Models in Systems Engineering: From Engineering Artifacts to Source of Competitive Advantage

Azad M. Madni

Abstract Models in systems engineering have existed in various forms dating back to the 1950s. They have been used by engineers to understand various types of phenomena, envision future systems, and generate engineering artifacts. Today the increasing complexity of operational missions and technological advances enabled in part by disciplinary convergence and wide access to data are having a dramatic impact on system modeling. Operational missions are becoming increasingly more complex with multiple sources of uncertainty and subject to a variety of disruptions. Technological advances paced by advances in semantic technologies, machine learning, AI, and applied analytics are transforming model development into a closed-loop process. The advent of Industry 4.0 and digital engineering (including digital twin and digital thread) is causing models to be viewed in an entirely new light. And the convergence of engineering with other disciplines is opening up a whole new way of developing system models. This paper presents a historical perspective on models over several decades and offers a vision of how recent developments are likely to shape the trajectory of system models in the future.

Keywords Engineering models · Deterministic models · Probabilistic models · Learning models · Digital engineering · Industry 4.0

1 Introduction

Models, which have been a mainstay of engineering, are becoming central to systems engineering (SE) with the advent of model-based systems engineering (Madni and Sievers 2018). The questions today are determining where system modeling is headed and what impact it is likely to have on systems and enterprises.

A. M. Madni (✉)
University of Southern California, Los Angeles, CA, USA
e-mail: azad.madni@usc.edu

A. M. Madni et al. (eds.), *Recent Trends and Advances in Model Based Systems Engineering*, https://doi.org/10.1007/978-3-030-82083-1_48

These are some of the questions that the SE community is interested in as SE is being transformed to address the needs and challenges of the twenty-first century.

At the outset, it is worth reminding ourselves of George Box's (1976; Box and Draper 1987) famous refrain, "All models are wrong, but some are useful." The entire quote is: "All models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind." Box followed up on this cautionary comment with a more insightful and actionable refrain: "Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful." This quote essentially introduces the concept of "model fidelity." In SE, model fidelity is largely driven by the phase in SE life cycle and the intended purpose of the model (i.e., questions the model is expected to help answer).

Against the foregoing backdrop, this paper reviews the chronology of models over the past 60 years – first in engineering, then in SE, and most recently in model-based systems engineering (MBSE). It examines recent business and technology trends that are likely to shape the trajectory of system modeling in the next decade and what they foreshadow for system modeling in the twenty-first century.

This paper is organized as follows. Section 2 discusses the history of models in engineering. Section 3 presents models and modeling advances in SE. Section 4 presents the expanding role of models in MBSE. Section 5 discusses the growing importance of ontologies, knowledge graphs, metamodels, and reference models in MBSE. Section 6 reviews how models have evolved over the last several decades. Section 7 takes a look over the horizon to portend future advances in models. Section 8 summarizes the key points made in this paper.

2 Models in Engineering

Models have been used to envision architectures and engineering artifacts from time immemorial. In the early days, models took the form of sketches, which were a prelude to building physical artifacts. Over the ensuing years, models started to become increasingly more structured. Over the past 50 years, the importance of standardized representation, syntactic correctness, consistent semantic conventions, and the need to enforce semantic consistency in models was gradually recognized. This recognition enabled models to progress beyond drawings and concept maps to computer-based representations with a standard lexicon (vocabulary), syntax (grammar), and semantics (meaning). As important, the use of computer-based models expanded to the understanding of real-world phenomena as models grew in sophistication. Today models are being used to study and build complex sociotechnical systems with the ability to dynamically adapt, learn, and improve (Madni et al. 2018a).

Models are fundamentally abstractions (i.e., a simplified representation of reality) in which the simplifications are achieved through purposeful suppression of irrelevant real-world details (i.e., details that do not contribute to answering questions at hand). Abstractions can take a variety of forms, including generalizations from specific instances, uniform suppression of details not relevant to the purpose of the model, and selective suppression of details (e.g., elimination/simplification of certain perspectives or functions) to reduce complexity of envisioned systems or phenomenon under study (Madni et al. 2018b).

