PROJECT PERFORMANCE INTERNATIONAL

CERTIFICATION TRAINING INTERNATIONAL

A PPI Company

SYSTEMS ENGINEERING NEWSLETTER

PPI SyEN 77 – 22 May 2019

brought to you by

Project Performance International (PPI)

systems engineering training for project success

PPI SyEN is an independent free newsletter containing informative reading for the technical project professional, with scores of news and other items summarizing developments in the field, including related industry, month by month. This newsletter and a newsletter archive are also available at www.ppi-int.com.

Systems engineering can be thought of as the problem-independent, solution technology-independent, life-cycle-oriented principles and methods, based on systems thinking, for defining, performing, and controlling the engineering effort within a technical project. The approach aims to maximize the benefit delivered to the enterprise, as influenced by the needs and values of the other applicable stakeholders.

If you are presently receiving this newsletter from an associate, you may wish to receive the newsletter directly in future by signing up for this free service of PPI, using the form at www.ppi-int.com. If you do not wish to receive future Systems Engineering Newsletters, please unsubscribe by clicking on the link at the bottom of this email.

We hope that you find this newsletter to be informative and useful. Please tell us what you think. Email us at syen@ppi-int.info.

The views expressed in externally authored articles are those of the author(s), and not necessarily those of PPI or its professional staff.

IN THIS EDITION

Read More…

6. News on Software Tools Supporting Systems Engineering

6.1 REUSE launches Traceability Studio

6.2 Phoenix Integration's ModelCenter MBSE

Read More…

7. Systems Engineering Publications

7.1 Engineering Systems: Meeting Human Needs in a Complex Technological World

7.2 Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach

7.3 Netcentric System of Systems Engineering with DEVS Unified Process

7.4 Introduction to the Theory of Complex Systems

Read More…

8. Education and Academia

8.1 University of Technology Sydney Center for Autonomous Systems

8.2 New Jersey Institute of Technology Newark, New Jersey USA

Read More…

9. Some Systems Engineering-Relevant Websites

Read More…

10. Standards and Guides

10.1 Peak Operational Concept Description (OCD) Standard Updated

10.2 Object Management Group Chairs Build Momentum for Technology Standards

Read More…

11. Some Definitions to Close On

Read More…

12. Conferences and Meetings

12.1 The featured event for this edition is: International Conference on Industrial, Production & Systems Engineering (ICIPSE)

Read More…

13. PPI and CTI News

Read More…

14. PPI and CTI Events

Read More…

15. Upcoming PPI Participation in Professional Conferences

Read More…

1. QUOTATIONS TO OPEN ON

"Systems Engineering is not a process. It is a set of principles, based on systems thinking, together with a set of process building blocks with which to implement those principles."

Robert John Halligan

"There are only two ways to live. One is as though nothing is a miracle. The other is as though everything is."

Albert Einstein

"Collaboration is a voyage from the known to the unknown with people of common commitment who both steer and follow."

John Wooden

2. FEATURE ARTICLE

2.1 Model-based Systems Engineering De-Mystified

by

Warren K. Vaneman

Naval Postgraduate School

Email: wvaneman@nps.edu

Abstract

Model-Based Systems Engineering (MBSE) is a mysterious concept that means many different things to different stakeholders. MBSE was envisioned to manage the increasing complexity within systems, by replacing traditional document-based system engineering with a model-based approach. However, more than a decade after MBSE was introduced many systems engineering efforts still default to a "document-like view" rather than integrated, "virtual," representation of the system. This paper discusses a revised definition for MBSE that supports system design and analysis, throughout all phases of the system lifecycle, and through the collection of modeling languages, model-based processes, structures, and

presentation frameworks used to support the discipline of systems engineering in a model-based or model-driven context.

I. Introduction

Complexity has been identified as one of the most critical problems in systems engineering. Advances in technology have led to larger, more complex systems, with an unprecedented level of functionality. As a result, there is a need for a clear, concise, way to express system designs clearly, and logically consistent, with the ability to express the emergent behavior of the system.

Model-Based Systems Engineering (MBSE) was conceived by the International Council on Systems Engineering (INCOSE) to address the increasing complexity of systems by transforming systems engineering from a document-based to model-based discipline. INCOSE (2007) defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout the later lifecycle phases." This definition captures the lifecycle perspective of systems engineering, but does not suggest how MBSE is different from traditional systems engineering, nor does it address the elements required from implementation. As a result, MBSE is often thought to be synonymous with the Systems Modeling Language (SysM), the Unified Modeling Language (UML), or various MBSE tools. Unfortunately, this thinking has hindered the systems engineering discipline from progressing to a true model-based discipline.

II. The Essence of MBSE

The objective of systems engineering is to facilitate a process that consistently leads to the development of successful systems (Long and Scott 2011). Model-Based Systems Engineering (MBSE) was envisioned to transform the reliance of traditional document-based work products to an engineering environment based on models. One can argue that systems engineering has always used models (i.e. diagrams, documents, matrices, tables, etc.) to represent systems. In these traditional document-based models, the system's entities were represented multiple times, making it difficult, if not impossible, to view the system holistically. The transformation to MBSE means more than using model-based tools and processes to create document-based models, but shifts the focus to a virtual system model of the system, where there exists a singular definition for any system element (Vaneman and Carlson 2019).

To illustrate the concept of a virtual model of a system, consider the dimensions of a systems engineering project (Figure 1) (Larson, et al. 2013, Vaneman and Carlson 2019), where the cube represents a system. The system has height, width, and depth. System height provides a decomposition from the highest system level down to components and parts. System width defines the lifecycle of the system, and provides insight across the entire system lifecycle from concept definition to disposal. System depth provides the complex relationships between systems, functions, requirements, etc.

This paper defines MBSE as the formalized application of modeling (static and dynamic) to support system design and analysis, throughout all phases of the system lifecycle, through the collection of modeling languages, structures, model-based processes, and presentation frameworks used to support the discipline of systems engineering in a model-based or model-driven context (Vaneman 2016).

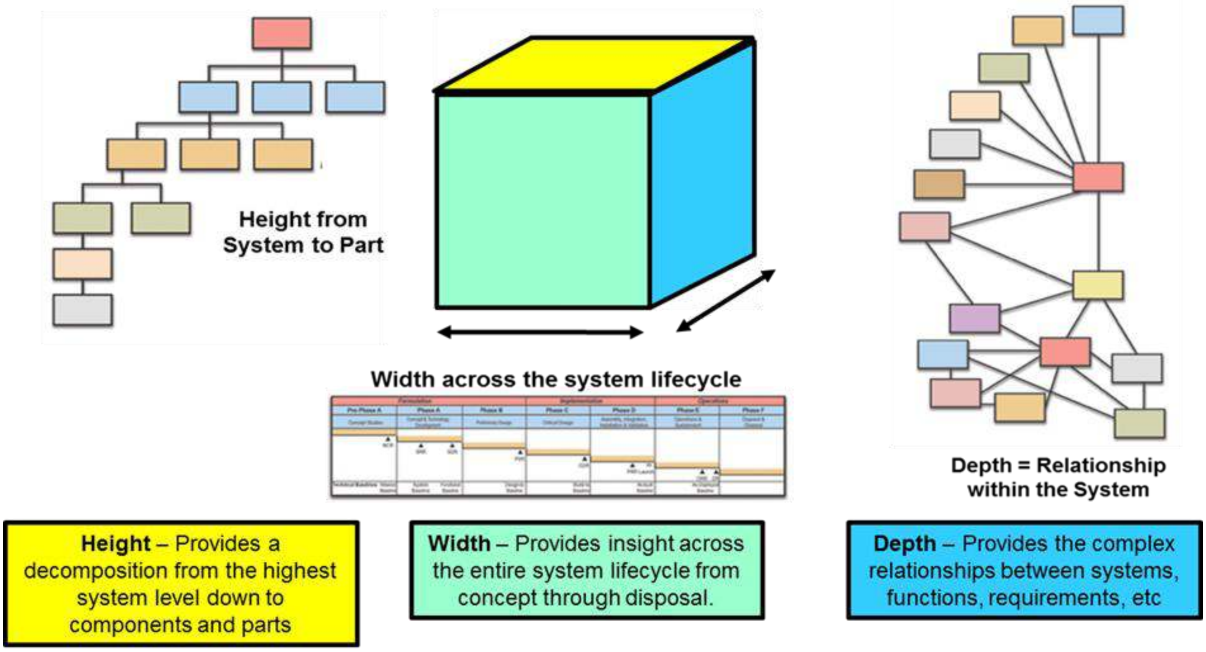


Figure 1: Dimensions of a Systems Engineering Project (Larson, *et al.* 2013; Vaneman and Carlson 2019)

Relevant definitions follow:

Modeling Languages – Serve as the basis of tools, and enable the development of system models. Modeling languages are based on a logical construct (visual representation) and/or an ontology. An ontology is a collection of standardized, defined terms and concepts and the relationships among the terms and concepts.

