



WHITEPAPER

# The Real Intent of Model-Based Systems Engineering (MBSE)— Keeping Up with Complexity

Written by Lou Wheatcraft

Senior Consultant & Managing Member at Wheatland Consulting, LLC  
and current co-chair of the INCOSE Requirements Working Group (RWG)

GREEN  
YELLOW



# Introduction

**Today's products are becoming increasingly complex and software intensive. This presents major challenges for organizations being able to effectively manage all the data, information, and artifacts across the lifecycle for these types of systems – and one of the key reasons that Model-Based Systems Engineering (MBSE) is fast becoming the standard approach to systems engineering for digital transformation today.**

Historically, organizations defined and recorded the data and information associated with the various artifacts in the form of “documents.” As products have become more complex and regulated, the sheer volume of documentation has become overwhelming; especially in terms of configuration management and change control. Because of the complexity, there are more people involved in the development of these systems spread out in different geographical locations. This results in many of the documents being developed and managed in silos with limited collaboration.

Because of these issues, it is nearly impossible to keep all the data and information contained within the various documents in sync, current, correct, and consistent resulting in no real single source of truth (SSoT). For highly regulated products, the amount of documentation that must be developed, maintained, and supplied to regulators to show compliance has become overwhelming. Inconsistencies in these documents can result in numerous negative outcomes and ultimately even a product that is not approved for use:

- The cost and time overhead associated with managing the data and information within documents consume a significant part of development cost, eating into profit margins.
- This additional overhead results in longer development times and time to market for many products, making a company less competitive.
- Quality suffers resulting in post product launch costs associated with increases returns, recalls, warranty work, and company image degradation associated with negative social media comments – all of which also eat into profits.
- A highly regulated product that is not approved for use can result in a company's stock price plummeting and, in some cases, the company going bankrupt.

This old, 20th century, document-centric approach is no longer effective for many of today's products of the 21st century. As a result, many organizations are seeking approaches such as MBSE to address these challenges.

An organization that is leading the move towards MBSE is the International Council of Systems Engineers (INCOSE) of which the author of this paper is an active member. As stated in the INCOSE Vision 2025, *“a constant theme throughout the evolution of systems engineering is the ever-increasing complexity of systems which can be observed in terms of the number of system functions, components, and interfaces and their non-linear interactions and emergent properties. Each of these indicators of complexity has increased dramatically over the last fifty years and will continue to increase due to the capabilities that stakeholders are demanding and the advancement in technologies that enable these capabilities. This complexity presents organizations with challenges that need to be addressed. Failures can result in tremendous loss.”*

Out of this search for a better way, MBSE has emerged as a popular approach to address these challenges.

In the following chapters, we will answer some of the most common questions around MBSE, including:

- What is MBSE?
- What is the real intent of MBSE?
- Is MBSE the same thing as SysML?



# What is MBSE?

If you ask 20 people to define MBSE, you might get 20 different answers.