Models can be descriptive, prescriptive, or predictive. *Descriptive* models represent or explain a phenomenon, problem situation, or system to enhance shared human understanding and facilitate collaboration. *Prescriptive* models specify required/desired behaviors or courses of action. *Predictive* models facilitate exploration and help illuminate future outcomes in response to what-if assumptions and decisions/actions. Model purpose (i.e., the questions we want the model to answer at a desired level of detail) determines the scope and fidelity of models needed.

Today models are used in engineering analysis and design to visualize envisioned systems or modifications to existing systems; specify structure and behavior of systems; and understand how parts of a system inter-relate and behave in relation to each other and the external world. Models are also used to guide system development, assemble parts, and identify/generate and evaluate alternatives during design. And, finally, models are used to maintain an audit trail of assumptions and design decisions during system development (Madni et al. 2019).

3 Models in Systems Engineering (SE)

Models in SE have generally followed the evolution of models in traditional engineering disciplines but with a time lag. Over the years, models have appeared in a variety of forms: a "back of the envelope" calculation to ballpark a solution; a sketch on a napkin to communicate a germinating idea or an evolving concept; a computational algorithm to describe a physical law that lends itself to mathematical description; a deterministic representation to describe systems with known cause and effect; a probabilistic representation to capture environmental uncertainties and uncertainties in the knowledge of the system state space, as well as to account for random events; a statistical model to parsimoniously summarize the data collected over space and time; an architectural model to depict system structure and behavior and conduct trade-off analyses among performance and quality attributes associated with a system in its operational context; a logical model to describe how entities relate to each other in implementation-independent form; a data model to represent data in abstract form and organize and standardize how the data elements relate to each other; and a "learning" model that employs supervised, unsupervised, and reinforcement learning to increase model accuracy at system "build-time" and during system "run-time" (i.e., operational use).

Table 1 presents an approximate timeline of modeling methods that have been employed over time to deal with increasing problem and system complexity. Modeling in SE began with the ubiquitous block diagram or black box model,

Table 1 Rough chronology of models in systems engineering

Modeling Construct Name	Year Originated/Discipline	Application within SE (approx.)
Black Box (block diagrams)	1945/electronic circuit theory	1950
Functional Flow Block Diagram	1955/systems engineering	1955
Petri Nets	1962/concurrent hardware communication	1977
Hidden Markov Model/POMDP	1965/operations research, robotics	2015
Reinforcement Learning	1965/psychology, robotics	2014
Structure Analysis and Design Technique	1969/software and systems engineering	1969
Linear Temporal Logic	1977/computer science	1980
Data Flow Diagram	1979/software engineering	1980
N2 Diagram/Design Structure Matrix	1980/software and hardware design	1990
IDEF0	1981/manufacturing	1982
Contract-Based Design	1986/design automation	2000
State M	1987/computation theory	1990
Axiomatic Design	1990/system design	2002
Unified Modeling Language (UML)	1996/software engineering	1997
Digital Twins	2002/manufacturing	2019
Flexible Contract Approach	2010/design automation	2014

in which blocks represented system components and the arcs between the blocks represented the exchange of information, energy, and physical artifacts. Similarly, the N2 diagram and later the design structure matrix (DSM) gained popularity as a parsimonious way to represent a system along with its interactions and dependencies. In systems engineering, the N2 diagram was interpreted from a functional perspective, with the components in the N2 diagram being replaced with major system functions. Thereafter, the functional flow block diagram (FFBD) was developed to capture the dynamics of system behavior in a multitier, time-sequenced flow diagram depicting a system's functional flow. In the meantime, the software engineering community was engaged in developing data flow diagrams (DFDs) to model the data flow aspects of software systems. In the 1969–1973 timeframe, Structured Analysis and Design Technique (SADT) came into being within systems engineering and software engineering methodologies to describe systems as a hierarchy of functions (Ross 1977). It was subsequently formalized and published as Integrated Computer-Aided Manufacturing (ICAM) Definition, or