Structure – Defines the relationships between the system's entities. These structures allow for the emergence of system behaviors and performance characterizations within the model.

Model-Based Processes – Provide the analytical framework to conduct the analysis of the system virtually defined in the model. The model-based processes may be traditional systems engineering processes such as requirements management, risk management, or analytical methods such as discrete event simulation, systems dynamics modeling, and dynamic programming.

Presentation Frameworks – Provide the framework for the logical constructs of the system data in visualization models that are appropriate for the given stakeholders. These visualization models take the form of traditional systems engineering models. These individual models are often grouped into frameworks that provide the standard views and descriptions of the models, and the standard data structure of architecture models.

The four components of MBSE are shown in Figure 2 (Vaneman 2016).

Maximum MBSE Effectiveness occurs at the convergence of the four components. Most MBSE tools strive to be within this. Model-Based Systems Engineering tools are general-purpose software products that use modeling languages, and support the specification, design, analysis, validation, and verification of complex system representations. Although some tools are treated as synonymous with MBSE, this paper does not discuss specific tools. Tools are generally popular for a period, and then are superseded by the “next best idea,” while the MBSE concepts presented here are meant to be fundamental and transcendent tools.

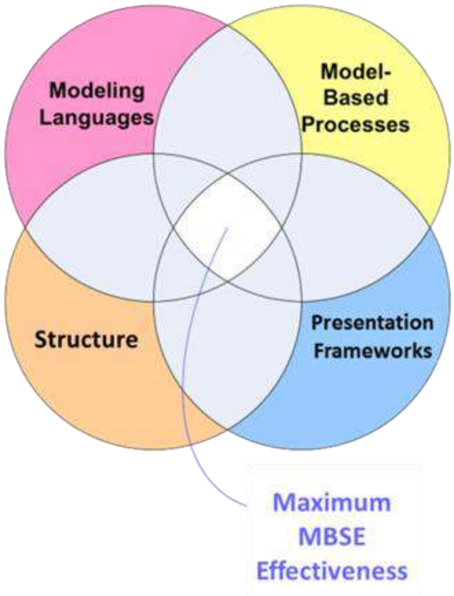


Figure 2: Four Components of MBSE (Vaneman 2016)

III. The MBSE Environment

Model-Based Systems Engineering was envisioned to transform systems engineering’s reliance on document-based work products to an engineering environment based on models. This transformation means more than using model-based tools and processes to create hard-copy text-based documents, drawings, and diagrams. In an MBSE environment, the model is a virtual representation of the system, where each system entity is represented as data, only once, with all necessary attributes and relationships of that entity being portrayed. The relationships developed between the system’s entities allows for concordance across the model.

Concordance is the ability to represent a single entity such that data in one view, or level of abstraction, match the data in another view, or level of abstraction, when talking about the exact same thing (Vaneman 2016). This singular representation of each entity allows the system to be explored from the various engineering and programmatic perspectives (viewpoints). A viewpoint describes data drawn from one or more perspectives and organized in a particular way useful to management decision-making. The compilation of viewpoints (e.g. capability, operational, system, programmatic viewpoints) represents the entire system, where the system can be explored as a whole, or from a single perspective.

The MBSE environment may consist of single or multiple tools and data repositories. Regardless, whether the environment is a single tool and data repository, or is composed of multiple tools and an integrated data repository, the four components must be implemented for the MBSE environment to be fully effective. A notional MBSE environment is shown in Figure 3.

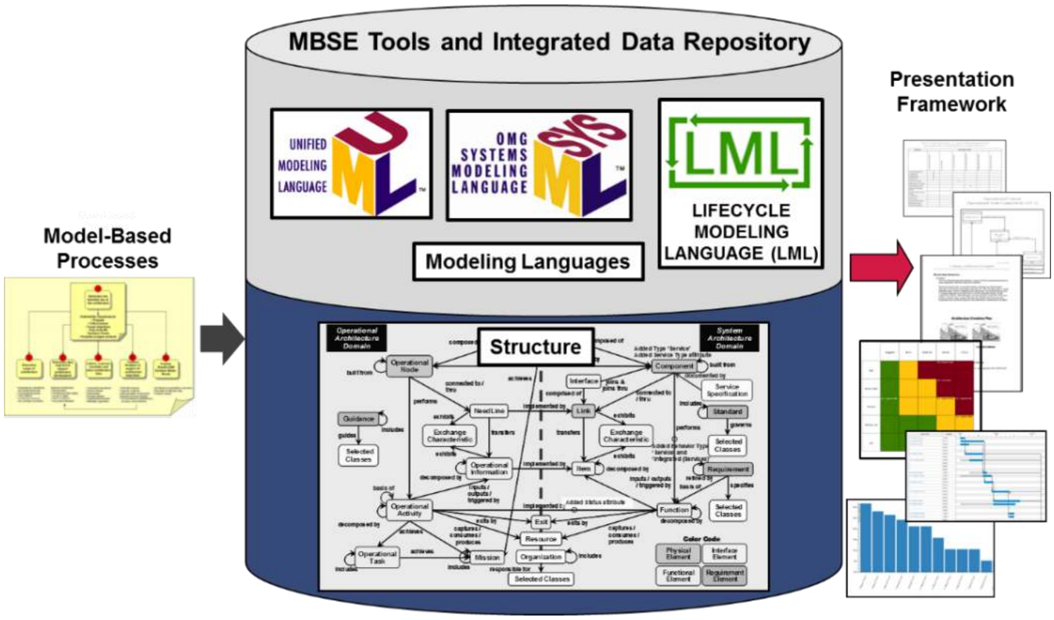


Figure 3: Notional MBSE Environment

A. Model-Based Processes

One of the goals of the Department of Defense (DoD) (USA) Digital Information Strategy is to "formalize the development, integration, and use of models to inform enterprise and program decision-making" (DASD(SE) 2018). In MBSE, the modeling process provides the analytical framework to conduct the analysis of the system virtually defined in the model. Model-Based Systems Engineering is not a new discipline designed to supersede traditional systems engineering, but a new way to address systems engineering problems such as architecting, requirements management, risk management, and analytical methods such as discrete event simulation, systems dynamics modeling, and dynamic programming. Model-based processes offer different analytical approaches and are used to address the various challenges throughout the system's lifecycle.

Model-based processes emphasize the model itself, specifically, the objects and relationships it contains, rather than the diagram to encourage better model development, usage, and decision-making. As such, model-based processes must focus on the entity as the atomic level to be modeled, with a holistic understanding of the system issues to be addressed. Thus model-based processes essentially occur from a bottom-up perspective that develops the entities and attributes first, and builds the relationships between the entities or various data types. Traditional systems engineering processes are developed from a top-down perspective with the diagram or document being the focus, and the lower levels of the system depicted within.

System entities also have attributes such as: described by physical dimensions; satisfy capabilities; perform functions; exhibit behavior; have cost; are governed by a schedule; and, may have risks, just to name the most common. Essentially, each modeled entity should fully represent their corresponding system element. Traditional systems engineering artifacts typically represent the system from only one or two of these dimensions.

This new paradigm can also serve as a bridge between system engineering and the related disciplines that occur throughout the lifecycle. For example, the systems engineering and operations research disciplines often address similar problems, with analytical processes rooted in their own discipline.

However, these communities often use different baselines of the same system when solving these problems. The MBSE environment allows each discipline to solve problems with their own methods, but provides a common baseline, that will facilitate consistencies among the understanding of the system.

B. Modeling Languages

The foundation of the MBSE environment is the modeling languages that enable the tools. Modeling languages are based on a visual representation (logical construct), an ontology, or both. The leading MBSE language is SysML, a general-purpose graphical modeling language for supporting requirements, design, analysis, and validation and verification, for systems that include hardware, software, information, process, and people. SysML has made tremendous progress towards advancing MBSE, but the lack of an ontology hinders the language from representing precise complex system relationships and interactions, and forces tool developers to define data schemas for tool execution. The systems engineering community, led by the Object Management Group (OMG), is currently developing SysML 2.0, which seeks a long-sought ontology (Friedenthal and Burkhart, 2015).

