

FUTURE DIRECTIONS FOR PRODUCT LIFECYCLE MANAGEMENT (PLM) AND MODEL-BASED SYSTEMS ENGINEERING (MBSE)



Sanford Friedenthal, SAF Consulting

Sponsored by Aras



Future Directions for Product Lifecycle Management (PLM) and Model-based Systems Engineering (MBSE)

SANFORD FRIEDENTHAL, SAF CONSULTING

1. INTRODUCTION

Industries such as automotive, aerospace, biomedical, and telecommunications, continue to face increasing system and product development challenges that require a strategic response. Organizations must innovate to advance technology at an accelerating rate in areas such as computing, networking, sensors, and materials, and insert these technologies into their product development. They must manage growing system complexity that results from increased system and software functionality and inter-connectedness. The need for new technologies and increases in system complexity are often in response to customer demands for smarter and more autonomous systems that require less human interaction and are more fault tolerant and secure. Organizations must provide this capability within the constraints of shortened development cycles and reduced costs, while relying on extended supply chains, all to meet increasing global competition.

An organization's capability to develop and evolve systems must address these challenges to stay competitive. The development process is complex in its own right and involves interaction and collaboration among many different engineering disciplines and other stakeholders across the lifecycle. The development process must also facilitate innovation, while at the same time, leverage previous design experience and knowledge that builds on existing products.

Today's multi-disciplinary engineering practices are often stove-piped and disconnected creating many inefficiencies, rework, and limit the opportunity for reuse. This paper explores how Product Lifecycle Management (PLM) and Model-based Systems Engineering (MBSE) are complementary approaches that can be part of an organizational strategy to address many of these development challenges. The combination of PLM and MBSE can facilitate the integration of engineering processes, tools, and data to improve productivity, quality, shorten cycle time, and reduce risk in the development process.

PLM is a general approach to manage processes and data throughout the lifecycle development of a system or product. MBSE is an approach that emphasizes the development of a system model that integrates across subsystems and engineering disciplines. Both the PLM and MBSE approaches reflect evolving practices from their own distinct roots.

2. PLM BACKGROUND

The PLM approach is an outgrowth of Product Data Management (PDM) applications that manage versions of product configuration data throughout the product lifecycle. The term PDM has traditionally been used to describe the applications for document and computer-aided design (CAD) data management, whereas PLM reflects the broader approach to managing the multi-disciplinary engineering data across the product lifecycle. The original PDM applications related parts in the product parts list, often called a bill of materials (BOM), to their geometric representation in mechanical CAD tools. This provided the means for controlling the configuration to release to manufacturing.

Over the years, the PDM applications continued to evolve to manage a broader scope of engineering data. This included managing the relationship between the mechanical CAD representation and the processes used to manufacture the parts, such as processes that are implemented by software to drive numerical controlled machines. The PDM applications also evolved to support more complex CAD representations as they transitioned from 2-dimensional representations to 3-dimensional representations. In addition, these applications began to manage the relationship between the mechanical CAD models and the structural, thermal, and other analysis models that rely on the geometrical representation.

In more recent years, the PDM applications began to manage relationships to engineering data that is not merely mechanical in nature. This includes managing the relationship to electrical design models of an electronics assembly, and managing the relationship to software design artifacts, which is sometimes referred to as Application Lifecycle Management (ALM). In addition, PDM applications now manage the relationship to requirements as well as design data. PDM applications support a PLM approach, particularly as they extend their scope to include engineering data that increasingly covers more of the 'Vee' process [1] from requirements to design to test to manufacturing.

3. MBSE BACKGROUND

The roots of systems engineering are often debated, but can be traced to operations research, industrial engineering, and foundational work in system science and general systems theory. It is generally agreed that the discipline of systems engineering evolved substantially from its application to the space program beginning in the 1950's. Systems engineering provides a holistic approach to the development of a system to achieve its mission objectives. A prime example of its application is the achievement in 1969 'of landing a man on the moon and returning him safely to the Earth.'