$IDEF_0$, in 1981 (Marca and McGowan 1987; Davis 1992; Mylopoulos 2004). The $IDEF_0$ representation was primarily championed by the USAF as a viable way to model systems. Not long after, several structured approaches emerged including structured programming, structured design, and structured analysis. It is worth noting that DFDs, N2 charts, and $IDEF_0$ diagrams all capture the same time-lapsed flow of information, energy, and physical artifacts among functions. Then came the recognition of the importance of system states and the advent of state machines (or state transition diagrams), which were adapted by several engineering disciplines to capture dynamic behavior. These methods were rapidly adopted and applied by the SE community to model system modes and states. It soon became evident that state machines suffered from a combinatorial explosion in their state space compromising their scalability. To ameliorate this problem, the SE community turned to heuristics, meta-rules, Petri nets, and Petri net variants (Zisman 1978). This strategy delayed the combinatorial explosion but did not eliminate it.

The past six decades have seen several contributions to systems modeling from a variety of disciplines such as electrical engineering, operations research, design automation, manufacturing, and software engineering. For example, formal modeling approaches for representing, analyzing, and designing systems originated in software engineering and design automation. The early modeling work relevant to SE, which drew on mathematical system representations, includes modeling formalisms (Tarski 1955), homomorphic relational structures (Klir 1991; Lin 1999), axiomatic design (Suh 1998), and structured analysis and design (Yourdon 1989).

It is interesting to note that many of the models being used in systems engineering had their origins in other disciplines such as electronic circuit theory, operations research, software engineering, design automation, robotics, and manufacturing. Also, some modeling approaches from other disciplines were adopted quickly by systems engineers, while others took more than a decade. This time lag was essentially a function of the need expressed by the SE community. For example, increasing system complexity and emphasis on system safety led systems engineers to employ formal and probabilistic methods to address verification and validation needs and uncertainties in knowledge of system states and the environment. Similarly, the advent of machine learning was only recently adopted by the SE community when it became apparent that many complex systems operate in uncertain, partially observable environments in which incoming information from sensor onboard vehicles and the environment help reduce the uncertainty in system models.

4 Models in Model-Based Systems Engineering (MBSE)

The historic use of models in SE is best characterized as "engineering with models." The basic idea is that models from different disciplines can be integrated to provide a solution to a problem that cuts across multiple disciplines (e.g., electrical, mechanical, thermal, optical). However, these models, based on different assumptions, were not designed with integration in mind. Model-based systems engineering (MBSE) is different from engineering with models. In MBSE, models represent an enduring and authoritative source of truth. MBSE replaces the traditional document-centric approach to SE while ensuring that documents can be produced on demand from the unified model and from the perspective of different stakeholders. This is similar to the automatic generation of code on demand in Model-Based Software Engineering. In MBSE, the models have shared assumptions and shared (or compatible) underlying ontologies and representations. Models in MBSE are centralized digital repositories that interconnect information from multiple sources and disciplines such that a change in one part of the model can be traced back to the original/derived requirement or use case. SysML, an extension of a subset of UML developed by the International Council on Systems Engineering (INCOSE) and subsequently advanced by Object Management Group (OMG) and INCOSE, became the popular system modeling language.

Over the past decade and a half, several MBSE methodologies have emerged (Estefan 2008). They include as follows: IBM Rational Unified Process (RUP) supported by IBM Rational Suite; INCOSE Object-Oriented Systems Engineering Method (OOSEM) developed with extensive aerospace involvement, which is supported by commercial SysML tools; Vitech's CORE product suite; JPL State Analysis Methodology; Dori's Object-Process Methodology (OPM); INCOSE MBSE Initiative, OMG's Model-Driven Architecture; and ISO/IEC 42010.

Today MBSE remains an important augmentation of SE as it continues to address additional phases of the system life cycle such as verification, validation, and testing. In this regard, the advent of digital twins (from digital engineering) can be expected to facilitate and accelerate system life cycle coverage (Madni et al. 2019).

