What is a 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*

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

☐

Contents

Definition of a Model

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

- a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or

process (DoD 1998);

- 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 the system engineer who must specify, design, analyze, and verify systems, as well as share information with other stakeholders. A variety of system models are used to represent different types of systems for different modeling purposes. Some of the purposes for modeling systems are summarized in the topic Why Model?, and a simple taxonomy of the different types of models are described in the topic Types of Models. The modeling standards topic refers to some of the standard system modeling languages and other modeling standards that support MBSE.

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 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 (SoI).

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 to express their meaning as explained in "On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models" (Guizzardi 2007).

Just as engineering drawings express the 3D structure of mechanical and architectural designs, conceptual models are the means by which systems are conceived, architected, designed, and built. The resulting models are the counterparts of the mechanical design blueprint. However, the difference is that, while blueprints are exact representations of physical artifacts with a precise, agreed-upon syntax and long tradition of serving as a means of communication among professionals, conceptual models are just beginning to make headway toward being a complete and unambiguous representation of a system under development. The articles in the special section of Communications of the Association for Computing Machinery (ACM) (Dori 2003) present the abstract world of systems analysis and architecting by means of conceptual modeling, and how to evaluate, select, and construct models.

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 specifies the model constructs and the rules for constructing the model. 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 relationships. 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 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 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(θ)* and *cos(θ)*, the meaning of the pendulum position is understood in terms of these formalisms.

Modeling Tools

Models are created by a modeler using modeling tools. 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 palette 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 International Council of Systems Engineering (INCOSE) (INCOSE Systems Engineering Vision 2020 2007) defines MBSE 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." 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 and implementation teams.

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

The jointly sponsored INCOSE/Object Management Group (OMG) MBSE Wiki provides additional information on the INCOSE MBSE Initiative, including some applications of MBSE and some key topics related to MBSE such as sections on Methodology and Metrics, and Model Management.

The Final Report of the Model Based Engineering (MBE) Subcommittee, which was generated by the the National Defense Industrial Association (NDIA) Modeling and Simulation Committee of the Systems Engineering Division, highlights many of the benefits, risks, and challenges of a model-based approach, and includes many references to case studies of MBE (NDIA 2011).

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 1979), Integration Definition for Functional Modeling (IDEF0) (Menzel and Maier 1998), and enhanced functional flow block diagram (eFFBD). Other behavioral modeling techniques include the classical state transition diagram, statecharts (Harel 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.

In 2008, Estefan conducted an extensive survey of system modeling methods, processes, and tools and documented the results in A Survey of Model-Based Systems Engineering (MBSE) Methodologies (Estefan 2008). This survey identifies several candidate MBSE methodologies and modeling languages that can be applied to support an MBSE approach. Additional modeling methods are available from the MBSE Wiki under the section on Methodology and Metrics. The modeling standards section refers to some of the standard system modeling languages and other modeling standards that support MBSE. Since Estefan's report, a number of surveys have been conducted to understand the acceptance and barriers to model-based systems engineering (Bone and Cloutier 2010, 2014; Cloutier 2015).

References

Works Cited

Bellinger, G. 2004. "Modeling & simulation: An introduction," in *Mental Model Musings*. Available at: http://www.systems-thinking.org/modsim/modsim.htm.

Bone, M. & Cloutier, R. 2010. "The current state of model based systems engineering: Results from the OMG™ SysML request for information 2009." 8h Conference on Systems Engineering Research, Hoboken, NJ, USA, March 17-19, 2010, Paper #1569270919. Available at http://www.omgsysml.org/SysML_2009_RFI_Response_S

ummary-bone-cloutier.pdf. Accessed May 26, 2019.

Cloutier, R. & Bone, M. 2014. "MBSE survey," Presented January 2015 INCOSE IW, Los Angeles, CA. Available at: http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_survey_results_initial_report_2015_01_24.pdf. Accessed May 26, 2019.

Cloutier, R. 2015. "Current modeling trends in systems engineering," *INCOSE Insight,* vol. 18, no. 2.

Chen, P. 1976. "The entity relationship model – Toward a unifying view of data," ACM *Transactions on Data Base Systems,* vol. 1, no. 1, pp. 9-36.

DoD. 1998. "'DoD modeling and simulation (M&S) glossary," in *DoD Manual 5000.59-M*. Arlington, VA, USA: US Department of Defense. January. P2.13.22. Available at http://www.dtic.mil/whs/directives/corres/pdf/500059m.pdf.

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

Dori, D. 2003. "Conceptual modeling and system architecting," *Communications of the ACM*, vol. 46, no. 10, pp. 62-65. Available at: http://esml.iem.technion.ac.il/site/wp-content/uploads/2011/02/article169.pdf.

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.

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.

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,* vol. 8, no. 3, pp. 231–74.

Jackson, M.A. 1975. *Principles of Program Design.* New

York, NY, USA: Academic Press.

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, pp. 209-241.

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

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

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

Yourdon E. and Constantine L.L. 1973. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. 1st Edition.

Primary References

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.

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.

INCOSE. 2007. *Systems Engineering Vision 2020.* Seattle, WA, USA: International Council on Systems Engineering. September 2007. INCOSE-TP-2004-004-02.

NDIA. 2011. *Final Report of the Model Based Engineering (MBE) Subcommittee*. Arlington, VA, USA: National Defense Industrial Association. Available at: http://www.ndia.org/Divisions/Divisions/SystemsEnginee ring/Documents/Committees/M_S%20Committee/Reports /MBE_Final_Report_Document_(2011-04-22)_Marked_Fin al_Draft.pdf

Additional References

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, NJ, USA: Prentice Hall.

Harel, D. 1987. "Statecharts: A visual formalism for complex systems," *Science of Computer Programming,* vol. 8, no. 3, pp. 231–74.

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

OMG. "MBSE Wiki." Object Management Group (OMG). Available at: http://www.omgwiki.org/MBSE/doku.php. Accessed 11 September 2011.

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 >
SEBoK v. 2.10, released 06 May 2024

This page was last edited on 2 May 2024, at 22:17.