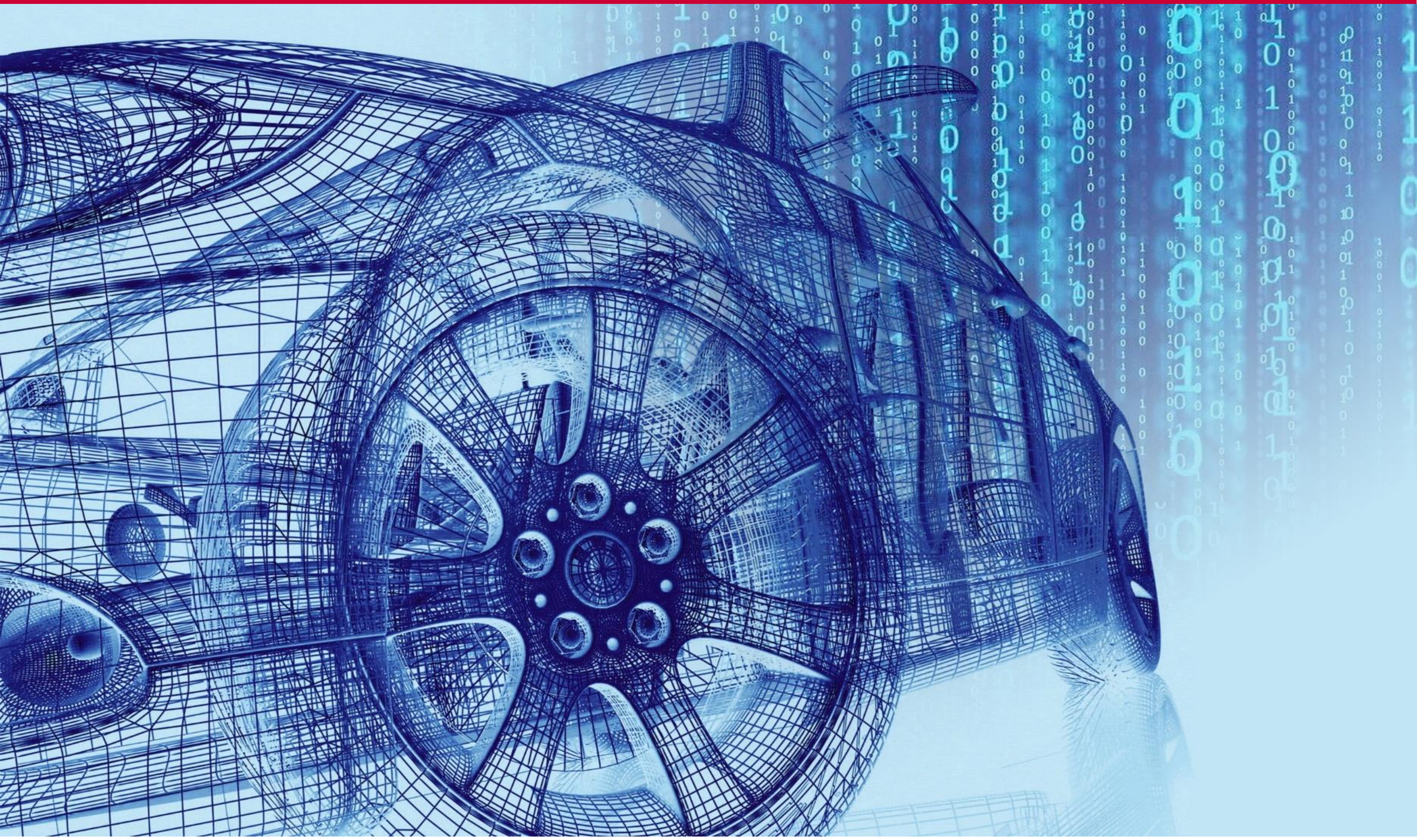


MBSE and the Business of Engineering

The Case for Integrating MBSE and PLM

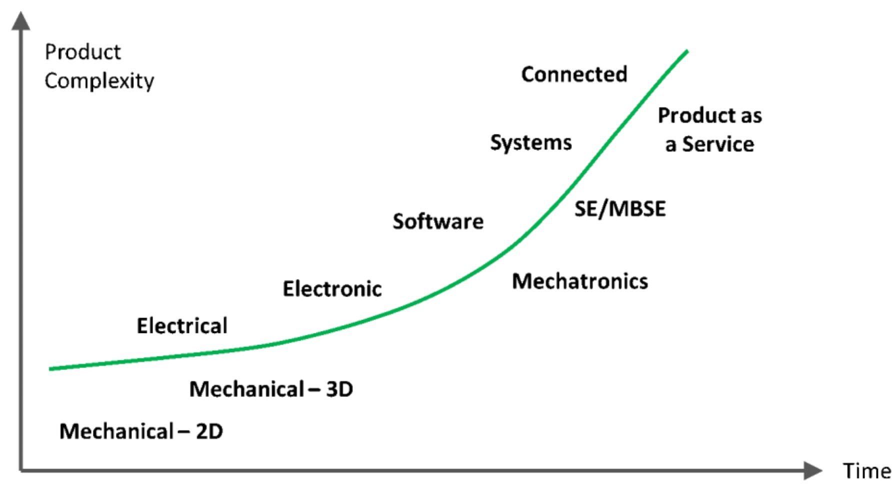
aras corp®

To succeed and win against sophisticated competitors, companies must excel at verifying their systems behavior and design.

EXECUTIVE SUMMARY

Manufacturers across all major industries are struggling to manage the ever-increasing complexity of modern products. They have become systems ─ even systems of systems ─ whose product designs require a mix of hardware, software, electronics and/or firmware.

Managing the system view of a product has traditionally been done with relatively simple tools, such as Excel, Visio, PowerPoint, etc. That simplicity disappears when managing the behavior of ‘systems of systems.’



To succeed and win against sophisticated competitors, companies must excel at verifying their systems behavior and design as products progress through definition, development and the complete post-manufacturing product lifecycle. Otherwise, product quality issues will emerge, putting brands, companies and their stakeholders at risk.

This Paper addresses the way in which robust system behavioral modeling can be integrated with downstream design practices to produce better, safer products. It is intended, through discussion of the latest advancement in system-level thinking, to help inform senior engineering professionals responsible for the cross-discipline lifecycle of their products.

