

Why Model?

Why Model?

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

Lead Author: Sanford Friedenthal, **Contributing Authors:** Dov Dori, Yaniv Mordecai

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



Contents

Purpose of a Model

Indicators of an Effective Model

Model Scope

Indicators of Model Quality

Model-Based Metrics

References

Works Cited

Primary References

Additional References

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, verification, and validation of a system, as well as to 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 are:

- **Characterizing an existing system:** Many existing systems are 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 system maintenance 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 new standards to withstand earthquakes.
- **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 which verify that 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 and software-in-the-loop testing. 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 other stakeholders. Models may be a basis for developing a simulator 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 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. Models can also enable the capture of families of products.

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 remainder of this section and the topics Types of Models, System Modeling Concepts, and Modeling Standards describe indicators of an effective model (Friedenthal, Moore, and Steiner 2012).

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 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, take off, 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 provide a representation sufficient to 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 to assess the modeling effort, and, in some cases, the overall systems engineering (SE) effort. Different types of models provide different types of information. In general, models provide information that enables one to:

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

Models can capture metrics similar to those captured in a traditional document-based approach to systems engineering, but potentially with more precision given the more accurate nature of models 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, R. Steiner, and M. Kaufman. 2012. *A Practical Guide to SysML: The Systems Modeling Language*, 2nd Edition. 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.

Primary References

Friedenthal, S., A. Moore, R. Steiner, and M. Kaufman. 2012. *A Practical Guide to SysML: The Systems Modeling Language*, 2nd Edition. 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. Accessed April 13 at <https://www.incose.org/ProductsPublications/techpublications/GuideMetrics>

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.5, released 15 October 2021

Retrieved from

"https://www.sebokwiki.org/w/index.php?title=Why_Model%3F&oldid=62516"

This page was last edited on 14 October 2021, at 03:06.