13th ANNUAL IEEE INTERNATIONAL SYSTEMS CONFERENCE Orlando, Florida, USA. April 8-11, 2019

Operational Needs of Model Centric Engineering

Note: Author unable to present paper so does not appear in proceedings, but is in IEEE Xplore at https://ieeexplore.ieee.org/document/8836783

Rick Dove
School of Systems & Enterprises
Stevens Institute of Technology*)*
Hoboken, NJ, USA
rdove@stevens.edu

Dr. Mark Blackburn
School of Systems & Enterprises
Stevens Institute of Technology
Hoboken, NJ, USA
mblackbu@stevens.edu

***Abstract*—Model Centric Engineering (MCE) platforms for mixed discipline systems engineering are the subject of various research and prototyping projects under various names, as well as commercially available in early stages of interoperable engineering tool platforms. Commercial platforms offer interoperability and data exchange with a variety of other engineering tools. Some have evolved from a vendor-offered SysML Tool base, others are vendor agnostic. Research prototypes are starting from a technical feature requirements base, exploring the potential and needs for MCE technology. None have started from the operational user needs, but rather focus on platform technical features. The operational users of most import aren't the end users – those that will use the platform in engineering activity; but rather those that must provide an MCE operational service to the end users that addresses all of their needs for acceptance and employment. This article explores why this is true and the range of operational service needs, and suggests basic requirements that should be addressed when developing or evolving MCE platforms.**

Keywords—MCE, system engineering, model based engineering

I. Introduction

"Model-centric engineering can be characterized as an overarching digital engineering approach that integrates different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity across disciplines throughout the life cycle. Industry is trending towards more integration of computational capabilities, models, software, hardware, platforms, and humans-in-the-loop." [1]

MCE in this article is concerned with a full life cycle systems engineering environment consisting of mixed discipline component engineering and systems engineering. Component engineering and systems engineering employ various analysis, modeling, and simulation tools. Systems engineering establishes the systems architecture, the component capability requirements, the interface protocols among interacting components, and the integrated test and evaluation criteria. Component engineers and systems engineers need to interoperate effectively to reduce rework caused by component and system interaction ignorance and misunderstandings. The tools involved with both systems engineering and component engineering need to interoperate effectively to reduce rework caused by conflicting models and data. Interoperation is therefore both an engineer-to-engineer and a tool-to-tool interaction. An interoperable engineer and tool environment provides a cohesive single source of truth.

As to full life cycle systems engineering (rather than development only): systems today are generally expected to provide sustained service over periods of time that span changes in their operational environment, necessitating changes in their operational capabilities; and changes in their engineering environment, necessitating a cohesive single source of truth independent of transient engineers and contractors.

A full operational environment for MCE will not spring to life overnight – it will not be embraced by engineers happy with what they know and what they do, unhappy with something new and disruptive to learn, unhappy with the need to think and know about system concepts beyond component capability that requires compromises. That's nothing new. What is new is a foreign engineering environment that has tighter linkage with system issues and more transparent work-in-process issues.

An MCE platform serves the systems engineering world much like an Integrated Development Environment (IDE), such as Eclipse [2], serves the software engineering world – but an MCE platform is much broader in concept, as it is a full life cycle engineering platform that integrates mixed discipline engineering and sustainment activity typically across a much larger community of engineers.

Software engineers wear two hats – they design and construct a product with an Eclipse-like environment that enables (enforces) object-oriented design of the software product, which facilitates agile development evolution and sustainment. Systems engineering doesn't have such a platform as yet. The MCE concept promises to provide that platform, allowing mixed-discipline engineers to design and construct (virtually with high fidelity models/simulations), and experience the results of full (and partial-in-process) systems integration. This is a breakthrough concept for systems engineering. Agile development for software relies on Eclipse-like IDEs – and enjoys incremental/iterative development/evolution facilitation. The MCE concept brings that to mixed-discipline engineering.

Software engineers employing an IDE are fairly self sufficient, able to connect tools of their own choosing to the core IDE infrastructure they are using, and allow other software engineers to access their constructed code. System

This work was funded by the Systems Engineering Research Center under contract HQ0034-13-D-0004.

and component engineers employing an MCE on large projects, such as defense contracts, don't have that same self-sufficiency – they are constrained (and enabled) by project and program standards for tools, methods, interoperability, and a coherent single source of truth employable in development as well as long-life sustainment.

