System Definition

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

Lead Authors: Alan Faisandier, Garry Roedler, Contributing Author: Rick Adcock

System definition activities are conducted to create and describe in detail a system-of-interest (SoI) to satisfy an identified need. The activities are grouped and described as generic processes, which consist of system requirements definition, system architecture definition, system design definition and system analysis. The architecture definition of the system may include the development of related logical architecture models and physical architecture models. During and/or at the end of any iteration, gap analysis is performed to ensure that all system requirements have been mapped to the architecture and design.

System definition activities build on the artifacts and decisions from concept definition, primarily the articulation of the mission of the (SoI), the needs and requirements of stakeholders, and preliminary operational concepts. See Life Cycle Processes and Enterprise Need for further detail on the transformation of needs and requirements from the business or enterprise and stakeholder levels of abstraction addressed in concept definition to the system and system element level of abstraction addressed in system definition.

The products of system definition activities (system requirements, architecture and design) are inputs to system realization.

The specific activities and sequence of system definition activities and their involvement with the life cycle activities of any system, and in particular the close integration with concept definition and system realization activities, will be dependent upon the type of life cycle model being utilized. See Applying Life Cycle

Processes for further discussion of the concurrent, iterative and recursive nature of these relationships.

Topics

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The Kas, in turn, are divided into topics. This KA contains the following topics:

- System Requirements
- System Architecture
- Logical Architecture Model Development
- Physical Architecture Model Development
- System Design
- System Analysis

See the article Matrix of Implementation Examples for a mapping of case studies and vignettes included in Part 7 to topics covered in Part 3.

System Views and System Elements

An engineered system solution to a defined concept includes a set of engineering elements, characteristics, and properties. These elements are grouped in two ways:

- Needs and requirements views
- Architecture and design views

Architecture views include the identification of the boundary and interfaces of a system-of-interest (SoI), which may then be further refined as a collection of system elements and their relationships.

Needs and Requirements Views

Requirements provide an overall view of the purpose and mission which the system as a whole is intended to satisfy, as well as a technology-independent view of what the system solutions(s) should do. They are conventionally organized into two types:

- Business or mission requirements and stakeholder requirements are defined and discussed in the Concept Definition KA.

- System requirements, which describe the functions which the system as a whole should fulfill in order to satisfy the stakeholder requirements and are expressed in an appropriate set of views, and non-functional requirements expressing the levels of safety, security, reliability, etc., which are called for. These collectively form the basis for verification later in the life cycle.

System requirements and stakeholder requirements are closely related. Neither can be considered complete until consistency between the two has been achieved, as demonstrated by traceability, for which a number of iterations may be needed.

The process activities that are used to identify, engineer and manage system requirements are described further in the System Requirements article in the KA.

Architecture and Design Views

A given engineered system is one solution that could address/answer a problem or an opportunity (represented through requirements views); the solution may be more or less complex. A complex solution cannot be comprehended with a single view or model, because of the characteristics or properties of the problem/solution (see system complexity). The characteristics are structured as types or entities; types are related to each other. An instantiation of the set of types can be understood as THE architecture of the system. The majority of interpretations of system

architecture are based on the fairly intangible notion of structure. Therefore, the system architecture and design is formally represented with sets of types or entities such as functions, interfaces, resource flow items, information elements, physical elements, nodes, links, etc. These entities may possess attributes/characteristics such as dimensions, environmental resilience, availability, reliability, learnability, execution efficiency, etc. The entities are interrelated by the means of relationships and are generally grouped into sets to represent views/models of the system architecture and design.

Viewpoints and views are sometimes specified in architecture frameworks. Views are usually generated from models. Many systems engineering practices use logical and physical views for modeling the system architecture and design.

- The **logical view of the architecture** supports the logical operation of the system all along its life cycle, and may include functional, behavioral, and temporal views/models. Operational scenarios refine the mission into a collection of functions and dynamic structures that describe how the mission is performed (behavior).

- The **physical view of the architecture** is a set of system elements performing the functions of the system. Those system elements can be either material or immaterial (e.g., equipment made of hardware, software and/or human roles).

The boundary of the system architecture depends on what engineers include within the scope of the SoI and outside of it. This decision marks the transition from the characterization of the problem context to the beginnings of solution definition.

Facing the potential number of system elements that constitute the physical architecture, sets of system elements can be grouped to form systems. The decomposition of the SoI (highest level) may include the decomposition of several layers of systems (intermediate levels of systems) until technological system elements (lowest level) are defined. Any layer of the decomposition may include systems and non-decomposable technological system elements. The relationship between each layer is recursive; as a system element is also an engineered system, it can be characterized in its turn using the previous views in its own context.

