4-9.

DAU. February 19, 2010. *Defense acquisition guidebook (DAG)*. Ft. Belvoir, VA, USA: Defense Acquisition University (DAU)/U.S. Department of Defense.

DUS(AT). 2008. *Systems Engineering Guide for Systems of Systems,*" version 1.0. Washington, DC, USA: Deputy Under Secretary of Defense for Acquisition and Technology (DUS(AT))/U.S. Department of Defense (DoD).

Gorod, A., B. Sauser, and J. Boardman. 2008. "System-of-Systems Engineering Management: A Review of Modern History and a Path Forward". IEEE *Systems Journal*, 2(4): 484-499.

Handy, C. 1995. "How Do You Manage People Whom You Do Not See? Trust and the Virtual Organization." *Harvard Business Review*.' *73(3) (May-June): 40-50*.

Handy, C. 1992. "Balancing Corporate Power: A New Federalist Paper". *Harvard Business Review*. 70(6) (November/December): 59-72.

Jain, P. and Dickerson, C. 2005. "Family-of-Systems Architecture Analysis Technologies." Proceedings of the 15th Annual International Council on Systems Engineering International Symposium, 10-15, July 2005, Rochester, NY, USA.

Jamshidi, M. "Theme of the IEEE SMC 2005" in IEEE SMC 2005 - International Conference on Systems, Man, and Cybernetics. Accessed 11 September 2011. Available at: http://ieeesmc2005.unm.edu.

Jamshidi, M. (ed.). 2009. *Systems of Systems Engineering – Innovations for the 21st Century*. Hoboken, NJ, USA: John Wiley and Sons.

Jamshidi, M. 2008. "System of Systems Engineering – New Challenges for the 21st Century". IEEE *Aerospace and Electronic Systems Magazine*. 23(5) (May 2008): 4-19.

Krygiel, A. J. 1999. *Behind the Wizard's Curtain: An Integration Environment for a System of Systems*. Arlington, VA, USA: C4ISR Cooperative Research Program (CCRP).

Maier, M. W. 1998. "Architecting Principles for Systems-of-Systems". *Systems Engineering*, 1(4): 267-84.

Sage, A., and C. Cuppan. 2001. "On the Systems Engineering and Management of Systems of Systems and Federations of Systems". *Information-Knowledge-Systems Management Journal*. 2(4) (December 2001): 325-45.

Primary References

Gorod, A., B. Sauser, and J. Boardman. 2008. "System-of-Systems Engineering Management: A Review of Modern History and a Path Forward." IEEE *Systems Journal*. 2(4): 484-499.

Jamshidi, M. editor. 2009. *Systems of Systems Engineering – Innovations for the 21st Century*. Hoboken, NJ: Wiley and Sons.

Jamshidi, M. 2008. "System of Systems Engineering – New Challenges for the 21st Century". IEEE *Aerospace and Electronic Systems Magazine*. 23(5) (May 2008): 4-19.

Maier, M. W. 1998. "Architecting Principles for Systems-of-Systems". *Systems Engineering*. 1(4): 267-84.

Sage, A. and C. Cuppan. 2001. "On the Systems Engineering and Management of Systems of Systems and Federations of Systems". *Information-Knowledge-Systems Management Journal*. 2(4) (December 2001): 325-45.

Additional References

No additional references have been identified for version 0.5. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Complexity

Complexity is one of the most important and difficult to define system concepts. Is a system's complexity in the eye of the beholder or is there inherent complexity? How should complexity be rigorously defined? How should it be measured? What are the consequences on systems engineering of dealing with higher complexity systems? Many questions abound. The knowledge about complexity is summarized in this article.

Defining System Complexity

Weaver (Weaver 1948) gives one of the earliest definitions as the degree of difficulty in predicting the properties of a system, if the properties of the system's parts are given. Does this simple definition describe a static property of a system artifact, or a dynamic property of systems in use to solve a problem? If complexity is related to an ability to understand systems, does it vary depending who is considering the system and why? How do these questions relate to the distinctions between Natural Systems (glossary), Social Systems (glossary) and Engineered Systems (glossary) or to the idea of a System Context (glossary)?

According to (Sheard and Mostashari 2008) complexity sits on a spectrum somewhere between order and chaos. In common usage chaos (glossary) is a state of disorder or unpredictability. A chaotic system has elements which are not interconnected and behave randomly with no adaptation or control. Chaos Theory (Kellert 1993) is applied to certain types of dynamic system (e.g. the weather) which, although they have structure and relationships, exhibit unpredictable behavior (glossary). These systems are deterministic; their future behavior is fully determined by their initial conditions with no random elements involved. However, their structure is such that (un-measurably) small perturbations in inputs or environmental conditions may result in unpredictable changes in behavior. This behavior is known as deterministic chaos, or simply chaos. Models of chaotic systems can be created and, with increases in computing power, reasonable predictions of behavior are possible at least some of the time. One might need to consider truly random or chaotic natural or social systems as part of the context of an engineered system, but such system cannot themselves be engineered.

Ordered systems have fixed relations between elements and are not adaptable. (Page 2009) cites a watch as an example of an ordered system. The components of a watch are based on similar technologies with a clear mapping between form and function. If the operating environment changes outside prescribed limits or one key component is removed, the watch will cease to perform its function. Although the watch may have many components, it can be regarded as complicated but not complex. Ordered systems occur as system components, and are the subject of traditional engineering. It is important to understand the limitations of such systems when using them in a complex system.

Complex systems sit between order and chaos with combinations of elements of different types arranged in relationships which provide more than one function. This can lead to multiple ways of achieving a given outcome. Complex systems adapt (glossary) to environmental changes or the loss of some elements. For example, if one element such as a doctor, piece of equipment or building infrastructure is removed from a hospital surgical unit, the remaining elements will continue to function as a unit, albeit with reduced effectiveness. This would be considered a complex system.

The inclusion of people in a system is often a factor in their complexity, due to the variability of human behavior as part of a system and the perceptions of people out-side the system. (Sheard and Mostashari 2011) sort the attributes of complexity into causes and effects. Attributes that cause complexity include: many pieces, nonlinear, emergent, chaotic, adaptive, tightly coupled, self-organized, decentralized, open, political (vs. scientific), and multi-scale. The effects of those attributes which make a system seem complex, often as perceived by people, include: uncertain, difficult to understand, unclear cause and effect, unpredictable, uncontrollable, unstable, unrepairable and unmaintainable, costly, and takes too long to build. (Sillitto 2009) refers to these as Objective and Subjective complexity and associates both with problem situations and system solutions.

Thus, complexity is a measure of how difficult it is to understand how a system will behave or to predict the consequences of changing it. It occurs when there is no simple relationship between what an individual element does and what the system as a whole will do, and when the system includes some element of adaptation or problem solving to achieve its goals in different situations. It can be based on objective attributes of the system or on subjective perceptions of system observers. This view of complex systems is very much the kind of system for which a Systems Approach (glossary) is essential.

Origins and Characteristics of Complexity

Many systems science authors have attempted to make sense of complexity; how does it differ from what is merely complicated or intricate, and how is it related to human perception or societal context? Weaver provided an early viewpoint categorizing organized and disorganized complexity (Weaver 1948). These categories and later reflections amongst others, such as (Flood and Carson 1993) and (Lawson 2010), provide the following complexity categorization:

- Organized simplicity occurs when there are a small number of essential factors and large number of less significant or insignificant factors. Initially, a situation may seem to be complex, but on investigation the less significant and insignificant factors are taken out of the picture and hidden simplicity is found. This is also the basis for the process of abstraction (glossary); creating systems of greater general applicability, but with lower level of detail.

- Organized complexity is prevalent in physical and abstract systems where the structure of the system is organized in order to be understood and thus amenable to scientists in describing complex behaviors as well as for structuring the engineering and life cycle management (glossary) of complex systems (Braha et al. 2006). There is a richness that must not be oversimplified.

- Disorganized complexity occurs when there are many variables that exhibit a high level of random behavior. It can also represent the product of not having adequate control over the structure of heterogeneous complex systems that have evolved due to inadequate control over the system during its life (complexity creep).

- People-related complexity, where perception fosters a feeling of complexity. In this context, humans become "observing systems". People can be viewed as system elements which contribute to the other types of complexity (Axelrod and Cohen 1999). The rational or irrational behavior of individuals in particular situations is of course a vital factor in respect to complexity (Kline 1995).

(Senge 1990) identifies two fundamental forms of engineered systems complexity; namely, detail complexity and dynamic complexity. Detail complexity arises from the number of systems elements and 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 use in

different problem scenarios.

(Sheard and Mostashari 2011) describe Structural, Dynamic and Socio-political complexity. Structural complexity looks at the system elements and relationships. In particular, structural complexity looks at how many different ways system elements can be combined, and thus the potential for the system to adapt to external needs. Dynamic Complexity considers the complexity which can be observed when systems are used to perform particular tasks in an environment. There is a time element to dynamic complexity. The ways in which systems interact in the short term is directly related to system behavior (glossary); the longer term affects of using systems in an environment is related to system evolution. Finally, Socio-political complexity considers the affect of individuals or groups of people on complexity. This will include the cognitive behavior of people in the system, multiple stakeholder viewpoints within a system context and social or cultural biases which add to the wider influences on a system context.

Characteristics of Complex Systems

According to (Page 2009), there are four characteristics of complex systems:

- Independence of system elements. That is, making their own decisions; although these decisions may be influenced by information from other elements and the adaptability algorithms it carries with it. (Sheard and Mostashari 2008) refer to this characteristic as "autonomous" components.
- Interconnectedness: between system elements. This may be via a physical connection, shared data or simply a visual awareness of where the other elements are and what they are doing as in the case of the flock of geese or the squadron of aircraft.
- Diversity simply means that system elements are different in some way, technologically or functionally. Element may be carrying different adaptability algorithms, for example.
- Adaptability is generally considered to be the most important characteristic of the elements of a complex system. Adaptability means that each element can do what it wants to do to support itself or the entire system. In the case of the human pilots, each pilot can make his or her own decisions to adjust to the mission of the whole squadron. (Sheard and Mostashari 2008) refer to this characteristic as self-organizing. Sheard and Mostashari also say that complex systems adapt to their environment. Adaptability can also be achieved with software. (Pollock and Hodgson 2004) describe how this can be done in a variety of complex system types including power grids and enterprise systems.

Complexity is also in many ways a human concept. Looking at a hospital surgical unit from the perspective of an experienced nurse, a member of the cleaning staff, a software engineer designing code for a piece of medical equipment, a typically educated patient or a patient from an African village flown into the hospital after a natural disaster, it is clear that the education, experience and knowledge of each person may radically change their understanding of the same system. Such factors as human values and beliefs, interests, capabilities as well as notions and perceptions of systems are determinants of complexity.

(Warfield 2006) developed a powerful methodology for addressing complex issues, particularly in the socio-economic field, based on a relevant group of people developing an understanding of the issue in the form of a set of interacting problems - what he called the "problematique". The complexity is then characterized by several measures, such as the number of significant problems, their interactions, and the degree of consensus about the nature of the problems. Thus, what becomes clear that how, why, where and by whom a system is used may all contribute to its complexity.

Some of this complexity can be reduced by education, training or familiarity with a system; some must be managed as part of a problem or solution. (Checkland 1999) argues that a group of stakeholders will have its own world views which lead them to form different, but equally valid, understandings of a system context. These differences cannot be explained away or analyzed out, but must be understood and considered in the formulation of problems or the creation of potential solutions.

Complexity and Context

The views of complexity are not independent when considered across a system Hierarchy (glossary). System Context (glossary) is a concept used to focus on an engineered system-of-interest, while still considering wider Holistic (glossary) system and environmental relationships. Problem situations and potential solutions may contain both subjective and objective complexity, while structural complexity at one level will be related to dynamic complexity at higher levels. People are involved in most system contexts, as system elements and as part of the operating environment. People are also involved with systems throughout the lifetimes of those systems.

(Sillitto 2009) considers the link between the types of complexity and system architectures, but this can be generalized to consider how to deal with complexity in the applications of a Systems Approach (see Applying the Systems Approach). Sheard and Mostashari 2011) also show how the different views of complexity map onto Product Systems (glossary), Service Systems (glossary) and Enterprise Systems (glossary); and to associated Development and Sustainment systems and Project organizations.

The definition of system complexity used in the SEBoK covers two views of complexity within a system context: the structural complexity of the system-of-interest and wider system; and the dynamic complexity when the system-of-interest is used as part of the wider system in different problem scenarios. The differing perceptions of this complexity by both individuals and social groups of people involved in creating,

using or interacting with a system is recognized. In many ways the Systems Approach exists to deal with these complexity issues.

References

Works Cited

Axelrod, R. and M. Cohen. 1999. *Harnessing Complexity: Organizational Implications of a Scientific Frontier*. New York, NY, USA: Simon and Schuster.

Braha, D., A. Minai, and Y. Bar-Yam (eds.). 2006. *Complex Engineered Systems: Science Meets Technology*. New York, NY, USA: Springer.

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

Flood, R. L., and E.R. Carson. 1993. *Dealing with Complexity: An Introduction to The Theory and Application of Systems Science"*, *2nd ed.*. New York, NY, USA: Plenum Press.

Lawson, H. W. 2010. *A Journey Through the Systems Landscape*. Kings College, UK: College Publications.

Kellert, S. 1993. *In the Wake of Chaos: Unpredictable Order in Dynamical Systems*, Chicago, IL, USA: University of Chicago Press. p. 32.

Kline, S. 1995. *Foundations of Multidisciplinary Thinking*. Stanford, CA, USA: Stanford University Press.

Page, Scott E. 2009. *Understanding Complexity*. Chantilly, VA, USA: The Teaching Company.

Pollock, J.T. and R. Hodgson. 2004. *Adaptive Information*. Hoboken, NJ, USA: John Wiley & Sons.

Senge, P.M. 1990. *The Fifth Discipline: The Art & Practice of The Learning Organization*. New York, NY, USA: Doubleday/Currency.

Sheard, S.A. and A. Mostashari. 2008. "Principles of Complex Systems for Systems Engineering." *Systems Engineering*, 12(4): 295-311.

Sheard, SA. and A. Mostashari. 2011. "Complexity Types: From Science to Systems Engineering." Proceedings of the 21st Annual of the International Council on Systems Engineering (INCOSE) International Symposium, 20-23 June 2011, Denver, Colorado, USA.

Sillitto H.G. 2009. "On Systems Architects and Systems Architecting: Some Thoughts on Explaining The Art and Science of System Architecting." Proceedings of the 19th Annual International Council on Systems Engineering (INCOSE) International Symposium, 20-23 July 2009, Singapore.

Weaver, W. 1948. "Science and Complexity." *American Science*. 36: 536-544.

Warfield, J.N. 2006. *An Introduction to Systems Science*. London, UK: World Scientific Publishing.

Primary References

Page, Scott E. 2009. *Understanding Complexity*. Chantilly, VA, USA: The Teaching Company.

Flood, R. L., & E.R. Carson. 1993. *Dealing with Complexity: An Introduction to The Theory and Application of Systems Science*, 2nd ed. New York, NY, USA: Plenum Press.

Sheard, S.A. and A. Mostashari. 2008. "Principles of Complex Systems for Systems Engineering". *Systems Engineering*, 12(4): 295-311.

Additional References

Ashby, W.R. 1956. *An Introduction to Cybernetics*. London, UK: Chapman and Hall.

Aslaksen, E.W. 2004. "System Thermodynamics: A Model Illustrating Complexity Emerging from Simplicity". *Systems Engineering*, 7(3). Hoboken, NJ, USA: Wiley.

Aslaksen, E.W. 2009. *Engineering Complex Systems: Foundations of Design in the Functional Domain*. Boca Raton, FL, USA: CRC Press.

Aslaksen, E.W. 2011. "Elements of a Systems Engineering Ontology". *Proceedings of SETE 2011*, Canberra, Australia.

Eisner, H. 2005. *Managing Complex Systems: Thinking Outside the Box*. Hoboken, NJ, USA: John Wiley & Sons.

Jackson, S., D. Hitchins, and H. Eisner. 2010. What is the Systems Approach? INCOSE *Insight* 13(1) (April 2010): 41-43.

MITRE. 2011. "Systems Engineering Strategies for Uncertainty and Complexity." *Systems Engineering Guide*. Accessed 9 March 2011 at [[1] (http://www.mitre.org/work/systems_engineering/guide/enterprise_engineering/comprehensive_viewpoint/sys_engineering_strategies_uncerta].

Ryan, A. 2007. "Emergence Is Coupled to Scope, Not Level, Complexity". A condensed version appeared in INCOSE *Insight*, 11(1) (January 2008): 23-24.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Emergence

According to (Checkland 1999, p. 314), emergence (glossary) is "the principle that entities exhibit properties which are meaningful only when attributed to the whole, not to its parts." Questions that arise from this definition include: What kinds of systems exhibit emergence? Under what conditions do they exhibit emergence? Are emergent properties predictable? Can emergent properties be planned? How does one achieve a planned emergent property? There are many varied and even conflicting views on emergence. This article presents what is believed to be the prevailing view. Some references for other views are also provided.

Overview of Emergence

According to (Hitchins 2007, p. 7) emergence is common in nature. The emergent property "self-awareness" results from the combined effect of the interconnected and interacting neurons that make up the brain. The pungent gas ammonia results from the combination of two odorless gases, hydrogen and nitrogen.

Hitchins also notes that technological systems can also exhibit emergence. For example, the performance of a radar system results from the action of all of its subsystems. Thus, emergence always occurs at the highest level of the system hierarchy. However, he points out that to the extent that the subsystems themselves can be considered systems, they also exhibit emergence. (Page 2009) also refers to emergence as a "macro-level property."

According to (Hitchins 2007, p. 27), emergence depends on the concept of holism (glossary) that holds that "an open system is a whole" and that "the whole is different from, and may be greater than, the sum of its parts. (Page 2009) says that emergence "refers to the spontaneous creation of order and functionality from the bottom up." (Checkland 1999) points out that the concept of holism is counter the idea of Reductionism (glossary) in which a system is built up from its parts.

(Bedau and Humphreys 2008) provide a comprehensive description of the philosophical and scientific background of emergence.

Emergent Properties

Emergent properties (glossary) are the resulting properties of emergence. Emergent properties may or may not be quantifiable. For complex systems they are generally not quantifiable. Typical emergent properties are agility (glossary) and resilience (glossary). Emergent properties may or may not be predictable as discussed below. In general, emergent properties of ordered systems are predictable, while those of complex systems are not.

Types of Emergence

According to (Page 2009) there are three types of emergence, two of which are complex, while the other is called "simple emergence."

Simple Emergence

According to Page (2009), simple emergence is the only type of emergence that can be predicted. Simple emergence occurs in non-complex systems (see the Complexity article). Sheard and Mostashari (2008) refer to such systems as "ordered." Page (2009) refers to non-complex systems as "equilibrium" systems. For example, the physics of aircraft flight is well-known. To achieve the emergent property of "controlled flight", all parts of the aircraft need to be considered. It cannot be achieved alone by considering only the wings or just the control system or the propulsion system. All three (plus other elements) must be considered.

Weak Emergence

According to Page (2009), weak emergence is emergence which is expected and presumably desired. However, since weak emergence is a product of a complex system, the actual level of emergence cannot be predicted. So how is the desired level of emergence achieved? According to (Jackson et al. 2010), the desired level of emergence can only be achieved by iteration. That is, the different design parameters of the system must be adjusted until the desired level of emergence is achieved. This can be accomplished through simulation or testing.

Strong Emergence

According to Page (2009), strong emergence is unexpected emergence. That is, emergence is not observed until the system is simulated or tested. Strong emergence may be evident in failures or shutdowns. For example, the US-Canada Blackout of 2003 as described by (US-Canada Power System Outage Task Force, 2004) was a case of cascading shutdown that resulted from the design of the system; however, there were no equipment failures. The shutdown was completely systemic (glossary). As Hitchins (2007, p. 15) points out, this example shows that emergent properties are not always beneficial.

A type of system particularly subject to strong emergence is the system of systems (SoS) (glossary). The reason for this is that the SoS, by definition, resulted from different systems that were designed to operate independently. When these systems are operated together, the interaction among the parts of the system results in unexpected emergence.

Other Theories on Emergence

Most sources, for example, Hitchins (2007, p. 7), state that emergence occurs at the highest level of the system architecture. Another theory advanced by Ryan (2007) contends that emergence is coupled to scope rather than system hierarchical levels. In Ryan's terms, scope has to do with spatial dimensions rather than hierarchical levels.