The operational users of most import aren't the end users – those that will use the platform in engineering activity; but rather those that must provide an MCE operational service to the end users. This service sustains and evolves MCE tool interoperability, and addresses all end-user needs for effective acceptance and employment.

This article explores the operational service needs, and suggests basic requirements that should be addressed when developing or evolving MCE platforms.

II. Literature Review

This article presents research work into operational/service needs for an engaging mixed discipline MCE platform, and presents relevant operational issues and objectives obtained from a literature review.

A. Sandia National Labs – Common Engineering Environment

The Common Engineering Environment (CEE) at Sandia National Laboratories in Albuquerque, New Mexico, is focused on simulation and analysis tools for engineering other than common Model Based Systems Engineering tools. Knowledge of the operation was limited to publicly available materials.

The following excerpts are quoted from [3], after 2+ years of CEE existence:

The Common Engineering Environment (CEE) is a set of Preferred Engineering Software, Infrastructure, and Support Services that enable a Highly Functional Engineering Environment with seamless and ubiquitous access to Computational Engineering Capabilities.

CEE Goals:

- Facilitate the identification and use of a common, preferred set of tools and infrastructure for computational engineering.
- Provide consistent and easy-to-use methods for customers to access, and use software tools.
- Provide consistent support for engineering tools and infrastructure in the preferred set.

Later in [4] there is a more detailed view:

"Sandia's vision for the CEE is a set of preferred engineering best practices, processes, training, tools and software, support services, and shared architecture that enables an integrated engineering environment with easy access to tools and capabilities, lab-wide efficiencies, and a disciplined approach to engineering."

Fig. 1 and Fig. 2 depict the tool support infrastructure and the service infrastructure.

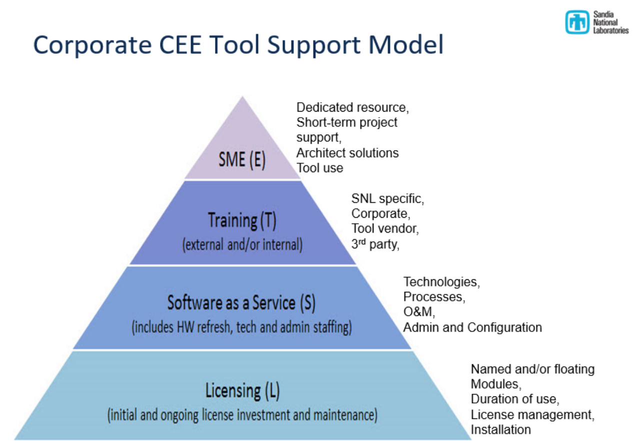

Fig. 1. Here we see the emphasis on a tool support infrastructure for engineers that will use tools licensed by CEE.

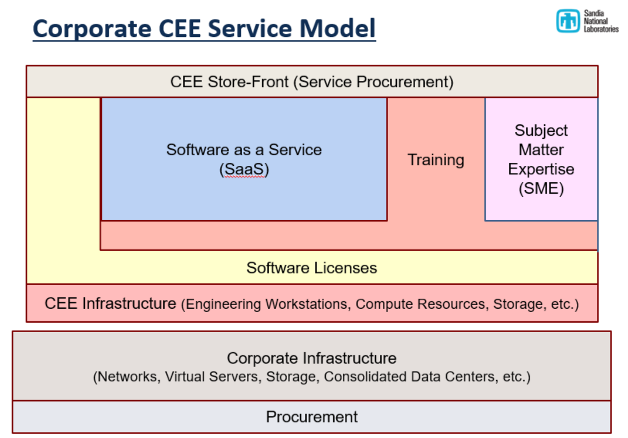

Fig. 2. Here we see the emphasis on a service infrastructure that provisions and provides tool capabilities for engineers

Relevant take away: the need for an organizational support and service capability to enable engineers to focus on their development efforts rather than tool acquisition, provisioning, tool access, training acquisition, and administrative efforts. The MCE capability cannot be supplied as an off-the-shelf product from a single-source product supplier, as it necessarily requires access to and integration with tools that may be project, program, or organizational specific. This implies either a need for an internal MCE service group or an equivalent contracted service capability.

