

System Implementation

System Implementation

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: John Snoderly, Alan Faisandier

System Implementation uses the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder needs and requirements and system requirements developed in the early life cycle phases. These system elements are then integrated to form intermediate aggregates and finally the complete system-of-interest (SoI). See System Integration.



Contents

Definition and Purpose

Process Approach

- Purpose and Principle of the Approach

- Activities of the Process

- Artifacts and Ontology Elements

- Methods, Techniques, and Tools

- Checking and Correctness of Implementation

References

- Works Cited

- Primary References

- Additional References

Definition and Purpose

Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). System elements are

made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing, the software realization processes of coding and testing, or the operational procedures development processes for operators' roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process. Figure 1 portrays how the outputs of system definition relate to system implementation, which produces the implemented (system) elements required to produce aggregates and the Sol.

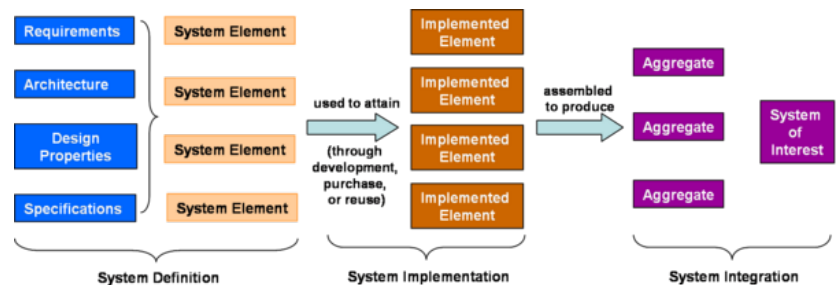


Figure 1. Simplification of How the Outputs of System Definition Relate to System Implementation, which Produces the System Elements Required to Produce Systems and Subsystems. (SEBoK Original)

Process Approach

Purpose and Principle of the Approach

During the implementation process, engineers apply the design properties and/or requirements allocated to a system element to design and produce a detailed description. They then fabricate, code, or build each individual element using specified materials, processes, physical or logical arrangements, standards, technologies, and/or information flows outlined in detailed descriptions (drawings or other design documentation). A system element will be verified against the detailed description of properties and validated against its requirements.

If subsequent verification and validation (V&V) actions or configuration audits reveal discrepancies, recursive

interactions occur, which includes predecessor activities or processes, as required, to mitigate those discrepancies and to modify, repair, or correct the system element in question. Figure 2 provides the context for the implementation process from the perspective of the U.S. Defense Acquisition University (DAU).

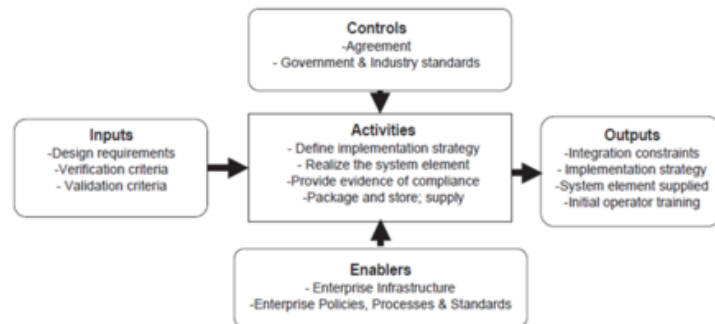


Figure 2. Context Diagram for the Implementation Process (DAU 2010). Released by the Defense Acquisition University (DAU)/U.S. Department of Defense (DoD).

Such figures provide a useful overview of the systems engineering (SE) community's perspectives on what is required for implementation and what the general results of implementation may be. These are further supported by the discussion of implementation inputs, outputs, and activities found in the National Aeronautics and Space Association's (NASA's) *Systems Engineering Handbook* (NASA 2007). It is important to understand that these views are process -oriented. While this is a useful model, examining implementation only in terms of process can be limiting.

Depending on the technologies and systems chosen when a decision is made to produce a system element, the implementation process outcomes may generate constraints to be applied on the architecture of the higher-level system; those constraints are normally identified as derived system requirements and added to the set of system requirements applicable to this higher-level system. The architectural design has to must take those constraints into account.

If the decision is made to purchase or reuse an existing system element, it has to must be identified as a constraint or system requirement applicable to the architecture of the higher-level system. Conversely, the implementation process may involve some adaptation or adjustments to the system requirement in order to be integrated into a higher-level system or aggregate.

Implementation also involves packaging, handling, and

storage, depending on the concerned technologies and where or when the system requirement needs to be integrated into a higher-level aggregate. Developing the supporting documentation for a system requirement, such as the manuals for operation, maintenance, and/or installation, is also a part of the implementation process; these artifacts are utilized in the system deployment and use phase. The system element requirements and the associated verification and validation criteria are inputs to this process; these inputs come from the architectural design process detailed outputs.

Execution of the implementation process is governed by both industrial and government standards and the terms of all applicable agreements. This may include conditions for packaging and storage, as well as preparation for use activities, such as operator training. In addition, packaging, handling, storage, and transportation (PHS&T) considerations will constrain the implementation activities. For more information, refer to the discussion of PHS&T in the System Deployment and Use article. The developing or integrating organization will likely have enterprise-level safety practices and guidelines that must also be considered.

