System Architecture Design Definition

System Architecture Design Definition

Author(s): Jeffrey Carter and Caitlyn Singam

The system architecture design defines system behavior and structure characteristics in accordance with derived requirements. The architecture ensures that the system elements operate together in the applicable operating environment to satisfy Stakeholder needs. Figure 1 illustrates the system design phenomenon based on Hamilton's Principle: a system is composed of interacting elements exchanging data, energy, or mass to impact the state of cooperating elements resulting in continuous, discrete, or emergent behaviors at progressive levels of aggregation or decomposition.

Error creating thumbnail: File missing

Figure 1. The System Phenomenon – Hamilton's Principle
(SEBoK Original)

Systems Engineering has traditionally applied intuitive domain-specific [e.g., aerospace, defense, automotive, consumer, ...] product practices emphasizing processes and procedures. The practices in conjunction with proficient writing skills manually organize design configuration information in a disparate collection of documents. The documents depict the system architecture design with textual description and graphical figures using unique organizational adopted notation without formal semantics. The documents require manual updating to synchronize design configuration information resulting in consistency issues in addition to cost and schedule inefficiencies.

This article describes a model-based system architecture design definition approach with the Systems Modeling Language [SysML] for product development, integration, and verification. This approach enables digital transformation design initiatives to employ Model-Based System Engineering [MBSE] practices similar to other design engineering disciplines [EE, ME, SW].

SysML provides industry standard graphical and textual [added with SysML-v2] notations with formal semantics to define system design requirements, behavior and structure characteristics with traceability to the associated requirements. The system design model provides an Authoritative Source of Truth [ASoT] digital repository of the project technical baseline including requirement, functional, logical, and physical design representations for each system, subsystem, and component design abstraction level. Integrated system design model simulation capabilities with verification criteria of end-to-end system "digital threads" enable evaluation of key performance parameters to verify design decision or discover and resolve design defects in digital environments. The system design model with integrated simulation capabilities provides a system "digital twin" as an authoritative representation of the physical system including digital thread end-to-end simulations with all the data, models, and infrastructure necessary to optimize the system design digitally and explore design alternatives for system improvements to evolving missions and threats. The approach is tailorable to satisfy specific project objectives.

☐

Contents

System Architecture Design Overview

System architecture design definition includes system behavioral and structural analyses to define a holistic

technical solution that satisfies Stakeholder needs. The model-based system architecture design definition approach provides an implementation for the ISO/IEC/IEEE-15288 Architecture Definition technical lifecycle process. The approach defines practices to develop, integrate, and employ digital system design models for cross-domain collaboration by applying product domain-specific design knowledge and expertise.

The system architecture design defines the structural elements and their behavior interactions in the applicable operating environment to satisfy Stakeholder needs. The architecture includes a project overview with representations of system requirements, functional behavior & interactions, logical and physical configuration items. Figure 2 illustrates the system architecture design definition approach.

Error creating thumbnail: File missing

Figure 2. System Architecture Design Definition Approach.
(SEBoK Original)

The following describes the model-based system architecture design definition approach activities and information for the digital system design model.

1. Provide a project overview by describing the scope of the project and the system/project boundaries.
   1. System mission description relative to the project's Stakeholder needs with requirements traceable to the system needs.
   2. Glossary of terms with descriptions of the design model organization, environment, development approach, SysML conventions and custom profiles
2. Define the system requirements and verification criteria derived from the Stakeholder Needs to specify the system capabilities, functions, interfaces, performance, characteristics, operating conditions and environments.
   1. System use cases reflect black-box functionality, solution constraints, boundaries, external human and system interfaces.
   2. Capabilities and requirements derived from the

Stakeholder's needs verified with engineering analysis, demonstration, inspection, and test results throughout the hierarchical system structure of elements.

3. Define the system functional hierarchical functions and subfunctions with their behavior interactions to satisfy Stakeholder needs. Each function defines input-to-output transformations to satisfy one or more allocated requirement(s).
    1. System use case elaboration identifies system capabilities with activity, sequence, and state machine diagrams to describe system functional operations decomposed into User tasks, system functions and subfunctions with the associated requirements and verification criteria.
    2. System element interactions define how the functions and subfunctions operate together via interfaces to provide capabilities.

4. Define the logical system structural elements to identify logical configuration items without a specific design implementation, referred to as the technology Platform Independent Model [PIM]. Each logical configuration item has a set of allocated functions and subfunctions with specified property attributes, interfaces, and functional performance requirements for programmatic make vs. buy decisions. Also, each logical element has a corresponding physical design element with additional lower-level design details. For example, an automobile design will specify a logical combustion engine with requirements to establish a contractual baseline for production/procurement.
    1. Recursive white-box analyses identify logical configuration items with allocated functions, subfunctions, and interfaces for implementation by the corresponding physical configuration item.
    2. Logical configuration items include hierarchical subsystems, assemblies, subassemblies, and components based on the system complexity.