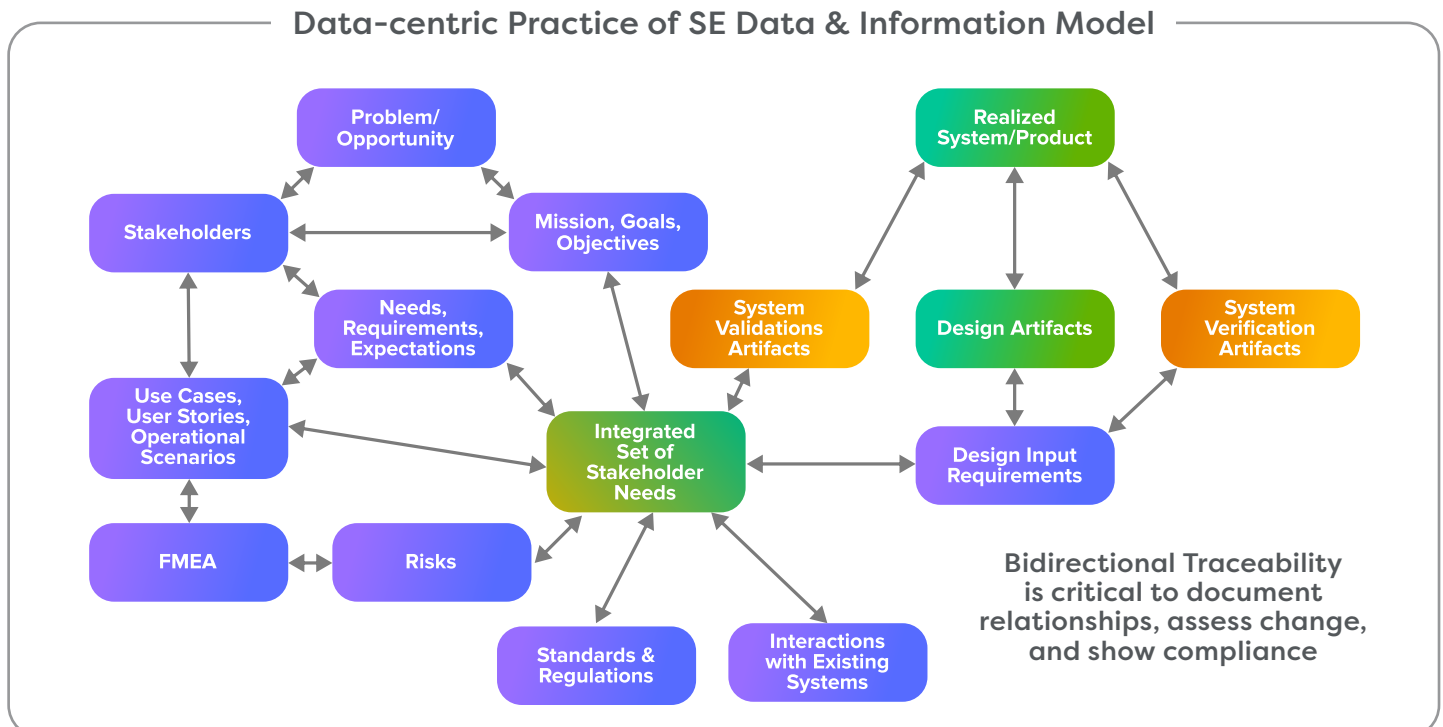
The core of confusion is the word “model.”

The INCOSE Systems Engineering Handbook (SE HB) states that “a model that represents a system and its environment is of particular importance to the systems engineer who must analyze, specify, design, and verify systems, as well as share information with other stakeholders. Different types of models are used to represent systems for different modeling purposes.”

In the INCOSE SE HB, the word “model” shows up over 680 times! The term is used to refer to the various kinds of models, visualizations of the data and information contained the various types of models, as well as documents, diagrams, drawings, or any other representation of a system.

In this context, a “model” can be any representation or abstraction of the system of interest that best suits the purpose for which it was created. Thus, there are various kinds of models including analytical, logical, behavioral, and architectural models, textual descriptions, as well as documents, functional flow diagrams, data flow diagrams, interface diagrams, architectural diagrams, drawings, or any other representation of a system.

The common denominator of all these representations is that they are all visualizations of the underlying data and information that is used to construct and represent the model.