Abbott (2006) does not disagree with the general definition of emergence as discussed above. However, he takes issue with the notion that emergence operates outside the bounds of classical physics. He says that "such higher-level entities…can always be reduced to primitive physical forces."

Practical Considerations

The requirement to iterate the design to achieve desired emergence, as discussed above, results in a design process that is itself more lengthy than one needed to design an "ordered" system. Hence, the Vee model cannot be executed in a single direction. Repeated iterations of the model are required. The result is that complex systems may be more costly and time-consuming to develop.

Application to Enterprise Systems

Enterprise systems exhibit emergence because these systems contain humans and are therefore complex. Enterprise systems will most likely exhibit weak and strong emergence, but it is unlikely that enterprises systems will exhibit simple emergence.

Sillitto (2008) discusses the application of the concept of emergent to "ultra large systems," which is essentially the same as an enterprise

system. Sillitto provides practical advice on how to define such a system with emergence as a consideration.

Application to Product Systems

Product systems can be either ordered or complex. Complex product systems will exhibit both weak and strong emergence. Ordered product systems will exhibit simple emergence.

Application to Service Systems

Like enterprise systems, service systems will exhibit weak and strong emergence. It is unlikely that a service system will exhibit simple emergence.

References

Works Cited

Abbott, R. 2006. "Emergence Explained: Getting Epiphenomena to Do Real Work". *Complexity*, 12(1) (September-October): 13-26.

Bedau, M.A. and P. Humphreys, P. (eds.). 2008. "Emergence" In *Contemporary Readings in Philosophy and Science*. Cambridge, MA, USA: The MIT Press.

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

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology*. Hoboken, NJ, USA: John Wiley & Sons.

Jackson, S., D. Hitchins, and H. Eisner. 2010. "What is the Systems Approach?" INCOSE *Insight* 13(1) (April 2010): 41-43.

Page, S.E. 2009. *Understanding Complexity*. The Great Courses. Chantilly, VA, USA: The Teaching Company.

Ryan, A. 2007. "Emergence is Coupled to Scope, Not Level." *Complexity*, 13(2) (November-December).

Sheard, S.A. and A. Mostashari. 2008. "Principles of Complex Systems for Systems Engineering." *Systems Engineering*. 12: 295-311.

Sillitto, H.G. 2008. "Design Principles for Ultra-Large-Scale Systems". Proceedings of the Second International Workshop on Ultra Large Scale Software Intensive Systems. Leipzig, Germany.

US-Canada Power System Outage Task Force. 2004. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. April, 2004. Washington-Ottawa. Available at https://reports.energy.gov/BlackoutFinal-Web.pdf.

Primary References

Hitchins, D. 2007. Systems Engineering: A 21st Century Systems Methodology. Hoboken, NJ, USA: John Wiley & Sons.

Page, S. E. 2009. Understanding Complexity. The Great Courses. Chantilly, VA, USA: The Teaching Company.

Additional References

Abbott, R. 2006. "Emergence Explained: Getting Epiphenomena to Do Real Work". *Complexity*, 12(1) (September-October): 13-26.

Bedau, M.A. and P. Humphreys (eds.). 2008. "Emergence" In *Contemporary Readings in Philosophy and Science*. Cambridge, MA, USA: The MIT Press.

Jackson, S., D. Hitchins, and H. Eisner. 2010. "What is the Systems Approach?" INCOSE *Insight*. 13(1) (April): 41-43.

Ryan, A. 2007. "Emergence is Coupled to Scope, Not Level". *Complexity*, 13(2) (November-December).

Sheard, S.A. and A. Mostashari. 2008. "Principles of Complex Systems for Systems Engineering." *Systems Engineering*. 12: 295-311.

Sillitto, H.G. 2008. "Design Principles for Ultra-Large-Scale Systems". Proceedings of the Second International Workshop on Ultra Large Scale Software Intensive Systems. Leipzig, Germany.

US-Canada Power System Outage Task Force. 2004. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. April, 2004. Washington-Ottawa. Available at https://reports.energy.gov/BlackoutFinal-Web.pdf.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Representing Systems with Models

[X]

This navigational tool is a prototype under development SEBoK 1.0. Comments are welcome for improvement.

A model (glossary) is an abstraction of a system that offers insight into that system's properties, such as behavior, appearance, or cost. Modeling is a common practice that is shared by most engineering disciplines and includes electrical circuit design models, three-dimensional computer-aided design models, software design models, and analytical models used to support power, thermal, structural, and embedded real-time analysis. Modeling systems are becoming increasingly important to the practice of systems engineering (glossary) as a means to enhance quality, productivity, and innovation, as well as reduce the cost and risk of systems development. Different types of models are needed to represent systems in support of the analysis, specification, design, and verification of systems. This knowledge
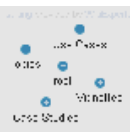
area provides an overview of models used to represent different aspects of systems.

Topics

The topics contained within this knowledge area include:

- What is a Model?
- Why Model?
- Types of Models
- System Modeling Concepts
- Modeling Standards

References

Works Cited

None.

Primary References

Dori, D. 2002. *Object-Process Methodology – A Holistic Systems Paradigm*. Berlin, Germany: Springer Verlag.

Estefan, J. 2008. *A Survey of Model-Based Systems Engineering (MBSE) Methodologies*, rev, B. Seattle, WA: International Council on Systems Engineering. INCOSE-TD-2007-003-02. Available at http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf.

Friedenthal, S., A. Moore, and R. Steiner. 2009. "Chapter 2". *A Practical Guide to SysML: The Systems Modeling Language*. Needham, MA, USA: OMG Press.

Guizzardi, G. 2007. *On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models*. Proceedings of the Databases and Information Systems IV Conference, Amsterdam, Netherlands. Available at http://portal.acm.org/citation.cfm?id=1565425.

INCOSE. 2007. *Systems Engineering Vision 2020*. Seattle, WA, USA: International Council on Systems Engineering. September 2007. INCOSE-TP-2004-004-02. Available at http://www.incose.org/ProductsPubs/products/sevision2020.aspx.

Wymore, A.W. 1993. *Model-Based Systems Engineering*. Boca Raton, FL, USA: CRC Press, Inc.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g.

Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

What is a Model?

This article provides foundational concepts, such as definitions of a model and a modeling language, and expresses their relationships to modeling tools (glossary) and model-based systems engineering (MBSE) (glossary).

Definition of a Model

There are many definitions of the word model (glossary). The following definitions refer to a model as a representation of selected aspects of a domain of interest (glossary) to the modeler:

- a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process (DoD 1994);
- a representation of one or more concepts that may be realized in the physical world (Friedenthal, Moore, and Steiner 2009);
- a simplified representation of a system at some particular point in time or space intended to promote understanding of the real system (Bellinger 2004);
- an abstraction of a system, aimed at understanding, communicating, explaining, or designing aspects of interest of that system (Dori 2002); and
- a selective representation of some system whose form and content are chosen based on a specific set of concerns; the model is related to the system by an explicit or implicit mapping (Object Management Group 2010).

In the context of systems engineering, a model that represents a system and its environment is of particular importance to help analyze, specify, design, and verify systems, as well as share this information with other stakeholders. A variety of system models are used to represent different types of systems for different modeling purposes.

A model can have different forms as indicated in the first definition above, including a physical, mathematical, or logical representation. A physical model can be a mockup that represents an actual system, such as a model airplane. A mathematical model may represent possible flight trajectories in terms of its acceleration, speed, position, and orientation. A logical model may represent logical relationships that describe potential causes of airplane failure, such as how an engine failure can result in a loss of power and cause the airplane to lose altitude, or how the parts of the system are interconnected. It is apparent that many different models may be required to represent a system of interest.

Modeling Language

A physical model is a concrete representation of an actual system that can be felt and touched. Other models may be more abstract representations of a system or entity. These models rely on a modeling language (glossary) to express their meaning as explained in "On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models" (Guizzardi 2007). These days, modeling languages are generally intended to be both human-interpretable and computer-interpretable, and are specified in terms of both syntax and semantics.

The abstract syntax (glossary) specifies the model constructs and the rules for constructing the model from its constructs. In the case of a natural language like English, the constructs may include types of words such as verbs, nouns, adjectives, and prepositions, and the rules specify how these words can be used together to form proper sentences. The abstract syntax for a mathematical model may specify constructs to define mathematical functions, variables, and their relationship. The abstract syntax for a logical model may also specify constructs to define logical entities and their relationships. A well-formed model abides by the rules of construction, just as a well formed sentence must conform to the grammatical rules of the natural language.

The concrete syntax (glossary) specifies the symbols used to express the model constructs. The natural language English can be expressed in text or Morse code. A modeling language may be expressed using graphical symbols and/or text statements. For example, a functional flow model may be expressed using graphical symbols consisting of a combination of graphical nodes and arcs annotated with text; while a simulation modeling language may be expressed using a programming language text syntax such as the C programming language.

The semantics (glossary) of a language define the meaning of the constructs. For example, an English word does not have explicit meaning until the word is defined. Similarly, a construct that is expressed as a symbol, such as a box or arrow on a flow chart, does not have meaning until it is defined. The language must give meaning to the concept of a verb or noun, and must give specific meaning to a specific word that is a verb or noun. The definition can be established by providing a natural language definition, or by mapping the construct to a formalism whose meaning is defined. As an example, a graphical symbol that expresses $\sin(x)$ and $\cos(x)$ is defined using a well-defined mathematical formalism for the sine and cosine function. If the position of a pendulum is defined in terms of $\sin(\theta)$ and $\cos(\theta)$, the meaning of the pendulum position is understood in terms of these formalisms.

Modeling Tools

Models are created by a modeler using modeling tools (glossary). For physical models, the modeling tools may include drills, lathes, and hammers. For more abstract models, the modeling tools are typically software programs running on a computer. These programs provide the ability to express modeling constructs using a particular modeling language. A word processor can be viewed as a tool used to build text descriptions using natural language. In a similar way, modeling tools are used to build models using modeling languages. The tool often provides a tool pallet to select symbols and a content area to construct the model from the graphical symbols or other concrete syntax. A modeling tool typically checks the model to evaluate whether it conforms to the rules of the language and enforces such rules to help the modeler create a well-formed model. This is similar to the way a word processor checks the text to see that it conforms to the grammar rules for the natural language.

Some modeling tools are commercially available products, while others may be created or customized to provide unique modeling solutions. Modeling tools are often used as part of a broader set of engineering tools which constitute the systems development environment. There is increased emphasis on tool support for standard modeling languages that enable models and modeling information to be interchanged among different tools.

Relationship of Model to Model-Based Systems Engineering

The INCOSE Systems Engineering Vision 2020 defines (MBSE) (glossary) as "the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" (2007). In MBSE, the models of the system are primary artifacts of the systems engineering process, and are managed, controlled, and integrated with other parts of the system technical baseline. This contrasts with the traditional document-centric approach to systems engineering, where text-based documentation and specifications are managed and controlled. Leveraging a model-based approach to systems engineering is intended to result in significant improvements in system specification and design quality, lower risk and cost of system development by surfacing issues early in the design process, enhanced productivity through reuse of system artifacts, and improved communications among the system development team.

In addition to creating models, the MBSE approach typically includes methods for model management (glossary) which aim to ensure that models are properly controlled and methods for model validation (glossary) which aim to ensure that models accurately represent the systems being modeled.

The jointly sponsored INCOSE/Object Management Group MBSE Wiki (http://www.omgwiki.org/MBSE/doku.php) provides additional information on the INCOSE MBSE Initiative.

Brief History of System Modeling Languages and Methods

Many system modeling methods and associated modeling languages have been developed and deployed to support various aspects of system analysis, design, and implementation. Functional modeling languages include the data flow diagram (DFD) (Yourdon and Constantine 1976), IDEF0 (Menzel and Maier 1998), and enhanced functional flow block diagram (EFFBD). Other behavioral modeling techniques include the classical state transition diagram, statecharts (Hare 1987), and process flow diagrams. Structural modeling techniques include data structure diagrams (Jackson 1975), entity relationship diagrams (Chen 1976), and object modeling techniques (Rumbaugh et al. 1991), which combine object diagrams, DFDs, and statecharts.

Some of the recent system modeling methods and languages evolved from these roots and are highlighted in A Survey of Model-Based Systems Engineering (MBSE) Methodologies (Estefan 2008). This survey identifies several candidate MBSE methodologies that can be applied to support an MBSE approach. The modeling standards section refers to some of the standard system modeling languages and other modeling standards that support MBSE.

References

Works Cited

Bellinger, G. 2004. *Modeling & Simulation: An Introduction*. Available at http://www.systems-thinking.org/modsim/modsim.htm.

Boehm, B. and W. May. 1988. "A Spiral Model of Software Development and Enhancement." IEEE *Computer* 21(5): 61-72.

Chen, P. 1976. "The Entity Relationship Model – Toward a Unifying View of Data." ACM *Transactions on Data Base Systems* 1(1): 9-36.

DoD. 1994. *DoD Modeling and Simulation (M&S) Management*. DoD Directive 5000.59. Arlington, VA: US Department of Defense. January 4, 1994.

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

Estefan, J. 2008. "A Survey of Model-Based Systems Engineering (MBSE) Methodologies," rev. B, Seattle, WA, USA: International Council on Systems Engineering. INCOSE-TD-2007-003-02. Available at http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf.

Friedenthal, S., A. Moore, and R. Steiner. 2009. *A Practical Guide to SysML: The Systems Modeling Language*. Needham, MA: OMG Press.

Guizzardi, G. 2007. "On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models" Proceedings of Seventh International Baltic Conference. Amsterdam, The Netherlands. Available at http://portal.acm.org/citation.cfm?id=1565425.

Harel, D. 1987. "Statecharts: A Visual Formalism for Complex Systems". *Science of Computer Programming*. 8(3): 231–74.

INCOSE. 2007. "Systems Engineering Vision 2020" , Seattle, WA, USA: International Council on Systems Engineering. INCOSE-TP-2004-004-02. Available at http://www.incose.org/ProductsPubs/products/sevision2020.aspx.

Jackson, M. A. 1975. *Principles of Program Design* New York, NY, USA: Academic Press.

Object Management Group. 2010. *MDA Foundation Model*. OMG document number ORMSC/2010-09-06.

Menzel, C. and R.J. Mayer. 1998. "The IDEF Family of Languages." In P. Bernus, K. Mertins, and G. Schmidt (eds.). *Handbook on Architectures for Information Systems*. Berlin, Germany: Springer-Verlag. p. 209-241.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson. 1990. *Object-Oriented Modeling and Design* Upper Saddle River, NJ: Prentice Hall.

Yourdon and Constantine. 1976. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Upper Saddle River, NJ: Prentice Hall.

Primary References

Guizzardi, G. 2007. "On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models." Proceedings of Seventh International Baltic Conference. Amsterdam, The Netherlands. Available at http://portal.acm.org/citation.cfm?id=1565425.

Estefan, J. 2008. *A Survey of Model-Based Systems Engineering (MBSE) Methodologies*, Rev. B. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TD-2007-003-02. Available at http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf.

INCOSE. 2007. *Systems Engineering Vision 2020*. Seattle, WA, USA: International Council on Systems Engineering. September 2007. INCOSE-TP-2004-004-02. Available at http://www.incose.org/ProductsPubs/products/sevision2020.aspx.

Additional References

Chen, P. 1976. "The Entity Relationship Model – Toward a Unifying View of Data." ACM *Transactions on Data Base Systems* 1(1): 9-36.

Downs, E., P. Clare, and I. Coe. 1992. *Structured Systems Analysis and Design Method: Application and Context*. Hertfordshire, UK: Prentice-Hall International.

Eisner, H. 1988. *Computer-Aided Systems Engineering*. Englewood Cliffs, NL, USA: Prentice Hall.

Harel, D. 1987. "Statecharts: A Visual Formalism for Complex Systems". *Science of Computer Programming*. 8(3): 231–74.

Jackson, M. A. 1975. *Principles of Program Design*. New York, NY, USA: Academic Press.

Kossiakoff, A. and W. Sweet. 2003. "Chapter 14" in *Systems Engineering Principles and Practice*. New York, NY, USA: Wiley and Sons.

OMG. "MBSE Wiki." Accessed 11 September 2011. Available at: http://www.omgwiki.org/MBSE/doku.php.

Object Management Group. 2010. *MDA Foundation Model*. Needham, MA, USA: OMG Press. OMG document number ORMSC/2010-09-06.

Oliver, D. T. Kelliber, and J. Keegan. 1997. *Engineering Complex Systems with Models and Objects*. New York, NY, USA: McGraw-Hill.

< Previous Article | Parent Article | Next Article >

Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Why Model?

System models (glossary) can be used for many purposes. This topic highlights some of those purposes in support of model-based systems engineering (glossary) (MBSE), and provides indicators of an effective model.

Purpose of a Model

Models are representations that can aid in defining, analyzing, and communicating a set of concepts. System models are specifically developed to support analysis, specification, design, and verification of a system, as well as communicate certain information. One of the first principles of modeling is to clearly define the purpose of the model. Some of the purposes that models can serve throughout the system life cycle (glossary) are highlighted below.

- **Characterizing an existing system:** Many existing systems may be poorly documented, and modeling the system can provide a concise way to capture the existing system design. This information can then be used to facilitate maintaining the system or to assess the system with the goal of improving it. This is analogous to creating an architectural model of an old building with overlays for electrical, plumbing, and structure before proceeding to upgrade it to withstand new earthquake standards.
- **Mission and system concept formulation and evaluation:** Models can be applied early in the system life cycle to synthesize and evaluate alternative mission and system concepts. This includes clearly and unambiguously defining the system's mission and the value it is expected to deliver to its beneficiaries. Models can be used to explore a trade-space by modeling alternative system designs and assessing the impact of critical system parameters such as weight, speed, accuracy, reliability, and cost on the overall measures of merit. In addition to bounding the system design parameters, models can also be used to validate that the system requirements meet stakeholder needs before proceeding with later life cycle activities such as synthesizing the detailed system design.
- **System design synthesis and requirements flowdown:** Models can be used to support architecting system solutions, as well as flow mission and system requirements down to system components. Different models may be required to address different aspects of the system design and respond to the broad range of system requirements. This may include models that specify functional, interface, performance, and physical requirements, as well as other non-functional requirements such as reliability, maintainability, safety, and security.
- **Support for system integration and verification:** Models can be used to support integration of the hardware and software components into a system, as well as to support verification that the system satisfies its requirements. This often involves integrating lower level hardware and software design models with system-level design models that verify system requirements are satisfied. System integration and verification may also include replacing selected hardware and design models with actual hardware and software products in order to incrementally verify that system requirements are satisfied. This is referred to as hardware-in-the-loop testing (glossary) and software-in-the-loop testing (glossary). Models can also be used to define the test cases (glossary) and other aspects of the test program to assist in test planning and execution.
- **Support for training:** Models can be used to simulate various aspects of the system to help train users to interact with the system. Users may be operators, maintainers, or potentially other stakeholders. Models may be a basis for developing a simulator (glossary) of the system with varying degrees of fidelity to represent user interaction in different usage scenarios.
- **Knowledge capture and system design evolution:** Models can provide an effective means for capturing knowledge about the system and retaining it as part of the organizational knowledge. This knowledge, which can be reused and evolved, provides a basis for supporting the evolution of the system, such as changing system requirements in the face of emerging, relevant technologies, new applications, and new customers.

Indicators of an Effective Model

When modeling is done well, a model's purposes are clear and well-defined. The value of a model can be assessed in terms of how

effectively it supports those purposes. The following three sections describe indicators of an effective model (Friedenthal, Moore, and Steiner 2009).

Model Scope

The model must be scoped to address its intended purpose. In particular, the types of models and associated modeling languages selected must support the specific needs desired to be met. For example, suppose models are constructed to support an aircraft's development. A system architecture model may describe the interconnection among the airplane parts, a trajectory analysis model may analyze the airplane trajectory, and a fault tree analysis model may assess potential causes of airplane failure.

For each type of model, the appropriate breadth, depth, and fidelity should be determined to address the model's intended purpose. The model breadth reflects the system requirements coverage in terms of the degree to which the model must address the functional, interface, performance, and physical requirements, as well as other non-functional requirements, such as reliability, maintainability, and safety. For an airplane functional model, the model breadth may be required to address some or all of the functional requirements to power up, takeoff, fly, land, power down, and maintain the aircraft's environment.