INCREASING PRODUCT COMPLEXITY AND THE NEED FOR MBSE

Systems engineering (SE) emerged as a means to manage large-scale, complicated product development programs, such as the US Navy's ballistic missile submarine program in the 1950s, and later, NASA's Apollo Program. As product complexity increased over the years, SE practices became the norm in aerospace programs.

Today, SE practices are critical across a broad range of products, including automobiles and other smart, connected products. That's because product complexity continues to rise, driven by new product designs that require behavioral and physical integration of hardware, software, electronics, firmware and embedded software that can receive in-service upgrades.

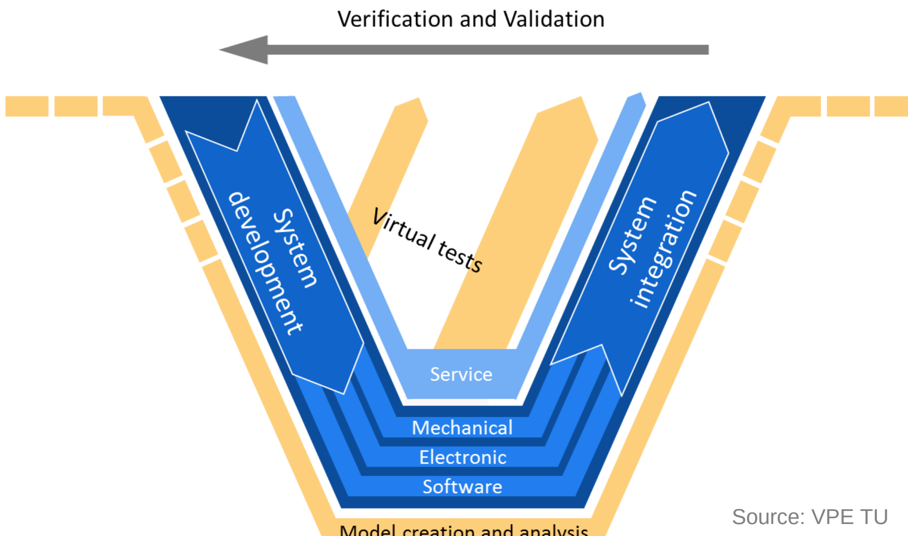
Consider three recent situations which could have been avoided through better systems design evaluation:

1. **Unintended Accelerations:** The electronic throttle control system (ETCS) in millions of cars caused extreme, and often deadly, acceleration due to faulty software. Additionally, black boxes recorded false 'driver action' data in ETCS-related fatal crash incidents.