An ontology is a collection of standardized, defined terms, and relationships between the terms, to capture the information that describes the physical, functional, performance, and programmatic aspects of a system (LML Steering Committee, 2015). The most common format for an ontology is the Entity, Relationship, and Attribute (ERA) data schema. Each entity has a defined relationship allowing it to represent system complexity, and may include multiple attributes to capture all of the dimensions of the system. An attribute is an inherent characteristic or quality that further describes an entity. The ERA approach allows for the efficient use of entities due to the attributes and relationships defined (LML Steering Committee, 2015).

Lifecycle Modeling Language is an example of entity, relationship, and attribute (ERA) based language. It was developed as an approach for incorporating visualization models and corresponding ontologies within the same framework (LML Steering Committee 2015). Table 1 shows the LML entities and their corresponding visualization models (Vaneman 2018).

C. Model Structure

Systems consist of "building blocks" and their relationships to each other that allow them to come together in a designed form that satisfies the desired capabilities and functionality. Model structures are probably the least understood aspect to MBSE. Structure defines the relationships between the system entities, establishes concordance within the model, and allows for the emergence of system behaviors and performance characterizations within the model. The relationships between the principal entities define structure, address complexity, and ensure system traceability across the model.

Current implementations often do not apply model structures efficiently, thereby causing data to be represented in the model more than once. This often results in data maintenance errors, leading to different representations of the same data in different views. In an MBSE environment, each entity is ideally represented in the model only once. Model structure is formed by the relationships within the data, where essentially every entity is related to every other entity (LML Steering Committee 2015). Figure 4 shows a partial set of relationships that form the basis for the entire ontology of LML.

Table 1: LML Entities and their Corresponding Visualization Models (Vaneman 2018)

LML Entity	**LML Model**
Action	Action Diagram
Artifact	Photo, Diagram, etc.
Asset	Asset Diagram
Resource (Asset)	Asset Diagram
Port (Asset)	Asset Diagram
Characteristic	State Machine, Entity-Relationship, and Class Diagrams
Measure (Characteristic)	Hierarchy, Spider, and Radar Charts
Connection	Asset Diagram
Conduit (Connection)	Asset Diagram
Logical (Connection)	Entity-Relationship Diagram
Cost	Pie/Bar/Line Charts
Decision	
Input/Output	State Machine Diagram
Location	Map
Physical (Location)	Geographic Maps
Orbital (Location)	Orbital Charts
Virtual (Location)	Network Maps
Risk	Risk Matrix
Statement	Hierarchy and Spider Charts
Requirement (Statement)	Hierarchy and Spider Charts
Time	Gantt Chart, Timeline Diagram
Equation	*Equation*

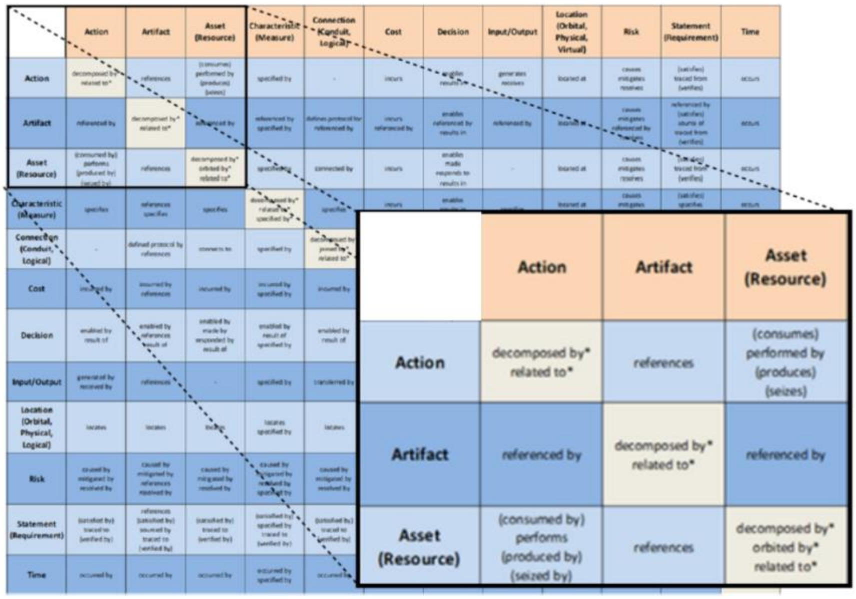


Figure 4: Partial View of LML Ontological Relationships

D. Presentation Framework

System data and models need to be portrayed in a manner that decision-makers can understand and use to make decisions. Presentation Frameworks provide the structure to represent system data visually in views that are appropriate for the given stakeholders. These visualization models take the form of traditional systems engineering models, and are typically grouped into frameworks that provide the standard views and descriptions of the models. Complexity in a model-based environment is significantly reduced by separating and characterizing systems issues into various data-driven viewpoints and views. The Department of Defense Architecture Framework (DoDAF) and the Zachman Framework are examples of frameworks that may be encountered within the systems engineering domain.

Regardless of the framework, the fundamental components of the structure are the same. Each framework provides the definitions, references, guidance and rules for structuring, classifying, and organizing the views of systems engineering data. For example, the Zachman Framework is a two-dimensional matrix classification schema that reflects the intersection between two classification types. The first type is known as interrogatives and asks the questions: What (data); How (function); Where (network); Who (people); When (time); and Why (motivation). The second classification type, known as transformations, considers what is needed to transform a concept into instantiations. The transformation perspectives are: Planner (scope); Owner (enterprise/business definition); Designer (system model); Builder (technology specification); and, Technician (detailed representation) (Zachman 2008).

One of the biggest misperceptions about presentation frameworks is that the views contained in them are unique to the framework. In fact, the visualization models contained in these frameworks are models widely used in systems engineering. Complexity in the model-based environment is significantly reduced by separating and characterizing systems issues into various date-driven viewpoints and views. The current presentation frameworks focus on the early systems engineering processes. For MBSE to be truly effective, the presentation frameworks must be extended to include data that is relevant across the system lifecycle (e.g. requirements, risk, verification and validation data, programmatic data).

Summary and Conclusions

Model-Based Systems Engineering represents a fundamental change in the systems engineering; from a document-based to model-based discipline. It requires an organizational mindset change in systems engineering processes, and a change in expectations of the data required to make decisions. A well-developed system model will be a virtual representation of the system, with relationships made between the various system entities, and exhibit a high level of concordance (Vaneman and Carlson 2019).

The challenges of implementing an MBSE environment extend past the technical challenges. To succeed at implementing MBSE, organizations also need a deliberate effort to transform the workforce and promote cultural change. Organizations must also realize that implementing MBSE often takes a leap of faith as it deviates from the proven analog processes (Vaneman and Carlson 2019).

List of Acronyms Used in this Paper

Acronym	Explanation
DoDAF	Department of Defense Architecture Framework
INCOSE	International Council on Systems Engineering
LML	Lifecycle Modeling Language
MBSE	Model-Based Systems Engineering
SysML	System Modeling Language
UML	Unified Modeling Language

References

Friedenthal, S. and R. Burkhart. *Evolving SysML and the System Modeling Environment to Support MBSE.* INSIGHT, 2015, v18, n2, pp. 39-41.

International Council on Systems Engineering (INCOSE). *Systems Engineering Vision 2020.* 2007.

Larson, W., D. Kirkpatrick, D. Verma, T. Date, T, and J.J. Sellers. *Space Systems Engineering.* Boston, MA: McGraw Hill, 2013.

Lifecycle Modeling Language Steering Committee. *Lifecycle Modeling Language (LML) Specification.* Document URL: http://www.lifecyclemodeling.org/spec/1.1, 2015.

Long, D. and Z. Scott. *A Primer for Model-based Systems Engineering (2nd Edition).* Blacksburg, VA: Vitech Corporation. 2011.

Office of the Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)). *Department of Defense Digital Engineering Strategy Guide.* Department of Defense, Washington, DC, June 2018.

Vaneman, W. K. *Enhancing Model-Based Systems Engineering with the Lifecycle Modeling Language.* Orlando, FL: Proceedings of the10th Annual IEEE Systems Conference, April 18-21, 2016.

Vaneman, W.K. *Evolving Model-Based Systems Engineering Ontologies and Structures.* Washington: D.C.: Proceedings of the 28th Annual INCOSE International Symposium, July 7-12, 2018.