Again, referring to the INCOSE SE HB:

*“MBSE is often contrasted with a traditional document-based approach to SE. In a document-based SE approach, there is often considerable information generated about the system that is contained in documents and other artifacts such as specifications, interface control documents, system description documents, trade studies, analysis reports, and verification plans, procedures, and reports. The information contained within these documents is often difficult to maintain and synchronize, and difficult to assess in terms of its quality (correctness, completeness, and consistency).”*

*“In an MBSE approach, much of this information is captured [electronically] in a system model or set of models. The system model is a primary artifact of the SE process. MBSE formalizes the application of SE through the use of models. The degree to which this information is captured in models and maintained throughout the life cycle depends on the scope of the MBSE effort. Leveraging an MBSE approach to SE is intended to result in significant improvements in system requirements, architecture, and design quality; lower the risk and cost of system development by surfacing issues early in the system definition; enhance productivity through reuse of system artifacts; and improve communications among the system development team.”*

Given there can be multiple types of models, and visualizations of the data and information in those models generated while progressing through the system life cycle processes, a 10,000-foot level view of SE from a data-centric perspective helps stakeholders understand the context in which various artifacts and their underlying data and information are generated and used.

MBSE is not really about drawing diagrams that contain boxes and lines but rather is about the underlying data and information model itself that enables consistency across the data and information that represents the various models and visualizations.



## What is the Real Intent of MBSE?

The goal of an organization adopting MBSE is to move from a 20th century document-centric practice of systems engineering (SE) toward a data-centric practice of SE to realize the real intent of MBSE which is to develop, maintain, and manage a data and information model of the system being developed along with a model of all the system life cycle process activities, resulting artifacts, and their underlying data and information.

*“MBSE, from a data-centric perspective, involves the formalized application of shareable sets of data to represent the SE work products and underlying data and information generated to support lifecycle concept maturation, needs and requirements definition, design, analysis, and verification and validation activities throughout the system life cycle, from conceptual design to retirement.”<sup>1</sup>*

With this data-centric perspective of SE, the capability to capture, manage, access data, and manage the interrelationships between SE work products can be accomplished through a variety of methodologies, which range from the establishment of a single relational database to a virtually integrated, but distributed, database by means of a federation (or data map/index) of disparate data sources.

This data and information can be captured using a variety of SE tools and applications. To effectively manage the development of our ever increasing complex, software-centric systems there are benefits to managing this underlying data and information in such a way it can be shared across the system lifecycle process activities, shared between the various SE tools used to create and manage this data and information, and shared between organizations involved in the development and operations of the system of interest. This sharing will help ensure correctness, consistency, and completeness of the data and information typical of our ever increasingly complex systems. It will also enable collaboration between not only the project team members, but also external stakeholders involved in the development of the system.

To manage the development of increasingly complex software-centric systems, managing data in a way that can be shared across the system lifecycle process activities, and between all key stakeholders involved, can ensure consistency and correctness.

SE cannot be effectively practiced when viewed from just one perspective (requirements, models, patterns, standards, industry specific application, etc.). To successfully practice SE, wise systems engineers recognize and use each perspective as appropriate to the activity they are performing. Based on their needs they choose the appropriate tool and visualization that will meet their needs.

The degree to which the data and information is captured and managed is driven by the needs of the organization and programs from a business and technical perspective based on the size and complexity of their programs, their product line, culture, processes, workforce, their diversity and complexity of supply chains, and types of engineering information that comprises a technical baseline for their system of interest.

## MBSE is NOT SysML!

To some practitioners, MBSE is equated to the use of SysML and other language-based modeling tools. However, as discussed above, MBSE is much more than using language-based modeling tools to develop analytical, logical, behavioral, and architectural models. To others, what they are calling MBSE is really Model-based Design (MBD).

**MBSE is much more than using language-based modeling tools to develop analytical, logical, behavioral, and architectural models.**

MBD design starts with a set of requirements and using various language-based modeling tools to architect and design a system, develop simulations to assess the behavior of the system, and then develop a set of design output specifications to which the system is built or coded. Often the MBD activities are completed in a silo separate from the other SE lifecycle process activities.