2. **Tesla's Autonomous Vehicle Crash:** Following the fatal crash of a Tesla autonomous vehicle, the company stated its need to evaluate "the design and performance of automated driving systems in the Tesla Model S."

3. **Boeing 787s' Generator Shutdown Hazard:** 787s had the potential to entirely lose their electrical power due to a software error (likely integer overflow) in an electrical system responsible for power generation.

Each of these hazardous occurrences calls for refinement to the 'V' model that reflects discipline-based product development flowing from a system design. Under the updated 'V' model version, systems engineers would begin with the system architecture (functional, logical and behavioral). Then, engineers from each discipline would proceed to design their individual portion of the product based on the systems design.



Source: VPE TU Kaiserslautern

Just as senior product development professionals recognize the need to evolve the 'V' model, so too have SE practices evolved. Their evolution is meant to address the shortcomings of SE tools in meeting the complex behavioral challenges of today's product design world. From hand-drawn block diagrams, to "dumb" tools like PowerPoint and Visio, new tools and practices have emerged to enable Model Based Systems Engineering (MBSE) methodologies. MBSE authoring tools like No Magic Cameo and IBM Rhapsody, integrated with simulation tools bring about a system behavioral abstraction required for the design of complex, connected products.

Such behavioral system abstraction must permeate, and be traceable between, all design domains for manufacturers of intelligent, smart products to succeed.

MBSE methodologies enable engineers to approach development of a design from several viewpoints: Requirements (R), Functional (F), Logical (L) and Physical (P). MBSE enables more robust systems engineering, because it results in models and their associated behavioral abstraction which, together, document systems design intent in a more easily understood manner.

LIMITATIONS OF MBSE

Several challenges frequently prevent organizations from realizing the full advantage of MBSE capabilities.

From an organizational standpoint, MBSE tools and the resulting behavioral models could potentially create another "island of automation." An SE team that uses MBSE tools and practices improves productivity within the team but finds itself further isolated from the rest of the engineering organization and processes. This is due to complex MBSE tools being used by a small number of engineering specialists whose models are not easy to disseminate and not easily understood by the rest of the organization – preventing the intent of the systems engineers from being followed to its fullest.

In addition, there are no established procedures in the industry today for creating a traceable linkage between MBSE model elements and discipline-specific hardware and software authoring tools and their data models – making it difficult to support robust traceability, verification and validation.

Another practical issue is that of synchronizing and reconciling multiple sub-models of a large and complex system. Rarely is there sufficient time or necessity to model every part of a system, but this can lead to conflicts between different models that are hard to identify until detailed design is well underway.

Finally, MBSE is not subject to the standard change control process, because MBSE lacks integration into the overall design and configuration management process. Changes to a product's initial requirements or behavioral model should reveal the impact of such changes on the hardware and software designs. Conversely, a change to a physical design or software code should allow engineers across disciplines to trace relationship of that change to the corresponding elements of the MBSE behavioral model.

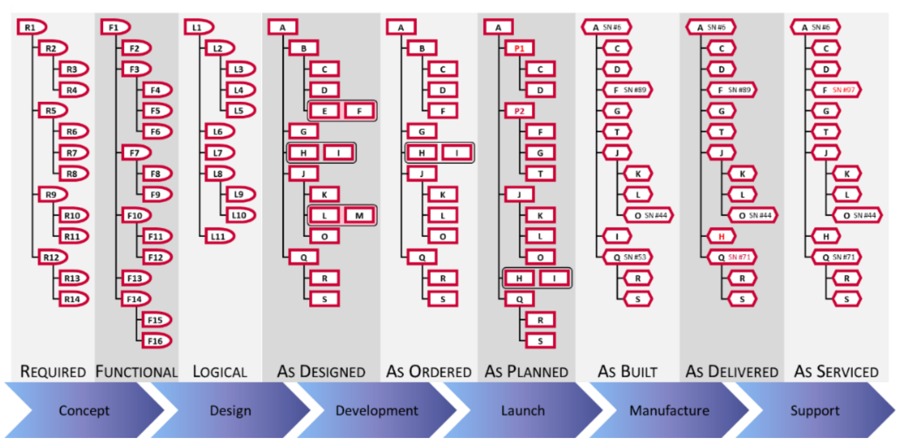
LEVERAGING USE OF MBSE THROUGHOUT THE PRODUCT LIFECYCLE

Product Lifecycle Management (PLM) has long been used to manage the development of hardware elements within complex products. A new breed of PLM has recently emerged ─ implemented as a platform and capable of managing across hardware and software disciplines as well as through the entire product lifecycle.