The model's depth indicates the coverage of system decomposition from the system context down to the system components. For the airplane example, a model's scope may require it to define the system context; ranging from the aircraft, the control tower, and the physical environment; down to the navigation subsystem and its components; such as the inertial measurement unit; and perhaps down to lower-level parts of the inertial measurement unit.

The model's fidelity indicates the level of detail the model must represent for any given part of the model. For example, a model that specifies the system interfaces may be fairly abstract and represent only the logical information content; such as aircraft status data, or it may be much more detailed to support higher fidelity information; such as the encoding of a message in terms of bits, bytes, and signal characteristics. Fidelity can also refer to the precision of a computational model, such as the time step required for a simulation.

Indicators of Model Quality

The quality of a model should not be confused with the quality of the design that the model represents. For example, one may have a high-quality, computer-aided design model of a chair that accurately represents the design of the chair, yet the design itself may be flawed such that when one sits in the chair, it falls apart. A high quality model should assist the design team in assessing the quality of the design and uncovering design issues.

Model quality is often assessed in terms of the adherence of the model to modeling guidelines and the degree to which the model addresses its intended purpose. Typical examples of modeling guidelines include naming conventions, application of appropriate model annotations, proper use of modeling constructs, and applying model reuse considerations. Specific guidelines are different for different types of models. For example, the guidelines for developing a geometric model using a computer-aided design tool may include conventions for defining coordinate systems, dimensioning, and tolerances.

Model-based Metrics

Models can provide a wealth of information that can be used for both technical and management metrics (glossary) to assess the modeling effort, and, in some cases, the overall systems engineering effort. Different types of models will provide different types of information. In general, models can provide information that enables one to:

- assess progress;
- estimate effort and cost;
- assess technical quality and risk; and
- assess model quality.

The models can be used to capture metrics that are similar to those captured in a traditional document-based approach to systems engineering, but potentially with more precision due to the more accurate nature of models when compared to documents. Traditional systems engineering metrics are described in *Metrics Guidebook for Integrated Systems and Product Development* (Wilbur 2005).

A model's progress can be assessed in terms of the completeness of the modeling effort relative to the defined scope of the model. Models may also be used to assess progress in terms of the extent to which the requirements have been satisfied by the design or verified through testing. When augmented with productivity metrics, the model can be used to estimate the cost of performing the required systems engineering effort to deliver the system.

Models can be used to identify critical system parameters and assess technical risks in terms of any uncertainty that lies in those parameters. The models can also be used to provide additional metrics that are associated with its purpose. For example, when the model's purpose is to support mission and system concept formulation and evaluation, then a key metric may be the number of alternative concepts that are explored over a specified period of time.

References

Works Cited

Friedenthal, S., A. Moore, and R. Steiner. 2009. "Chapter 2" in *A Practical Guide to SysML: The Systems Modeling Language*. Needham, MA, USA: OMG Press.

Wilbur, A., G. Towers, T. Sherman, D. Yasukawa, and S. Shreve. 2005. *Metrics Guidebook for Integrated Systems and Product Development*. Seattle, WA, USA: International Council on Systems Engineering (INCOSE). INCOSE-TP-1995-002-01. Available at http://www.incose.org/ProductsPubs/products/metricsguidebook.asp.

Primary References

Friedenthal, S., A. Moore, and R. Steiner. 2009. "Chapter 2" in *A Practical Guide to SysML: The Systems Modeling Language*. Needham, MA, USA: OMG Press.

Wilbur, A., G. Towers, T. Sherman, D. Yasukawa, and S. Shreve. 2005. *Metrics Guidebook for Integrated Systems and Product Development*. Seattle, WA, USA: International Council on Systems Engineering (INCOSE). INCOSE-TP-1995-002-01. Available at http://www.incose.org/ProductsPubs/products/metricsguidebook.asp.

Additional References

No additional references have been identified for SEBoK 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Types of Models

There are many different types of models (glossary) expressed in a diverse array of modeling languages and tool sets. This article offers a taxonomy for the many model types and highlights how different models must work together to support broader engineering efforts.

Model Classification

There are many different types of models and associated modeling languages to address different aspects of a system. Since different models serve different purposes, a classification of models can be useful for selecting the right type of model for the intended purpose and scope.
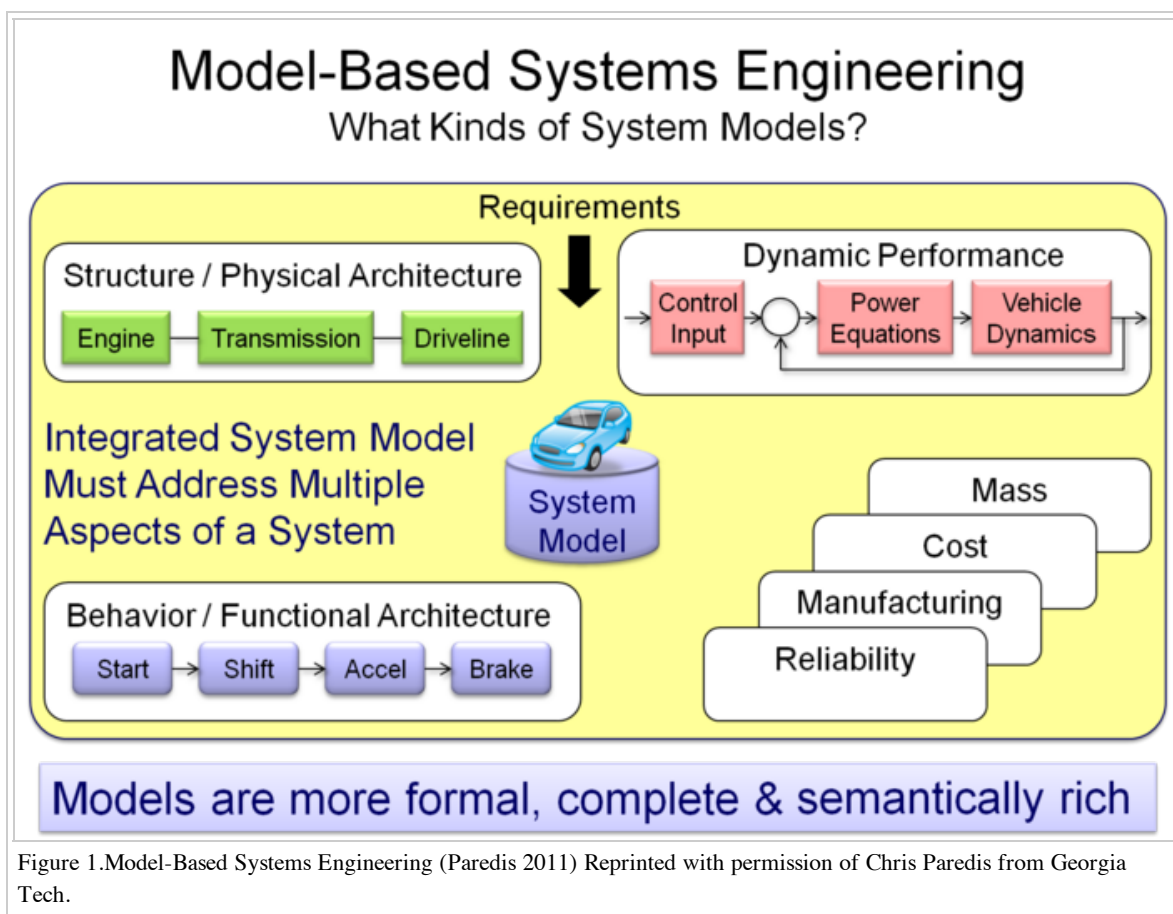
Formal versus Informal Models

Since a system model is a representation of a system, many different expressions that vary in degrees of formalism could be considered models. In particular, one could draw a picture of a system and consider it a model. Similarly, one could write a description of a system in text, and refer to that as a model. Both examples are representations of a system. However, unless there is some agreement on the meaning of the terms, there is a potential lack of precision and the possibility of ambiguity in the representation.

The primary focus of system modeling is to use models supported by a well-defined modeling language (glossary). While less formal representations can be useful, a model must meet certain expectations for it to be considered within the scope of model-based systems engineering (MBSE). In particular, the initial classification distinguishes between informal and formal models as supported by a modeling language with a defined syntax and the semantics for the relevant domain of interest.

Physical Models versus Abstract Models

The "Department of Defense Modeling and Simulation (M&S) Glossary" asserts that "a model can be [a] physical, mathematical, or otherwise logical representation of a system" (1998). This definition provides a starting point for a high level model classification. A physical model (glossary) is a concrete representation that is distinguished from the mathematical and logical models, both of which are more abstract representations of the system. The abstract model (glossary) can be further classified as descriptive (similar to logical) or analytical (similar to mathematical). Some example models are shown in Figure 1.

Figure 1.Model-Based Systems Engineering (Paredis 2011) Reprinted with permission of Chris Paredis from Georgia Tech.

Descriptive Models

A descriptive model (glossary) describes logical relationships, such as the system's whole-part relationship that defines its parts tree, the interconnection between its parts, the functions that its components perform, or the test cases that are used to verify the system requirements. Typical descriptive models may include those that describe the system architecture or computer-aided design models that describe the three dimensional geometric representation of a system.

Analytical Models

An analytical model (glossary) describes mathematical relationships, such as differential equations that support quantifiable analysis about the system parameters. Analytical models can be further classified into dynamic and static models. Dynamic models describe the time-varying state of a system, whereas static models perform computations that do not represent the time varying state of a system. A dynamic model may

represent the performance of a system, such as the aircraft position, velocity, acceleration, and fuel consumption over time. A static model may represent the mass properties calculation or a reliability prediction.

Hybrid Descriptive and Analytical Models

A particular model may include descriptive and analytical aspects as described above, but models may favor one aspect or the other. The logical relationships of a descriptive model can also be analyzed, and inferences can be made from reasoning about the system. Nevertheless, logical analysis provides different insights than a quantitative analysis of system parameters.

Domain-specific models

Both descriptive and analytical models can be further classified according to the domain that they represent. The following classifications are partially derived from the presentation on *OWL, Ontologies and SysML Profiles: Knowledge Representation and Modeling* (Jenkins 2010):

- properties of the system, such as reliability, mass properties, power, structural, or thermal models;
- design and technology implementations, such as electrical, mechanical, and software design models;
- subsystems and products, such as communications, fault management, or power distribution models; and
- system applications, such as information systems, automotive systems, aerospace systems, or medical device models.

A single model may include multiple domain categories from the above list. For example, a reliability, thermal, and/or power model may be defined for an electrical design of a communications subsystem for an aerospace system, such as an aircraft or satellite.

System Models

System models can be hybrid models that are both descriptive and analytical. They often span several modeling domains that must be integrated to ensure a consistent and cohesive system representation. As such, the system model must provide both general-purpose system constructs and domain-specific constructs that are shared across modeling domains.

A system model is the conceptual model that describes and represents a system. A system comprises multiple views, such as planning, requirement (analysis), design, implementation, deployment, structure, behavior, input data, and output data views. These models are required to describe and represent all of these views.

Wayne Wymore is credited with one of the early efforts to formally define a system model using a mathematical framework in *A Mathematical Theory of Systems Engineering: The Elements* (1967). Wymore established a rigorous mathematical framework for designing systems in a model-based context. A summary of his work can be found in A Survey of Model-Based Systems Engineering (MBSE) Methodologies.

Simulation versus Model

The term simulation, or more specifically computer simulation (glossary), refers to an analytical model that can be executed by a computing infrastructure. The computer simulation includes the analytical model and the computing infrastructure, as well as the initial conditions required to execute the model. There are many different types of computer simulations. Computer simulations can be characterized based on the following characteristics:

- stochastic or deterministic;
- steady-state or dynamic;
- continuous or discrete; and
- local or distributed.

Simulations may often be integrated with actual hardware, software, and operators of the system, to evaluate how actual components and users of the system perform in a simulated environment.

In addition to representing the system and its environment, the simulation must provide efficient computational methods for solving the equations. Simulations may be required to operate in real time, particularly if there is an operator in the loop. Other simulations may be required to operate much faster than real time and perform thousands of simulation runs to provide statistically valid simulation results. Several computational and other simulation methods are described in Simulation Modeling and Analysis (Law 2007).

Visualization

Computer simulation results and other analytical results often need to be processed so they can be presented to the users in a meaningful way. Visualization techniques and tools are used to display the results in various visual forms, such as parametric relationships that may include a simple plot of the state of the system versus time. Another example of this occurs when the input and output values from several simulation executions are displayed on a response surface showing the sensitivity of the output to the input. Additional statistical analysis of the results

may be performed to provide probability distributions for selected parameter values. Animation is often used to provide a virtual representation of the system and its dynamic behavior. For example, animation can display an aircraft's three-dimensional position and orientation as a function of time, as well as project the aircraft's path on the surface of the Earth as represented by detailed terrain maps.

Integration of Models

Many different types of models may be developed as artifacts of a MBSE effort. Many other domain specific models are created for component design and analysis. The different descriptive and analytical models must be integrated in order to fully realize the benefits of a model-based approach. The role of MBSE as the models integrate across multiple domains is a primary theme in the INCOSE Systems Engineering Vision 2020 (2007).

As an example, system models can be used to specify the components of the system. The descriptive model of the system architecture may be used to identify and partition the components of the system and define their interconnection or other relationships. Analytical models for performance, physical, and other quality characteristics, such as reliability, may be employed to determine the required values for specific component properties to satisfy the system requirements. An executable system model (glossary) that represents the interaction of the system components may be used to validate that the component requirements can satisfy the system behavioral requirements. The descriptive, analytical, and executable system model must ensure they represent different facets of the same system.

The component designs must satisfy the component requirements that are specified by the system models. As a result, the component design and analysis models must have some level of integration to ensure the design model is traceable to the requirements model. The different design disciplines for electrical, mechanical, and software each create their own models that represent different facets of the same system as well. It is evident that the different models must be sufficiently integrated to ensure a cohesive system solution.

In order to support the integration, the models must establish semantic interoperability (glossary) to ensure that a construct in one model has the same meaning as a corresponding construct in another model. In addition to sharing common definitions when referring to the same thing, the information must also be exchanged from one modeling tool to another.

An approach to achieve semantic interoperability is to use model transformations (glossary) between different models. This approach defines a transformation to establish correspondence between the concepts in one model and the concepts in another. In addition to establishing correspondence, the tools must have a means to exchange the model data in order to share the information. There are multiple means for exchanging data between tools, including file exchange, use of application program interfaces (API), and a shared repository.

The use of modeling standards for modeling languages, model transformations, and data exchange is an important enabler to achieve integration across modeling domains.

References

Works Cited

DoD. 1998. "'DoD Modeling and Simulation (M&S) Glossary" in *DoD Directive 5000.59-M*. Arlington, VA, USA: US Department of Defense. January.

Rouquette, N. and S. Jenkins. 2010. *OWL Ontologies and SysML Profiles: Knowledge Representation and Modeling*. Proceedings of NASA-ESA PDE Workshop. June 2010. Available at http://www.congrex.nl/10m05post/presentations/pde2010-Jenkins.pdf.

Wymore, A. 1967. *A Mathematical Theory of Systems Engineering: The Elements*. New York, NY, USA: John Wiley.

Wymore, A. 1993. *Model-Based Systems Engineering*. Boca Raton, FL, USA: CRC Press.

Primary References

Law, A. 2007. *Simulation Modeling and Analysis*, 4th ed. New York, NY, USA: McGraw Hill.

Wymore, A. 1993. *Model-Based Systems Engineering*. Boca Raton, FL, USA: CRC Press.

Additional References

Estefan, J. 2008. *Survey of Candidate Model-Based Systems Engineering (MBSE) Methodologies,* Revision B. Pasadena, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TD-2007-003-02. Available at http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf.

INCOSE. 2007. *Systems Engineering Vision 2020*. Seattle, WA, USA: International Council on Systems Engineering. September 2007.

INCOSE-TP-2004-004-02. Available at http://www.incose.org/ProductsPubs/products/sevision2020.aspx.

Rouquette, N. and S. Jenkins. 2010. *OWL Ontologies and SysML Profiles: Knowledge Representation and Modeling*. Proceedings of the NASA-ESA PDE Workshop, June 2010. Available at http://www.congrex.nl/10m05post/presentations/pde2010-Jenkins.pdf.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

System Modeling Concepts

A system model (glossary) represents aspects of a system and its environment. As explained in the article Types of Models, there are many different types of models, reflecting the diverse purposes for which people build them. It is useful to have a common way to talk about the concepts underlying the many different types of models (e.g., many modeling techniques enable understanding system behavior (glossary), while others enable understanding system structure (glossary)). This article highlights several common systems modeling concepts.

Abstraction

Perhaps the most fundamental concept in systems modeling is abstraction (glossary), which concerns hiding unimportant details in order to focus on essential characteristics. Systems worth modeling have more details than can reasonably be modeled. Apart from the sheer size and structural complexity that a system may possess, a system may be behaviorally complex as well, with emergent properties, non-deterministic behavior, and other difficult-to-characterize properties. Consequently, models must focus on a few vital characteristics in order to be computationally and intellectually tractable. Modeling techniques address complexity through various forms of abstraction. For example, a model may assume that structural and other characteristics of many individual instances of a particular type of component are all the same, ignoring small order differences between individual instances that occur in real life. In that case, those differences are assumed to be unimportant to modeling the structural integrity of an aggregation of those components. Of course, if that assumption is wrong, then the model could lead to false confidence in that structural integrity. Two key concepts to model different levels of abstraction are (a) view and viewpoint and (b) black-box and white-box modeling, as described below. Different modeling languages and tools employ other techniques as well.

View and Viewpoint

IEEE 1471, a standard for architecture modeling, defines "view" and "viewpoint" as follows:

1. View (glossary): A representation of a whole system from the perspective of a related set of concerns
2. Viewpoint (glossary): A specification of the conventions for constructing and using a view; a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis

Even though IEEE 1471 is focused on architectural models, the concepts of view and viewpoint are general and could apply to models for other purposes as well. The viewpoint specifies the stakeholders and their concerns and provides the conventions for constructing a view to address those concerns. The view represents aspects of the system that address the stakeholder concerns. Models can be created to represent the different views of the system. A systems model should be able to represent multiple views of the system to address a range of stakeholder concerns. Standard views may include requirements, functional, structural, and parametric views, as well as a multitude of discipline-specific views to address system reliability, safety, security, and other quality characteristics.

Black-box and White-box Models

A very common abstraction technique is to model the system as a black-box (glossary), which only exposes the features of the system that are

visible from an external observer, and hide the internal details of the design. This includes stimulus response characteristics and other black box physical characteristics, such as the system mass or weight. A white-box (glossary) model of a system shows the internal structure and behavior of the system. Black box and white-box modeling can be applied to the next level of design decomposition in order to create a black-box and white-box model of each system component.

Concept Model

A concept model is the set of concepts used to describe models and the relationships between those concepts (e.g., view and viewpoint). It is, in effect, a language for talking about models. A system concept model is used to describe models about systems and includes its requirements (glossary), behavior (glossary), structure (glossary), and properties (glossary). In addition, a system concept model is accompanied by a set of definitions for each concept. Sometimes system concept models are defined using an Entity Relationship diagram or a UML class diagram.

A preliminary concept model for systems engineering (Systems Engineering Concept Model) was developed in support of the integration efforts between the development of the OMG Systems Modeling Language (SysML) and the ISO AP233 Data Exchange Standard for systems engineering (ISO 2010). The concept model was originally captured in an informal way but the model and associated concepts were rigorously reviewed by a broad representation from the systems engineering community, including members from the INCOSE, AP233 and SysML development teams.

A fragment from the top level Systems Engineering Concept Model is included in Figure 1. This model provides concepts for requirements, behavior, structure and properties of the system, as well as other concepts common to systems engineering and project management, such as Stakeholder (glossary). The concept model is augmented by a well-defined glossary of terms called the semantic dictionary. The concept model and semantic dictionary served as a key input to the requirements for the OMG Systems Modeling Language that was called the UML for Systems Engineering Request for Proposal.
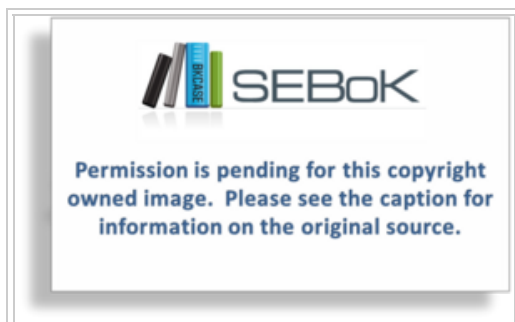

Figure 1. Fragment of the object management group system concept model. (INCOSE 2003, Slide 3) Source available at http://www.incose.org/practice/techactivities/wg/mdsd

A concept model is sometimes referred to as a meta-model (glossary), domain meta-model, or schema, and can be used to specify the abstract syntax of a modeling language (refer to the MDA Foundation Model (OMG 2010)). Several other systems engineering concept models have been developed but not standardized. Future standardization efforts should establish a standard systems engineering concept model. The model can then evolve over time as the systems engineering community continues to formalize and advance the practice of systems engineering.