B. NASA/JPL:

The issues of transformation and infusion have bearing on the operational requirements as full-scale cutover to an MCE-like environment is neither recommended in the literature nor practical. Many sources were found and read for potential influence on this research; but generally little had direct and relevant bearing on establishing requirements for an operational environment beyond those that follow.

1) Integrated Model Centric Engineering (IMCE)

An Integrated Model Centric Engineering (IMCE) Concept of Operations document [5] focused on areas of "largest beneficial impact on project development and operations, as well as those that are needed for early infrastructure development." The document was intended for two distinct audiences: The IMCE Architecting team, and potential adopters, users and funders of IMCE capabilities. Goals and objectives for the ConOps were articulated in the document as follows:

Goal 1: The IMCE Team understands and is able to articulate to project users why project users should, and how they can, use IMCE to accomplish a model-centric project development.

Objective 1A: Collect, synthesize and document significant problem areas in the current flight project development environment, and how IMCE addresses them.

Objective 1B: Describe, through development of selected scenarios, how people supporting a JPL flight project would interact with models, model-centric tools, and each other throughout the project life cycle.

Objective 1C: Discover and document the characteristics of a flight project lifecycle as executed in a model-centric environment.

Goal 2: The IMCE Team has valid use cases upon which to base the IMCE Architecture.

Objective 2A: Deliver the Concept of Operations with form and content appropriate for the IMCE Architecting work and the Architecture Description Document.

Relevant takeaway: JPL architected, modeled, and developed a Concept of Operations for their product development "enterprise" as a precursor for development of an IMCE capability. This helped them identify operational requirements before designing a solution. Notably, they collected problem areas in the before-IMCE development approach and showed how IMCE would address them (internal document) – which amounts to an evidence-based value proposition for users.

2) Early Formulation Model Centric Engineering

A subsequent paper [6] provided 12 lessons learned in the transformation to IMCE that inform our interest in the nature of the operational environment. Selected relevant excerpts from 8 of the 12 lessons-learned follow:

- **Unity of Leadership is Essential.** In the first infusions, management support for the effort must be clear and consistent (verbatim from the paper).
- **Early Efforts Draw on a Limited Pool of Talent.** Similarly to the "Unity of Leadership" above, the first infusions will not have the benefit of an engineering pool with ubiquitous modeling skills (verbatim from the paper).
- **Leverage Learning with Synergistic Work.** With a limited pool of modeling talent and three projects in need of modeling experts they allowed the experts to participate in all three simultaneously, which they believed outweighed the lack of full time commitment on a single project. "We have found this belief to be fully validated: the learning that has been shared between the three efforts has been enormously beneficial for all, and has clearly accelerated the institutional infusion.
- **Team Organization Matters.** While we have found that descriptive modeling can be done by almost anyone with the basic training, the additional rigor and consistency needed for quantitative analysis requires us to designate a smaller team of people who are modeling experts and who can apply best practice to the official configuration managed project models. Presently we have a core modeling team of a half dozen or so, within a larger team of 20 or so engineers. The experienced systems engineers provided guidance to keep the modeling focused on providing useful information, as well as mentoring of the core modelers who tended to be more junior. Frequent (daily) interactions were crucial to getting useful products (verbatim from the paper).
- **Everyone Needs Training, but to Different Levels.** In the usage model above it is clear that all three groups need to receive training commensurate with their level of interaction with the models. Different levels of modeling familiarity are required, thus resulting in different levels of training. Working with IMCE, we have constructed a set of classes that addresses all three user-type groups (verbatim from the paper).
- **Models Evolve.** The model needed in concept formulation is very different than the model needed in detailed design, or in operations. Models need to evolve and grow, and sometimes shrink. This should be the focus of model reuse along the project lifecycle. … It is clear that the more a model can be a self-contained, internally self-consistent, and an intuitive description of the concept, the more informative it will be. Moreover, the more the analysis can be separated from the model, the more reusable it will be (verbatim from the paper).
- **First Description, Then Analysis.** The more the analysis can be separated from the model, the more reusable it will be. For our mass analysis we have achieved a high degree of separation of the model from the analysis, and as a result we are able to run exactly the same mass analysis script on all three of our mission option models (verbatim from the paper).
- **Real Examples are Powerful.** Trying to describe to stakeholders and potential collaborators what MBSE looks and feels like has proven to be rather difficult and not very effective. We have found that many people 'get it' for the first time only when they see an actual example (verbatim from the paper).