By integrating MBSE and PLM, product development organizations can overcome many of the current challenges of MBSE. MBSE-PLM integrations would allow them to:

- Communicate systems design intent and capture early feedback (by extracting MBSE outputs in easy-to-view formats and disseminating them to downstream, discipline-specific engineers)
- Create a contextual environment for multiple system sub-models, allowing conflicts to be identified earlier
- Create a complete digital thread that links requirements through design, manufacturing and service assets (via a traceable linkage from an MBSE model to discipline-specific hardware and software development outputs)
- Put MBSE models under formal revision and change control once detail design has started

MBSE-PLM integration will give organizations greater control over product design and development consistency. This is critical to producing complex products in a multinational environment, as they will be able to control access to MBSE models across the extended enterprise and geographic boundaries. These models can be linked to appropriate revisions of downstream assets in order to adhere to configuration rules, and the models can persist as long as organizations deem necessary.



REFERENCE ARCHITECTURE FOR MBSE-PLM INTEGRATION

A reference architecture for the MBSE-PLM integration discussed in this paper has been created by the combined efforts of several organizations: Technical University Kaiserslautern, XPLM Solution GMBH, No Magic, IBM and Aras Corporation. Implementations utilized No Magic's Cameo and IBM's Rhapsody for SysML authoring, and Aras for PLM management.

MBSE elements (i.e. functional blocks, logical blocks, diagrams, etc.) are instantiated in PLM and contain metadata that relates the elements to one another based on the original SysML definitions (SysML is a System Modeling Language utilized by many of the MBSE tools). As such, the architecture supports top-down design, parallel development and design reuse. Moreover, this approach makes the MBSE model subject to all PLM controls, such as configuration management, change control, versioning, workflow and more. An important aspect of PLM control, particularly in light of system failure catastrophes, is that system model changes cannot be propagated to PLM without active ECO (Engineering Change Order) authorization. This prevents design change errors that would impact a product design's current configuration.

THIS TYPE OF INTEGRATION FREES INDIVIDUAL SE TEAMS TO FOCUS ON SEPARATE ASPECTS OF AN OVERALL SYSTEM WITHOUT LOSING TRACK OF CROSS-DEPENDENCIES BETWEEN THESE MODELS.

Managing individual MBSE model structures in PLM encourages reuse of previously architected sub-systems by keeping them under strict configuration and release controls. Similarly, PLM management of the basic model structures allows systems engineers to integrate (or relate) individual models to each other as an overall large-scale system model. This type of integration frees individual SE teams to focus on separate aspects of an overall system without losing track of cross-dependencies between these models.

Integration of MBSE to PLM also allows requirements to be managed in a more robust manner. Requirements evolve like everything else, and thus, should be subject to formal change control. This means that requirements should be managed centrally in PLM, not in a domain-specific authoring tool like MBSE. Furthermore, centrally managed requirements encourage reuse and simplify uniformity of their content.

The PLM environment also provides visualization tools that enhance and streamline inter-team workflows. Visualizations better communicate design intent, give downstream engineers visibility of the systems model and provide them with an easier and instantaneous feedback mechanism. For example, by managing MBSE and physical product structures ("RFLP") within PLM, engineering teams gain automatic generation of SysML-like diagrams. These diagrams express model behavior in way that is easy to comprehend by the rest of the design community. As everything else in PLM, these types of diagrams can be tailored by task-specific needs and controlled by PLM access permissions. This relieves systems engineers from having to manage end-user deliverables while ensuring full synchronization of the diagrams with the desired product configuration context.

Finally, different organizations rely on different MBSE workflows and/or use cases. For example, space applications are typically unique per project, and therefore, always start with a new MBSE-driven definition of the system architecture prior to a physical implementation. The automotive industry, by contrast, typically starts with an existing architecture/platform already defined in PLM which they use as a "baseline" for the new platform. In this case, MBSE is used to evaluate the impact of changes rather than to model a completely new behavior. These different MBSE use cases demonstrate the importance of establishing a robust and tight integration between MBSE and PLM.