Systems engineering is used to support the specification, design, analysis, and verification of the system, and must be initiated early in the lifecycle to have the largest impact on product development. Trade-off analysis is performed throughout the lifecycle to identify and evaluate alternatives, and select a balanced system solution among competing considerations that span the design elements (i.e., subsystems and components), and span different facets of the design, and different phases of the lifecycle. The different facets of design include functionality, performance, physical, and other quality characteristics such as safety, reliability, producibility, maintainability, as well as lifecycle cost and risk. The different phases of the lifecycle include development, operation, manufacturing, support, and disposal. To achieve this balance, systems engineering is inherently multi-disciplinary.

In addition to trade-off analysis, many other systems engineering techniques are commonly used to support the specification, design, analysis, and verification of systems. These include requirements traceability analysis, functional analysis, architecture design, performance simulation, and risk management, to name a few. Many different kinds of design and analysis models are also used as part of the systems engineering toolkit.

Over the last decade, increased emphasis has been placed on developing an integrated system model to capture information about the system. This MBSE approach contrasts with a more traditional document-based approach that captures similar information in documents, such as specifications, interface control documents, architecture description documents, and others. The system model is intended to ensure a more precise, consistent, and traceable system design that reflects the multi-disciplinary design considerations.

The system model is one of many engineering models that include hardware and software design models, and diverse simulation and analysis models. However, the system model can play a key role in facilitating integration by capturing shared aspects of the system that are used by other engineering models. This includes information related to systems, subsystems and components such as their requirements, function, states, interfaces, key system properties, and their inter-relationships.

4. PRODUCT DEVELOPMENT LIMITATIONS

As noted in the introduction, the product development challenges require organizations to leverage rapidly changing technologies, and to manage increasing system complexity within development constraints. However, this challenge is made more difficult within an organization because many engineering processes and tools have evolved from their discipline-specific roots without an over-arching integrated set of processes, tools, and data that brings together the workforce in an efficient and effective way.

The engineering processes span the entire product development lifecycle that includes conceptual design, preliminary, design, detailed design, and integration and test. These processes continue to be applied through product manufacturing and support to address defects, technology insertion, and evolving customer needs. The lack of integration in each phase results in inefficiencies, defect insertion, and limits the ability to assess change impacts and respond to changes.

Integrating the development processes is essential from Day 1 on a project. Right from the start, engineers from each discipline must begin working to meet early project milestones such as the procurement of long-lead items. The lack of integration across disciplines can result in early design decisions that can have significant adverse downstream cost and schedule impacts.

An organization's engineering change process can be an indicator of the level of integration of their multi-disciplinary engineering processes. The change process includes identifying the need for the change, assessing the potential impact of the change, implementing the change, and must be managed to ensure proper orchestration among the development team. This often requires program and engineering management to coordinate analysis, design, and test activities that are performed by many engineering disciplines that include systems, software, electrical, mechanical, reliability, safety, test, manufacturing, and others. Each engineer uses their discipline-specific engineering process, tools, and data, and often relies on informal inter-discipline communications and manual exchange of data. This lack of integration can significantly limit the effectiveness of the overall process. Some contributing factors that limit integration are highlighted below.

Tool integration limitations.

Each tool is developed to support a specific task, and there has been limited demand from the user community for tool vendors to enable information exchange across multi-disciplinary tools.

Lack of shared vocabulary.

Each discipline has its own vocabulary and associated concepts which is essential for each discipline. However, there is considerable information that must be shared across disciplines. Little attention is given to establishing a shared vocabulary to enable communications across disciplines.

Lack of architecture baseline.

The system architecture is often understood by the chief engineer on a project. However, architecture products are often not captured and maintained as part of the technical baseline. As a result, it is more difficult for the engineering disciplines to understand the impact of a change across the system.

Inflexible change process across the lifecycle.

It is important to maintain a level of control over design decisions beginning early in design that supports concept exploration and broad architectural design trade-offs. However, the change process and configuration control are often not introduced until much later in the design.

Limited design reuse.

Specification and design artifacts are often developed with little consideration for reuse. This is often due to lack of motivation for reuse by a particular project, the lack of guidance for how to design for reuse, and the lack of infrastructure to support reuse, such as libraries of reusable assets at the enterprise level.