5. Define he physical system structural elements to identify a specific design implementation for each logical structural element, referred to as a technology Platform Specific Model [PSM]. The physical configuration items comply with the allocated requirements and have a specific manufacturer's part number identification. For example, an automobile

will have a specific V8 gas, diesel, or propane combustion engine in accordance with the requirements and include additional physical engine component configuration items [e.g., pistons, spark plugs, cam shaft, ...].

1. Recursive design analyses with electrical, mechanical, and software engineering to identify the specific design implementation for physical configuration items [i.e., manufacture and part number identification].

2. Physical configuration items provide a design implementation for the corresponding logical element including the required property attributes and functional operations.

System Architecture Design Definition Model

Figure 3 provides an example for the organization of the digital system design model. The design model provides an Authoritative Source of Truth [ASoT] repository for a project's system technical baseline including requirements, functional, logical, and physical design representations for each system, subsystem, and component design abstraction level. The design model also provides design libraries for product-line management and reuse.

Error creating thumbnail: File missing

Figure 3. System Architecture Design Model Organization Example. (SEBoK Original)

The digital system design model enables the following:

- Specifies system behavior and structure characteristics with traceability to the associated requirements.
- Provides an Authoritative Source of Truth [ASOT] digital repository of a project's approved technical baseline across engineering, production, and sustainment including requirement, functional, logical, and physical design representations for each system, subsystem, and component design abstraction level for collaboration and decision making.
- Provides product design libraries to adapt product designs to stakeholder needs with new, modified, and existing system capabilities.
- Integrated system design model simulation

capabilities with verification test cases of end-to-end system "digital threads" enable evaluation of key performance parameters to verify design decision, discover and resolve design defects in digital environments before the expense of producing prototypes.

- Assessment of design changes in all model usages for compliance, with any issue(s) flagged for corrective action.

- Provides a system design model with integrated simulation capabilities providing a system "digital twin" representing an authoritative representation of the physical system including digital thread end-to-end simulations with all the data, models, and infrastructure necessary to optimize the system design digitally and explore design alternatives for system improvements to evolving missions and threats.

- Provides design model scripts to export functional & interface specifications, design & requirements traceability and design descriptions reports.

System Architecture Design Definition Model Viewpoints and Views

The architecture model includes multiple views of the system design features with the information necessary to address each stakeholder's viewpoint and concerns. Figures 4a and 4b provide typical system views and viewpoints which are tailorable by specific projects, if necessary.

Error creating thumbnail: File missing

Figure 4a: System Architecture Design Model Viewpoints and Views. (SEBoK Original)

Error creating thumbnail: File missing

Figure 4b: System Architecture Design Model Viewpoints and Views. (SEBoK Original)

System Architecture Design Definition Modeling Ecosystem

Digital system design model integration within an engineering development ecosystem including electrical, mechanical, software and specialty engineering models enables system-level simulations. The system simulations provide initial design verification in digital environments to discover and resolve design defects before producing physical prototypes. Figure 5 provides an example of a model-based system design ecosystem.

Figure 5. Model-based System Design Ecosystem Example.
(SEBoK Original)

The digital system design model provides an executable representation of a system in terms of the hierarchical structural elements and their behavior interactions to satisfy Stakeholder needs in the applicable operating environment.

- Hierarchical structural elements include subsystems, assemblies, subassemblies, and components enabling separation of concerns between each design abstraction level.
- Element interactions involve the exchange of data, energy, force, or mass that trigger state changes in the cooperating elements resulting in discrete, emergent, or continuous behaviors at progressive levels of aggregation or decomposition.

The executable design model enables digital simulations to confirm system functional performance, accuracy, timeliness, and stability with verification criteria. In addition, the system response to operating condition variations is evaluated including deterministic and stochastic disturbances to discover design defects and

enhancements before the expense of producing physical prototypes. The governing principle is to develop digital system design models with high-fidelity simulation capabilities to realistically emulate systems to create a digital twin with digital threads to evaluate design alternatives in virtual computing environments without producing prototypes as shown in Figure 6.

Error creating thumbnail: File missing

Figure 6. System Architecture Design Definition and Verification. (SEBoK Original)

- Digital threads are analytical frameworks providing end-to-end system simulation representations to evaluate key performance parameters in virtual environments by exchanging information between different modeling tools across the lifecycle.
- Digital twins are authoritative representations of physical systems including the digital thread end-to-end connections with all the data, models, and

infrastructure needed to create and optimize a system's lifecycle digitally. Digital twins enable project team collaboration, system simulation performance assessments, design change impact evaluations, and product-line management reuse libraries.