The benefits of MBSE-PLM integration can be summarized in three main areas:

- Traceability across all instantiations of "R", "F", "L, and "P" to provide a single source of product data, post-manufacture
- Design reuse and sharing of previously released sub-sections of a system
- Large-scale integration and traceability of the individual system models in an overall ‘system-of-systems’
- Accurate system model views (diagrams) directly from the appropriate configuration, including all design and maintenance stages. These are must-have views for performing change impact analysis and traceability.
- Forward and backward integration that streamlines product information flows across all stakeholder teams

CONCLUSION

Increasing complexity, particularly caused by software-driven products that can be updated in the field, is driving the need for through-life product configuration management. PLM platforms, when integrated with multiple discipline-based tools and other enterprise systems, can achieve through-life configuration management.

While MBSE methodologies are critical to management of today’s system complexities at the system architecture level, it is the integration of MBSE with PLM that allows the benefits of MBSE to be fully exploited throughout detail design and subsequent lifecycle phases. This creates a much more robust design environment ─ the key element to ensuring high-quality products.

APPENDIX – PUTTING THE PIECES TOGETHER

As the design industry and solution providers are grappling with the overall issue of digital thread (traceability) and digital twin (digital representation of a physical system) that spans system lifecycle states (from MBSE artifacts to a Serial Number of a unit in the field), two implementation approaches have emerged:

- Informal links between data elements of various authoring tools and data repositories (ex: OSLC technology)
- Formal relationships between various system abstractions, elements, and domains (ex: PLM platforms).

These two methodologies are often misunderstood as competing, when in fact they are complementary. In many MBSE implementations organizations tend to focus only on a selected domain and phase of the overall product development process. This narrow focus will result in inadvertently creating costly barriers and limitations that are challenging to overcome down the road. We have found the best practice is to allow the firm's unique business systems, processes and user stories to dictate the specific blend of the implementation options.

One way to understand the potential impact of a poorly blended implementation is by considering MBSE in the context of Configuration Management. When the firm is in the early stages of product development (ex: initial MBSE authoring), model changes tend to have minimal impact on the subsequent implementation phases. As development progresses into detailed design (e.g., in mechanical, electronic, software, etc.), changes in the MBSE model have an increasing large effect on the product's digital thread.

The impact of a change in the block diagram may drive changes in multiple design domains as well as other associated systems. Consider the difference between navigating informal associations vs. traceability of formal dependencies. Engineers depend on the ability to trace associations between various design elements to explore impact of change as well as design options. Effectively capturing and designing your MBSE-PLM integration for this use case will drive the blend of implementation methods for successful lifecycle management as well as multi-disciplinary design.

This puts the value of informal links (ex: OSLC) and formal relationships (ex: PLM platforms) in a proper context:

- Informal links provide engineers with the flexibility to incorporate disparate tools, data models, and data repositories. However, these links do not provide (or are not expected to provide) details of their context beyond that of a typical URL.
- Formal relationships, by contrast, provides a rich semantic context that provide engineers with enhanced traceability throughout product configuration – all the way down to a serial number. The traceability includes data such as change history, configuration history, variant configurations, lifecycle states, etc. PLM platforms also provide additional benefits of access control impossible to enforce via informal links (access, visualization, changes, checkout, etc.).

ABOUT ARAS

Aras® offers the best Product Lifecycle Management (PLM) software for global businesses with complex products and processes. Advanced PLM platform technology makes Aras more scalable, flexible and resilient for the world's largest organizations, and a full set of applications provide complete functionality for companies of all sizes.

By rethinking the way PLM is designed, Aras has taken a fundamentally different approach with a focus on the Business of Engineering. Aras solutions support processes for global product development, systems engineering, multi-site manufacturing, supply chain, quality and more.

Companies running Aras include Airbus, Boeing, Bombardier, GE, Hitachi, Honda, Kawasaki, Magna, Microsoft, Motorola, Nissan, TOSHIBA, Xerox, the US Army and hundreds of others worldwide.

aras corp®

300 Brickstone Square
Andover, MA 01810
USA
[978] 691-8900 Phone
[978] 794-9826 Fax
info@aras.com
www.aras.com