Vaneman, W.K. and R. Carlson. *Model-Based Systems Engineering Implementation Considerations.* Orlando, FL: Proceedings of the13th Annual IEEE Systems Conference, April 8-11, 2019.

Zachman, J.A. *The Concise definition of the Zachman Framework.* Zachman International, Inc. https://www.zachman.com/about-the-zachman-framework.

About the Author

Warren Vaneman has more than 32 years of leadership and systems engineering experience from various positions within the Intelligence Community. He is also a Retired Navy Reserve Captain. He has a B.S. from the State University of New York Maritime College, an M.S. in Systems Engineering, and a Ph.D. in Industrial and Systems Engineering from Virginia Tech, and a Joint Professional Military Education Phase 1 Certificate from the Naval War College. He is a Certified Systems Engineering Professional (CSEP) by the International Council on System Engineering (INCOSE).

3. ADDITIONAL ARTICLE

3.1 The Past and the Future of Systems Engineering

by

Olivier L. de Weck

Journal of INCOSE

Editor-in-Chief

Editor's Note: The May 2018 issue of *Systems Engineering*, the Journal of the International Council on Systems Engineering (INCOSE), was the 20th Anniversary Special Issue. The Editor-in-Chief, Olivier L. de Weck, dedicated this issue to the past and the future of systems engineering. "Oli" provides an informative introduction and eight articles that explore the roots of *Systems Engineering* as a field of research and practice, crystallize the progress and challenges over the last two decades, and set our sights to the next twenty years of evolution of the *Systems Engineering* discipline. This article provides de Weck's introduction to this special issue.

Published with permission.

***Systems Engineering* is not just a field of research and practice;**

it is truly an endless journey.

We have transformed our way of working in *Systems Engineering* from the classical V-approach to more agile methods where requirements are discovered along the way, instead of being frozen at the start. We are abandoning documents in favor of integrated and executable system models. We now see *Systems Engineering* in a more evolutionary way than we did 20 years ago. Systems are no longer just conceived, designed, implemented, and operated in a linear fashion to satisfy stakeholder needs. They are ever-changing, coalescing into systems-of-systems driven by dynamic technological, economic, and political forces and they require us to constantly reassess, upgrade, and evolve them over time. This is why designing systems for specific desired lifecycle properties such as *resilience*, *sustainability,* and *evolvability* is more important today than ever before. This has transformed *Systems Engineering* from a rather narrow and technocratic field into a broad vision of *engineering of socio-technical systems*, a vision that many of us refer to as *Engineering Systems*[1].

Systems Engineering is a discipline whose responsibility it is to create and operate technologically enabled systems that satisfy stakeholder needs throughout their life cycle. Systems engineers reduce ambiguity by clearly defining stakeholder needs and customer requirements, they focus creativity by developing a system's architecture and design, and they manage the system's complexity over time. Considerations taken into account by systems engineers include, among others, quality, cost and schedule, risk and opportunity under uncertainty, manufacturing and realization, performance and safety during operations, training and support, as well as disposal and recycling at the end of life.

Has Systems Engineering progressed over the last 20 Years?

Systems Engineering is at the nexus of understanding real problems in society and in markets and clarifying what is needed, the channeling of creativity, the bundling of technical expertise, the expression of architecture, the emergence of inspiring and competitive design, the harmonization of humans and machines, and the sustainable transition to industrialization at scale. These are truly high stakes.

But has *Systems Engineering* progressed at all over the last 20 years?

To me, *Systems Engineering* seems at once the same as it was 20 years ago, but on the other hand it is also almost unrecognizable. The use of sophisticated models and simulations at multiple levels of abstraction, the ability to model not only functions and performance but also lifecycle properties and lifecycle costs, the understanding of what are *systems-of-systems* and the application of systems engineering not only to technical artifacts but *enterprises* and *socio-technical systems* are at the forefront of this transformation.

And yet, while the internet and more recently the internet-of-things have clearly had an enormous impact on society, it is mainly computer scientists and software engineers and not necessarily systems engineers that lay claim to this grand success. This begs the question: Are we treading water in *Systems Engineering* as a scientific discipline? Are we actually ahead of our time? What has *Systems Engineering* really contributed to society? Is it true that *Systems Engineering* is only really noticed when it is done poorly or not at all? As we collectively ponder the question whether and to what extent *Systems Engineering* has progressed over the last 20 years we could take two somewhat extreme positions. Let's call these positions "P" and "O".

[1] De Weck, O. L., Roos, D., & Magee, C. L. (2011). *Engineering systems: Meeting human needs in a complex technological world*. MIT Press. See the first article in the SE Publications Section of this issue for more information concerning this book.

Position P (=pessimistic view).

Systems Engineering should be mainly about the optimal blending of hardware, software and humans to maximize value to individuals, enterprises and society. When ISO standard 15288 was first published in 1998, there was much hope that *Systems Engineering* would fully blend and integrate these aspects and that software engineering would radically transform how *Systems Engineering* is done. However, we have been sorely disappointed. Twenty years later *Systems Engineering* is still being viewed by most practitioners through the lens of the classic "V" model with a waterfall approach, gathering of system requirements upfront, etc. … and consequently many large software and systems projects still fail for the same reasons they failed twenty years ago. While there has been progress in auto-generating code for embedded software in many products, this has not radically changed how we practice and teach *Systems Engineering*. The great advances of the internet, internet of things (IoT) and industry 4.0 are not a credit to *Systems Engineering* but to others who have avoided the overly burdensome, bureaucratic, process-centric and outdated views of *Systems Engineering*.

Position O (=optimistic view)

While *Systems Engineering* was a rather new discipline twenty years ago - rooted in the tradition of large and expensive defense and aerospace systems - it is today radically transformed. Systems are now developed in an agile way whereby software, hardware, and manufacturing are co-designed in spirals or sprints, quickly tested, and continuously improved using immediate customer feedback. Rather than waiting for heavy certification procedures and approvals, systems are deployed rapidly by the thousands or even by the millions and data on actual usage is collected in real-time, with big data analytics, deep machine learning and fast improvement cycles now commonplace. *Systems Engineering* has had a major impact on these developments and is today the recognized *primus inter pares discipline* of engineering where all functions, interfaces and interactions between people, software, and hardware come together. *Systems Engineering* is today effectively the parent discipline of *Software Engineering* and has fully absorbed its best practices and fundamental principles which are clearly articulated and generally accepted.

Having read these two diametrically opposed statements, what do *you* think?

Have we already reached the *INCOSE Vision 2025*[2] today or are we well behind schedule? Understanding where as a discipline we truly fall between the *"P"* and the *"O"* and how these views may differ across domains, organizations and generations of systems engineers is the main ambition of this special issue.

Following are summaries of the articles in the 20th Anniversary Special Issue. Each can be accessed at the link provided: https://onlinelibrary.wiley.com/toc/15206858/2018/21/3.

In his article "A historical perspective on systems engineering" *Eric Honour* explores the impetus for creating the journal in 1998. As former President of INCOSE during its formative period he shares a unique perspective on the birth of NCOSE, which later became INCOSE, and the conception of the *Systems Engineering* journal in 1998. He tells us in depth about prior failed attempts at creating an archival journal for *Systems Engineering* and how the recruitment of Professor Andy Sage as founding editor along with an initial critical mass of contributors was essential to achieving a sustainable and high

[2] URL: https://www.incose.org/products-and-publications/se-vision-2025

quality publication. Eric is a highly respected scholar of *Systems Engineering* himself and has – among other contributions - provided us with a quantified understanding of the relationship between the effort spent on *Systems Engineering* activities in a project or program and its eventual benefits, what some call the return-on-investment (ROI) of *Systems Engineering*.

The second article is provided by *Sarah Sheard*, a Principal Engineer at the Software Engineering Institute (SEI) at Carnegie Mellon University. She is also an INCOSE Fellow and an Associate Editor of this journal and has been an outstanding member of our Editorial Board for many years. Sarah has done an in-depth analysis of two specific years (2000 and 2015) of the journal's papers and has looked in particular at the role of *software* in systems. What emerges from her work is a fascinating story of evolution of our field in its greater context of information and software systems. Her findings are validated with an extensive lexicographic analysis using both the *Compendex* and *Inspec* databases. Software has been a critical enabler of systems over the last 20 years and the fraction of a system's functionality executed by software has steadily increased to the point where *Systems Engineering* and *Software Engineering* are now almost indistinguishable. This article examines the role of software in *Systems Engineering* scholarship and practice over the last 20 years. The movement from rather isolated systems dominated by mechanical subsystems and human operators to larger, increasingly automated, and cloud-based systems-of-systems is also evident in this paper.