5. LOOKING FORWARD WITH PLM/MBSE

Engineering is an approach to develop solutions in response to a need. This requires continuous innovation and design iteration to converge on a solution. It requires a highly complex and interactive process where the previous decision informs the next, and is distinct from a cookie-cutter process that produces the same output each time it is performed. The ability to facilitate innovation and change, and manage this change in a disciplined manner across a multi-disciplinary development team, is at the heart of an efficient and effective engineering process. This requires that **each member of the distributed team has access to the right data in the right form at the right time**. The need for collaborative engineering practices across globally distributed teams is emphasized in the INCOSE Systems Engineering Vision 2025 [2]. Integrating the engineering processes, tools, and data as indicated in Figure 1, is essential to achieve this collaboration.

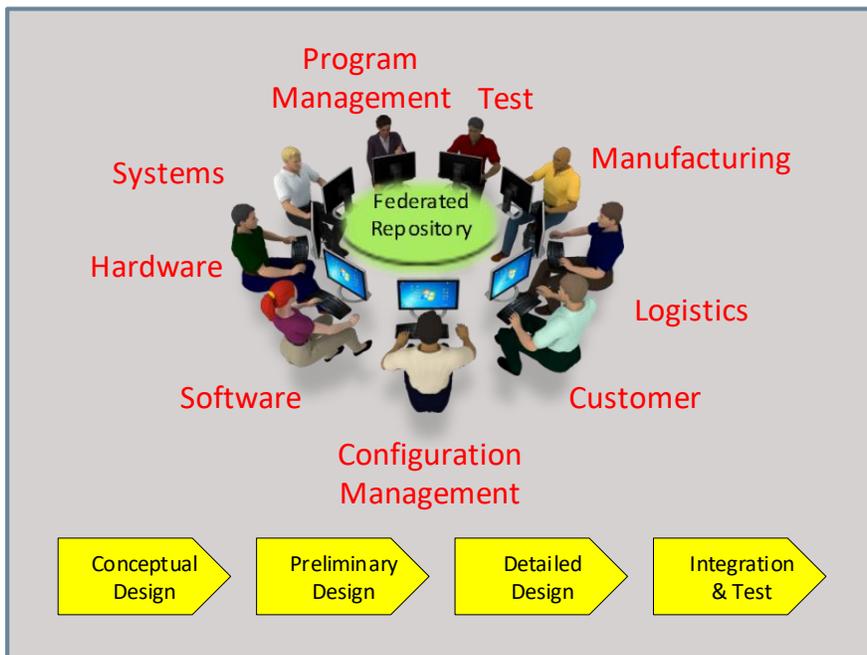


Figure 1. Integrating Process, Tools, and Data across the Lifecycle

Source: Image derived from NDIA MBE Final Report [3]

Although PLM and MBSE have evolved independently, they have reached a common cross-road where these two complementary approaches can enhance integration of the engineering effort to help address the product development challenges. PLM can manage the process and data across the lifecycle, and MBSE can facilitate integration of the multi-disciplinary engineering data. Over the last few years, there has been increased emphasis by both the PLM and MBSE community to leverage the combined approach of PLM and MBSE [4, 5, 6]. Some features of a combined PLM and MBSE approach that can enable this integration are summarized below.

Early start. The approach should be introduced at the beginning of any product development lifecycle and continue throughout.

Integrated workflow. The workflow should orchestrate across multi-disciplinary engineering processes that cross global boundaries so that each team member is made aware of the relevant changes to the technical baseline, and gets the right data at the right time.

Controlled data access. Protecting the engineering data as an organizational asset is more critical than ever in light of cyber threats, and must be managed as part of the organization information technology (IT) infrastructure strategy across the global supply chain.

Lifecycle configuration management. Managing the product configuration, including the use of metadata associated with versions, revisions, and variants, is at the heart of a PLM approach. This ensures the right system design configuration is made available to all team members so they work from a common technical baseline. In addition, the change history is maintained across the lifecycle as part of the digital thread to provide provenance of the design, and enable design decisions to be revisited when the need arises.

Adaptive change process. The workflow and configuration management should adapt to the lifecycle phase and needs of the project to enable more rapid change with lighter controls during earlier phases of a design, and more rigorous control of the technical baseline as the design matures.

Rapid and comprehensive change impact assessment. The approach should leverage the traceability provided by the system model to enable the more rapid and accurate assessment of the impact of change on requirements, design, simulation and analysis, test, and other parts of the technical baseline.