References

Works Cited

ISO. 2010. *OMG System Modeling Language (OMG SysML)*, version 1.2. Needham, MA, USA: Object Management Group.

OMG. 2010. *MDA Foundation Model*. Needham, MA, USA: Object Management Group. Document number ORMSC/2010-09-06.

Primary References

ANSI/IEEE. 2000. *Recommended Practice for Architectural Description for Software-Intensive Systems*. New York, NY, USA: American National Standards Institute (ANSI)/Institute of Electrical and Electronics Engineers (IEEE), ANSI/IEEE 1471-2000.

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

Guizzardi, G. 2007. "On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models". Proceeding of the 2007 Conference on Databases and Information Systems IV. Available at http://portal.acm.org/citation.cfm?id=1565425.

INCOSE. 2003. *Systems Engineering Concept Model. Draft 12 Baseline*. Seattle, WA: *International Council on Systems Engineering*. Available at http://syseng.omg.org/SE_Conceptual%20Model/SE_Conceptual_Model.htm.

OMG. 2003. *UML for Systems Engineering Request for Proposal*. Needham, MA: Object Management Group. OMG document number ad/2003-3-41. Available at http://www.omg.org/cgi-bin/doc?ad/2003-3-41.

Additional References

OMG. 2010. *MDA Foundation Model*. Needham, MA, USA: Object Management Group. Document number ORMSC/2010-09-06.

< Previous Article | Parent Article | Next Article >

Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Modeling Standards

Different types of models (glossary) are needed to support the analysis, specification, design, and verification of systems. The evolution of modeling standards enables the broad adoption of Model-Based Systems Engineering (MBSE).

Motivation for Modeling Standards

Modeling standards play an important role in defining agreed-upon system modeling concepts that can be represented for a particular domain of interest, and they enable integration of different types of models across domains of interest. Modeling standards are extremely important to support MBSE, which must integrate across disciplines, products, and technologies.

Standards for system modeling languages also can enable cross-discipline, cross-project, and cross-organization communication. This communication offers the potential to reduce training requirements for practitioners who need to learn about a particular system, and enables the reuse of system artifacts. Standard modeling languages also provide a common foundation for advancing the practice of systems engineering as do other Systems Engineering Standards.

Types of Modeling Standards

Many different standards apply to systems modeling. Modeling standards include standards for modeling languages, data exchange between models, and for transformation of one model to another to achieve semantic interoperability, as well as more general modeling standards. Each type of model can be used to represent different aspects of a system, such as representing the set of system components and their interconnections and interfaces, or representing a system to support performance analysis or reliability analysis.

Following is a partial list of representative modeling standards, including a common acronym for many and a reference where more information can be found.

Modeling Languages for Systems

Descriptive models. These standards apply to general descriptive modeling of systems.

- Functional Flow Block Diagram (FFBD) (Oliver, Kelliher, and Keegan 1997)

- Integration Definition for Functional Modeling (IDEF0) (NIST 1993)
- Object Process Diagrams (OPD) and Object Process Language (OPL) (Dori 2002)
- Systems Modeling Language (SysML) (OMG 2010a)
- Unified Profile for DoDAF and MODAF (UPDM) (OMG 2011e)
- Web ontology language (OWL) (W3C 2004b)

Analytical models and simulations. These standards apply to analytical models and simulations.

- Distributed Interactive Simulation (DIS) (IEEE 1998)
- High-Level Architecture (IEEE 2010)
- Modelica (Modelica Association 2010)
- Semantics of a Foundational Subset for Executable UML Models (FUML) (OMG 2011d)

Data Exchange Standards

These standards enable the exchange of information between models.

- Application Protocol for Systems Engineering Data Exchange (ISO 10303-233) (AP-233) (ISO 2005)
- Requirements Interchange Format (ReqIF) (OMG 2011c)
- XML Metadata Interchange (XMI) (OMG 2003a)
- Resource Description Framework (RDF) (W3C 2004a)

Model Transformations

These standards apply to transforming one model to another to support semantic interoperability.

- Query View Transformations (QVT) (OMG 2011b)
- SysML-Modelica Transformation (OMG 2010c)
- SysML-OPM Transformation (Grobshtein and Dori 2008)

General Modeling Standards

These standards provide general frameworks for modeling.

- Model-driven architecture (MDA®) (OMG 2003b)
- IEEE 1471-2000 - Recommended Practice for Architectural Description of Software-Intensive Systems (ANSI/IEEE 2000) (ISO/IEC 2007)

Other Domain-Specific Modeling Standards

Software design models

These standards apply to modeling application software and/or embedded software design.

- Architecture Analysis and Design Language (AADL) (SAE 2009)
- Modeling and Analysis for Real-Time and Embedded Systems (MARTE) (OMG 2009)
- Unified Modeling Language (UML) (OMG 2010b)

Hardware design models

These standards apply to modeling hardware design.

- VHSIC Hardware Description Language (VHDL) (IEEE 2008)

Business process models

These standards apply to modeling business processes.

- Business Process Modeling Notation (BPMN) (OMG 2011a)

References

Works Cited

ANSI/IEEE. 2000. Recommended Practice for Architectural Description for Software-Intensive Systems. New York, NY: American National Standards Institute (ANSI)/Institute of Electrical and Electronics Engineers (IEEE), ANSI/IEEE 1471-2000.

Grobshtein, Y. and Dori, D. 2008. "Evaluating Aspects of Systems Modeling Languages by Example: SysML and OPM". *Proceedings of the 2008 INCOSE Annual International Symposium*. Utrecht, Netherlands.

IEEE. 1998. *Distributed Interactive Simulation (DIS)*. Institute for Electrical and Electronic Engineers. Available at http://standards.ieee.org/develop/project/1278.2.html.

IEEE. 2008. *VHSIC hardware description language (VHDL)*. IEEE Standard 1076-2008. Institute of Electrical and Electronics Engineers. Available at http://standards.ieee.org/findstds/standard/1076-2008.html.

IEEE. 2010. *Standard for High Level Architecture, IEEE Standard 1516*. Institute for Electrical and Electronic Engineers. Available at http://standards.ieee.org/develop/intl/intlstds.html

ISO. 2005. *Application Protocol for Systems Engineering Data Exchange*. (ISO 10303-233). International Organization for Standardization. Available at http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=55257.

ISO/IEC. 2007. Systems and Software Engineering — Recommended Practice for Architectural Description of Software-intensive Systems, International Organization for Standardization/International Electrotechnical Commission, September 12, 2007. ISO/IEC 42010:2007. Available at http://www.iso.org/iso/catalogue_detail.htm?csnumber=45991.

Modelica Association. 2010. *Modelica® - A Unified Object-Oriented Language for Physical Systems Modeling, Language Specification, Version 3.2*. Modelica Association. Available at https://www.modelica.org/documents/ModelicaSpec32.pdf.

NIST. 1993. *Integration Definition for Functional Modeling (IDEF0)*. National Institute for Standards and Technologies. Available at http://www.idef.com/IDEF0.htm.

Oliver, D., Kelliher, T., and Keegan. J. 1997. *Engineering Complex Systems with Models and Objects*. New York, NY, USA: McGraw Hill. Available at http://www.incose.org/ProductsPubs/DOC/EngComplexSys.pdf.

OMG 2003a. *XML Metadata Interchange (XMI),* Version 1.1. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/XML/.

OMG. 2003b. *Model driven architecture (MDA®),* Version 1.0.1. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/mda.

OMG. 2009. *Modeling and Analysis for Real-Time and Embedded Systems (MARTE),* Version 1.0. Object Management Group. Available at http://www.omg.org/spec/MARTE/1.0/.

OMG. 2010a. *OMG Systems Modeling Language (SysML),* Version 1.2. Needham, MA, USA: Object Management Group. Available at http://www.sysml.org/docs/specs/OMGSysML-v1.2-10-06-02.pdf.

OMG. 2010b. *Unified Modeling Language™ (UML),* Version 2.. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/UML/.

OMG. 2010c. *SysML-Modelica Transformation Specification*, Beta Version. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/SyM/.

OMG. 2011a. *Business Process Modeling Notation (BPMN),* Version 2.0. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/BPMN/2.0/ .

OMG. 2011b. *Query View Transformations (QVT),* Version 1.1. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/QVT/1.1/.

OMG. 2011c. *Requirements Interchange Format (ReqIF),* Version 1.0.1. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/ReqIF/.

OMG. 2011d. *Semantics of a Foundational Subset for Executable UML Models (FUML),* Version 1.0. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/FUML/1.0/.

OMG. 2011e. *Unified Profile for DoDAF and MODAF (UPDM),* Version 1.1. Needham, MA, USA: Object Management Group. Available at http://www.omg.org/spec/UPDM/.

SAE. 2009. *Architecture Analysis & Design Language (AADL)*. Warrendale, PA, USA: SAE International. Available at

http://standards.sae.org/as5506a/.

W3C. 2004a. *Resource Description Framework (RDF),* Version 1.0. World Wide Web Consortium. Available at http://www.w3.org/RDF/.

W3C. 2004b. *Web ontology language. (OWL).* World Wide Web Consortium. Available at http://www.w3.org/2004/OWL.

Primary References

No primary references have been identified for version 0.75. Please provide any recommendations on primary references in your review.

Additional References

Dori, D. 2002. *Object-Process Methodology – A Holistic Systems Paradigm.* Berlin, Germany: Heidelberg; New York, NY, USA: Springer Verlag. http://www.amazon.com/Object-Process-Methodology-Dov-Dori/dp/3540654712

Friedenthal, S., A. Moore, and R. Steiner. 2009. "Chapter 1" in *A Practical Guide to SysML: The Systems Modeling Language.* Needham, MA, USA: OMG Press.

Fritzon, P. 2004. *Object-oriented modeling and simulation with Modelica 2.1.* New York, NY, USA: Wiley Interscience and IEEE Press.

Grobshtein, Y. and D. Dori. 2011. "Generating SysML Views from an OPM Model: Design and Evaluation" in *Systems Engineering*, DOI 10.1002/sys.20181. Available at: http://esml.iem.technion.ac.il/site/wp-content/uploads/2011/02/GeneratingSysMLViewsFromAnOPMModel.pdf.

Paredis, C. J. J., et al. 2010. "An overview of the SysML-modelica transformation specification". Proceedings of the 20th Annual International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July 2010, Chicago, IL.

Weilkiens, T. 2008. *Systems Engineering with SysML/UML.* Needham, MA, USA: OMG Press.

< Previous Article | Parent Article | Next Article >

Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Systems Approach

This Knowledge Area (KA) provides a guide to an approach to complex problems and opportunities based on Systems Science (glossary)and the application of Systems Thinking (glossary). This knowledge is not specific to Systems Engineering, but is part of a wider systems body of knowledge. We have not attempted to capture all of the system knowledge here, but to identify those aspects relevant to the systems engineering body of knowledge.

The framework of activities and principles described in the KA are mapped to the other sections of the SEBoK to provide a guide to the system foundations of Systems Engineering (glossary) practices.

Topics

The topics contained within this knowledge area include:

- Overview of the Systems Approach
- Engineered System Context

- Identifying and Understanding Problems and Opportunities
- Synthesizing Possible Solutions
- Analysis and Selection between Alternative Solutions
- Implementing and Proving a Solution
- Deploying, Using, and Sustaining Systems to Solve Problems
- Acquirer/Supplier Responsibility
- Applying the Systems Approach

What is a Systems Approach?

The term Systems Approach (glossary)defined by authors such as Churchman (Churchman 1979) is an aspect of Systems Thinking through which we consider a System from the view point of those outside its boundary. In particular, we can gain significant insight by understanding the "enemies" of viewing the system holistically. While this is an important foundation for the system thinking which underpins the Systems Approach, this Knowledge Area uses the term to describe both systemic understanding and action as discussed below.

The Systems Approach (glossary) is defined as a set of principles for applying Systems Thinking (glossary) to Engineered System (glossary) contexts. According to Jackson et al (2010, pp. 41-43), the Systems Approach is a problem solving paradigm. It is a comprehensive problem understanding and resolution approach based upon the principles of systems thinking; and utilizing the concepts and thinking tools of Systems Science (glossary) along with the concepts inherent in engineering problem solving. It incorporates a Holistic (glossary) systems view of the system that includes the larger context of the system, including engineering and operational environments, stakeholders, and full life cycle. This Systems Approach forms the foundation for the descriptions and standards which define the practices of Systems Engineering. Systems engineering-related competency models generally agree that a fully capable systems engineer must employ Systems Thinking when undertaking these practices.

According to Checkland (1999, 318), Systems Thinking is "*an epistemology which, when applied to human activity is based on basic ideas of systems.*"

Senge (1990) provides an expanded definition as follows: "*Systems thinking is a discipline for seeing wholes. It is a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static "snapshots." It is a set of general principles -- distilled over the course of the twentieth century, spanning fields as diverse as the physical and social sciences, engineering, and management. During the last thirty years, these tools have been applied to understand a wide range of corporate, urban, regional, economic, political, ecological, and even psychological systems. And systems thinking is a sensibility for the subtle interconnectedness that gives living systems their unique character.*"

These sources and others discussed in the Systems Thinking knowledge area associate at least three different ideas to Systems Thinking. The first is a set of concepts to assist in learning how to think in terms of systems; the second is a set of principles to ensure a holistic approach to problems whatever there domain and the third is a set of tools to support the approach. The Systems Approach utilises all three of these aspects of Systems Thinking.

In the Introduction to SEBoK Part 2 we make the following distinctions:

- Systems Thinking (glossary) is a fundamental set of ideas which encapsulate a way of thinking about something as a set of related systems.
- Systems Science (glossary) is an interdisciplinary field of science that studies the nature of complex systems in nature, society, and science. It aims to develop interdisciplinary foundations, which are applicable in a variety of areas, such as engineering, biology, medicine and social sciences. The outputs of this work can be summarized by:
    - A System of Systems Concept (glossary), defining ideas which can be used to better understand the systemic nature of a situation or thing, independent of its application domain
    - A number of Systems Methodologies , which organize these ideas into a problem exploration and/or solution creation approach. These methodologies are often based upon a particular world view or paradigm, which influences the approach they take and the models they use.
- Systems Approach (glossary) is a combination of the above into a generic problem resolution approach which spans the whole life of an Engineered System (glossary) context, as part of the exploration and resolution of one or more real world problem situations or opportunities.

Note, the definitions above have been produced for the SEBoK to provide a framework of ideas to help understand the relationships between systems ideas and the practice of Systems Engineering. While they are not inconsistent with the literature in outline some authors have used different combinations of terminology to express them. It is our aim for the above ideas to help SEBoK readers to understand the literature and its relevance to them, without becoming too confused by slight differences, or even contradictions, in the detailed language used.

This Knowledge Area describes the Systems Approach in the context of applying Systems Thinking to an engineered system through its whole life.

References

Works Cited

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

Churchman, C. West. 1979. The Systems Approach and Its Enemies. New York: Basic Books.

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?". INCOSE *Insight*. 12(4).

Jackson, S., D. Hitchins, and H. Eisner. 2010. "What is the Systems Approach?". INCOSE *Insight*. 13(1): 41-43.

Senge, P. M. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, Doubleday/Currency.

Primary References

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

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?". INCOSE *Insight*. 12(4).

Jackson, S., D. Hitchins, and H. Eisner. 2010. "What is the Systems Approach?". INCOSE *Insight*. 13(1): 41-43.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Senge, P. M. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, Doubleday/Currency.

Additional References

No additional reference have been identified for SEBoK 0.75. Please provide any recommendations on additional reference in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Overview of the Systems Approach

This article considers the question, "what is the systems approach and how does it relate to systems engineering (SE)?". This question is discussed in general terms here and expanded upon in the other articles, which detail parts of the approach. The final article in this knowledge area, Applying the Systems Approach, then returns to this question and considers the dynamic aspects of how the approach is used and how this relates in detail to elements of systems engineering.

Definition and Uses of Systems Approach

According to Ryan (2008), the systems approach and SE formed and grew somewhat independently, but are highly related and compatible. Both are based on the concepts of systems thinking (glossary) that produce a broader understanding of the challenges being faced, the environment, and all of the related parts of the solution. Good SE process and lifecycle descriptions are based upon the systems approach. The best systems engineers must be good systems thinkers and should understand and apply a systems approach to all challenges and problems that they face when conducting SE.

The systems approach is essential when reductionist assumptions (i.e., the whole system has properties derived directly from the properties of their components) no longer apply to the system of interest (SoI). Emergent properties at the system level, which cannot be derived from a summation of the subsystem properties, necessitate a holistic systems approach. The systems approach is often invoked in applications beyond product systems. For example, the systems approach can be used in the educational domain. According to Biggs (1993), the system of interest includes "the student, the classroom, the institution, and the community." In fact, as the founder of systems thinking and systems science, Ludvig von Bertalanffy (1968), points out, "Systems are everywhere."

The goal of the systems approach is to understand the organization of ideas and to view problems and solutions holistically. It views the organization of ideas as a compound concept that incorporates structure and behavior of related systems associated with a problem and its possible solutions. It has resulted in a loosely connected set of techniques, where each technique contributes some insight on systemic aspects of problem or solution systems and helps us deal with different aspects of complexity (glossary). Because no single technique provides a complete understanding, knowledge of the limits of applicability of individual techniques is central to any systems approach. Most "situation systems" and "respondent systems" (see Figure 1) that systems engineers deal with require a combination of both hard and soft methods. An integration of hard and soft methods is an important aspect of the systems approach.

Lawson (2010) describes the relationship among the systems approach, systems thinking, and SE as a mindset to "think" and "act" in terms of systems. Developing this mindset is promoted by several paradigms including the System Coupling Diagram (glossary), which includes the elements "situation system", "respondent system", and "system assets" (see Figure 1).
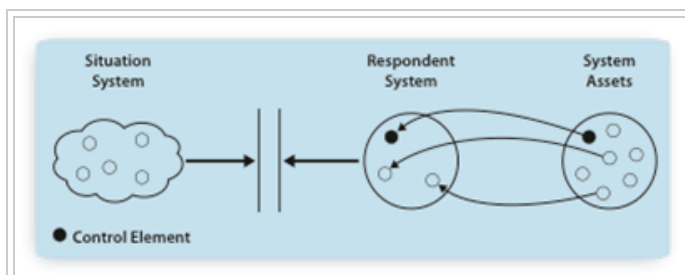

Figure 1. System Coupling Diagram (Lawson 2010) Reprinted with permission of Harold "Bud" Lawson.

- The situation system is the problem or opportunity situation, either unplanned or planned. The situation may be the work of nature, man-made, a combination of both natural and man-made, or a postulated situation that is to be used as a basis for deeper understanding and training (e.g., business games or military exercises).
- The respondent system is the system created to respond to the situation. The parallel bars indicate that this system interacts with the situation and transforms the situation into a new situation. Based on the situation that is being treated, a respondent system can have several names, such as project, program, mission, task force, or in a scientific context, experiment. One of the system elements of this system is a control element that directs the operation of the respondent system in its interaction with the situation. This element is based on an instantiation of a control system asset; e.g., a command and control system or a control process of some form.
- System assets are the sustained assets of one or more enterprises to be used in response to situations. System assets must be adequately managed throughout the life cycle so they will perform their function when instantiated in a respondent system. These assets are the primary objects for systems engineers. Examples of assets include value-added products or services, facilities, instruments and tools, and abstract systems, such as theories, knowledge, processes, and methods.

Lawson's model portrays one aspect of the systems approach and is applicable to understanding a problem; it organizes the resolution of that problem and creates and integrates any relevant solutions to enable that solution. Since the systems approach is a mindset prerequisite to SE, the premise is that projects and programs executed with this mindset are more likely to solve an idenfitied problem or take advantage of an identified opportunity.