Azad Madni from the University of Southern California (USA) and Michael Sievers from NASA's Jet Propulsion Laboratory (JPL) provide us with a very important and realistic view of the state of the art in Model Based Systems Engineering (MBSE) today (See the previous article). JPL in particular is a clear leader in MBSE with a very ambitious adoption of MBSE on the Europa Clipper and approximately twenty other missions. In its MBSE roadmap, JPL relies not only on commercial software tool vendors, but also on extensive in-house software development. MBSE is often cited as one of the major innovations in Systems Engineering over the last two decades. As systems continue to grow in scale and complexity, the Systems Engineering community has increasingly turned to Model Based Systems Engineering to manage complexity, maintain consistency, and assure traceability during system development. The authors argue that it is different from "engineering with models," which has been a common practice in the engineering profession for decades. MBSE on the other hand is a holistic, Systems Engineering approach centered on the evolving system model, which serves as the "sole source of truth" about the system. Besides providing concrete evidence for the advancement of MBSE, Madni and Sievers also highlight the main blocking points and research opportunities in order for MBSE to unleash its full potential in the future.

The issue of uncertainty is at the heart of the article "Sensitivity analysis methods for mitigating uncertainty in engineering system design" by *Chelsea Curran*, *Doug Allaire* and *Karen Willcox* of MIT. In their seminal paper they introduce a rigorous approach to uncertainty quantification and mitigation in complex engineering systems using a design computation framework. A new entropy-based sensitivity analysis methodology is introduced, which apportions output uncertainty into contributions due to not only the variance of input factors and their interactions, but also to features of the underlying probability distributions that are related to distribution shape and extent. The ability to set realistic targets and mitigate programmatic, cost, schedule and technical risks remains one of the most formidable challenges for Systems Engineering. For me this paper points the way to a future understanding of Systems Engineering that relies on rigorous data analysis, statistics, and deliberate uncertainty management. This is a long way from the past, where Systems Engineering was mainly based on “rules of thumb” or simplistic deterministic models of average system behavior.

Another important trend in *Systems Engineering* related to uncertainty has been the notion of emergence where the social and the technical parts of a system interact and co-evolve with each other. Emergence is a subtle phenomenon that creates macroscopic behaviors in systems that cannot easily be predicted by linear or quasi-linear interactions of its constituent elements. A particularly important class of methods for studying emergence in systems is Agent-based Modeling (ABM) as explained by *Babak Heydari* and *Michael Pennock* of the School of Systems and Enterprises at Stevens Institute of Technology in their paper titled “Guiding the Behavior of Socio-Technical Systems: The Role of Agent-Based Modeling”. Dynamic interactions are increasingly becoming an essential part of design and governance of many emerging systems such as sharing economy platforms and critical infrastructures. This paper makes the case for an agent-based approach to support the design and governance of socio-technical systems and asserts that agent-based modeling and simulation, with certain suggested revisions, can be a powerful methodology in this domain. An important contribution here is a perspective on the often controversial topic of model validation for Agent-Based Modeling (ABM).

In their article “System Architecting and Design Space Characterization” *Ali Raz, Bob Kenley* and *Dan DeLaurentis* of Purdue University skillfully build on prior work and connect *Systems Architecting* with design space exploration. The article provides a process for system architecting that incorporates a holistic approach for architecture design space characterization by integrating decision alternatives in functional, physical, and allocational design spaces and accounting for interactions. The design space characterization is made an integral part of the system architecting process and a set-theoretic framework is developed for managing an extensive design space. The design space characterization problem is formulated as identification of the significant decisions variables and quantification of their impact on the system objectives. A Design of Experiments framework---utilizing Analysis of Variation (ANOVA) and Range Tests---is presented to holistically characterize the system architecture design space including the interactions between system form, function, operations, and design decisions.

One of the questions many critics (and even proponents) of *Systems Engineering* have asked is this: *How do we know if or when Systems Engineering is working well?* How do we measure its impact or its benefits and how can we make sure Systems Engineering is based on real evidence? In other words, how do we introduce a new level of rigor when it comes to implementing existing and new *Systems Engineering* principles, methods and tools? This is a topic that has been asked for a long time in medicine. How do we know that a certain medication, treatment or intervention is effective? Can we justify large health care expenses with evidence as to their efficacy?

In their contribution “Evidence based Systems Engineering” *Duane Hybertson, Mimi Hailegiorghis, Kenneth Griesi, Brian Soeder* and *William Rouse* of MITRE Corporation and Stevens Institute of Technology propose an evidence-based approach to Systems Engineering, similar to medicine where the effectiveness of a new framework or intervention should be assessed rigorously using statistics on measures of outcomes and carefully chosen control groups. This thought-provoking framework includes the combining of scientific knowledge with data analytics to produce best evidence, and then applying mediating context factors to the evidence to make *Systems Engineering* decisions. Lessons learned from evidence-based practice in other domains, especially law and medicine, are described and incorporated into evidence-based systems engineering (EBSE).

The final article is provided by a representative of the new generation of *Systems Engineering* scholars, *David Broniatowski* of George Washington University, who uses the analogy of the famous *Tower of Babel* to explain the metamorphosis from *Systems Engineering* to *Engineering Systems* alluded to earlier. His article is titled "Building the tower without climbing it: Progress in engineering systems". Specific attention in his analysis is given to approaches that emphasize the roles of abstraction hierarchies, contextual interpretation, knowledge sharing, and expertise. The paper also briefly addresses the perceived tradeoff between academic rigor and practical utility – a perennial concern in the *Systems Engineering* field.

4. SYSTEMS ENGINEERING NEWS

4.1 Call for Applications: Editor-in-Chief of IEEE Reliability Magazine

The IEEE Reliability Society is seeking an Editor-in-Chief (EIC) for its newly approved publication, Reliability Magazine (R-Mag).

Requirements

The EIC must be a highly accomplished professional with a specialty within the scope of the Reliability Society and possess strong leadership skills. The EIC should have experiences serving on an IEEE editorial board, and preferably has had extensive experiences as an IEEE magazine AEIC or AE. She/he must be a member of the IEEE Reliability Society when assuming the EIC position.

Reporting to the Vice President for Publications, the duties of the EIC include:

- Recruiting, mentoring and managing the editorial board consisting of associate editors
- Setting the editorial direction for the periodical
- Planning and finalizing each issue of the periodical for production
- Making final publication decisions for manuscripts
- Communicating with authors and answering magazine-related questions
- Working with IEEE production staff
- Representing the publication to the Reliability Society at appropriate events

Duration of Commitment

- As per the Reliability Society policy, the inaugural EIC will be appointed to an initial 3-year term with the possibility of being appointed to serve a second (final) term.

Process for Applying

Interested candidates should submit an application packet consisting of the following materials:

- A current CV or résumé which should include past editorial experiences and service activities to the IEEE, the Reliability Society, and related or similar organizations
- A 250-350-word statement of the applicant's vision for R-Mag
- Names of 3 references (one of which should note the individual's abilities to serve as an editor)
- A letter of support from the individual's supervisor (if applicable)
- Application packages should be sent to Dr. Steven Li, Chair of the R-Mag Editorial Search Committee.

Extended application deadline is May 31, 2019.

Review of applications will begin immediately and a decision will be made by mid-June, 2019.

4.2 Hyperloop One Seeks a New Systems Integration Manager

Hyperloop One is looking for a Systems Integration Manager who will play a key role in setting the standards for integration for Virgin Hyperloop One's transportation system of the future that will be used around the world. The Hyperstructures team is responsible for the design, development and integration of structures, mechanisms and systems of the Hyperloop Infrastructure that make travel on the transportation system of the future possible. As Systems Integration Manager, successful applicant will be instrumental in developing a safe, cost effective, and robust transportation system. We are seeking candidates with broad skillsets and various levels of experience to join our team of qualified, diverse individuals at our Los Angeles facility.