The logical and physical representations of the system architecture are mapped onto each other. The interactions between system elements are defined by interfaces whose complexity strongly depends on the way the system architecture and design is defined. The relationships between the outputs of concept definition and the system solution, as well as the range of other views of a system that are available to describe a more complete set of characteristics between the system elements are discussed further in the Logical Architecture Model Development and Physical Architecture Model Development sections of system definition.

System Synthesis and Decomposition

System definition is managed through methodical synthesis of the SoI into systems and system elements. Solution synthesis may be top-down or bottom-up, as discussed in Synthesizing Possible Solutions. However it is done, as the system architecture definition advances, a decomposition of systems and system elements emerges; this forms a system breakdown structure (SBS). For project management purposes, every system of the SBS may be included in a *building block*, a notion introduced in (ANSI/EIA 1998), also called *system blocks*.

Stakeholder requirements and system requirements exist at all layers of the SBS. In ISO/IEC/IEEE 29148 *Systems and Software Engineering - Requirements Engineering* (ISO 2011), these layers are known as levels of abstraction. Along with systematically introducing layers of systems, the architecture and design process manages the transformation of the system requirements through levels of abstraction. Figure 1 illustrates this approach.
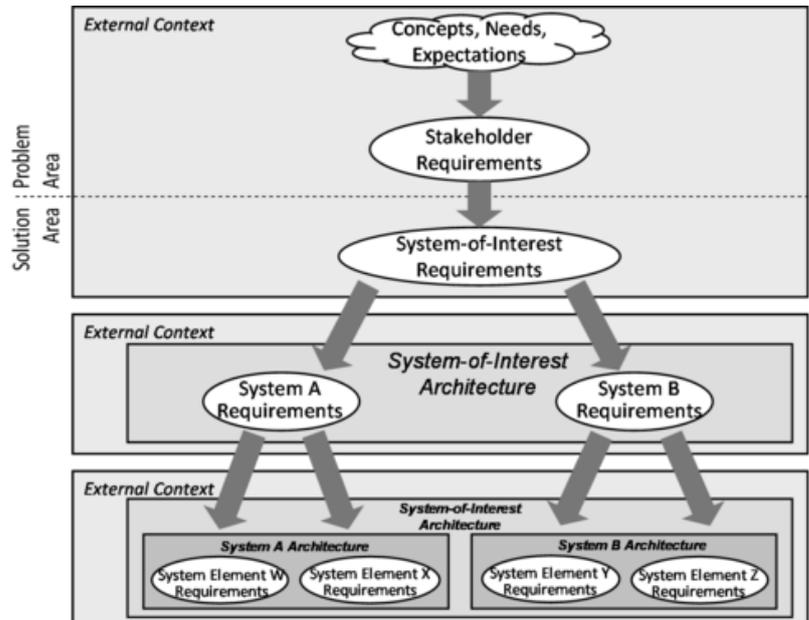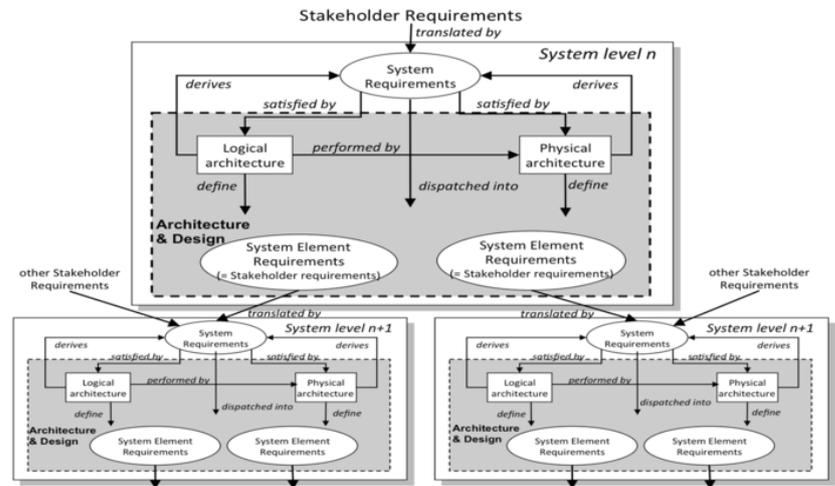
Figure 1. Top-down Development of Architecture and Design, and Requirements (Faisandier 2012). Permission granted by Sinergy'Com. All other rights are reserved by the copyright owner.