Relevant takeaway: leadership commitment is crucial; a core support team is crucial, one that both assists projects and trains project personnel; and model reuse is facilitated by modeling methods that separate analysis from models.

3) IMCE Infusion at JPL

A later presentation on IMCE infusion at JPL [7] addresses enablers, barriers, and challenges experienced at JPL. Selected portions of relevance follow:

Enablers: innovative engineers, long development cycles, experience with challenges of project knowledge management, organizational combination of systems engineering and software engineering, strong management support.

Infusion Barriers: conservative engineering philosophy due to unforgiving nature of the environments in which we operate, maturity of the integrated tools, big learning curve for experienced practitioners.

Addressing Challenges: work through the line organization to take ownership, embed MBSE-enabled engineers in projects as infusion agents.

Institutional Capability in 2014: exposed a cadre of engineers (200+) to concept of system modeling, with 50+ in practice, deployed a set of system engineering modeling standards and a lexicon to be used for developing and structuring system models, consultation services for 20+ projects applying IMCE developed capabilities, model repository (Teamserver) deployed and in use by multiple tasks and projects, capture re-usable modeling design patterns.

Relevant takeaway: enterprise transformation to a model-based engineering operational environment is necessarily a phased approach that incrementally learns from experience and experimentation; engineers will learn, value, and adopt system modeling approaches (if they understand a realistic beneficial value proposition); creation and utilization of local modeling standards is crucial; a core and competent support team for project consulting is necessary.

C. Boeing

A recounting of the MBSE infusion process at Boeing [8] summarized MBSE Process Needs, and MBSE Tool Needs, selectively excerpted as follows:

Process Needs: success stories to help promote the benefits, methods to measure impact of MBSE, training to develop good modelers, methods to ensure persistence of MBSE after the advocates move on.

Tool Needs: lower user entry barriers – more intuitive user interfaces, support for hundreds of globally distributed users, support for data reference libraries and data reuse.

Relevant takeaway: a support group that promotes benefits, reinforces values with evidence of usefulness, and provides training; tools with low entry barriers; reusable data/model libraries; and scalability for large user and data bases.

D. Thales Group

From the abstract of [9]: "This article presents a case study on applying model-based systems engineering (MBSE) methodologies under real-life conditions. We present how engineers tailored existing MBSE methods and tools to both address the complexity factors of nuclear power plants engineering, and to contribute to the comprehensiveness of the design and safety assessment. We also provide feedback on the application of MBSE approaches and their key benefits on projects' execution."

From the summary of findings and conclusion, verbatim:

Findings:

- A better communication and definition of responsibilities scopes between stakeholders: technical exchanges with detailed design teams, transversal disciplines such as safety or human factors, and other systems' architecture teams, were more productive when supported by common and normalized graphical representations provided by Arcadia/Capella.
- A unique source of information on systems architecture: Models encapsulate the key information and become the reference database for architecture-related topics, easing information capitalization for example, the extract of Interface Control Document tables.
- A fast learning curve: the Arcadia/Capella concepts and diagrams are rather well adapted to the nuclear engineering population mostly composed of engineers who [have] not been exposed to modelling approaches such as UML or SysML. The strong coupling between the method and the tool, and the availability of multiple productivity tools, are of great value for engineers.

Conclusion: This article showed how the application of MBSE methods and tools may have a positive impact on nuclear power plants' engineering task, and particularly during architectural definition and design. We witnessed benefits both in supporting the technical production, (by contributing to the exhaustiveness of design, safety justifications, and third-party assessments) and in the daily interactions between engineering teams (by providing a common and normalized graphical representation, and by introducing concepts that make teams work in a more collaborative and agile way). To guarantee these benefits, future users shall conjointly perform a tailoring of MBSE concepts and models to cope with their discipline and project-specific constraints.

Relevant takeaway: Mixed-discipline engineering teams embraced values from graphical architecture models, inter-team communications, and mixed daily collaboration meetings; and mixed-discipline engagement was facilitated by tools with fast learning curves.