However, this approach does not address the true intent of MBSE. The data and information model that must be developed must represent all the SE artifacts generated during the product development process activities across all lifecycle stage activities. Thus, information associated with the elicitation of stakeholder needs and requirements, lifecycle concept definition, analysis, and maturation, deriving an integrated set of needs, and transforming those needs into sets of design input requirements, defining an architecture, designing the system, verification the system meets those requirements, and validation that the system meets the needs must also be captured within the data and information model. In addition, relationships and dependencies of the individual data items must be captured (traceability) in order to help determine and ensure consistency across all lifecycle stages as well as help assess the impact of changes made to any of the data items. Design inputs must be traceable to design outputs.



# Benefits of Implementing Systems Engineering from a Data-Centric Perspective<sup>1</sup>

**A data-centric perspective of SE complements the system life cycle process activities by enabling the opportunity for system development with increased quality, lower cost, and lower risk. Adopting MBSE and implementing a data-centric perspective enables organizations to realize the following benefits:**

- Meet the challenges associated with increasing complexity for current and future systems discussed in the previous section.
- Provide greater consistency of all products because any single piece of design data and information can be expressed authoritatively within integrated/federated, shareable sets of data that can later be referred to by others for decisions or formation of other work products.
- Provide better visibility into the principal characteristics of the whole integrated system because multiple views from the data and information model can be created that succinctly address specific stakeholder needs, concerns, and interests.
- Provide greater congruence and configuration management between documentation and reality. Differing views of the underlying data and information model can be automatically generated into SE artifacts, reducing the effort to keep the artifacts and their underlying data and information up to date and consistent, resulting in artifacts that match the best available, current data and information.
- Establish “single source of truth.” Single source of truth (SSoT) represents the official state or baseline version of data and information — regardless of what someone says or thinks, no matter what they “remember” or what perspective they have concerning what is being done or built or a decision that was made, if it isn’t in the project’s configuration managed, data and information model, it isn’t the truth. (Requires all the underlying data and information to be configuration managed and kept current and consistent.)
- Facilitate the navigation, traceability, and interrogation of data and information across all system life cycle stage activities. Managers and engineers can have access to the correct, complete, and consistent data and information more quickly, and on an as-needed basis, without going through manual distribution or search processes.
- Enable the reuse of SE and PM artifacts and underlying data and information. Considerable time and expense can be saved when an organization can reuse SE and PM data and

information and not have to start from scratch for each new project (brown field systems). This reuse ability is key to effective product line management.

- Facilitate the management of the stakeholder needs, requirement definition, design, build/code, system verification, and system validation activities in an integrated, consistent manner. Data and information associated with verification and validation activities across all system life cycle process activities can have higher quality and provide greater insight concerning the status of verification and validation activities. This makes it easier to show compliance and that stakeholder needs are being met.
- Reduce the costs associated with erroneous design and resulting rework. Analysis of the SE work products and underlying data and information can reveal a flaw or inconsistency as soon as it is created, enabling correction before downstream work is done, work that would be invalid, and costly and time consuming to correct if the upstream mistake were not corrected immediately. This also helps to avoid huge expenses associated with recalls, returns, warranty work, and negative comments on social media.
- Facilitate the identification of interactions (interfaces), helping to ensure the system of interest can be successfully integrated into the macro system it is a part and reducing integration issues and costly rework and schedule slips associated with these issues.
- Provide identification, management, interoperability, and integration of artifacts and underlying data and information across business or organizational elements needed to support program budget and schedule goals. For example, with the ability to metatag data, information, and work products, they can directly be linked to the Work Breakdown Structure (WBS), budget, schedule, and risk management activities.
- Ensure data and information needed by programs and projects (e.g., for milestones, gate reviews, mission operations, risk mitigation, and anomaly investigations, decisions, and outcomes) are identified and managed to provide traceability of the data and information used in decision-making.
- Use measures to better manage SE activities across all system life cycle process activities. Measures allow managers and systems engineers to monitor trends, assess progress, identify issues to help ensure the system being developed will meet stakeholder needs and expectations.