The responsibilities of the integration manager include:

- Developing Systems Integration and Interface Management Plans
- Establishing and maintaining a systematic, documented, comprehensive, and verifiable integration process for wayside systems to be applied throughout the duration of the development life cycle
- Developing and managing interface control documents
- Coordinating and managing interface to vendors and contractors
- Coordinating and managing interfaces to other disciplines and internal teams
- Reviewing designs from internal teams as well as from external consultants

Applications can be made via LinkedIn

4.3 Industrial Research Position on Domain-Specific Languages for MBSE Still Open

A full-time Knowledge Transfer Research (KTP) Associate position is still open for application until 23 May 2019. Primarily based at Smith & Nephew (S&N) in Hull, the successful applicant will lead a KTP project between S&N and the University of York. The applicant will be employed by The University of York but will be based primarily at Smith & Nephew in Hull, with visits to the company's other sites as required.

Through this KTP, the successful applicant will lead the transition of S&N processes into a new design and development methodology for medical devices where requirements, risks, and safety cases are captured using structured models (e.g. SysML for requirements, GSN/SACM for safety cases), and are shared with stakeholders using a centralized version control repository. This will allow S&N to streamline activities such as change tracking, to conduct automated model analysis, and to automate manual processes.

The main objectives of the role are to carry out research and select, extend or develop appropriate modelling languages and standards for the different activities (requirements management, risk and safety case analysis) involved the development lifecycle of NPWT devices, and to design and deploy an optimal development lifecycle model for efficient collaborative modelling.

Key responsibilities of the role include understanding existing development processes, becoming familiar with the regulatory environment, evaluating available modelling technologies and collaborative version control systems, developing Domain-Specific Languages that can support the development lifecycle of NPWT devices, and assembling a collaborative modelling environment.

More Information

Informal enquiries may be made to:

Professor Dimitris Kolovos

Mr Yeswanth Gadde

4.4 Cities Need 'System of Systems' to Break Silos in Managing IoT Data

Extract by

Chris Teale

April 8, 2019

If cities are to effectively manage the vast amounts of data they collect from internet of things (IoT) devices, they need to take a holistic approach and break down silos - speakers said during a panel discussion at Smart Cities Connect in Denver last week.

All too often, city leaders are guilty of thinking of IoT initiatives — like smart lighting and smart parking — as isolated strategies. Instead, Rob Silverberg, Dell EMC's chief technology officer for digital communities, said cities should use a "system of systems" in their IoT and data collection strategy and think about how, for example, smart parking fits into a wider goal of having smart transportation.

"What [cities are] starting to realize is that they're implementing silos," Silverberg told Smart Cities Dive in an interview after the panel discussion. "As they start to look at it more strategically, some of the cities have decided to establish more of a platform approach."

According to an estimate given during the panel, there could be as many as 200 billion connected devices worldwide by 2031, and it will be imperative for governments to be able to manage all the data gathered and use it in ways that will make its residents' lives better. As cities experiment with initiatives like smart parking and smart lighting, they will need to work across departments to ensure that decisions are being made in the best way possible.

More Information

4.5 Laboratory Testing for Integrated Aircraft Systems

Extract by

Ben Sampson

April 8, 2019

Engineers in the aerospace industry are always under pressure to shorten development times for new aircraft. One way to achieve this is to reuse proven designs, architectures and components. But with new technology, societal trends, customer demands and commercial pressures, there will always be a need for development. Aircraft are becoming more complex, varied and interconnected, with more electronics and electrical systems on board.

The onus therefore falls on engineers to innovate better and quicker ways of evaluating and certifying aircraft. Flight testing is an expensive and time-consuming process, so engineers are aiming to test more of an aircraft's systems at higher levels of integration on the ground, before flight testing.

The forthcoming Boeing 777X, which completed its maiden flight last month, provides a good example of the way in which laboratory testing is playing a greater role in modern commercial aircraft development.

Boeing Test & Evaluation (T&E) built one of its most complex laboratories ever to test the 777X and verify its systems' maturity before the first flight. The 11,600-square foot (1,000 square meter) Airplane Zero Lab features an accurate recreation of the 777X flight deck, integrated with all of the airplane's components and testing equipment in one place. Engineers are using the laboratory to run the 777X's systems in a virtual environment before its first flight.

More Information

4.6 Tutorial Submission Deadline for INCOSE ASEC Extended

The deadline for tutorial submissions for INCOSE's Annual Systems Engineering Conference (ASEC) is 24 May 2019. ASEC is INCOSE UK's flagship event to take place from 19-20 November brings together a wide range of professionals from a variety of backgrounds, with the common interest of building upon their Systems Engineering knowledge and sharing ideas with their peers.

The format of 'tutorial' can cover a range of options: from a 'set of instructions to complete a task' to 'an interactive problem-solving workshop'. A typical ASEC tutorial attracts between five and twenty delegates who are interested in trying something new or perhaps finding a different perception on something already familiar to them (and maybe looking for a break from back-to-back sessions!)

INCOSE UK is looking for around four tutorial options which will be selected from the proposals received. Proposers are welcome to submit more than one proposal, but each proposal must be self-contained so that it can be assessed independently. It is vital that the tutorial style is explicitly expressed so that delegates clearly understand whether they will be expected to plaster flip charts with Post-it notes or simply sit and listen.

Successful presenters will be awarded a single one-day pass for ASEC 2019. If you wish to field more than one presenter, then additional presenters must register for the event in the usual way. You should be prepared to present on either 19 or 20 November 2019. The information in your proposal will be incorporated into promotional material for the event and used to ensure that the required facilities are provided. INCOSE UK will contact the proposer if any clarification is required.

The full tutorial proposal must include the following information:

- Title of the tutorial
- Name and contact details of the proposer and the presenter
- Aims and objectives of the tutorial / workshop
- Overview of content
- Course material to be provided
- Facilities required (Note: presenters should source their own laptop, but rooms will be equipped with projectors and flip charts)
- Credentials of presenter(s)
- Intended audience/potential interest
- Maximum number of attendees

More Information

4.7 Vitech Webinar: MBSE 2.0 A New Performance Level for Engineering Enterprises

A new Vitech webinar on achieving new performance levels for engineering enterprises with MBSE 2.0 is available on Vitech's website.

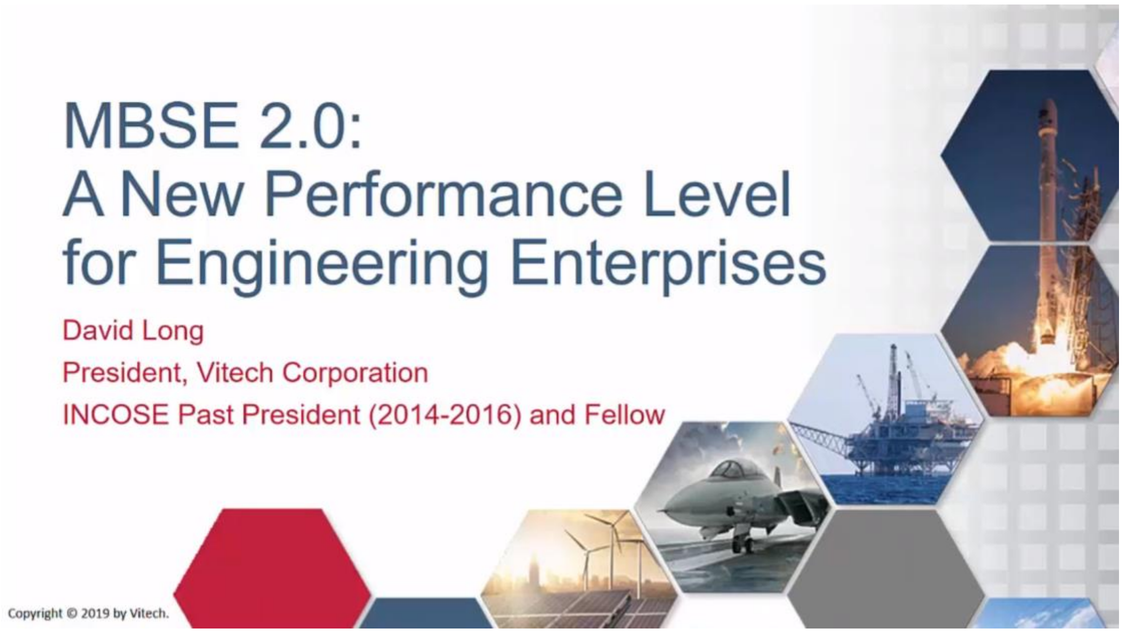


Image Source

Taken from Vitech's website: 'Concurrent model-based systems engineering is becoming critical to a successful engineering outcome. It requires that systems engineers and subject matter experts collaborate on design solutions. In order to facilitate this, it is necessary to adopt a methodology and a toolset based on a proven systems-metamodel, enabling the design team to instantiate their system model in an enterprise-class database. With a robust API that connects with the tools and processes of subject matter experts and other engineering disciplines, concurrent model-based systems engineering becomes indispensable in addressing the engineering challenges of our time. This is the power and promise of MBSE 2.0.'