System Architecture Design Definition Illustrative Example

Figure 7 provides an example automobile product to illustrate how the digital design model can specify, analyze, and design the system. The scope of the example is an automotive powertrain subsystem consisting of an engine, drivetrain, and four-wheel assemblies. The drivetrain assembly consists of transmission, driveshaft, differential, and two axle shaft rod subassemblies. The intent is to demonstrate the system architecture design definition approach and design solution accuracy is not a high priority.

Error creating thumbnail: File missing

Figure 7. Automobile Product Design Example. (SEBoK Original)

Figure 8 provides the automobile system context and use

cases. The diagram defines the automobile system boundary, Stakeholder capability needs, and external interfaces with. The design analyses will focus on the "Drive Vehicle" capability which includes the "Control Vehicle Acceleration" function allocated to a logical and physical system "Powertrain" subsystem.

- The powertrain subsystem is composed of engine, drivetrain, and wheel assemblies.
- The engine assembly components interact to generate torque by consuming fuel and air.
- The drivetrain assembly is composed of transmission, driveshaft, differential, and two axle rod shaft subassemblies. The drivetrain interacts with the engine to distribute amplified torque to the wheels.
  - The transmission subassembly amplifies the engine torque.
  - The driveshaft subassembly delivers the amplified torque to a differential.
  - The differential subassembly transfers the delivered torque to axle rod shaft.
  - The axle shaft rod subassembly provides the amplified torque [rotational force] to the four wheels.
- The four-wheel assemblies interact with the road surface to apply the torque and move the vehicle.
- The Driver's interaction with the vehicle determines the speed and direction of travel.
- The vehicle speed and direction of travel is affected by environmental conditions.
- A Mechanic uses an external diagnostic tool to read vehicle fault codes during maintenance.
- The automobile's fuel is filled at an external fuel [gas-diesel-propane] or electrical charging station.

Error creating thumbnail: File missing

Figure 8. Automobile System Context and Use Cases Example. (SEBoK Original)

The following three sub-articles [functional, logical, and physical architecture] illustrate the model-based system architecture design definition approach for the automobile system example to define a holistic solution. The functional architecture article has been provided. The logical and physical architecture articles will be updated in future SEBoK releases.

References

Works Cited

Bill Schindel; *SEBoK Part-3: Systems Engineering STEM Overview*

Will Roper. *There is No Spoon: The New Digital Acquisition Reality*. US Air Force: Assistant Secretary of the Air Force, 07-October 2020

U.S. DOD; *Digital Engineering Strategy*; Office of the Deputy Assistant Secretary of Defense for Systems Engineering. June 2018.

Primary References

INCOSE, *Systems Engineering Handbook – 5th Edition: A Guide for System Life Cycle Processes and Activities*, John Wiley & Sons Ltd., 2023

ISO/IEC/IEEE-15288, *Systems and Software Engineering: System Life Cycle Processes*, 2023

ISO/IEC/IEEE-42010; Software, *Systems and Enterprise – Architecture Description*; 2022

OMG; *Systems Modeling Language [SysML] Standard – v1.7*; August 2022

OMG; *Systems Modeling Language [SysML] Standard – v2 beta 2*; April 2024

OMG; *Model Driven Architecture Guide*; 2003

Lenny Delligatti; *SysML Distilled: A brief Guide to the Systems Modeling Language* Addison-Wesley Professional; 2013

Sanford Friedenthal, Alan Moore, Rick Steiner; *A Practical Guide to SysML: The Systems Modeling Language – 3rd Edition*; Morgan Kaufmann; October 2014

Tim Weilkiens, Jesko G. Lamm, Stephan Roth, Markus Walker; *Model-Based System Architecture – 2nd Edition*; John Wiley & Sons Ltd., 2023

Tim Weilkiens; *Systems Engineering with SysML/UML: Modeling, Analysis, Design*; Morgan Kaufmann; 2011

Mark Maier; *The Art of Systems Architecting – 3rd Edition*; CRC Press; 2009

Jon Holt, Simon Perry; *SysML for Systems Engineering*; The Institution of Engineering and Technology; 2008

Additional References

Charles Wasson; *System Analysis, Design, and*

Development – Concepts, Principles, and Practices; John Wiley & Sons; 2006

Katsuhiko Ogata; *Modern Control Engineering*; Prentice-Hall Inc; 1970

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.10, released 06 May 2024