5 Growing Importance of Ontologies, Knowledge Graphs, Metamodels, and Reference Models

Two key problems being addressed by the SE community today are to eliminate the miscommunications that frequently occur within SE teams and to assure interoperability among models and between information systems of collaborators. This focus led to the growing importance of ontologies, knowledge graphs, metamodels, and reference models.

Ontologies are thesauri of words representing concepts, the relationships among them, and the rules that help with model correctness checking (Sowa 1996, 2011). The model checking rules help with identification of gaps and semantic inconsistencies in system models. Ontologies have been a subject of study in the systems engineering community as a means to reduce modeling complexity, facilitate model verification, and enhance interoperability. From a data perspective, ontologies are semantic data models that define the types of entities in a particular domain, the relationship among the entities, and the properties that can be used to describe the entities. Ontologies are generic data models in that they only model generic types of entities that share certain properties but do not include information about specific entities in the domain. For example, an ontology might focus on generic vehicles, attempting to capture characteristics that most vehicles might have. By capturing information in this way, the ontology can be used to describe other vehicles in the future. An ontology comprises three main elements: classes, which are distinct types of entities that exist in the domain; relationships, which link any two classes; and attributes, which are properties that describe an individual class. When classes are linked through relationships, the ontology can be visualized as a graph.

A *knowledge graph* acquires and integrates information into an ontology and applies a reasoner to derive new knowledge (Ehrlinger and Wöß 2016). In other words, knowledge graphs are instantiations of ontologies. Using an ontology as an organizing framework, real data about specific entities in the domain can be added to create a knowledge graph. When data about specific entities are added for all entities in the ontology, a knowledge graph emerges. In other words, a knowledge graph is created when an ontology is used as an organizing construct for real-world data. Thus,

$$\boxed{\text{Ontology} + \text{Data} = \text{Knowledge Graph}}$$

Metamodels define the abstract syntax (i.e., grammar) of model description languages (Sprinkle et al. 2014). For example, the Unified Modeling Language (UML) metamodel defines the abstract syntax of various UML diagrams. More generally, metamodels express the logical syntactical structures that domain-specific models need to conform to for scalability, reuse, and extensibility. Metamodels are concerned with defining the symbols and structure for a predefined class of problems, along with rules that operate on the symbols. These properties allow the instantiation of a model from a metamodel. Thus, a metamodel defines the general structure, constraints, and symbols that can be used to model a system. Since metamodels do not specify the semantics of models, they do not have stand-alone use. However, ontologies and metamodels are complementing and synergistic. Specifically, an ontology can represent concepts and relationships formally using the structure provided by the metamodel. While ontologies may not use a metamodel, those that do will have certain desirable properties (e.g., interoperability, reuse, syntactic correctness, semantic consistency).

Reference models are abstract frameworks or domain-specific ontologies consisting of an interlinked set of clearly defined concepts produced by authoritative sources within defined stakeholder communities. A reference model can represent business functions and system components as long as they constitute a complete set. The terms in the reference model can be used to communicate ideas clearly among members of the SE community from vastly different backgrounds. The reference model is distinct from, but can include, related taxonomies, entities, and relationships to reveal hierarchies (e.g., system hierarchy, system architecture) relevant to stakeholders.

6 How Have Models Changed over the Last Several Decades?

After more than 50 years, system modeling has evolved in several important ways shown below:

- The starting point for modeling has changed from choosing a modeling construct to starting with a detailed analysis of needs to derive system modeling requirements which are then used to determine the right combination of models needed to model the system of interest.
- The scope of modeling has expanded – from a single system to networked systems, system of systems, and enterprises.
- Models have grown in sophistication – from deterministic to stochastic, probabilistic, and learning models.
- Engineering models used to be rooted in the engineering discipline. Today they are drawing on other disciplines such as biology, cognitive science, social science, economics, and entertainment arts.
- Earlier models used to be "data hungry." They needed complete information before they could provide value. Today models can cope with partial information and still provide value.
- Modeling used to be an open-loop process. Today modeling is being transformed to a closed-loop process that improves model completeness and accuracy based on data from collection assets, machine learning, and data analytics techniques. For example, virtual system models can now incorporate data from the corresponding physical system and become a digital twin (Madni et al. 2019).
- System representations have expanded from fixed structures to flexible representations which are needed to respond to systemic problems and adapt to external disruptions.
- System models are becoming increasingly more formal and rigorous to enable verification and validation, support simulation-based testing, and facilitate reasoning, interoperability, and reuse.
- Models are beginning to incorporate the capability to explain system behavior, an important characteristic that is key to model acceptance and trust in the engineering community. Explanation capability is needed for black box models, while interpretability is needed for glass box models.
- The SE community is much more cost conscious, with an emphasis on economic value derived from transitioning to MBSE (Madni and Purohit 2019).
- Industry view of models has changed from viewing them solely as engineering artifacts to viewing them as knowledge assets and a source of competitive advantage.

Today models are expanding into the behavioral domain. The human is no longer modeled as a transfer function, optimal controller, or utility maximizer. Rather, the human is modeled with an awareness of strengths (e.g., ability to generate creative options, rapid context awareness) and limitations (e.g., cognitive limitations, biases, tendency to lose focus). The advent of cyber-physical-human systems is a driver in

Table 2 System models: pre-2005 and today

System models	Pre-2005	Today
Comparison factors		
Starting Point	a modeling construct (e.g., IDEF0, SADT)	requirements derived from needs and mapped to appropriate combination of models
Focus	single system	networked systems, SoS, enterprise
Methods	deterministic (mostly)	deterministic, stochastic, probabilistic
Multidisciplinary Emphasis	minimal	significant
Model Requirements	need complete information	can work with partial information (e.g., POMDP)
Process	open loop	closed loop
Representation	fixed	flexible
Correctness Proof	not available	available
Rigor	structured representation; static correctness checking	formal representation; use of ontology and metamodel; support for formal reasoning
Learning	a priori supervised learning	in situ unsupervised and reinforcement learning
Explanatory Capability	none	some; distinguishes between interpretability (glass box models) and explainability (black box models)
Emphasis on ROI	modest	significant
Industry View	engineering artifact	knowledge asset; source of competitive advantage

this regard. As important, the age-old thinking of "humans versus machines" has been replaced by "humans and machines" with a growing emphasis on augmented intelligence (Madni 2020a).

Table 2 provides a comparison of pre-2005 system models and system models today.

7 Looking over the Horizon

With systems continuing to grow in complexity and missions continuing to become increasingly more challenging, the versatility and value of a model depend on its ability to provide useful information despite incomplete information; ability to acquire and reflect valid information pertaining to key system characteristics and behaviors of interest; ability to support simplifications (e.g., assumptions, approximates) while retaining requisite fidelity to provide correct answers; and ability to validate its outputs. Importantly, models require real-world measurements to test the validity of their predictions and explanations and for validation of outputs. It may not be feasible to meet these requirements in circumstances where input conditions cannot be adequately controlled or input and control conditions cannot be replicated.

With the recent surge in interest to transform and underpin SE with formal methods (e.g., linear temporal logic, contract-based design), three necessary characteristics of models surfaced: provably correct representation essential in applications where safety is paramount; flexible representation to support agility and resilience; and evidence-based learning to complete and refine models. Learning ability is crucial when operating in partially observable environments in which information about the system and the environment becomes incrementally available during mission execution. In response to these requirements, probabilistic learning models emerged including the flexible contract approach with the capacity to learn (Sievers and Madni 2016; Madni 2018a). This construct combines traditional contracts, partially observable Markov decision process (POMDP), reinforcement learning, and heuristics to strike an effective balance between model verifiability and flexibility (Sievers and Madni 2017; Madni et al. 2018a, b).