Alternatively, the video may be viewed on Vitech's YouTube channel.

4.8 Speakers Announced for Building NATO NAFv4 Architectures Using UAF

NATO has recently adopted version 4 of its Enterprise Architecture Framework - NAFv4 - which adopted UAF DMM as one of two meta-models for building NAF architectures. This information session explores how to leverage **MBSE** with Architecture modelling in an integrated and disciplined approach, supporting NATO needs and enabling the modernization of complex systems (**Systems of Systems, C4I systems and complex industrial systems**).

This free half-day session brings together thought leaders from NATO, national governments, system integrators, the **UAF® development team**, and practitioners. Discussions will address the challenges, strategies, current and emerging practices, and will inform the user community about the path forward. Attendees will have the opportunity to exchange experiences and ideas.

Agenda Highlights

- **A NATO ACaT Perspective on architecting and the need to share architecture information**
  Kevin Wallis, UK Deputy Chief Architecture and UK Lead Architect for "International by Design", the National Chairperson of the NATO C3B Architecture Capability Team (ACaT) and the lead for the Federated Mission Networking Capability Architecture & Requirements Syndicate

- **An Introduction to the Unified Architecture Framework® (UAF)**
  Graham Bleakley, Ph.D. - Principal Consultant, Systems Engineering & Architecture Frameworks, IBM and OMG UAF Co-Chair
  Aurelijus Morkevicius, Ph.D. - Head of Solutions, No Magic, Inc., and OMG UAF Co-Chair

- **Update from The Open Group NATO/NAF ArchiMate Mapping Working Group**
  Eugene McSheffrey, Principal Business Consultant, MEGA International
  Dr. Harmen van den Berg, Director, BiZZdesign

- **NAF in Practice - Use of the NATO Architecture Framework in National Capability Development**
  Christian Freihoff, System Architect IT-System Bundeswehr, Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support, Germany

- **MBSE Acquisition Reference Model (ARM) - Lowering the barrier to MBSE during acquisition using UAF – UAF WG Update**
  Laura Hart - Research Engineer Principal, Lockheed Martin

- **Electric Roads - modelling the system of systems for the Electric Road Project**
  Lars-Olof Kihlström Syntell AB, Ida Karlsson Syntell AB
  ***Co-authors**: Bilin Chen Syntell AB, Håkan Sundelin RISE*

- **Model Based Risk Assessment (MBRA) - Mechanized and automated best practices of risk analysis in the age of cyber**
  Dr. Nikolai Mansourov, CTO, KDM Analytics

For the preliminary agenda and free registration, click here.

To learn more about the OMG UAF, click here.

For information about OMG membership, click here.

5. FEATURED ORGANIZATIONS

5.1 NASA Engineering and Safety Center

At the core of the NESC is an established knowledge base of technical specialists pulled from the ten NASA Centers and from a group of partner and organizations external to the Agency. This ready group of engineering experts is organized into discipline areas called Technical Discipline Teams (TDT). TDT members are from other NASA organizations, industry, academia, and other government agencies. By drawing on the mind of leading engineers across the country, the NESC consistently optimizes its processes, deepens its knowledge base, strengthens its technical capabilities, and broadens its perspectives, thereby further executing its commitment to engineering excellence. The NESC Technical Discipline Teams include, but are not limited to: aero sciences, human factors, nondestructive evaluation, propulsion and sensors/instrumentation.

The NESC strives to set the example for the Agency of providing full and appropriate documentation of every activity that is performed. Along with each report, lessons learned are communicated to Agency leadership and to engineers through avenues such as the Agency lesson learned system. Another important function of the NESC is to engage its proactive investigations to identify and address potential concerns before they become major problems. To further this goal, the NESC is currently leading NASA's efforts for independent data mining and trend analysis. The NESC has established a Data Mining and Trending Group that includes representatives from all NASA Centers as well as external experts. This group ensures that results are maximized and that the NESC comprehensively learns from previous efforts.

More Information

5.2 Hellenic Society for Systemic Studies (HSSS)

HSSS is a scientific non-profit society whose goal is to advance the Systemic Studies with the use of systemic multi-methodologies applied effectively in real life up to date problems. HSSS promotes the unity of sciences by spreading out the Systems Approaches for managing complexity.
The goal of HSSS is the promotion of Systemic Analysis, more specifically (taken from their website):

1. The growth and dissemination of methodologies and applications of Systemic Analysis in Science, Technology and Administration

2. The recognition of Systemic Analysis in the Public Sector, private enterprises and Organizations of the Public and Private Sector

3. The consolidation of the prestige of a Systemic Analyst

4. The promotion of research and teaching on Systemic Analysis in [academic] education and professional training

5. The exchange and distribution of information relative to Systemic Analysis in Greece and internationally

6. The provision of possibilities to its members for professional and scientific advancement

The realization of the goal of HSSS is sought (among others) with the following means:

1. Implementation of lectures, discussions, seminars and organization of conferences and other scientific activities

2. Publication of a scientific journal, information bulletin and other publications

3. Collaboration with national and international scientific societies that seek similar aims

More information

6. NEWS ON SOFTWARE TOOLS SUPPORTING SYSTEMS ENGINEERING

6.1 REUSE launches Traceability Studio

Traceability Studio is a software tool that enables the definition and implementation of trace links between two sources of information of different types. Users can trace links between key processes to be efficient and effective, such as V&V, requirements definition, architecture definition, design definition or risk management, among others, as defined in the ISO/IEEE 15288/12207 standards. Therefore, trace links between work products is a compulsory activity within the Systems Engineering processes, which can be monitored and controlled with the Traceability Studio.

Traceability Studio provides a framework for managing trace links between all types of information, covering pairs of (accessible) electronic work product items. Current Systems Engineering practice implies almost always managing ecosystems of several software tools. In this context, providing electronic traceability between heterogeneous and isolated work product items becomes a “connectivity” challenge, which is the main goal of Traceability Studio by helping its users overcome this challenge.

More Information

6.2 Phoenix Integration's ModelCenter MBSE

Phoenix Integration will soon release an entirely new analysis integration framework for MagicDraw and CAMEO Systems Modeler called ModelCenter MBSE. Phoenix Integration states that ModelCenter MBSE for MagicDraw has been designed from the ground up to be more flexible and easier to use. It allows engineers to integrate any set of analysis tools with a MagicDraw SysML model to validate system behavior, verify requirements satisfaction, and optimize the system design.

More Information

7. SYSTEMS ENGINEERING PUBLICATIONS

7.1 Engineering Systems: Meeting Human Needs in a Complex Technological World

by

Olivier L. de Weck, Daniel Roos, and Christopher L. Magee

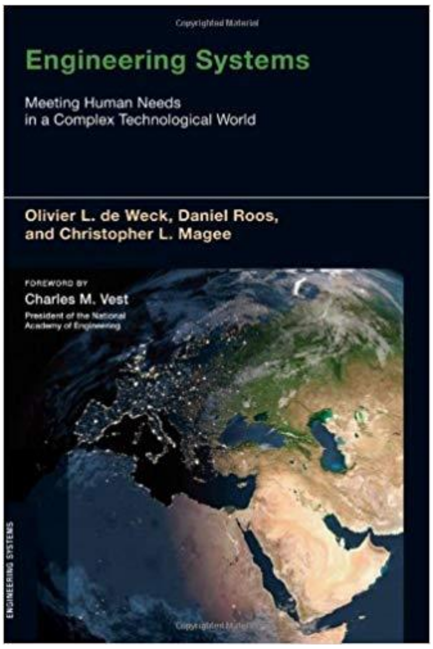


Image Source

From the Amazon Website:

Engineering, for much of the twentieth century, was mainly about artifacts and inventions. Now, it's increasingly about complex systems. As the airplane taxis to the gate, you access the Internet and check email with your PDA, linking the communication and transportation systems. At home, you recharge your plug-in hybrid vehicle, linking transportation to the electricity grid. Today's large-scale, highly complex sociotechnical systems converge, interact, and depend on each other in ways engineers of old could barely have imagined. As scale, scope, and complexity increase, engineers consider technical and social issues together in a highly integrated way as they design flexible, adaptable, robust systems that can be easily modified and reconfigured to satisfy changing requirements and new technological opportunities.