The systems approach described in the SEBoK uses the following general problem understanding and resolution activities:

1. identify and understand the relationships between the potential problems and opportunities in a real world situation;
2. fully understand and describe a selected problem or opportunity in the context of its wider system and its environment;
3. synthesize viable system solutions to a selected problem or opportunity situation;
4. analyze and choose between alternative solutions for a given time/cost/quality version of the problem;
5. provide evidence that a solution has been correctly implemented and integrated; and
6. deploy, sustain, and use a solution to help solve the problem (or exploit the opportunity).

All of the above are considered within a life cycle (glossary) framework which may need concurrent, recursive (glossary) and Iterative (glossary) applications of some or all of the systems approach. Activities 1 and 6 are part of the business cycles of providing stakeholder value (Ring 2004) within an enterprise, whereas activities 2-5 can be mapped directly to product, service, and enterprise engineering life cycles. A distinction is made here between the normal business of an enterprise and the longer-term strategic activities of Enterprise Systems

Engineering.

When the systems approach is executed in the real world of an engineered system (glossary), a number of engineering and management disciplines emerge, including SE. SEBoK Parts 3 and 4 contain a detailed guide to SE with references to the principles of the systems approach as relevant.

SEBoK Part 5 provides a guide to the relationships between SE and the organizations, and Part 6 provides a guide to the relationship between SE and other disciplines. More detailed discussion of how the systems approach relates to these engineering and management disciplines is included in the Applying the Systems Approach topic in this knowledge area.

References

Works Cited

Biggs, J.B. 1993. "From Theory to Practice: A Cognitive Systems Approach". *Journal of Higher Education & Development*. Available from http://www.informaworld.com/smpp/content~db=all~content=a758503083.

Boardman, J. and B. Sauser 2008. *Systems Thinking - Coping with 21st Century Problems*. Boca Raton, FL, USA: CRC Press.

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

Edson, R. 2008. *Systems Thinking. Applied. A Primer*. Arlington, VA, USA: Applied Systems Thinking (ASysT) Institute, Analytic Services Inc.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Senge, P.M. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY, USA: Doubleday/Currency.

Ryan, A. 2008. "What is a Systems Approach?" Journal of Non-linear Science.

Ring J. 2004. "Seeing an Enterprise as a System". INCOSE *Insight*. 6(2) (January 2004): 7-8.

vonBertalanffy, L. 1968. General Systems Theory. New York, Ny, USA: George Braziller, Inc.

Primary References

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

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

Senge, Peter. M. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York: Doubleday/Currency.

Additional References

Biggs, J.B. 1993. "From Theory to Practice: A Cognitive Systems Approach". *Journal of Higher Education & Development*.. Available from http://www.informaworld.com/smpp/content~db=all~content=a758503083.

Edson, R. 2008. *Systems Thinking. Applied. A Primer*. Arlington, VA, USA: Applied Systems Thinking (ASysT) Institute, Analytic Services Inc.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Engineered System Context

Engineered Systems

The ideas of both an engineered system (glossary) and a systems context (glossary) are discussed in the Systems Thinking knowledge area.

The single most important principle of the systems approach (glossary) is that it is applied to an engineered system context and not to a single system (INCOSE 2011). The systems approach includes models and activities useful for the understanding, creation, use, and sustainment of engineered systems to enable the realization of stakeholder needs. Disciplines that use a systems approach (such as systems engineering (SE)) consider an engineered system context that defines stakeholder needs and looks for the best ways to provide value by applying managed technical activities to one or more selected engineered systems of interest (SoI) (glossary).

Generally, three specific types of engineered system contexts are recognized in SE: product systems (glossary), service systems (glossary), and enterprise systems (glossary). One of the key distinctions between product, service, and enterprise SE is how widely the SoI boundary is drawn.

Engineered System Context

We use the idea of an engineered system context to define an engineered system of interest, and to capture and agree on the important relationships between it, the systems it works directly with, and the systems which influence it in some way. All applications of a systems approach (and hence of SE) are applied to a system context, and not to an individual system.

A system context can be constructed around the following set of open system relationships (Flood and Carson 1993):

- The **Narrower System of Interest** (NSoI) is the system of direct concern to the observer. The focus of this system is driven by the scope of authority or control with implicit recognition that this scope may not capture all related elements.
- The **Wider System of Interest** (WSoI) describes a logical system boundary containing all of the elements needed to fully understand system behavior. The observer may not have authority over all of the elements in the WSoI, but will be able to agree upon the relationships between WSoI elements and NSoI elements.
- The WSoI sits in an **environment**. The immediate environment contains engineered, natural, or social systems with which the WSoI (and thus some elements of the NSoI) directly interacts with, exchanging material, information, and/or energy to achieve its goals or objective.
- A **Wider Environment** completes the context and contains systems which have no direct interaction with the SoI, but which might influence decisions related to it during its life cycle.
- "Some Theoretical Considerations of Mathematical Modeling" (Flood 1987) extends this context to include a **Meta-System** (MS) which sits outside of the WSoI and exercises direct control over it.

The choice of the SoI boundary for particular activities depends upon what can be changed and what must remain fixed. The SoI will always include one or more NSoI but may also include WSoI and MS if appropriate, such as when considering a service or enterprise system.

How is a system context applied to different kinds of engineered system problems?

For lower level, less complex, systems the WSoI can represent levels of product system (glossary) hierarchy, e.g., an engine management unit as part of an engine, or an engine as part of a car.

The WSoI in a system context may encapsulate some aspects of system of systems (SoS) (glossary) ideas for sufficiently complex (glossary) systems. In these cases, the WSoI represents a collection of systems with their own objectives and ownership which the NSoI must cooperate with towards a shared goal, e.g., a car and driver contributing to a transport service (glossary).

This view of a system of systems context as a means to support the engineering of a NSoI product system (glossary) is one way in which a systems approach can be applied. We can also apply it directly to the system of systems, e.g., a flexible, multi-vehicle transport service, or transport as part of a commercial retail enterprise (glossary). In this case, the NSoI aspect of the context no longer applies. The WSoI will consist of a set of cooperating systems, each of which might be changed or replaced to aid in the synthesis of a solution. The context may also need to represent **loose coupling** with some systems moving in or out of the context depending on the need, or **late binding** with systems joining the context only at, or close to, delivery of the service.

Thus, a context allows an observer to take a reductionist view of the SoI he/she has direct concern for and authority over while providing the system relationships and influences needed to enable them to keep a holistic (glossary) view of the consequence of any actions they take.

Product System Contexts

The distinction between a product (glossary) and a product system (glossary) is discussed in Types of Systems.

A product system context would be one in which the SoI is the product itself. The Wider System context for a product system can be a higher level of product hierarchy or a service or an enterprise system which uses the product directly to help provide value to the user.

A significant aspect of a product systems context is a clear statement of how the product is intended to be used and its actual delivery to the acquirer. The customer will be required to accept the system, typically through a formal process, against the agreed use.

If we apply a systems approach to a product context, we do it with the purpose of engineering a narrow system product to be integrated and used in a wider system product hierarchy, or to enable the delivery of a wider system service directly to a user by an enterprise.

Note, this view of the relationship between product and service is specific to Product Systems Engineering. While some engineering of the acquirer's static service system may occur, this is done with a product focus. The definition of service system, as associated with Service Systems Engineering, describes a more dynamic view of service systems. This is discussed more fully in the next section.

Service System Context

Services are activities that cause a transformation of the state of an entity (people, product, business, and region or nation) by mutually agreed terms between the service provider and the customer (Spohrer 2008). The distinction between service and a service system is discussed in Types of Systems.

A service system context is one in which the SoI is the Service system. This SoI contains all of the technology, infrastructure, people, resources, etc. needed to enable the service. The wider SoI describes the enterprise providing the service and its relationship with other services to provide enterprise success.

If we apply a Systems Approach to a service system we do it with the purpose of engineering a service system to enable the outcomes required by an enterprise to satisfy its clients. When operating in the service system context, we should be free to consider all options to provide the service, providing they fit within the constraints of the enterprise. This will include interfaces to other services, people, and resources in the enterprise. If an option for providing the service makes use of existing products or resources within or outside of the enterprise, we must ensure they are available for this use, and that this does not adversely affect other services. Part of getting the right service may require the negotiation of changes to the wider enterprise context, but this must be by agreement with the relevant authority.

For a service system, and when considering the service system context, the value is realized only through service transactions. The end-user co-creates value at the time of the request to use the service. One may have a smart phone (the product) but if one wants to make a flight reservation, the service system is composed of many service system entities (the caller, the person called, the smart phone, the access network, the core Internet Protocol (IP) network, the Internet Service provider (ISP), the World Wide Web, data centers, etc.) that are all necessary to enable the service. When one makes a reservation and then books the flight, the value has been created.

This definition of a service system, as associated with dynamic IT services, is discussed more fully in the Service Systems Engineering topic of Part 4.

Enterprise Systems Context

The distinction between an enterprise and an enterprise system is discussed in Types of Systems.

An enterprise system context is one in which the SoI is the enterprise system. This system contains all of the technology, infrastructure, people, resources, etc. needed to enable the service. The wider SoI describes the business environment within which the enterprise sits.

It is worth noting that an enterprise context is not equivalent to an "organization" according to this definition. This is a frequent misuse of the term enterprise. An enterprise includes not only the organizations that participate in it, but also the people, knowledge, and other assets, such as processes, principles, policies, practices, doctrine, theories, beliefs, facilities, land, and intellectual property that compose the enterprise.

An enterprise may contain or employ service systems along with product systems. An enterprise might even contain sub-enterprises.

Enterprise systems are unique when compared to product and service systems in that they are constantly evolving, they rarely have detailed configuration controlled requirements, they typically have goals of providing shareholder value and customer satisfaction (these goals are constantly changing and are difficult to verify), and they exist in a context (or environment) that is ill-defined and constantly changing. The enterprise systems engineer must consider and account for these factors in their processes and methods.

Both product and service systems require an enterprise system context to create them and an enterprise to use the product system and deliver services, either internally to the enterprise or externally to a broader community. Thus, the three types of engineered system contexts are linked in all instances, regardless of which type of system the developers consider the object of the development effort and which is delivered to the customer.

References

Works Cited

Flood, R. L. 1987. "Some Theoretical Considerations of Mathematical Modeling." In Problems of Constancy and Change (proceedings of 31st Conference of the International Society for General Systems Research, Budapest), Volume 1, p. 354 - 360. Flood, R. L. and E.R. Carson. 1993. Dealing with Complexity: An Introduction to the Theory and Application of Systems Science, 2nd ed. New York, NY, USA: Plenum Press.

INCOSE. 2011. *INCOSE Systems Engineering Handbook*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2003-002-03.2.1.

Spohrer, J. 2008. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline-Outline & References." *International Journal of Information Systems in the Service Sector*. 1(3) (May).

Primary References

Chang, C.M., 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley and Sons.

Tien, J.M. and D. Berg. 2003. "A Case for Service Systems Engineering". *Journal of Systems Science and Systems Engineering*. 12(1): 13-38.

INCOSE. 2011. *INCOSE Systems Engineering Handbook*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2003-002-03.2.1.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press.

Rouse, W.B. 2005. "Enterprise as Systems: Essential Challenges and Enterprise Transformation". *Systems Engineering*. 8(2): 138-50.

Additional References

Martin J. N. 1997. *Systems Engineering Guidebook*. Boca Raton, FL, USA: CRC Press.

ANSI/EIA. 2003. *Processes for Engineering a System*. Philadelphia, PA, USA: American National Standards Institute (ANSI)/Electronic Industries Association (EIA). ANSI/EIA 632-1998.

Rouse, W.B. 2009. "Engineering the Enterprise as a System". In *Handbook of Systems Engineering and Management*, 2nd ed. A.P. Sage

and W.B. Rouse (eds.). New York, NY, USA: Wiley and Sons.

Bernus, P., L. Nemes, and G. Schmidt (eds.). 2003. *Handbook on Enterprise Architecture*. Heidelberg, Germany: Springer.

Valerdi, R. and D.J. Nightingale. 2011. "An Introduction to the Journal of Enterprise Transformation." *Journal of Enterprise Transformation* 1(1):1-6.

DeRosa, J. K. 2006. "An Enterprise Systems Engineering Model." Proceedings of the 16th Annual International Council on Systems Engineering, 9-13 July 2006, Orlando, FL, USA.

Chang, C.M., 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley and Sons.

Katzan, H. 2008. *Service Science*. Bloomington, IN, USA: iUniverse Books.

Maglio P., S. Srinivasan, J.T. Kreulen, and J. Spohrer. 2006. "Service Systems, Service Scientists, SSME, and Innovation." *Communications of the ACM*. 49(7) (July).

Tien, J.M. and D. Berg. 2003. "A Case for Service Systems Engineering". *Journal of Systems Science and Systems Engineering*. 12(1): 13-38.

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press.

Joannou, P. 2007. "Enterprise, Systems, and Software—The Need for Integration." IEEE *Computer*. 40(5) (May): 103-105.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Identifying and Understanding Problems and Opportunities

This article considers the activities of the systems approach (glossary) when related to the identification and exploration of problems or opportunities in detail. Any of the activities described below may need to be considered concurrently (glossary) with activities used in the systems approach. The final article in this knowledge area, Applying the Systems Approach, considers the dynamic aspects of how these activities are used as part of the systems approach and how this relates in detail to elements of systems engineering (SE).

The phrase "problem or opportunity" used herein recognizes that the "problem" is not always a negative situation and can also be a positive opportunity to improve a situation.

Introduction

According to Jenkins (1969), the first step in the systems approach is "the recognition and formulation of the problem." The systems approach described in the SEBoK is predominantly a hard system (glossary) approach. The analysis, synthesis, and proving parts of the approach assume a problem or opportunity has been identified and agreed upon and a "new" engineered system (glossary) solution is needed.

However, the systems approach does not have to apply to the development and use of a newly designed and built technical solution. Abstract or experimental solutions to potential problems might be explored to help agree on a problem context. Solutions may involve reorganizing

existing system of system contexts, or the modification or re-use of existing products and services. The problem and opportunity parts of the approach overlap with soft system (glossary) approaches. This is discussed in more detail below.

One dimension of system complexity (glossary) is that the opportunity situation may be difficult to fully understand, and that system solutions may not solve the problem the first time, but may increase the understanding of both problem issues and ideas regarding what solutions to try next.

Hence, problem exploration and identification is often not a one-time process that specifies the problem, but is used in combination with solution synthesis and analysis to progress towards a more complete understanding of problems and solutions over time (see Applying the Systems Approach for a more complete discussion of the dynamics of this aspect of the approach).

Problem Exploration

Soft system thinking does not look for "the problem", but considers a problematic situation. Forming systems views of this situation can help stakeholders better understand each other's viewpoints and provide a starting point for directed intervention in the current system context (glossary). If a full soft systems intervention, such as a soft systems methodology (SSM) (Checkland 1999), is undertaken it will not include formal analysis, synthesis, and Proving. However, the SSM method was originally based on hard methodologies, in particular one presented by Jenkins (1969). It follows the basic principles of a systems approach: "analyzing" conceptual models of shared understanding, "synthesizing" intervention strategies, and "proving" improvements in the problematic situation.

Often, the distinction between hard and soft methods is not as clear cut as the theory might suggest. Checkland himself has been involved in applications of SSM as part of the development of information system design (Checkland and Howelwell 1998). It is now agreed upon by many that while there is a role for a "pure soft system" approach, the service (glossary) and enterprise (glossary) problems now being tackled can only be dealt with successfully by a combination of soft problematic models and hard system solutions. Mingers and White (2009) give a number of relevant examples of this. In particular they reference "Process and Content: Two Ways of Using SSM" (Checkland and Winters 2006). Thus, it might be expected that future engineered system problems will be stated, solved, and used as part of a predominately soft intervention, placing pressures on the speed of development needed in the solution space. This is discussed more fully in the article Life Cycle Models.

The critical systems thinking (glossary) and multi-methodology approaches (Jackson 1985) take this further by advocating a "pick and mix" approach within which the most appropriate models and techniques are chosen to fit the problem rather than following a single methodology (Mingers and Gill 1997). Thus, even if the hard problem identification approach described below is used, some use of the soft system techniques (such as rich pictures, root definitions, or conceptual models) should be considered within it.

Problem Identification

Hard system thinking is based on the idea that a problem exists which can be stated by one or more stakeholders in an objective way. This does not mean that hard systems approaches start with a defined problem. Exploring the potential problem with key stakeholders is still an important part of the approach.

According to Blanchard and Fabrycky (2006, 55-56), defining a problem is sometimes the most important and difficult step. In short, a system cannot be defined unless it is possible to clearly describe what it is supposed to accomplish.

According to Edson (2008, 26-29), there are three kinds of questions that need to be asked to ensure we fully understand a problem situation. How difficult or well understood is the problem? Problems can be "tame," "regular," or "wicked." The answer to this question will help define the tractability of the problem.

- For tame problems, the solution may be well defined and obvious.
- Regular problems are those that are encountered on a regular basis. Their solutions may not be obvious, thus serious attention should be given to every aspect of them.
- Wicked problems (glossary) (Rittel and Webber 1973) cannot be fully solved, or perhaps even fully defined, and it is not possible to understand the full effect of applying systems to the problem.

Who or what is impacted? There may be elements of the situation that are causing the problem, elements that are impacted by the problem, and elements that are just in the loop. Beyond these factors, what is the environment and what are the external factors that affect the problem? In examining these aspects, the tools and methods of systems thinking can be productively applied.

What are the various viewpoints of the problem? Does everyone think it is a problem? Perhaps there are conflicting viewpoints. All these viewpoints need to be defined. Persons affected by the system, who stand to benefit from the system, or can be harmed by the system, are called stakeholders. Wasson (2006, 42-45) provides a comprehensive list of stakeholder types. The use of soft systems models, as discussed above, can inform an important part of this. Describing a problem using situation views can be useful when considering these issues, even if a single problem perspective is selected for further consideration.

Operations research (glossary) is a hard systems method which concentrates on solving problem situations by deploying known solutions. The problem analysis step of a typical approach asks questions about the limitation and cost of the current system to identify efficiency improvements that need to be made (Flood and Carson 1993).

Traditional systems engineering (SE) (glossary) methods tend to focus more on describing an abstract model of the problem, which is then used to develop a solution in terms of the benefits stakeholders expect to see (Jenkins 1969). The expectation is often that a new solution must be created, although this need not be the case. Jenkins suggests that SE is just as applicable to a redesign of existing systems. A clear understanding of stakeholder expectations in this regard should inform an understanding of part of the problem. Do stakeholders expect a new solution or modifications to their existing solutions, or are they genuinely open to solution alternatives which consider the pros and cons of either. Such expectations will inform the suggestion of solution alternatives, as discussed in the Synthesizing Possible Solutions article.

An important factor in defining the desired stakeholder outcomes, benefits, and constraints is the operational environment, or scenario, in which the problem or opportunity exists. Armstrong (2009, 1030) suggests two scenarios: the first is the descriptive scenario, or the situation as it exists now, and the second is the normative scenario, or the situation as it may exist sometime in the future.

All of these aspects of problem understanding can be related to the concept of a system context.

Problem Context

The Engineered System Context article identifies a way by which a complex system situation can be resolved around a system of interest (SoI) (glossary). The initial identification of a "problem context" can be considered as the outcome of this part of the systems approach.

The systems approach should not consider only soft or hard situations. More appropriately, a problem or opportunity should be explored by aspects of both. In general, the application of the systems approach with a focus on engineered system contexts will lead to hard system contexts in which an identified SoI and required outcome can be defined.

An initial description of the wider SoI and environment serves as the problem or opportunity problem scope. Desired stakeholder benefits are expressed as outcomes in the wider system and some initial expression of what the SoI is intended for may be identified. Jenkins (1969) defines a problem formulation approach where one

- states the aim of the SoI;
- defines the wider SoI;
- defines the objectives of the wider SoI;
- defines the objectives of the system; and
- defines economic, informational, and other conditions.

In a hard system problem context, a description of a logical or ideal system solution may be included. This ideal system cannot be implemented directly, but describes the properties required of any realizable system solution.

To support this problem or opportunity description, a soft context view of the SoI will help ensure wider stakeholder concerns are considered. If a soft system context has been defined, it may include a conceptual model (Checkland 1999) which describes the logical elements of a system that resolve the problem situation and how they are perceived by different stakeholders. Unlike the hard system view, this does not describe the ideal solution, but provides an alternative view on how aspects of any solution would be viewed by potential stakeholders.

In problem contexts with a strong coercive (glossary) dimension, the problem context should include an identification of the relative power and the importance of stakeholders.

The problem context should include some boundaries on the cost, time to deployment, time in use, and operational effectiveness needed by stakeholders. In general we are describing both the full problem context, and an agreed version of the problem to be tackled next (see Applying the Systems Approach).