Activities of the Process

The following major activities and tasks are performed during this process:

- **Define the implementation strategy** - Implementation process activities begin with detailed design and include developing an implementation strategy that defines fabrication and coding procedures, tools and equipment to be used, implementation tolerances, and the means and criteria for auditing configuration of resulting elements to the detailed design documentation. In the case of repeated system element implementations (such as for mass manufacturing or replacement elements), the implementation strategy is defined and refined to achieve consistent and repeatable element production; it is retained in the project decision database for future use. The implementation strategy contains the arrangements for packing, storing, and supplying the implemented element.
- **Realize the system element** - Realize or adapt and produce the concerned system element using the implementation strategy items as defined above.

Realization or adaptation is conducted with regard to standards that govern applicable safety, security, privacy, and environmental guidelines or legislation and the practices of the relevant implementation technology. This requires the fabrication of hardware elements, development of software elements, definition of training capabilities, drafting of training documentation, and the training of initial operators and maintainers.

- **Provide evidence of compliance** - Record evidence that the system element meets its requirements and the associated verification and validation criteria as well as the legislation policy. This requires the conduction of peer reviews and unit testing, as well as inspection of operation and maintenance manuals. Acquire measured properties that characterize the implemented element (weight, capacities, effectiveness, level of performance, reliability, availability, etc.).
- **Package, store, and supply the implemented element** - This should be defined in the implementation strategy.

Artifacts and Ontology Elements

This process may create several artifacts such as:

- an implemented system
- implementation tools
- implementation procedures
- an implementation plan or strategy
- verification reports
- issue, anomaly, or trouble reports
- change requests (about design)

This process handles the ontology elements shown in Table 1 below.

Table 1. Main Ontology Elements as Handled within System Element Implementation. (SEBoK Original)

Element	Definition
	Attributes (examples)

An implemented element is a system element that has been implemented. In the case of hardware it is marked with a part/serial number.

**Implemented
Element**

Identifier, name, description, type (hardware, software application, software piece, mechanical part, electric art, electronic component, operator role, procedure, protocol, manual, etc.)

**Measured
Property**

A measured property is a characteristic of the implemented element established after its implementation. The measured properties characterize the implemented system element when it is completely realized, verified, and validated. If the implemented element complies with a design property, the measured property should equal the design property. Otherwise one has to must identify the difference or non-conformance which treatment could conclude to modify the design property and possibly the related requirements, or to modify (correct, repair) the implemented element, or to identify a deviation.

Identifier, name, description, type (effectiveness, availability, reliability, maintainability, weight, capacity, etc.), value, unit, etc.

The main relationships between ontology elements are presented in Figure 3.

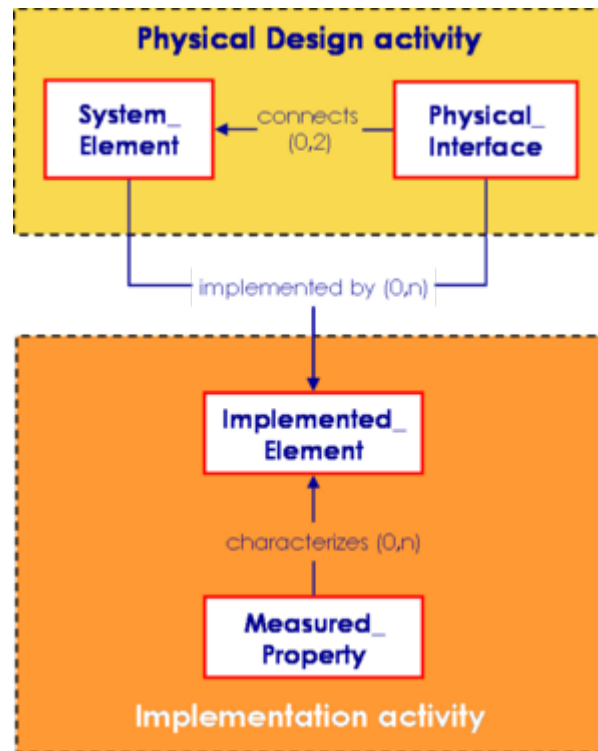


Figure 3. Implementation Elements Relationships with Other Engineering Elements. (SEBoK Original)

Methods, Techniques, and Tools

There are many software tools available in the implementation and integration phases. The most basic method would be the use of N-squared diagrams as discussed in Jeff Grady's book *System Integration* (Grady 1994).

Checking and Correctness of Implementation

Proper implementation checking and correctness should include testing to determine if the implemented element (i.e., piece of software, hardware, or other product) works in its intended use. Testing could include mockups and breadboards, as well as modeling and simulation of a prototype or completed pieces of a system. Once this is completed successfully, the next process would be system integration.

References

Works Cited

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

Grady, J.O. 1994. *System integration*. Boca Raton, FL, USA: CRC Press, Inc.

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

Primary References

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

Grady, J.O. 1994. *System Integration*. Boca Raton, FL, USA: CRC Press, Inc.

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. ISO/IEC/IEEE 15288:2015.

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

Additional References

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

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.10, released 06 May 2024

Retrieved from

"https://sandbox.sebokwiki.org/index.php?title=System_Implementation&oldid=71536"

This page was last edited on 2 May 2024, at 22:39.