E. Multiple USA Federal Agencies

A USA OSD-SE report on Digital Model-based Engineering [10] offers challenges and recommended mitigation from a US inter-agency task team (DoD, DHS, VA, FAA, and NASA).

The team identified challenges an organization might encounter when looking to infuse DMbE:

- Assessing value added to the organization. Not all DMbE practices will be applicable to every situation in every organization; and not all implementations will have positive results.
- Overcoming organizational and cultural hurdles.
- Adopting contractual practices and technical data management.
- Redefining configuration management. The DMbE environment changes the range of configuration information to be managed to include performance and design models, database objects, as well as more traditional book-form objects and formats.
- Developing IT infrastructure. Approaches to implementing critical, enabling IT infrastructure capabilities must be flexible, reconfigurable, and updatable.
- Ensuring security of the single source of truth.
- Potential over-reliance on quantitative data over qualitative data. Executable/computational models and simulations generally incorporate and generate quantitative vice qualitative data.

"Prerequisites for the infusion of DMbE include management support/advocacy, technical capability readiness, and organizational/cultural willingness (or lack of resistance) to adopt a new methodology. Some level of management support is essential, and having a management champion or advocate is better still. This support may be gained through education and exposure to examples and benefits of DMbE. Encouraging and facilitating organizational and cultural change is often a challenge. Education, training, and access to the necessary tools, applications, and aids can be helpful. In general, lowering barriers to adoption and implementation is necessary. Helpful in all these cases is a clear statement or vision of a future state of the use of DMbE and of the approach or roadmap aligned with the vision going forward to identify avenues of infusion into normal business activities.

"Organizations interested in infusing DMbE must recognize and identify the need and must be willing to make the necessary changes in established processes, tools and methods, and workforce. This results in a multifaceted approach that begins with the recognition that a change will have a positive outcome in the resultant capability, the staff makeup, and/or the speed of execution, with the expectation of higher precision and discovery of defects early on in a project's lifecycle. In other words, transition is a process, not an event.

"It is understandable that the multifaceted approach will have to be planned and will not occur instantaneously. A willingness to change is accompanied by planning for the transition, identification of the stakeholder population among the adopters, and the understanding of what a successful change looks like.

Relevant takeaway: necessary prerequisites to infusion include a clear vision of the future state, management support/advocacy, education and training, and lowering barriers to adoption and implementation is necessary. Importantly, a compelling need that can be appreciated by users must be identified.

F. SERC and Related MIT Research

A workshop was held in 2015 [11] at the Massachusetts Institute of Technology in support of the Systems Engineering Research Center (SERC) IMCSE program to investigate four interests (1) imagine an ideal world; (2) current state practice; (3) need for research; (4) emerging research; and (5) recommendations for gathering knowledge.

One participant offered this summary level statement as to the ideally envisioned experience of the individual: "An intuitive experience that generates deep insights across the area of relevant decisions that balances time, resources and the desired confidence in the decision outcome."

Key themes of relevance emerged in the topic "Imagine an Ideal World," with verbatim selections from the report as follows:

- Ease of Interaction. The individual interacting with a model will find it intuitive and the effort involved will be commensurate with the value the model provides. Novice users will be able to rapidly learn and benefit from use of modeling environments.
- Enabling Human-Human Interaction. Model-centric environments will support collaborative decision making and design with near real-time human to human interaction.
- Guided Interaction. Interactions with models will provide guided assistance for viewing models from standpoint of other stakeholders. … There will be assisted capabilities and wizards for model library curation, model composability, model interrogation, and stakeholder role playing.
- Model Re-Usability. The environment will be adaptable for the culture of the organization to enable effective reuse with confidence in the model and its appropriateness for the situation. Finding suitable models and reusing them in the individual's unique model-based environment will be easily accomplished. … Effective digital curation will enable preserving, discovering and reusing appropriate models.

Relevant takeaway: aptly summarized in the four bullet points above.

Many additional papers and presentations related to the SERC IMCSE project by Donna Rhodes, most with coauthors, were reviewed for relevance. Generally these go into research detail of the four key themes above, without additional relevance for this literature review.