References

Works Cited

Armstrong, Jr., J.E., 2009. "Issue Formulation." In A.P. Sage and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management*, 2nd edition. Hoboken, NJ, USA: Wiley.

Blanchard, B. and W.J. Fabrycky. 2006. *Systems Engineering and Analysis*. Upper Saddle River, NJ, USA: Prentice Hall.

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

Checkland, P. and S. Holwell. 1998. *Information, Systems and Information Systems: Making Sense of the Field*. Hoboken, NJ, USA: Wiley.

Checkland, P. and M. Winter. 2006. "Process and Content: Two Ways of Using SSM". *Journal of Operational Research Society*. 57(12): 1435-1441.

Edson, R. 2008. *Systems Thinking*. *Applied*. *A Primer*. Arlington, VA, USA: Applied Systems Thinking (ASysT) Institute, Analytic Services Inc.

Flood, R. L. and E.R. Carson 1993. *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science,* 2nd ed. New York, NY, USA: Plenum Press.

Jackson, M. 1985. "Social Systems Theory and Practice: The Need for A Critical Approach". *International Journal of General Systems*. 10: 135-151.

Jenkins, G. M. 1969. The Systems Approach. The Journal of Systems Engineering 1 (1).

Mingers, J. and A. Gill. 1997. *Multimethodology: Theory and Practice of Combining Management Science Methodologies*. Chichester, UK: Wiley.

Mingers, J. and L. White. 2009. *A Review of Recent Contributions of Systems Thinking to Operational Research and Management Science*, Working Paper 197. Canterbury, UK: Kent Business School.

Rittel, H. and M. Webber. 1973. "Dilemmas in a General Theory of Planning." In *Policy Sciences*. 4:155–169.

Wasson, C. S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: Wiley.

Primary References

Blandhard, B. & W.J. Fabrycky. 2006. *Systems Engineering and Analysis*. Upper Saddle River, NJ, USA: Prentice Hall.

Edson, R. 2008. *Systems Thinking*. *Applied*. *A Primer*. Arlington, VA, USA: Applied Systems Thinking (ASysT) Institute, Analytic Services Inc.

Additional References

Armstrong, Jr., J.E., 2009. "Issue Formulation." In A.P. Sage and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management*, 2nd edition. Hoboken, NJ, USA: Wiley.

Blanchard, B. and W.J. Fabrycky. 2006. *Systems Engineering and Analysis*. Upper Saddle River, NJ, USA: Prentice Hall.

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

Checkland, P. and S. Holwell. 1998. *Information, Systems and Information Systems: Making Sense of the Field*. Hoboken, NJ, USA: Wiley.

Checkland, P. and M. Winter. 2006. "Process and Content: Two Ways of Using SSM". *Journal of Operational Research Society*. 57(12): 1435-1441.

Flood, R. L. and E.R. Carson 1993. *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science,* 2nd ed. New York, NY, USA: Plenum Press.

Jackson, M. 1985. "Social Systems Theory and Practice: The Need for A Critical Approach". *International Journal of General Systems*. 10: 135-151.

Jenkins, G. M. 1969. "The Systems Approach." In Beishon, J. and G. Peters (eds). *Systems Behaviour,* 2nd eidtion. New York, NY, USA: Harper and Row.

Mingers, J. and A. Gill. 1997. *Multimethodology: Theory and Practice of Combining Management Science Methodologies*. Chichester, UK: Wiley.

Mingers, J. and L. White. 2009. *A Review of Recent Contributions of Systems Thinking to Operational Research and Management Science*, Working Paper 197. Canterbury, UK: Kent Business School.

Wasson, C. S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: Wiley.

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "Exploring a Problem or Opportunity". No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Synthesizing Possible Solutions

This article considers the activities of the systems approach (glossary) related to the synthesis (glossary) of possible solution options. Any of the activities described below may need to be considered concurrently (glossary) with other activities in the systems approach. The final article in this knowledge area, Applying the Systems Approach, considers the dynamic aspects of how these activities are used as part of the systems approach and how this concept relates to elements of systems engineering (SE).

Synthesis Overview

System synthesis is an activity within the systems approach to describe one or more system solutions based upon a problem context:

1. to define options for a system of interest (SoI) (glossary) with the required properties and behavior for an identified problem or opportunity context;
2. to refine the SoI to the point where it can be judged to be potentially realizable within any time line or cost of risk described as part of the problem context; and
3. to assess the properties and behavior of each candidate solution in its wider system context.

Synthesis defines possible solutions and may make some overall judgment regarding the feasibility of said solutions. The detailed judgment on whether a solution is suitable for a given iteration of the systems approach is made using the Analysis and Selection between Alternative Solutions activities.

Essential to synthesis is the concept of holism (glossary) discussed by Hitchins (2009). It states that a system must be considered as a whole and not simply as a collection of its elements. The holism of any potential solution system requires that the properties of the whole be determined by considering the behavior of the whole and not simply the accumulation of the properties of the elements. The latter process is known as reductionism and is the opposite of holism. Hitchins (2009, 60) states: "The properties, capabilities, and behavior of a system derive from its parts, from interactions between those parts, and from interactions with other systems."

When the system is considered as a whole, properties called emergent properties often appear (see Emergence). These properties are often difficult to predict from the properties of the elements alone. They must be evaluated within the systems approach to determine the complete set of performance levels of the system. According to Jackson (2010), these properties can be considered in the design of a system, but to do so, an iterative approach is required.

In complex systems, individual elements will adapt to the behavior of the other elements and to the system as a whole. The entire collection of elements will behave as an organic whole. Therefore, the entire synthesis activity, particularly in complex systems, must itself be adaptive.

Hence, synthesis is often not a one-time process of solution design, but is used in combination with problem understanding and solution analysis to progress towards a more complete understanding of problems and solutions over time (see Applying the Systems Approach for a more complete discussion of the dynamics of this aspect of the approach).

Problem or Opportunity Context

System synthesis needs the problem or opportunity that the system is intended to address to have been identified and described or, more likely for any non-trivial system, the problem or opportunity needs to be identified and understood concurrently with solution synthesis activities.

As discussed in Identifying and Understanding Problems and Opportunities the systems approach should not consider strictly soft or hard situations. In general, the application of the systems approach with a focus on engineered system (glossary) contexts will lead to hard system (glossary) contexts in which an identified System of Interest (SoI) and required outcome can be defined. Even in these cases, a soft context view of the SoI context will help ensure wider stakeholder concerns are considered.

The problem context should include some boundaries on the cost, time to deployment, time in use, and operational effectiveness needed by stakeholders. In general, the goal is not to synthesize the perfect solution to a problem, but rather to find the best available solution for the agreed version of the problem.

Synthesis Activities

The following activities provide an outline for defining the SoI: identification of the boundary of a system, identification of the function of system elements, identification of system elements, division of elements into smaller elements, grouping of elements, and identification of the interactions among the elements.

The activities of systems synthesis are built on the idea of a balanced reduction vs. holism approach as discussed in What is Systems Thinking?. It is necessary to divide system elements and functions to create a description of the SoI which is realizable, either through combinations of available elements or through the design and build of new elements. However, if the system is simply decomposed into smaller and smaller elements, the holistic nature of systems will make it more and more difficult to predict the function and behavior of the whole. Thus, synthesis progresses through activities which divide, group, and allocate elements and then assess the resulting complete system's properties in context.

Identification of the Boundary of a System

Establishing the boundary of a system is essential to synthesis and the determination of the system's interaction with its environment and with other systems and the extent of the SoI. Buede (2009, 1102) provides a comprehensive discussion of the importance of, and methods of, defining the boundary of a system in a SE context.

Identification of the Functions of the System

The function of a system at a given level of abstraction is critical to synthesis since the primary goal of the synthesis activity is to propose realizable system descriptions which can provide a given function. The function of a system is distinct from its behavior as it describes what the system can be used for or is asked to do in a larger system context.

Buede (2009, 1091-1126) provides a comprehensive description of functional analysis in a SE context.

Identification of the Elements of a System

System synthesis calls for the identification of the elements of a system. Typical elements of an Engineered System Context may be physical, conceptual, or processes. Physical elements may be hardware, software, or humans. Conceptual elements may be ideas, plans, concepts, or hypotheses. Processes may be mental, mental-motor (writing, drawing, etc.), mechanical, or electronic (Blanchard and Fabrycky 2006, 7).

In addition to the operational elements of the system under consideration (i.e., a SoI), ISO/IEC 15288 (2008) also calls for the identification of the enabling systems. These are systems utilized at various stages in the life cycle, e.g., maintenance and other systems that support operational elements in order to solve problems or realize an opportunity.

Today's systems often include existing elements. It is rare to find a true "greenfield" system in which the developers can specify and implement all new elements from scratch. "Brownfield" systems, wherein legacy elements constrain the system structure, capabilities, technology choices, and other aspects of implementation, are much more typical (Boehm 2009).

Division of System Elements

System synthesis may require elements to be divided into smaller elements. The division of elements into smaller elements allows the systems to be grouped and leads to the SE concept of physical architecture as described by Levin (2009, 493-495). Each layer of division leads to another layer of the hierarchical view of a system. As Levin points out, there are many ways to depict the physical architecture, including wiring diagrams, block diagrams, etc. All of these views depend on arranging the elements and dividing them into smaller elements. According to the principle of recursion, these decomposed elements are either terminal elements, or are able to be further decomposed. The hierarchical view does not imply a top-down analytical approach to defining a system. It is simply a view. In the systems approach, levels of the hierarchy are defined and considered recursively with one level forming the context for the next.

Grouping of System Elements

System synthesis may require that elements can be grouped. This leads to the identification of the sub-systems essential to the definition of a system. Synthesis determines how a system may be partitioned and how each subsystem fits and functions within the whole system. The largest group is the SoI, also called the relevant system by Checkland (1999, 166). According to Hitchins (2009, 61), some of the properties of a SoI are as follows: the SoI is open and dynamic, the SoI interacts with other systems, and the SoI contains sub-systems. The SoI is brought together through the concept of synthesis.

Identification of the Interactions among System Elements

System synthesis may require the identification of the interactions among system elements. These interactions lead to the SE process of interface analysis. Integral to this aspect is the principle of interactions. Interactions occur both with other system elements, and also with external elements and the environment. In a systems approach, interfaces have both a technical and managerial importance. Managerial aspects include contracts between interfacing organizations. Technical aspects include the properties of the physical and functional interfaces. Browning (2009, 1418-1419) provides a list of desirable characteristics of both technical and managerial interface characteristics.

System synthesis will include activities to understand the properties of system elements, the structure of proposed system solutions, and the resultant behavior of the composed system. A number of system concepts for describing system behavior are discussed in System Concepts. Note: to fully understand a system's behavior, we must consider the full range of environments in which it might be placed and its allowable state in each. According to Page (2009), in complex systems, the individual elements of the system are characterized by properties which enhance the systems as a whole, such as their adaptability.

Defining the System of Interest

Flood and Carson (1993) identify two ways to identify system boundaries: a bottom-up, or **structural approach**, which starts with significant system elements and builds out, and a top down, or **behavioral approach**, in which major systems needed to fulfill a goal are identified and then the work flows downward. They identify a number of rules proposed by Beishon (1980) and Jones (1982) to help in the selection of the best approach.

In either case, the ways in which system elements are refined, grouped, and allocated must be driven towards the synthesis of a realizable system solution description. Note: a realizable solution must consider elements which are either already available, can be created from existing system elements, or are themselves described as system contexts which will need to be synthesized at a future point. In the third case, it is one of the outcomes of the Analysis and Selection between Alternative Solutions activities to assess the risk that a given element may not be able to be synthesized in the required timescales or cost budget.

A top down approach might start with a system boundary and an overall description of system functions. Through the repeated application of element identification, division, grouping, and allocation of functions, a complete description of the elements needed for the SoI can be defined. In this case, the choice of system elements and allocation of functions may be guided by pre-defined ways of solving a given problem or identified system patterns (reference needed). For example, one might start with the need to provide energy to a new housing project and propose solution options based around connections to an existing power grid, local power generators, renewable energy sources, increased energy efficiency, etc.

A bottom up approach starts with major elements and interactions. Again, division, grouping, and identification allows the build of a full system description able to provide all the necessary functions, at which point the final SoI boundary can be set. In this case, the choice of system elements and groupings will be driven by ensuring the major system elements can be formed together into a viable system whole. For example, there may be a need to replace an existing delivery vehicle and produce solution options considering vehicle ownership/leasing, driver training, petrol, diesel or electric fuel, etc.

The systems approach aspect of synthesis leads to SE terms such as "design" and "development." Wasson (2006, 390-690) describes synthesis from a SE point of view. White (2009, 512-515) provides a comprehensive discussion of methods of achieving design synthesis. The systems approach treats synthesis at the abstract level while the SE process definitions provide the concrete steps.

References

Works Cited

Beishon, J. 1980. *Systems Organisations: the Management of Complexity*. Milton Keynes; Open University press.

Blanchard, B. and W.J. Fabrcky. 2006. *Systems Engineering and Analysis*. Upper Saddle River, NJ: Prentice Hall.

Boehm, B. 2009. "Applying the Incremental Commitment Model to Brownfield System Development". Proceedings of the 7th Annual Conference on Systems Engineering Research (CSER), Loughborough, UK.

Browning, T.R. 2009. "Using the Design Structure Matrix to Design Program Organizations". In Sage, A.P. and W.B. Rouse (eds.).

Handbook of Systems Engineering and Management, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

Buede, D.M. 2009. "Functional Analysis". In Sage, A.P. and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management,* 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

Checkand, P. 1999. *Systems Thinking, Systems Practice*. New York, NY, USA: John Wiley & Sons.

Flood, R. L. and E.R. Carson. 1993. *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, 2nd ed. New York, NY, USA: Plenum Press.

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?" INCOSE *Insight*. 12(4) (December 2009): 59-63.

INCOSE. 1998. "INCOSE SE Terms Glossary." INCOSE Concepts and Terms WG (eds.). Seattle, WA, USA: International Council on Systems Engineering.

Jackson, S., D. Hitchins and H. Eisner. 2010. "What is the Systems Approach?". INCOSE *Insight*. 13(1) (April 2010): 41-43.

Jackson, S., D. Hitchins and H. Eisner. 2010. "What is the Systems Approach?". INCOSE *Insight*. 13(1) (April 2010): 41-43.

Jones, L. 1982, "Defining System Boundaries in Practice: Some Proposals and Guidelines." *Journal of Applied Systems Analysis*, 9: 41-55.

Levin, A.H. 2009. "System Architectures". In Sage, A.P. and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

Page, S.E. 2009. "Understanding Complexity". The Great Courses. Chantilly, VA, USA: The Teaching Company.

Shorter Oxford English Dictionary on Historical Principles, 3rd ed. s.v. "Analysis." Oxford, UK: Oxford University Press. 1973.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

White, Jr., K.P. 2009. "Systems Design." In Sage, A.P. and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

Primary References

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?" INCOSE *Insight*. 12(4) (December 2009): 59-63.

Jackson, S., D. Hitchins and H. Eisner. 2010. "What is the Systems Approach?" INCOSE *Insight*. 13(1) (April 2010): 41-43.

ISO/IEC 2008. Systems and software engineering -- System life cycle processes. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2008.

Additional References

Blanchard, B. and W.J. Fabrcky. 2006. *Systems Engineering and Analysis*. Upper Saddle River, NJ: Prentice Hall.

Browning, T.R. 2009. "Using the Design Structure Matrix to Design Program Organizations". In Sage, A.P. and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management,* 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

Buede, D.M. 2009. "Functional Analysis". In Sage, A.P. and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management,* 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

INCOSE. 1998. "INCOSE SE Terms Glossary." INCOSE Concepts and Terms WG (eds.). Seattle, WA, USA: International Council on Systems Engineering.

Levin, A.H. 2009. "System Architectures". In Sage, A.P. and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

Page, S.E. 2009. "Understanding Complexity". The Great Courses. Chantilly, VA, USA: The Teaching Company.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

White, Jr., K.P. 2009. "Systems Design." In Sage, A.P. and W.B. Rouse (eds.). *Handbook of Systems Engineering and Management*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "Synthesis of a System".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Analysis and Selection between Alternative Solutions

This article considers the activities of the systems approach (glossary) related to the analysis and selection of a preferred solution from the possible options. Any of the activities described below may need to be considered concurrently (glossary) with other activities in the systems approach. The final article in this knowledge area, Applying the Systems Approach, considers the dynamic aspects of how these activities are used as part of the systems approach and how this relates to elements of systems engineering (SE).

System Analysis

System analysis is an activity within the systems approach that evaluates one or more system artifacts created during the activities involved in Synthesizing Possible Solutions, such as

- the definition of assessment criteria based on the required properties and behavior of an identified problem or opportunity system situation;
- the assessment of the properties and behavior of each candidate solution in comparison to the criteria; and
- the comparison of the assessments of the candidate solutions and identification of any that could resolve the problem or exploit the opportunities, along with the selection of candidates that should be further explored.

As discussed in Synthesizing Possible Solutions, the problem context for an engineered system (glossary) will include a logical or ideal system solution description. It is assumed that the solution which "best" matches the ideal will be the most acceptable solution to the stakeholders. Note, as discussed below, the "best" solution should include an understanding of cost and risk, as well as effectiveness. The problem context may include a soft system (glossary) conceptual model describing the logical elements of a system to resolve the problem situation and how these are perceived by different stakeholders (Checkland 1999). This soft context view will provide additional criteria for the analysis process, which may become the critical issue in selecting between two equally effective solution alternatives.

Hence, analysis is often not a one-time process of solution selection, but is used in combination with problem understanding and solution synthesis to progress towards a more complete understanding of problems and solutions over time (see Applying the Systems Approach for a more complete discussion of the dynamics of this aspect of the approach).

Effectiveness Analysis

Effectiveness studies use the problem or opportunity system context as a starting point.

The effectiveness of a synthesized system solution will include performance criteria associated with the system's primary functions. These are derived from the system's purpose, to enable the realization of stakeholder needs in one or more, wider system contexts.

For a product system (glossary) there are a set of generic non-functional qualities that are associated with different types of solution patterns or technology, e.g., safety, security, reliability, maintainability, usability, etc. These criteria are often explicitly stated as parts of the domain knowledge of related technical disciplines in technology domains.

For a service system (glossary) or enterprise system (glossary) the criteria will be more directly linked to the identified user needs or enterprise goals. Typical qualities for such systems include agility, resilience, flexibility, upgradeability, etc.

In addition to assessments of the absolute effectiveness of a given solution system, systems engineers must also be able to combine effectiveness with the limitations of cost and timescales included in the problem context. In general, the role of system analysis is to identify the proposed solutions which can provide some effectiveness within the cost and time allocated to any given iteration of the systems approach (see Applying the Systems Approach for details). If none of the solutions can deliver an effectiveness level that justifies the proposed investment, then it is necessary to return to the original framing of the problem. If at least one solution is assessed as sufficiently effective, then a choice between solutions can be proposed.

Trade-off studies

In the context of the definition of a system, a trade-off study consists of comparing the characteristics of each candidate system element to determine the solution that globally balances the assessment criteria in the best way. The various characteristics analyzed are gathered in cost analysis, technical risks analysis, and effectiveness analysis (NASA 2007). Each class of analysis is the subject of the following topics:

- Assessment criteria (glossary) are used to classify the various candidate solutions. They are either absolute or relative. For example, the maximum cost per unit produced is cc$, cost reduction shall be x%, effectiveness improvement is y%, and risk mitigation is z%.
- **Boundaries** identify and limit the characteristics or criteria to be taken into account in the analysis (e.g., the kind of costs to be taken into account, acceptable technical risks, and the type and level of effectiveness).
- **Scales** are used to quantify the characteristics, properties, and/or criteria and to make comparisons. Their definition requires knowledge of the highest and lowest limits, as well as the type of evolution of the characteristic (linear, logarithmic, etc.).
- An assessment score (glossary) is assigned to a characteristic or criterion for each candidate solution. The goal of the trade-off study is to succeed in quantifying the three variables (and their decomposition in sub-variables) of cost, risk, and effectiveness for each candidate solution. This operation is generally complex and requires the use of models.
- The **optimization** of the characteristics or properties improves the scoring of interesting solutions.