Technical data dependency management. Manage dependencies between data that is shared by different members of the engineering team using different models to ensure change coordination. At any given point in the lifecycle, the authoritative source of the data must be clearly identified and managed such that a change is driven from a single source. For example, the mass properties of the system may be used by many different engineering disciplines to support different analysis in different tools, but there should be a single source of this data that can be modified and propagated to other users of this data. The authoritative source of a particular type of data, such as the mass properties, can change over the lifecycle.

Lifecycle metrics tracking. Provide on-going lifecycle metrics related to engineering change, size, and complexity, and other key indicators of design maturity, that are inherently available from managing the continuous change process and traceability across the engineering design.

Strategic reuse for product family and variant design. Provide ready access to enterprise reusable assets to support product family design and design variants. The ability to leverage reusable assets across an organization requires high levels of engineering maturity. This includes the practices to design for reuse, and the organizational infrastructure to manage these assets and make them available to other projects. The system model can include logical abstractions to represent a range of products and other design elements that can be used to facilitate identification of candidate design configurations. The relevant metadata about the design elements can be used to evaluate applicability.

6. TOOL INTEGRATION

Although PLM and MBSE are two key elements that can significantly enable integration as described above, tool integration is a third major element needed to achieve the benefits of an integrated environment. PDM applications have traditionally integrated data at a file level, such as managing the relationship between a part number and document or CAD file. As indicated in Figure 2, a more granular level of data exchange between tools is needed to support change impact assessment and other design trade-offs and analysis. The figure highlights the need to maintain relationships between highly granular design information that includes requirements, architecture, simulation, geometry, test, and other data. The shared data across disciplines must be supported by a common data model. The application of MBSE produces a system model that contains shared data and their inter-relationships that reflect a common vocabulary with formal semantics for describing the system architecture. However, this data can only be shared among the disciplines if their tools enable the exchange of this data.

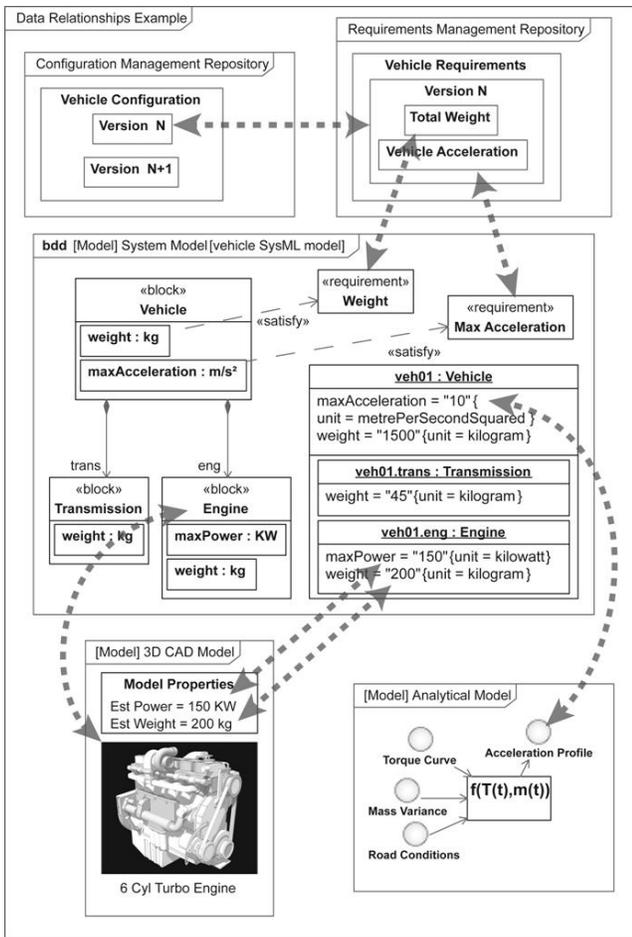


Figure 2. Data relationships between multi-disciplinary engineering models. Source: SAF Consulting

There often exists some localized integration between tools, such as between 3D CAD tools and thermal and structural analysis tools. The integration gap is between tools that span other disciplines such as systems, software, electrical, mechanical, reliability, safety, test, manufacturing, and others. These tools are often distributed and encompass a diverse range of specialized tools provided by many different tool vendors. The use of standards can enable tool integration and data exchange, but no single standard provides the total answer.