As shown in Figure 1:

- The white ovals represent requirements at decreasing levels of abstraction, and the arrows represent the transformation of those requirements through the levels using the architecture and design process. Stakeholder expressions of needs, expectations, and constraints are transformed into stakeholder requirements.
- The next transformation crosses the boundary between the problem and solution areas by converting stakeholder requirements into system requirements, reflecting the bounded solution space.
- At the SoI level, the system architecture is developed, which serves to identify systems and system elements and establishes how they operate together to address the SoI requirements.

This approach is applied recursively for each level of abstraction/decomposition recognizing that the same generic processes are applied at multiple levels of abstraction. At any level of this decomposition, one or more solution options may be presented as system architectures. The process by which the solution which best fits the system requirements, associated stakeholder needs and wider life cycle concerns is selected and justified is discussed in the System Analysis process.

Figure 2 below portrays the engineering that occurs in each system block. As necessary, system elements are defined through sets of system element requirements, which become inputs to other system blocks (*level n+1*). The approach is then recursively applied using the system definition processes.

Figure 2. Recursive Instantiation of Definition Processes (Faisandier 2012). Permission granted by Sinergy'Com. All other rights are reserved by the copyright owner.

At the *n+1* level, the systems or system elements may also collect other stakeholder requirements that are directly pertinent to this level of architecture and design. Processes within each system are generic but unique in local purpose, scope and context.

See Applying Life Cycle Processes for a discussion of the iterative and recursive application of system requirements and architecture processes, and Life Cycle Processes and Enterprise Need for further detail on the transformation of needs and requirements to system and system element levels of abstraction.

The different aspects of how systems thinking is applicable to system definition are discussed in SEBoK Part 2. In particular, see discussion of the recursive nature of systems and engineered system contexts in Engineered System Context; the contrast between top-down and bottom-up approaches in Synthesizing Possible Solutions and the role of solution architecture options and selection in Analysis and Selection between Alternative Solutions.

References

Works Cited

ANSI/EIA. 1998. *Processes for Engineering a System.* Philadelphia, PA, USA: American National Standards Institute (ANSI)/Electronic Industries Association (EIA), ANSI/EIA-632-1998.

Faisandier, A. 2012. *Systems Architecture and Design.* Belberaud, France: Sinergy'Com.

ISO/IEC/IEEE. 2011. *Systems and Software Engineering - Requirements Engineering.* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 29148.

ISO/IEC/IEEE. 2011. *Systems and Software Engineering - Architecture Description.* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

Primary References

ANSI/EIA. 1998. *Processes for Engineering a System.* Philadelphia, PA, USA: American National Standards Institute (ANSI)/Electronic Industries Association (EIA), ANSI/EIA 632-1998.

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.

INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-118-99940-0.

ISO/IEC. 2007. *Systems Engineering – Application and Management of The Systems Engineering Process.* Geneva, Switzerland: International Organization for Standards (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 26702:2007.

ISO/IEC/IEEE. 2015. *Systems and Software Engineering - System Life Cycle Processes.* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers

(IEEE). ISO/IEC/IEEE 15288:2015.

ISO/IEC/IEEE. 2011. *Systems and Software Engineering - Requirements Engineering*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/ Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 29148.

ISO/IEC/IEEE. 2011. *Systems and Software Engineering - Architecture Description*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

Martin, J.N. 1997. *Systems Engineering Guidebook: A Process for Developing Systems and Products,* 1st ed. Boca Raton, FL, USA: CRC Press.

NASA. 2007. *Systems Engineering Handbook.* Washington, D.C., USA: National Aeronautics and Space Administration (NASA), NASA/SP-2007-6105.

Additional References

Baldwin, C.Y. and K.B. Clark. 2000. *Design Rules*. Cambridge, MA, USA: MIT Press.

Buede, D.M. 2009. *The Engineering Design of Systems: Models and Methods*. 2nd ed. Hoboken, NJ, USA: John Wiley & Sons Inc.

Faisandier, A. 2012. *Systems Architecture and Design*. Belberaud, France: Sinergy'Com.

Hatley, D.J., and I.A. Pirbhai. 1987. *Strategies for Real-Time System Specification*. New York, NY, USA: Dorset House Pub.

MOD. 2010. *MOD Architecture Framework,* Version 1.2.004. London, UK: UK Ministry of Defence. Available at:
http://www.mod.uk/DefenceInternet/AboutDefence/What WeDo/InformationManagement/MODAF/.

id=65494"