A decision-making process is not an accurate science and trade-off studies have limits. The following concerns should be taken into account:

- subjective criteria (e.g., if the component has to be beautiful, what constitutes a "beautiful" component?);
- uncertain data (e.g., since inflation has to be taken into account to estimate the cost of maintenance during the complete life cycle of a system, how can a systems engineer predict the evolution of inflation over the next five years?); and
- sensitivity analysis. A global assessment score associated to every candidate solution is not absolute; thus, it is recommended that a robust selection is gathered by performing a sensitivity analysis that considers small variations of assessment criteria values (weights). The selection is robust if the variations do not change the order of scores.

A thorough trade-off study specifies the assumptions, variables, and confidence intervals of the results.

Systems Principles of System Analysis

From the discussions above, the following general principles (glossary) of systems analysis can be defined:

- Systems analysis uses assessment criteria based upon a problem or opportunity system description.
  - These criteria will be based around an ideal system description that assumes a hard system (glossary) problem context can be defined.
  - The criteria must consider required system behavior and properties of the complete solution in all of the possible wider system contexts and environments.
  - These must consider non-functional issues such as system safety, security, etc.
  - This idea system description may be supported by soft system (glossary) descriptions from which additional "soft" criteria may be defined, e.g., a stakeholder preference for or against certain kinds of solutions and relevant social, political, or cultural conventions to be considered in the likely solution environment, etc.
- As a minimum, the assessment criteria should include the constraints on cost and time scales acceptable to stakeholders.
- Trade studies provide a mechanism for conducting analysis of alternative solutions.
  - A trade study should consider a "system of assessment criteria" with appropriate awareness of the limitations and dependencies between individual criteria.

- Trade studies need to deal with both objective and subjective criteria. Care must be taken to assess the sensitivity of the overall assessment to particular criteria.

Linkages to other topics

Applying the Systems Approach

References

Works Cited

Checkland, P.B. 1999. *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley & Sons Ltd.

NASA. 2007. *Systems Engineering Handbook*, Revision 1. Washington, DC, USA: National Aeronautics and Space Administration (NASA). NASA/SP-2007-6105.

Primary References

ISO/IEC 2008. *Systems and software engineering -- System life cycle processes*. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2008.

Jackson, S., D. Hitchins and H. Eisner. 2010. "What is the Systems Approach?" INCOSE *Insight*. 13(1) (April 2010): 41-43.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional reference in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "Systems Analysis Approach".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Implementing and Proving a Solution

This article considers the activities of the systems approach (glossary) related to the implementation and proving of a preferred solution. Any of the activities described below may need to be considered concurrently (glossary) with other activities involved with the systems approach. The final article in this knowledge area, Applying the Systems Approach, considers the dynamic aspects of how these activities are used as part of the systems approach and how this relates to elements of systems engineering (SE).

Proving the System Overview

This topic covers both the sub-topics of verification and validation.

Verification

Verification is the determination that each element of the system meets the requirements of a documented specification (see principle of elements). Verification is performed at each level of the system hierarchy. In the systems approach this topic pertains to the more abstract level of providing evidence that the system will accomplish what it was meant to do. In systems engineering this topic pertains to providing quantitative evidence from tests and other methods for verifying the performance of the system.

Validation

Validation is the determination that the entire system meets the needs of the stakeholders. Validation only occurs at the top level of the system hierarchy. In the systems approach this topic pertains to the more abstract level of making sure the system meets the needs of the stakeholders. In systems engineering, this topic pertains to the detailed demonstrations and other methods to satisfy the stakeholders.

In a systems engineering context, Wasson (2006, 691-709) provides a comprehensive guide to the methods of both system verification and system validation.

Linkages to other topics

The systems approach topic is linked to the systems engineering topics of verification and validation.

References

Works Cited

Jackson, S., D. Hitchins and H. Eisner. 2010. "What is the Systems Approach?" INCOSE *Insight*. 13(1) (April 2010): 41-43.

Wasson, C. S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

Primary References

Jackson, S., D. Hitchins and H. Eisner. 2010. "What is the Systems Approach?" INCOSE *Insight*. 13(1) (April 2010): 41-43.

Additional References

Wasson, C. S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "Proving a System".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

Deploying, Using, and Sustaining Systems to Solve Problems

This article considers the activities of the systems approach related to the deployment, sustainment, and use of a solution. Any of the activities described below may need to be considered concurrently (glossary) with the other activities in the systems approach. The final article in this knowledge area, Applying the Systems Approach, considers the dynamic aspects of how these activities are used as part of the systems approach and how this relates in detail to elements of systems engineering (SE).

Engineered systems are eventually owned by an individual, team, or enterprise. Those who own the system during development may not be the ones who own the system when it is in operation. Moreover, the owners may not be the users; e.g., service systems may be used by the general public but owned by a specific business that is offering the service. The transition of a system from development to operations is often itself a complex task, involving such activities as training those who will operate the system, legal actions to complete the transfer, and establishing logistical arrangements so that the operators can keep the system running once the transition is completed.

There is very little in the literature pertaining to the application of the systems approach to this phase of the life cycle. However, a basic premise of this knowledge area is that the systems approach pertains to all phases of a system's life cycle. Hence, to properly build systems to solve problems or for other uses, it can be inferred that the systems approach pertains to the deployment, utilization, and the sustainment of the systems. Many of the references in this topic are from SE rather than from the systems approach. Hence, it can be assumed that aspects of the systems approach exist on a more abstract level and can be linked to SE aspects.

Transition

Transferring custody of the system of interest (SoI), and responsibility for its support from one organization to another, is often called transition (INCOSE 2011). Transition of a product system (glossary) includes the integration of the system into the acquiring organization's infrastructure.

Transition includes the initial installation of a system while ensuring that it is compatible with the wider system and does not cause any significant wider system issues. This process of acceptance and release for use varies between domains and across businesses and enterprises and can be thought of as an initial assessment of the system's effectiveness (glossary (Hitchins 2007). Generally, transition may be considered to have two parts: (1) ensuring that the new system interoperates (glossary) with the systems around it and (2) ensuring the resulting system is safe and has other critical operational properties.

It is particularly important to have considered emergent properties (glossary) when a new system is added to the existing organization's system of systems (SoS) (glossary) network, as well as the complexity (glossary) of the organization into which the new system is transitioned (see also Complexity). The more complex the receiving organization is, the more challenging the transition will be, and the greater the likelihood of unintended interactions and consequences from the new system's insertion. Dealing with the consequences of this complexity starts in transition and continues into operation, maintenance, and disposal.

Transition of a service system (glossary) is often performed in two stages. First, the service system infrastructure is accepted and released. Second, each realization of the service is accepted and released. There can be significant problems during the second stage if the required responsiveness of the service does not leave sufficient time to ensure that the service meets necessary functional and quality attributes, including interoperability, safety, and security. (See Service Systems Engineering).

Transition can also require its own enabling systems (glossary), each of which can be realized using a systems approach.

Operation

Use of the system to help enable delivery of user services is often called "operations" (INCOSE 2011). A system's effectiveness is normally considered throughout the operational life of a system. For a complex system, emergent behavior (glossary) should be considered in three ways:

1. to identify and plan for emergent properties within the system realization process;
2. to incorporate mechanisms for identifying and handling unexpected emergent properties within the system during its use; and
3. to provide necessary procedures for dealing with wider system consequences of unexpected emergent properties in the enterprise; e.g., emergency responses or medical first aid.

Operations require their own enabling systems, each of which can be realized using a systems approach.

Maintenance

The purpose of maintenance is to sustain the system through its useful life (INCOSE 2011). In system terms, maintenance implements systems to deal with entropy (glossary) and maintaining the system of interest in a viable (glossary) state. Since an open system (glossary) maintains its existence by continual exchange of energy, information, and materiel with its environment, one aspect of its maintenance must be the management of resources in the environment.

Hitchins (2007) describes generic approaches to resource management and viability management based on system concepts. Resource management identifies the need to consider the acquisition, storage, distribution, conversion, and disposal of resources. Viability management should consider systems to maintain homeostasis (glossary) and means for ensuring resilience (glossary) to environmental disturbance and adaptability (glossary) to environmental change.

Maintenance will require its own enabling systems, each of which can be realized using a systems approach. Maintenance success is more likely if it is considered as part of the system concept and design well before the system enters service.

Disposal

The purpose of disposal is to remove a system element from the operational environment with the intent of permanently terminating its functions, and to deal with any waste products left behind by the system. NASA (2007) describes disposal from the vantage point of NASA space and ground systems.

Disposal requires its own enabling systems, each of which can be realized using a systems approach. As with maintenance, a large part of successful disposal requires related issues to have been considered early on in the system's life cycle.

References

Works Cited

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology*. Hoboken, NJ, USA: John Wiley and Sons.

INCOSE. 2011. *INCOSE Systems Engineering Handbook*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2003-002-03.2.1.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Primary References

INCOSE. 2011. *Systems Engineering Handbook*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2003-002-03.2.1.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

Please note that in version 0.5, this article was titled "Owning and Making Use of a System".

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Acquirer/Supplier Responsibility

The following sections summarize various aspects of ownership and acquisition covered by such sources as INCOSE (2011). They briefly discuss these aspects in terms of key life cycle processes. The activities described below need to be considered concurrently (glossary) through the system's life, as discussed in Applying the Systems Approach. The implementation of these principles is described in Systems Engineering and Management.

Products, Services, and Enterprises

Most often, the terms "product" and "service" describe the thing exchanged through a customer and supplier agreement. This might be a commercial agreement, one funded publicly by a charity, or some other arrangement. One of the differences between a product and service is that a product is an artifact acquired to be used towards the achievement of an outcome while a service is an outcome supplied directly to a user.

The terms "customer" and "user" are often used interchangeably in engineering and management disciplines. The INCOSE Handbook (INCOSE 2011) makes the following specific distinctions between people associated with a system:

- The **acquirer** is the stakeholder that acquires or procures a product or service from a supplier.
- The **supplier** is an organization or individual that enters into an agreement with the acquirer to supply a product or service.
- The **Operator** is an individual or organization that contributes to the functionality of a system and draws on knowledge, skills, and procedures to enable said functionality.
- The **User** is an individual or group that benefits from a system during its utilization.

Product systems, consisting of hardware, software, and humans, have traditionally been the focus of systems engineering (SE) efforts. These systems are delivered to the acquirer and operated to accomplish the goals that led to the requirements for the system. These requirements were derived from the need to provide services to one or more users as part of an enterprise. The delivery (supplying) of a service implies the direct delivery of an outcome often related to the delivery of products, e.g., a maintenance, training, or cleaning service. This is not the same as the delivery of a service system (see the discussion below).

In traditional SE, the term "service" or "service system" describes the wider system context which describes the acquirer's need to deliver user value. In this case, the service system is a fixed system definition of how the acquiring enterprise will use products to enable the delivery of services to users. Product systems are designed to be integrated and operated as appropriate to enable this service to be maintained or improved as required. In this view, a service system is static and contains dedicated products, people, and resources. That is, hierarchies of products are engineered to provide acquirers with the ability to offer predefined services to users.

More recently, the term "service systems" has been used to describe a system that is engineered so enterprises can offer services directly to users without the need to hold all of the necessary products and services within the enterprise itself. This requires the expansion of the definition of a "supplier" as follows:

- A **product supplier** is an organization or individual that enters into an agreement with an acquirer to supply a product or related product support services.

- A **service system supplier** is an organization or individual that enters into an agreement with an acquirer to supply a service system.
- A **service supplier** is an organization or individual that enters into an agreement with a user to supply a service.

These service systems tend to be configured dynamically to deal with problems that traditional static services find difficult. This view of a service system employs "late binding" with product systems not owned by the enterprise, but used to enable the service to be offered as closely to a given timeline as possible. This is the definition of a service system used in the Service Systems Engineering topic in Part 4.

Acquirer/Supplier Agreements

Lawson (2010) provides a perspective on what it means to own systems, trading in system products and services, and the implications of supply chains in respect to the value added and ownership. INCOSE (2011) defines two life cycle processes related to acquisition and supply. The acquisition process includes activities to identify, select, and reach commercial agreements with a product or service supplier.

In many larger organizations, there is a tradition of system ownership vested in individuals, or in some cases, enterprise entities (groups or teams). Ownership implies the authority and responsibility to create, manage, and dispose of a system of interest (SoI) (glossary, as well as sometimes to operate the SoI.

Product Acquire/Supply

In some industries, a supplier works directly with an acquirer to help understand the acquirer's needs and then engineer one or more products to satisfy these needs. In some cases, a single supplier will provide the complete product system, and in other cases, a supply chain will be formed to deliver product systems with a system integrator to ensure they fit together and integrate into the wider context. This is a theoretical view of Product Systems Engineering in which the context is fixed and the product is designed to fit into it. A good systems engineer may well suggest changes to the enterprise as a better way to solve the problem and then modify the product system's requirements accordingly. However, at some point, an agreed context will be set and a product system developed to work within it.

For many commercial products, such as mobile phones, a supplier creates a representative user profile to generate the requirement and then markets the product to real users once it is realized. In these cases, the other elements of the systems approach are performed by the acquirer/user and may not follow formal SE processes. It is important that a product supplier takes this into account when choosing the way in which to engineer a system as additional help or support services may need to be offered with the purchased product. The idea of a supplier offering support services for users with a type of product purchased elsewhere (e.g., an auto-mechanic servicing different makes of cars) begins to overlap with the service systems (glossary) context, as discussed in the next article.

For an institutionalized infrastructure in which SoIs are entirely owned by an enterprise or parties thereof, the entire life cycle management responsibility, including operation, is often vested with the system owners. These systems belong to the system asset portfolio of an enterprise or multiple enterprises and provide the system resources, including the planned systems that are developed during life cycle management.

Service Acquire/Supply

Service systems (glossary) need not own the individual products and services which they deliver to their users and customers. The service system includes the means to identify and gain access to appropriate products or services when needed. These may not be enterprises in the traditional sense, but may include products embedded with the user (e.g., mobile phone hardware). They may be enterprises which offer infrastructure services to a wide range of different technologies or application domains. This can mean that the **transition**, **operation**, **maintenance**, and **disposal** activities associated with system ownership cannot be embedded in the acquiring enterprise, but need to be treated as separate system services. More detail can be found in Product Systems Engineering, Service Systems Engineering, and Enterprise Systems Engineering.

The service systems engineer helps the service supplier create and sustain the service system, which can be used to discover, integrate, and use specific versions of generic products or services when needed. The realization of service systems requires the ability to make use of product systems. However, these product systems are developed and owned outside of the service system. The service system must be able to gain access to a product or service when needed, and to interface with it effectively. The use of open interface standards, such as standard power supplies, interface connections (e.g., USB), or file formats (e.g., PDF) can help make this easier.

Enterprise Evolution

A useful distinction between product system design and enterprise system design is that "enterprise design does not occur at a single point in time like the design of most systems. Instead, enterprises evolve over time and are constantly changing, or are constantly being designed" (Giachetti 2010, xiii).

The enterprise developer may also aim to optimize back stage processes (the internal operations) of an organization or an institution by exploiting advances in technology (particularly IT and processes). In these cases, the engineered systems are enterprise systems.

Enterprises may offer products (goods) and/or services and the product SE must not only look at the development and delivery of the products, but also at the alignment and optimization of the product delivery within the enterprise objectives. Similarly, in service SE, the main focus is on intangible value delivery to the end-customer (externally focused: front stage) where internal and external processes must be synchronized. However, with the rapid advances in information and communications technologies (ICT), in many cases the boundaries between internal and external processes are quite blurred. Current research on SE is extending product methods, processes, and tools into the enterprise transformation and service innovation fields to exploit advances in business process methodologies and technologies.

Enterprise SE must do the engineering, not only of the enterprise itself, but may also need to get involved in the engineering of the service systems and products systems that the enterprise depends on in order to achieve its goals.

References

Works Cited

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press.

Hitchins, D. 2007. *Systems Engineering: A 21st Century Systems Methodology*. Hoboken, NJ, USA: John Wiley and Sons.

INCOSE. 2011. *INCOSE Systems Engineering Handbook*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2003-002-03.2.1.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Primary References

No primary references have been identified for version 0.75. Please provide any recommendations on primary reference in your review.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional reference in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Applying the Systems Approach

The Systems Approach is described in the SEBoK primarily through five groups of activities which describe the application of systems thinking (glossary) to an engineered system (glossary) throughout the life of that system. This article describes how the systems approach relates to both the dynamics of problem resolution and stakeholder value over time, as well as to the levels of system relationship, detailed management, and the engineering activities this implies.

Activity Mapping

The Systems Approach activities are directly related to the high level technical processes defined in Part 3, Systems Engineering and Management, of the SEBOK.

- Identifying and Understanding Problems and Opportunities relates to Concept Definition.

- Synthesizing Possible Solutions and Analysis and Selection between Alternative Solutions relate to System Definition.
- Implementing and Proving a Solution relates to System Realization.
- Deploying, Using, and Sustaining Systems to Solve Problems relates to Product and Service Life Management.

The principles defined in each of the systems approach activities and how they help shape the technical processes in each of these areas are discussed in Part 3.

Application Principles

Concurrency

Normally, the activities of the systems approach should be applied concurrently (glossary), reflecting their interrelationships and dependencies. Ring (1998) defines a system value cycle with three levels that a systems approach must consider to deliver real world benefit. These three levels are stakeholder value, system problem situation (or purpose), and system solution (or intervention). Value will only be fully realized when it is considered within the context of time, cost, and other resource issues appropriate to key stakeholders. A developer must consider not only what to do, but when and how much to do in order to provide real value (Senge 1990).

A greater understanding of the value that an engineered system is to offer enables more effective application of the systems approach, which in turn enables the problem situation to be agreed to and appropriate system interventions to be created, deployed, and used (Ring 1998). All of these activities happen concurrently. Efficient application of the systems approach requires awareness of constrained time and funding resources to realize the desired value. In systems engineering (SE) and management practices this leads to the creation of the following (INCOSE 2011):

- Life Cycles (glossary): Stakeholder value and problem resolution described as a set of life cycle stages (glossary) over which problems can be explored and resolved, and resources can be managed.
- Life Cycle Processes (glossary): Systems of activities focused on creation and sharing of knowledge associated with the systems approach, which can be employed to promote a holistic (glossary) approach over a life cycle.

For the generic systems approach, the following fundamental principles apply:

1. A life cycle has groups of stages which cover understanding stakeholder value; exploration of a problem situation (see System Definition); creation of a system solution, including analysis, synthesis, and proving (see System Realization); and System Deployment and Use.
2. Life cycle processes define a system of engineering and management activities based on the detailed information needed to ensure a systems approach across a life cycle (e.g., requirements, architecture, verification, and validation).
3. Activities in any of the processes may be employed in all of the stages to allow for appropriate concurrency.
4. The sequence and control of the life cycle stages and concurrent process activities must be tailored (glossary) to the problem situation and commercial environment (Lawson 2010), thus leading to the selection of an appropriate life cycle model (glossary).
5. Appropriate management activities must be included in the life cycle to ensure consideration of time, cost, and resource drivers.
6. In focusing on the creation of a specific system of interest (SoI) to provide solutions within the cycle, it is important to recognize the need to employ the right balance between reductionism (glossary) and holism (glossary) by considering the appropriate system context (glossary).

The ways in which this idea of concurrent process activity across a life cycle has been implemented in SE are discussed in System Life Cycle Process Models: Vee and System Life Cycle Process Models: Iterative.

Iteration

The systems approach can be applied in an iterative (glossary) way to move towards an acceptable solution to a problem situation within a larger cycle of stakeholder value.

Hitchins (2009) defines the principle of *adaptive satisficing* wherein a systems approach is applied iteratively to move towards the resolution of real world stakeholder needs. The system value cycle (Ring 1998) can be expressed as six groups of questions to cycle around value, problem, and solution questions that can be related to the systems approach:

1. What values do Stakeholders want/need? = Explore problems and opportunities
2. What system outcomes could improve this value? = System analysis
3. What system can provide these outcomes? = Synthesis
4. Has such a system been created? = Proving (verification)
5. Has the system been deployed to achieve the outcomes? = Proving (validation)
6. Do these outcomes provide the expected improvement in value? = Ownership and use

This cycle continues until stakeholders are satisfied or resources are exhausted.

The above questions focus on the iteration of the systems approach to deliver stakeholder goals within an enterprise context. Whether the first and last questions in this cycle are applications of SE or applications of a systems approach to the enterprise is a matter of debate within the systems community (see Enterprise Systems Engineering and Systems of Systems (SoS)).