A non-SERC paper by Donna Rhodes [12] does offer some new points of relevance. From that paper, verbatim: "A common approach to undertaking a model-centric program is to send all the engineers through tool training. Younger engineers may quickly learn to use modeling tools, but lack the experiential knowledge of engineering of products and systems. The members with years of experience, on the other hand, may find use of modeling software tools to be non-intuitive to the point where they spend more time on tool mechanics and less on decision making. Any discomfort and distrust of models and modeling toolsets can have negative impacts on the engineering effort."

Relevant takeaway: reinforcement of thought expressed frequently in the other reviews that tool ease of learning and use is especial crucial for older engineers with cultural legacy.

III. Tool Vendor Interview

Research than moved on to investigating the current plans of tool suppliers for interoperable modeling platforms. Mathew Hause, of PTC, agreed to spend three days in February, 2018 on this discussion. Hause has a recognized and deserved non-biased, non-selling, reputation for educating the community at large on MBSE and related standards (he sits on a number of MBSE-related standards committees). Hause was quick to point out that a lot of what PTC is doing for tool-interoperability is also being done with Dassault's Cameo as well.

Hause made it very clear that tool vendors are not interested in integrated-tool platform environments, but rather have high interest in tool and data interoperability.

Standards work on Open Services for Life Cycle Collaboration (OSLC) provides interoperability linkage and methods used by many tool vendors at this point. An excellent overview is available at (Szarazi 2014).

Relevant takeaway: Multi-tool interoperability is beginning to mature, principally due to standards work on Open Services for Life Cycle Collaboration (OSLC). Whether or not a proprietary core from a tool vendor is best for an organization, rather than a generic, vendor independent open source approach is a decision that warrants consideration. Reference [9] discusses usage and benefits of employing the open source Capella [13] systems engineering modeling platform and its companion Arcadia methodology.

IV. MCE Service Environment

Software engineering developers typically use an Integrated Development Environment (IDE), such as Eclipse. Eclipse is an open source platform under configuration control by individual developers that can interoperate with a variety of other tools of an individual developer's choice through Eclipse plug ins. A software developer's work combines both design and fabrication (code deployment) activities Large projects with multiple developers working on different parts of a software system produce code that can interoperate with code developed by other developers. Little interoperation of software component design occurs.

An MCE platform is quite different, with quite different needs. Fundamentally an MCE platform system will be a collection of interoperable design and simulation tools configured specifically by a service group for a program or project. A local service organization is required to enable and facilitate project wide usage, and to evolve capability and features needed as a project evolves.

The MCE service organization is the primary, front end, user of the MCE system. Their job is to enable and facilitate usage by the engineering users. How well they can accomplish that task will be determined and constrained by the design of the MCE system.

Considerations for MCE operational service support needs have been itemized in the "relevant take away" comments made in the preceding section.

V. Discussion and Opinion

Basic early versions of MCE platforms are available from commercial suppliers such as PTC and Dassault (with the acquisition of No Magic). More advanced MCE platforms are the work-in-process of various research and feature-prototyping projects, such as [1]; partly to overcome vendor-centric issues and limitations, but mainly to innovate new capabilities.

Research and prototyping work is generally focused on functional features: interoperability infrastructure, interoperability with new tools and applications, single-source-of-truth, knowledge of conflicts among engineering activities in process, trade-off and decision making support, and stakeholder-relevant communications and progress status.

To date, little has been focused on the operational requirements. Operational requirements include MCE facilitation of configuration engineering, sustainment, evolution, and stakeholder/user engagement. This research focused on user engagement requirements. Requirements identified in this summary are at the capability level rather than the feature level, with the intent to actionably inform a variety of MCE interests including researchers, feature prototype developers, MCE platform developers and suppliers, and tool-acquisition decision makers.

An organization, program, or project can mandate MCE usage based on logical expectations of benefit to the collective whole. That will likely meet strong resistance in some circles and un-involved compliance-marching in others. It is necessary to have MCE usage deliver perceived benefits above cost to all parties of the desired interaction. Current focus is on enabling interaction among tools. Additional focus needs to be on enabling and facilitating human interaction.