In the light of methodological advances and ongoing integration of MBSE with digital engineering, systems modeling can be expected to evolve in new and exciting directions. We already see evidence of formal methods being introduced within the MBSE rubric. Specifically, the concept of ontologies from computer science is being introduced into MBSE to enhance semantic consistency, enhance interoperability, and formalize scope with respect to the system modeling activity. In particular, ontologies can be expected to play important roles in answering stakeholder/user questions by capturing key concepts and relationships from use cases of interest and supplemented by expert knowledge. The scope of system modeling can be expected to expand to cover probabilistic modeling, formal modeling, modeling with incomplete or partial information, and learning models (i.e., supervised, unsupervised, and reinforcement learning). Models can be expected to have richer semantic foundations to reflect new perspectives made possible by disciplinary convergence. These advances and enhancements will enable more detailed questions to be answered earlier in the system's life cycle. With growing convergence of engineering with entertainment arts, it will be possible to transform system models into stories that can be executed in simulation or in virtual worlds (Madni et al. 2014; Madni 2015). Importantly, enterprises are beginning to increasingly rely on their suppliers, application providers, and tool vendors to create sustainable competitive advantage in their respective markets. This reliance calls for seamless interoperability. The latter can be achieved through models based on domain ontologies with interoperability being enabled by creating a semantic layer between enterprises and their technology/tool providers (Madni 2020b). As a result, system models are no longer being viewed as engineering artifacts but rather as knowledge assets and a source of competitive advantage for organizations.

8 Summary

Models have been a mainstay of systems engineering for several decades. However, the types of models and the value they provide have changed dramatically. The types of models used for system modeling have evolved considerably driven in large part by the increasing complexity of the system and the environment and advances made in formal and probabilistic methods, machine learning, and applied analytics. These advances have transformed modeling from being a one-shot open-loop activity to an iterative closed-loop activity informed by evidence and results of machine learning. Importantly, the historical view of models as engineering artifacts has changed dramatically. Today they are viewed as sources of competitive advantage. The competitive advantage results from the ability to reuse models, completely or in part, to rapidly achieve interoperability in risk-mitigated fashion with third-party applications and tools and enable the use of ontologies and metamodels. The growing importance and adoption of MBSE in major organizations coupled with the advent of digital engineering make digital twin-enabled MBSE especially effective for model-based V&V. This paper has addressed both modeling problems and how far along systems modeling has advanced as a result of problem pull and enabled by advances in system modeling and ongoing convergence of systems modeling with machine learning, data analytics, and entertainment arts (Madni 2018b). This trend can be expected to continue and grow in the future. As a result of these advances, systems models are becoming knowledge assets and a source of competitive advantage in various industries.

References

Box, G.E.P. 1976. Science and Statistics. *Journal of American Statistical Association* 71 (356): 791–799.

Box, G.E.P., and N.R. Draper. 1987. *Empirical Model Building and Response Surfaces*. New York: Wiley.

Davis, W.S. 1992. *Tools and Techniques for Structured Systems Analysis and Design*. Addison-Wesley. ISBN 0-201-10274-9.

Ehrlinger, L., and W. Wöß. 2016. Towards a Definition of Knowledge Graphs. In *SEMANTICS*, September 13–14, Leipzig, Germany.

Estefan, J. 2008. INCOSE Survey of MBSE Methodologies, USA, WA, Seattle: INCOSE TD 2007-003-02.

Klir, G. 1991. *Facets of Systems Science*. New York: Plenum.

Lin, Y. 1999. *General Systems Theory: A Mathematical Approach*. New York: Kluwer Academic/-Plenum.

Madni, A.M. 2015. Expanding Stakeholder Participation in Upfront System Engineering Through Storytelling in Virtual Worlds. *Systems Engineering* 18 (1): 16–27.

———. 2018a. Formal Methods for Intelligent Systems Design and Control. In *AIAA SciTech Forum, 2018 AIAA Information Systems, AIAA InfoTech@Aerospace*, Kissimmee, Florida, January 8–12.

———. 2018b. *Transdisciplinary Systems Engineering: Exploiting Convergence in a Hyperconnected World (forward by Norm Augustine)*. Springer, September.