The systems approach can be applied to multiple systems within an engineered system context, as discussed below. At each level, the approach may be applied iteratively to cycle between what is needed and versions of the solutions within a life cycle model. In general, these iterations are one of three types (Adcock 2005):

- Sequential: With iteration between the stages to solve detailed issues as they arise, a single application of the systems approach is sufficient.
- Incremental (glossary): Successive versions of the sequential approach are necessary for a solution concept. Each increment adds functionality or effectiveness to the growing solution over time.
- Evolutionary (glossary): A series of applications of the sequential approach for alternative solutions intended to both provide stakeholder value and increase problem understanding. Each evolutionary cycle provides an opportunity to examine how the solution is used so these lessons learned can be incorporated in the next iteration.

These aspects of the systems approach form the basis for life cycle models.

Recursion

The stakeholder value, problem resolution, and system creation aspects of the system value cycle may each require the use of a focused systems approach. These might be soft systems (glossary) to prove a better understanding of a situation, product systems (glossary) and/or service systems (glossary) solutions to operational needs, enabling systems (glossary) to support an aspect of the product or service life cycle, or enabling systems used directly by the enterprise system (glossary).

Each of these systems may be identified as a system of interest (SoI) and require the application of the systems approach. This application may be sequential (the start of one system approach dependent on the completion of another) or parallel (independent approaches which may or may not overlap in time), but will often be recursive in nature.

Recursion is a technique borrowed from computer science. In computer science recursion occurs when a function calls itself repeatedly to logically simplify an algorithm. In a recursive application applied to systems, the systems approach for one system of interest is nested inside another. Examples include cases where

- trades made at one level of the system require trades to be made for a system element;
- the analysis of a system requires analysis of a system element;
- the synthesis of a solution system requires one or more sub-system elements; and
- the verification of a product system requires verification of a system element.

In each case, the "outer" system approach may continue in parallel with the "inner" to some extent, but will be dependent on key outcomes for its own progress.

As with all recursive processes, at some stage the application of the approach must reach a level at which it can be completed successfully. This then "rolls up" to allow higher levels to move forward and eventually complete all nested applications successfully.

The INCOSE *Systems Engineering Handbook* (INCOSE 2011) describes a recursive application of SE to levels of system elements (glossary) with each application representing a system project (glossary). Martin (1997) describes the recursive application of SE within a product system hierarchy until a component (glossary) level is reached, at which point procurement of design and build processes can be used to create solution elements.

The principle of recursive application and how it relates to life cycle models is described in Systems Engineering and Management.

Linkages to other topics

This topic is linked to Mission Analysis, Stakeholder Needs and Requirements, System Realization, Architectural Design: Logical, and Architectural Design: Physical.

References

Works Cited

Blanchard, B. and W.J. Fabrycky. 2006. *Systems Engineering and Analysis*. Upper Saddle River, NJ, USA: Prentice Hall.

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

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?" INCOSE *Insight*. 12(4).

INCOSE. 2011. *INCOSE Systems Engineering Handbook*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering. INCOSE-TP-2003-002-03.2.1.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Martin J. N. 1997. *Systems Engineering Guidebook*. Boca Raton, FL, USA: CRC Press.

Ring, J., 1998. "A Value Seeking Approach to the Engineering of Systems." Proceedings of the IEEE Conference on Systems, Man, and Cybernetics. p. 2704-2708.

Senge, P. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY, USA: Doubleday/Currency.

Primary References

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

Hitchins, D. 2009. "What are the General Principles Applicable to Systems?" INCOSE *Insight*, 12(4).

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Additional References

Blanchard, B. and W.J. Fabrycky 2006. *Systems Engineering and Analysis*. Upper Saddle River, NJ: Prentice Hall.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Systems Challenges

Evolving societal needs creates new challenges for current systems engineering practices. For example, as the world becomes increasingly networked, business and government applications require integrated systems that exhibit intelligent and adaptive behavior. Also, while advances in technology allow us to do bigger projects with greater vision, trying to do more also exposes us to more and different forms of failure. These and other trends apply to all levels of government, industry, and academia, at the highest organizational levels down to the individual. The trends create new requirements on systems such as the use of new technology to create complex systems that are super efficient, eco-friendly, and people friendly; and systems engineering needs to adjust to handle these new requirements.

Topics

This knowledge area describes some of these challenges, from the viewpoint of systems. It also includes references to current research work being done in universities, research centers, and laboratories to respond to these new challenges. The Systems Challenges knowledge area contains the following topics:

- Complex System Challenges
- Dynamically Changing Systems

Range of Systems Challenges

Systems challenges can be categorized in each of the following system related areas:

- Concepts/Principles challenges: Related to systems theory. Concepts provide meaning for principles which provide basic truths. A system of "system concepts" describes common system attributes that span different domains (Ackoff 1971; Hitchins 2009; Lawson 2010).
- Properties/Features challenges: Related to a system's inherent relationships, structures, and behaviors.
- Modeling challenges: Related to the modeling of systems. System models are representations of the system.
- Approach challenges: These are focused on methods for addressing systems problems.

Examples of challenges inherent in each area are listed in the following sections.

Concepts/Principles Challenges

"Research issues Raised By The Guide to The Systems Engineering Body of Knowledge" (Adcock et al. 2011) provides a summary of reviewer comments on the version 0.25 of the SEBoK related to SE research. One area identified in the area of complexity deals with improving systems engineering around:

- technical complexity (caused by multiple system components)
- dynamic complexity (interaction between agents of system over time)
- behavioral complexity (caused by multiple goals and stakeholders)

Properties / Features Challenges

One primary challenge area in today's environment for system properties/features is resilience. Building resilience into a system is difficult when nodes of the system do not inherently cooperate, such as with a large infrastructure that has components from various states, regions, or localities. Other challenges include cultural differences and affordability. See Resilience Engineering.

Modeling Challenges

"Research Issues Raised By The Guide to The Systems Engineering Body of Knowledge" (2011) refers to the challenge of integrating engineering models across SE process areas and between levels of the system hierarchy. This can be expanded to the challenge of model integration:

- across multiple disciplines, which is a common occurrence in systems engineering
- throughout the development life cycle, from concept development to requirements to design/development to test/verification
- between development, production, and operation phases
- from the top system level down to the lowest unit in the system design hierarchy
- based on varying problem types or problem contexts

Approach Challenges

Systems approach challenges include the integration of various systems approaches. For example, one challenge is the integration of specialty disciplines and systems engineering. The integration of systems engineering with other disciplines extends to project management, software engineering, and other design and management disciplines.

References

Works Cited

Ackoff, R.L. 1971. "Towards a System of Systems Concepts." *Management Science*. 17(11).

Adcock, R., A. Squires, D. Olwell, and N. Hutchison. 2011. "Research Issues Raised By The Guide to The Systems Engineering Body of Knowledge". Proceedings from the Ninth Annual Conference on Systems Engineering Research (CSER), 14-16 April 2011, Redondo

Beach, CA.

Hitchins, D. 2009. "What are The General Principles Applicable to Systems?" INCOSE *Insight*. 12(4).

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College.

Primary References

Jamshidi, M. (ed.). 2009. *Systems of Systems Engineering – Innovations for the 21st Century*. Hoboken, NJ, USA: Wiley and Sons.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Complex System Challenges

Challenges in designing complex systems include:

- managing interdependencies
- balancing interactive social and technical effects
- understanding system behavior, including:
  - assessing system architecture functionality
  - understanding emergence
  - predicting unintended effects
- developing complex system models

Following is an overview of each of these challenges.

Managing Interdependencies

In a complex system or System of Systems (SoS) (glossary), constituent components or systems are highly interdependent. Changes in one part can have an impact throughout the system and possibly affect the entire system life cycle. Therefore, it is important to understand the interdependencies, discover the hidden interdependencies, and effectively manage them. One way to effectively manage the interdependencies is by managing interfaces. Chen and Han (2001) propose an Architecture Evolution Environment (AEE) (glossary) approach for SoS evolution. AEE can be defined as a set of architecture interfaces of all component systems involved. Here the interest is the externally observable characteristics of the component system rather than the internal organization. Such an approach can be used to

document and manage the knowledge base of interfaces. Algorithms for discovering hidden dependencies can be built on such managed knowledge.

In order to effectively implement AEE, a well-designed structure to architecture knowledge is needed. The interfaced-based architecture development process proposed by Dagli and colleagues (2009, 101) introduces some computational approaches to handle interfaces. The method leverages interfaces to generate architectural solutions. The process defines an architecture component by identifying the interfaces that a component needs to provide in the operating environment, in interaction with its peer components. The process advocates the use of the object-oriented modeling approach to define systems in various levels of abstraction, which make the process generic and broadly applicable. This object-oriented modeling approach also facilitates multiple forms of representation, data exchange, and computation.

For SoS, because of the complex interdependency of constituent systems, the classical point optimization method, which optimizes individual systems, does not apply well. SoS is more of a collaborative system in which the boundaries for optimization need to be expanded despite frequent uncertainty and conflicts.

Balancing Interactive Social and Technical Effects

Another dimension to be considered is the joint design of social and technical equilibria. The challenge is to effectively include interactive social and technical effects into the design. The Intelligent Transportation System (ITS) of U.S. Department of Transportation is an example (Sill et al. 2011). When given real-time traffic conditions and route recommendations, drivers may choose to follow the recommendation or not, and driver preferences may be aggressive (i.e., to reduce travel time) or conservative (i.e., to minimize variance). Such decisions will have an impact on the entire system and on the drivers themselves. For example, reduced mean travel time may lead to increased traffic (as more drivers choose to follow the fast route) or prolonged routes (as some drivers may not mind driving farther if they can drive fast). Such a shift in travel patterns eventually will increase traffic load and possibly drive the system to its earlier state. Alternatively, more stable travel time creates opportunities for business models to exploit.

Understanding System Behavior

Assessing System Architecture Functionality

The impacts of decisions made during the selection of the system architecture propagate throughout the entire system life cycle. Given the complexity of modern system concepts, it is difficult to understand and predict the full range of functionality that will be exhibited. This unpredictability begins during the architecting phase when the system concept still exhibits much ambiguity in its design details. System architects may be challenged to perform realistic objective assessments of an architecture concept, yet the need for such assessments persists. Progress has been made in the area of model-based architecture assessment using computational constructs such as canonical design primitives, comparative analysis, and fuzzy inference (J. P. Dauby and C. H. Dagli 2011, 244). These approaches are proving useful in exposing integration sensitivity in system concepts.

Understanding Emergence

The subject of emergence (glossary) has seen renewed interest given its implications in systems engineering. A number of specific sources of emergent behavior and emergent quality attributes have been described in a systems context (Dauby and Dagli 2011). These sources can be categorized as either components or interfaces. The overarching commonality is that each source is part of the architecture of the system.

Predicting Unintended Effects

Coupling variables are those factors that exist between systems and the operational environment. These couplings frequently produce unintended effects. Nonlinearity is exhibited when the total system response is not the superposition of the individual responses of the components. This results from internal coupling variables producing covariant dependencies. The rules of engagement govern the interaction and constraints between system components. They specify the protocols to be followed and conditions that must be met for intended system operation. These rules are frequently oversimplified, ill-defined, or misunderstood – a condition commonly known as loopholes. Undefined engagement scenarios represent an opportunity for unverified and unexpected behavior.

Developing Complex System Models

Improper system representations also contribute to the challenges associated with complex systems. Performance models are frequently too simplistic or are valid only in a limited region of operation. Behavioral models frequently describe principal actor or component behaviors too simplistically, and omitting stimulus/response behavior can result in poor predictions of system operation. Cognitive models describe human thought as rule-based and logical. An appreciation for psychological and sociological factors can account for otherwise emergent occurrences – especially in the context of underlying motives. Decision making and game theoretic constructs are useful for increasing the realism of complex system behaviors.

References

Works Cited

Chen, P. and J. Han. 2001. "Facilitating System-of-Systems Evolution with Architecture Support." Proceedings of the 4th International Workshop on Principles of Software Evolution. 20-21 September 2010. Antwerp, Belgium.

Dagli, C.H., A. Singh, J. P. Dauby, and R. Wang. 2009. "Smart Systems Architecting: Computational Intelligence Applied to Trade Space Exploration and System Design." *Systems Research Forum*. 3(2):101–119.

Dauby, J.P. and C. H. Dagli. 2011. "The Canonical Decomposition Fuzzy Comparative Methodology for Assessing Architectures." IEEE *Systems Journal*. 5(2): 244-255.

Dauby, J.P. and C. H. Dagli. 2011. "Architecting for Emergence by Identifying and Characterizing Its Sources." Proceedings From the Ninth Annual Conference on Systems Engineering Research (CSER). 14-16 April 2011. Redondo Beach, CA, USA.

Sill, S., B. Christie, A. Diephaus, D. Garretson, K. Sullivan, and S. Sloan. 2011. *Intelligent Transportation Systems (ITS) Standards Program Strategic Plan for 2011–2014*. Washington, DC, USA: U.S. Department of Transportation. April 2011. FHWA-JPO-11-052.

Primary References

Jamshidi, M. (ed). 2009. *Systems of Systems Engineering – Innovations for the 21st Century*. Hoboken, NJ, USA: Wiley and Sons.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Dynamically Changing Systems

Intelligent transportation systems, the global information grid, and the smart power grid are examples of large-scale complex systems. Increasing interconnectivity among large-scale socio-technical systems is giving rise to complex system behavior. These systems need to continually change and adapt to a changing set of needs and requirements, and they must keep pace with rapidly evolving technology. A broad spectrum of missions and operational modes requires these systems to adapt and evolve, often in real time.

Complex Adaptive Systems

Complex systems possess a unique set of attributes that can be used to differentiate between and define these systems. These attributes may include:

- interdependence
- independence
- distributed
- cooperative
- competitive, and
- adaptive.

These systems comprise multiple interdependent subsystems that are independent systems in their own right. The interconnected elements exhibit system-level behaviors that are greater than the capabilities of any individual subsystem and are often unpredictable. Such behavior is known as emergence (glossary). Complex adaptive systems also have the ability to learn from previous experiences, self-organize, and mold their behavior to adapt to positive or negative changes in their environments. Long-term forecasting of complex system behavior is not possible due to the phenomenon of emergence; however, short-term forecasting is possible provided the conditions within which the system will operate can be specified with some level of certainty. The properties of emergence and adaptation are key to the in-built resilience displayed by complex dynamic systems.

Dynamically Changing Environmental and Operational Conditions

The dynamically changing environmental and operational conditions require a system that will be effective for the duration of the needs but evolve to new systems. This challenging new demand has led to a new operational style: Instead of designing systems from scratch, business or government obtains the best systems available and integrates them into evolving systems to meet dynamically changing needs. These systems are distinct as they integrate within them complex systems that are assembled into a larger collaborative system structure to achieve a common purpose.

In this environment, architectural constraints imposed by existing systems have a major effect on the system capabilities, requirements, failures, and emergent behavior (glossary). This fact is important, as it complicates the systems engineering and architecting activities. Hence, architecture becomes a dominating concept in these evolving systems. Since the technology base and organizational and human needs are changing, systems engineering processes are required to become evolutionary. Components and functions are added, removed, and modified as owners of the evolving system architecture (glossary) experience and use the system (Dagli and Kilicay-Ergin 2009). This is a new systems challenge that needs to be adressed. Here are the basic questions:

- How can we handle complexity (glossary)?
- How can we achieve large-scale design?
- How can we design with hidden interdependencies?
- How can we guard against hidden cascading failures?
- How can we create interoperability (glossary)?
- How can we ensure evolutionary growth?

The dynamically changing environmental and operational conditions of present-day systems has generated the need for system architectures that not only perform effectively for their intended life cycle but also are capable of evolving to new system architectures as their mission changes. Representing a paradigm shift from the traditional static architectures, dynamic architectures enable a system to change while in operation. These architectures can automatically adapt to changes in their environment without disrupting service. Components of such systems are capable of self-organizing and automatically adapting to change. Systems with a majority of self-organizing components are resilient to changes and display hidden robustness to catastrophic events.

Architecting Process

In order to adapt to changing needs, a dynamically changing system must be designed for flexibility (glossary). The system should be engineered such that resources can be rapidly allocated in the event of altered operating conditions. The subsystems of such systems should be appropriately segregated to allow each subsystem to evolve independently while still contributing to the functioning of the whole. Traditionally systems architects have focused on creating systems that are robust to small changes and always operate at equilibrium. Robustness is built into these systems based on the assumption that all future system behavior can be predicted and controlled. Overlooking emergence (glossary) during the architecting process leads to systems that are rigid and lack the ability to cope with emergent behaviors arising from shifts in their operational environments. Dynamically changing systems represent a departure from this static viewpoint to systems that embrace change and self-adapt to it. Unfortunately, the current body of knowledge in systems research is not sufficient for effective design and operation of these types of systems. There is a need to push the boundaries of systems architecting research to meet the challenges imposed by new demands.

Architecting dynamic systems poses many challenges when compared with classical systems architecting. The fundamental processes remain the same as the systems architecting processes in terms of scoping, aggregation, partitioning, and certification, but they take place at a meta-

level. Although the process shares common properties, there are various deviations from classical systems architecting. The architecting of dynamic systems spans multiple abstraction layers and domains to foster collaborative functions among independent systems.

Collaboration requires more emphasis on interface architecting, which brings along many challenges such as interoperability, scalability, and security issues. Different architecting tools are required at high levels of complexity. Dynamic complex systems require a good balance of the architecting tools of heuristics, analytical techniques, and integrated modeling. Specifically, model-centric frameworks and executable models become important tools for analysis and architecting as they provide insights into system emergent behavior. Understanding emergence will play a critical role in the system architects' ability to design robust and resilient systems.

System architects can benefit from the property of emergence by designing components that can self-adapt and self-organize to changing environmental conditions. A dynamically changing meta-architecture can be defined as a collection of different complex adaptive systems that are readily available to be plugged into the evolvable net-centric communications architecture. The evolving physical architecture is created by interconnecting operationally and managerially independent systems. This meta-architecture evolves to meet the changes in system requirements and objectives. It is the dynamically changing architecture that creates the best complex systems, although data and the communication architecture is a necessity for the system to function.

The challenge is to identify the right collection of systems that will collaborate to satisfy the client requirements. This shifts the focus from component and individual system level to meta-level architecting (Dagli and Kilicay 2007). This "plug and play" concept of assembling and organizing coalitions from different systems provides flexibility to respond to changing operational and environmental situations but requires a high level of interoperability in the information architecture that supports the coalition units.

The production model used for the Boeing 787 illustrates the idea of creating meta-architectures:

> A total of forty two national and international companies are connected to the global company information grid. Each company is operationally and managerially independent and has different organizational architectures and legacy systems. The focus of architecting for this domain is to choose the right collection of component manufacturing companies to produce Boeing 787. The emphasis is on interface architecting to achieve collaborative functions among these manufacturing companies. The Boeing information grid (the GIG) provides the net-centric capabilities, and the way these independent companies are connected to the Boeing information grid creates the meta-architecture for Boeing enterprise. The information grid architecture is the backbone of the system architecture, but the system's virtual manufacturing meta-architecture can achieve its mission through the independent companies connected to the information grid. As the requirements or fuzzy system attributes change or evolve over time, new manufacturing companies are added to this information grid. Hence, the meta-architecture is evolutionary. The architecting process spans through different abstraction layers; in this case functional and mission layers. (Dagli and Kilicay-Ergin 2009, 88)

References

Works Cited

C. Dagli and N. Kilicay. 2007. "Understanding Behavior of System of Systems Through Computational Intelligence Techniques." Paper presented at the 1st Annual IEEE Systems Conference. 9-12 April 2007. Honolulu, HI, U.S.A.

Dagli, C.H. and N. Kilicay-Ergin. 2009. "System of Systems Architecting" in *System of Systems Engineering: Innovations for The 21st Century*. Mo Jamshidi (ed.) Hoboken: John Wiley and Sons. p. 77-101.

Primary References

Jamshidi, M. (ed.). 2009. *System of Systems Engineering – New Challenges for the 21st Century*. Hoboken, NJ, USA: John Wiley and Sons.

Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article (Part 3) >

Comments from SEBoK 0.5 Wiki

SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Retrieved from "http://www.sebokwiki.org/075/index.php?title=Part_2&oldid=33054"
Categories: Part │ Part 2 │ Knowledge Area │ Topic │ Systems Thinking │ System Science │ Representing Systems with Models │ Systems Approach │ Part 3 │ Systems Challenges

- This page was last modified on 15 March 2012, at 11:06.
- This page has been accessed 11 times.

- MediaWiki Consulting provided by WikiExpert.com