A. Value Proposition

An MCE platform facilitates the development and evolution throughout the life cycle of a digital twin. It provides an interoperable model for systems engineers. It provides an integration test and evaluation environment for component engineers. It provides a training environment for operations personnel. It provides a source of current knowledge and state-of-product for maintenance personnel. It lives throughout the product life cycle. At core, it supports planning, design, and tradeoff decisions with a single source of truth. It's principle values are in the reduction of rework in all life cycle stages and in clear stakeholder communications.

Core values of an MCE platform appear to be in rework reduction, stakeholder communications, life cycle support from a digital twin, and both target system (product) and operational (process) agility. Among other things, these core values elaborate to decision and trade-off support, and single source of truth throughout the life cycle.

Rework reduction should appeal to the customer, the systems engineers, and the multi-discipline engineers; whereas couching the value proposition in high-level organizational systems engineering benefit is likely heard as something completely different in traditional engineering instead of a naturally welcome productivity enhancement easily assimilated as an augmentation to current practice.

REFERENCES

[1] M. Blackburn, "Transforming systems engineering through model-centric engineering," Technical Report SERC-2017-TR-110, Systems Engineering Research Center, August 8, 2018.

[2] Wikipedia, "Eclipse (software)," accessed October 9, 2018. https://en.wikipedia.org/wiki/Eclipse_(software)

[3] R. Goebel and C. Pavlakos, "A common engineering environment (CEE) implementation," Sandia National Laboratories, Report SAND2012-4154 C. 2012. https://osti.gov/scitech/servlets/purl/1117233&usg=AOvVaw0aozgFaWsHvNf1T0tVxsDO

[4] C. Pavlakos, "Common engineering environment (CEE)," Sandia National Labs. November 2014. Presentation made to and available from Mark Blackburn, mblackbu@stevens.edu.

[5] T. J. Bayer, M. Bennett, C. L. Delp, D. Dvorak, J. S. Jenkins, and S. Mandutianu, "Update – concept of operations for integrated model-centric engineering at JPL," IEEE 2011 Aerospace Conference, Big Sky, MT, March 5-12, 2011. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.667.4638&rep=rep1&type=pdf

[6] T. J. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, D. Wagner, "Early formulation model-centric engineering on NASA's Europa mission concept study," INCOSE International Symposium, Rome, Italy, July 9-12, 2012. https://trs.jpl.nasa.gov/bitstream/handle/2014/42873/12-1191_A1b.pdf

[7] C. Y. Lin, "Integrated model-centric engineering: Infusing MBSE at JPL," INCOSE International Workshop, MBSE Workshop, Torrance, CA, January 25-26, 2014. www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2014:mbse_infusion-jpl.pptx

[8] B. Sheeley, R. Malone, J. Palmer, and R. Carson. "MBSE implementation across diverse domains at The Boeing Company," Presentation made to INCOSE MBSE Working Group, January 25-26, 2014, www.incose.org/docs/default-source/enchantment/180214carsonron-mbseimplementationacrossdiversedomainsattheboeingcompany-slides.pdf?sfvrsn=2

[9] J. Navas, P. Tannery, S. Bonnet, and J. Voirin, "Bridging the gap between model-based systems engineering methodologies and their effective practice – a case study on nuclear power plants systems engineering," INCOSE INSIGHT 21(1):17-20, March 2018. https://onlinelibrary.wiley.com/doi/epdf/10.1002/inst.12185, or with INCOSE membership: https://connect.incose.org/Library/InsightMagazine/

[10] MBSE Infusion Task Team, "Digital model-based engineering: expectations, prerequisites, and challenges of infusion," Interagency Working Group on Engineering Complex Systems (IAWG), Office of the Deputy Assistant Secretary of Defense, Systems Engineering. March, 2017. https://www.acq.osd.mil/se/docs/DMbE-20170524-0602.pdf

[11] D. H. Rhodes and A. M. Ross, "Interactive model-centric systems engineering (IMCSE) pathfinder workshop report," Systems Engineering Research Center, February 28, 2015. www.dtic.mil/get-tr-doc/pdf?AD=ADA626620

[12] D. H. Rhodes, "Using human-model interaction heuristics to enable model-centric enterprise transformation," IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, April 23-26, 2018.

[13] Polar Systems, "Open source solution for model-based systems engineering," http://www.polarsys.org/capella/