———. 2020a. Exploiting Augmented Intelligence in Systems Engineering and Engineered Systems. *INSIGHT Special Issue, Systems Engineering and AI*, March.

———. 2020b. Minimum Viable Model to Demonstrate the Value Proposition of Ontologies for Model Based Systems Engineering. In *2020 Conference on Systems Engineering Research (CSER)*, October 8–10.

Madni, A.M., and S. Purohit. 2019. Economic Analysis of Model Based Systems Engineering. In *MDPI Systems, special issue on Model-Based Systems Engineering*, February.

Madni, A.M., and M. Sievers. 2018. Model-Based Systems Engineering: Motivation, Current Status, and Research Opportunities, Systems Engineering. In *Special 20th Anniversary Issue*, Vol. 21, Issue 3.

Madni, A.M., M. Sievers, and D. Erwin. 2019. Formal and Probabilistic Modeling in the Design of Resilient Systems and System-of-Systems. In *AIAA Science and Technology Forum*, San Diego, California, January 7–11.

Madni, A.M., M. Spraragen, and C.C. Madni. 2014. Exploring and Assessing Complex System Behavior Through Model-Driven Storytelling. In *IEEE Systems, Man and Cybernetics International Conference, invited special session Frontiers of Model Based Systems Engineering*, San Diego, CA, October 5–8.

Madni, A.M., M. Sievers, A. Madni, E. Ordoukhanian, and P. Pouya. 2018a. Extending Formal Modeling for Resilient System Design. *Insight* 21 (3): 34–41.

Madni, A.M., M. Sievers, and C.C. Madni. 2018b. Adaptive Cyber-Physical-Human Systems: Exploiting Cognitive Modeling and Machine Learning in the Control Loop. *Insight* 21 (3): 87–93.

Madni, A.M., C.C. Madni, and D.S. Lucero. 2019. Leveraging Digital Twin Technology in Model-Based Systems Engineering. In *MDPI Systems, special issue on Model-Based Systems Engineering*, February.

Marca, D., and C. McGowan. 1987. *Structured Analysis and Design Technique*. McGraw-Hill. ISBN 0-07-040235-3.

Mylopoulos, J. 2004. Conceptual Modelling III. Structured Analysis and Design Technique (SADT).

Ross, D.T. 1977. Structured Analysis (SA): A Language for Communicating Ideas. *IEEE Transactions on Software Engineering* SE-3 (1): 16–34.

Sievers, M., and A.M. Madni. 2016. Agent-Based Flexible Design Contracts for Resilient Spacecraft Swarms. In *AIAA Science and Technology 2016 Forum and Exposition*, San Diego, CA.

———. 2017. Contract-Based Byzantine Resilience for Spacecraft Swarm. In *2016 AIAA Science and Technology Forum and Expo*, Grapevine, Texas, January 9–13.

Sowa, J.F. 1996. Top-Level Ontological Categories. *International Journal of Human-Computer Studies* 43 (5/6): 669–686.

———. 2011. Ontology Metadata and Semiotics. In *Conceptual Structures: Logical, Linguistic, and Computational Issues, LNAI 1867*, ed. B. Ganter and Mineau, 55–81. Berlin: Springer.

Sprinkle, J., B. Rumpe, H. Vangheluwe, and G. Karsai. 2014. Metamodeling: State of the Art and Research Challenges. *arXiv*, September.

Suh, N.P. 1998. *Axiomatic Design Theory for Systems in Research in Engineering Design*. Vol. 10, 189–209. Berlin: Springer.

Tarski, A. 1955. Contributions to the Theory of Models I II II. *Nederlandse Akademie Wetenschappen Proceedings Series A* 57: 572–581.

Yourdon, E. 1989. *Modern Structured Analysis*. Upper Saddle River: Yourdon Press.

Zisman, M. 1978. *Use of Production Systems for Modeling Asynchronous Concurrent Processes, Pattern-Directed Inference Systems*, 53–68. Academic.