An organization should identify their critical tool and data integration needs and identify the relevant standards to address these needs. Over the last several years, several tool and data integration standards have emerged that can potentially be part of the solution. These include the evolving STEP standards, Open Services for Lifecycle Collaboration (OSLC), Functional Mockup Interface (FMI), and many of the modeling standards such as the Systems Modeling Language (SysML®), Unified Architecture Framework (UAF®), and others. In addition to selecting the right standards, the tool environment should be architected like any other complex system to achieve its objectives.

7. SUMMARY

Organizations are continuing to face ever increasing product development challenges and must develop strategies on how to deal with these challenges in order to stay competitive. The integration of processes, tools, and data is a necessary part of an organizational strategy to enable collaborative engineering and the associated product development efficiencies, effectiveness, and innovation. Both PLM and MBSE are approaches that have their own roots, but over the last few years, have become recognized as complementary approaches that can provide higher levels of integration.

The PLM and MBSE approach should be brought to bear early in the lifecycle and integrated across engineering disciplines. The goal is to ensure that each member of the distributed team has access to the right data in the right form at the right time. Beyond PLM and MBSE, the strategy must address how to integrate data and models across a diverse set of engineering tools to achieve these benefits and should leverage standards where practical.

Implementing an integration strategy is not done in one step, but should be done incrementally as part of the organization's improvement process. This enables the organization to leverage their existing environments, and make managed changes that are defined, piloted, and deployed. In addition, organizations can leverage the broader industry experience from professional associations, standards bodies, tool vendors, and academia to advance their capability.

REFERENCES

- [1] Forsberg Kevin, Mooz Harold. *Application of the “Vee” to Incremental and Evolutionary Development*. St. Louis: Proceedings of the Fifth Annual International Symposium of the National Council on Systems Engineering; July 1995
- [2] *Systems Engineering Vision 2025, World in Motion*, INCOSE Publication, July, 2014, available at: <http://www.incose.org/AboutSE/sevision>
- [3] *Final Report of the Model Based Engineering (MBE) Subcommittee*, NDIA Systems Engineering Division, M&S Committee, February 10, 2011, available at: <https://www.ndia.org/-/media/sites/ndia/meetings-and-events/3187-sullivan/divisions/systems-engineering/modeling-and-simulation/reports/model-based-engineering.ashx>
- [4] *Model Lifecycle Management for MBSE*, Amit Fisher, Sanford Friedenthal, Mark Sampson, Lonnie VanZandt, John Palmer, Mike Nolan, Michael Loeffler, Manas Bajaj, Krista Hovey, Laura Hart, Proceedings of the 24th Annual International Symposium of the International Council on Systems Engineering Conference Proceedings, available at: http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:model_lifecycle_management_for_mbse_v4.pdf
- [5] *10 theses about MBSE and PLM Challenges and Benefits of Model Based Engineering (MBE)*, German Chapter of INCOSE, PLM4MBSE Working Group, July 2015, available at: http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:plm4mbse_position_paper_v1_0_july_2015.pdf
- [6] [SysML v2 Requirements Support Document-2017-09-23 \(OMG Doc # syseng/2017-09-01\)](#), OMG SysML v2 Requirements Working Group, September 23, 2017

ACKNOWLEDGEMENTS

The author wishes to acknowledge ARAS for sponsoring the writing of this paper. However, this paper expresses the views of the author. The author also would like to thank Pawel Chadzynski from ARAS, Harald Eisenmann from Airbus, and Chris Paredis from Georgia Institute of Technology for their excellent insights which were considered in writing this paper.

AUTHOR BIO

Sanford Friedenthal is an industry leader in Model-based Systems Engineering (MBSE) and an independent consultant. In his previous position as a Lockheed Martin Fellow, he led the corporate engineering effort to enable Model-Based Systems Development (MBSD) and other advanced practices. In this capacity, he was responsible for developing and implementing strategies to institutionalize the practice of MBSD across the company, and providing direct MBSE support to multiple programs.

His experience includes the application of systems engineering throughout the system lifecycle from conceptual design through development and production on a broad range of systems. He has been a systems engineering department manager responsible for ensuring that systems engineering is implemented on programs.

He also was a leader of the industry team that developed SysML from its inception through its adoption by the OMG. Mr. Friedenthal also led the International Council on Systems Engineering (INCOSE) MBSE Initiative, and is co-author of 'A Practical Guide to SysML'.