# Parting Thoughts

In order to effectively develop today's increasingly complex and software intensive systems and avoid the many negative consequences of the outdated document-centric practice of SE, organizations must move to a data-centric practice of SE which is the real intent of MBSE. Organizations need to develop a level of organizational SE capability that will enable them to realize the benefits listed above.

Since one size doesn't fit all, an organization needs to assess the SE capabilities that best fit its domain, product line (degree of complexity), and culture. The level of SE capability an organization establishes needs to be tailored to the size and complexity of systems developed by the organization, whether small, medium, or large projects.

Based on the needs of the organization and the level of SE capability, they will need to choose the appropriate SE toolset, update their processes, and train their people in these tools and processes.

## REFERENCES

<sup>1</sup> INCOSE INCOSE-TP-2018-001-01, 2018, Integrated Data as a Foundation of Systems Engineering, prepared by the Requirements Working Group, INCOSE.

<sup>2</sup> INCOSE-TP-2010-006-03, 2019, Guide to Writing Requirements, prepared by the Requirements Working Group, INCOSE.

<sup>3</sup> Wheatcraft, L., Ryan, M., Dick, J., Llorens, J., 2019. "The Need for an Information-based Approach to Requirement Development and Management", paper and poster presentation at INCOSE IS 2019.



## ABOUT THE AUTHOR

Lou Wheatcraft is a senior consultant and managing member of Wheatland Consulting, LLC. Lou is an expert in systems engineering with a focus on needs and requirements development, management, verification, & validation. Lou provides consulting and mentoring services to clients on the importance of well-formed needs & requirements helping them implement needs & requirement development and management processes, reviewing and providing comments on their needs and requirements, and helping clients write well-formed needs & requirements.

Specialties include: Understanding and documenting the problem; defining project & product scope; defining and maturing system concepts; assessing, mitigating, & managing risk; documenting stakeholder needs; transforming needs into well-formed design input requirements; allocation, budgeting, and traceability; interface management, requirement management; & verification and validation.

Lou's goal is to help clients practice better systems engineering from a needs & requirements perspective across all life cycle stages of system/product development. Getting the needs & requirements right upfront is key to a successful project. Poor needs & requirements can triple the chances of project failure.

Lou has over 50 years' experience in systems engineering, including 22 years in the United States Air Force. Lou has taught over 200 requirement seminars over the last 21 years. Lou supports clients from all industries involved in developing and managing systems and products including aerospace, defense, medical devices, consumer goods, transportation, and energy.

Lou has spoken at Project Management Institute (PMI) chapter meetings and INCOSE conferences and chapter meetings. Lou has published and presented many papers concerning needs and requirement for NASA's PM Challenge, INCOSE, INCOSE INSIGHT Magazine, and Crosstalk Magazine. Lou is a member of INCOSE, past Chair and current Co-Chair of the INCOSE Requirements Working Group (RWG), a member of the Project Management Institute (PMI), the Software Engineering Institute (SEI), the World Futures Society, and the National Honor Society of Pi Alpha Alpha.

Lou has a BS degree in Electrical Engineering from Oklahoma State University; an MA degree in Computer Information Systems; an MS degree in Environmental Management; and has completed the course work for an MS degree in Studies of the Future from the University of Houston – Clear Lake.



## ABOUT JAMA SOFTWARE

Jama Software® is focused on maximizing innovation success in multidisciplinary engineering organizations. Numerous firsts for humanity in fields such as fuel cells, electrification, space, software-defined vehicles, surgical robotics, and more all rely on Jama Connect® requirements management software to minimize the risk of defects, rework, cost overruns, and recalls. Using Jama Connect, engineering organizations can now intelligently manage the development process by leveraging Live Traceability™ across best-of-breed tools to measurably improve outcomes. Our rapidly growing customer base spans the automotive, medical device, life sciences, semiconductor, aerospace & defense, industrial manufacturing, consumer electronics, financial services, and insurance industries. To learn more, please visit us at [jamasoftware.com](http://jamasoftware.com).