Engineering Systems offers a comprehensive examination of such systems and the associated emerging field of study. Through scholarly discussion, concrete examples, and history, the authors consider the engineer's changing role, new ways to model and analyze these systems, the impacts on engineering education, and the future challenges of meeting human needs through the technologically enabled systems of today and tomorrow.

Format: Kindle, Hardcover, Paperback

Publisher: The MIT Press (October 21, 2011)

ISBN:

ISBN-10: 0262016702

ISBN-13: 978-0262016704

More Information

7.2 Emergent Behavior in Complex Systems Engineering:

A Modeling and Simulation Approach

by

Saurabh Mittal, Saikou Diallo, and Andreas Tolk

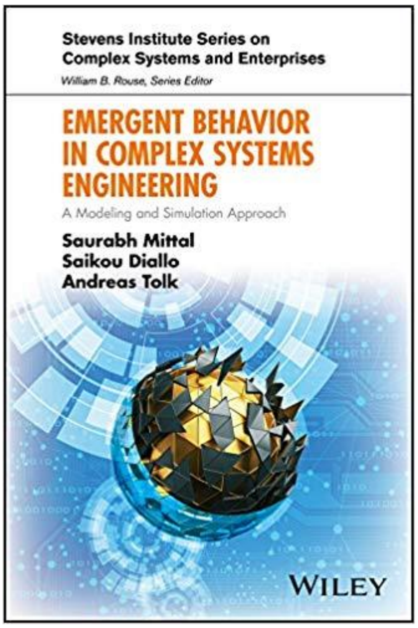


Image Source

From the Amazon Website:

In *Emergent Behavior in Complex Systems Engineering*, the authors present the theoretical considerations and the tools required to enable the study of emergent behaviors in manmade systems. Information Technology is key to today's modern world. Scientific theories introduced in the last five decades can now be realized with the latest computational infrastructure. Modeling and simulation, along with Big Data technologies are at the forefront of such exploration and investigation.

The text offers a number of simulation-based methods, technologies, and approaches that are designed to encourage the reader to incorporate simulation technologies to further their understanding of emergent behavior in complex systems. The authors present a resource for those designing, developing, managing, operating, and maintaining systems, including system of systems. The guide is designed to help better detect, analyze, understand, and manage the emergent behavior inherent in complex systems engineering in order to reap the benefits of innovations and avoid the dangers of unforeseen consequences. This vital resource:

- Presents coverage of a wide range of simulation technologies
- Explores the subject of emergence through the lens of Modeling and Simulation (M&S)
- Offers contributions from authors at the forefront of various related disciplines such as philosophy, science, engineering, sociology, and economics
- Contains information on the next generation of complex systems engineering

Written for researchers, lecturers, and students, *Emergent Behavior in Complex Systems Engineering* provides an overview of the current discussions on complexity and emergence, and shows how systems engineering methods in general and simulation methods in particular can help in gaining new insights in complex systems engineering.

Format: Kindle, Hardcover

Publisher: Wiley; 1 edition (April 17, 2018)

ISBN:

ISBN-10: 1119378869

ISBN-13: 978-1119378860

More Information

7.3 Netcentric System of Systems Engineering with DEVS Unified Process

by

Saurabh Mittal and José L. Risco Martín

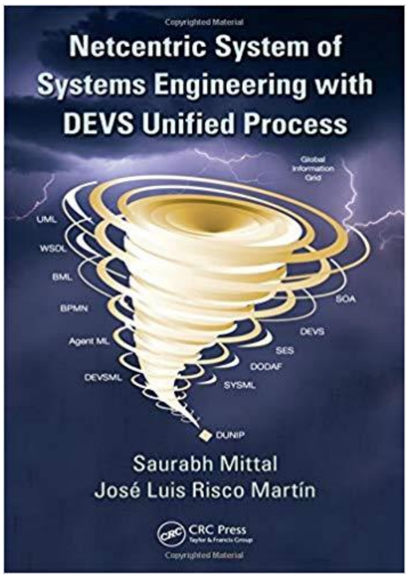


Image Source

From the Amazon Website:

In areas such as military, security, aerospace, and disaster management, the need for performance optimization and interoperability among heterogeneous systems is increasingly important. Model-driven engineering, a paradigm in which the model becomes the actual software, offers a promising approach toward systems of systems (SoS) engineering. However, model-driven engineering has largely been unachieved in complex dynamical systems and netcentric SoS, partly because modeling and simulation (M&S) frameworks are stove-piped and not designed for SoS composability. Addressing this gap, **Netcentric System of Systems Engineering with DEVS Unified Process** presents a methodology for realizing the model-driven engineering vision and netcentric SoS using DEVS Unified Process (DUNIP).

The authors draw on their experience with Discrete Event Systems Specification (DEVS) formalism, System Entity Structure (SES) theory, and applying model-driven engineering in the context of a netcentric SoS. They describe formal model-driven engineering methods for netcentric M&S using standards-based approaches to develop and test complex dynamic models with DUNIP. The book is organized into five sections:

- Section I introduces undergraduate students and novices to the world of DEVS. It covers systems and SoS M&S as well as DEVS formalism, software, modeling language, and DUNIP. It also assesses DUNIP with the requirements of the Department of Defense's (DoD) Open Unified Technical Framework (OpenUTF) for netcentric Test and Evaluation (T&E).

- Section II delves into M&S-based systems engineering for graduate students, advanced practitioners, and industry professionals. It provides methodologies to apply M&S principles to SoS design and reviews the development of executable architectures based on a framework such as the Department of Defense Architecture Framework (DoDAF). It also describes an approach for building netcentric knowledge-based contingency-driven systems.

- Section III guides graduate students, advanced DEVS users, and industry professionals who are interested in building DEVS virtual machines and netcentric SoS. It discusses modeling standardization, the deployment of models and simulators in a netcentric environment, event-driven architectures, and more.

- Section IV explores real-world case studies that realize many of the concepts defined in the previous chapters.

- Section V outlines the next steps and looks at how the modeling of netcentric complex adaptive systems can be attempted using DEVS concepts. It touches on the boundaries of DEVS formalism and the future work needed to utilize advanced concepts like weak and strong emergence, self-organization, scale-free systems, run-time modularity, and event interoperability.

This groundbreaking work details how DUNIP offers a well-structured, platform-independent methodology for the modeling and simulation of netcentric system of systems.

Format: Kindle, Hardcover, Paperback

Publisher: CRC Press; 1 edition (March 31, 2017)

ISBN:

ISBN-10: 1138076597

ISBN-13: 978-1138076594

More Information

7.4 Introduction to the Theory of Complex Systems

by

Stefan Thurner, Rudolf Hanel, and Peter Klimek

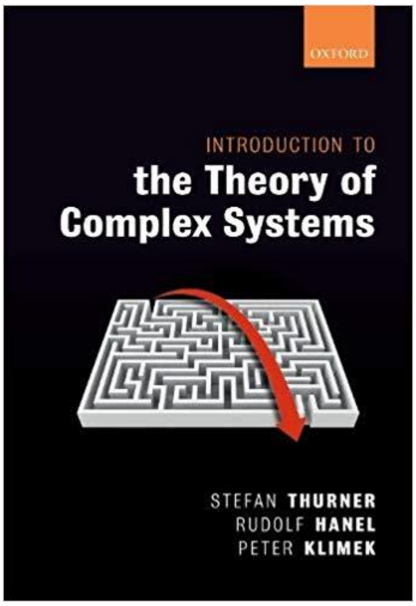


Image Source

From the Amazon.com Website:

This book is a comprehensive introduction to quantitative approaches to complex adaptive systems. Practically all areas of life on this planet are constantly confronted with complex systems, be it ecosystems, societies, traffic, financial markets, opinion formation and spreading, or the internet and social media. Complex systems are systems composed of many elements that interact strongly with each other, which makes them extremely rich dynamical systems showing a huge range of phenomena. Properties of complex systems that are of particular importance are their efficiency, robustness, resilience, and proneness to collapse.

The quantitative tools and concepts needed to understand the co-evolutionary nature of networked systems and their properties are challenging. The book gives a self-contained introduction to these concepts, so that the reader will be equipped with a toolset that allows them to engage in the science of complex systems. Topics covered include random processes of path-dependent processes, co-evolutionary dynamics, dynamics of networks, the theory of scaling, and approaches from statistical mechanics and information theory. The book extends beyond the early classical literature in the field of complex systems and summarizes the methodological progress made over the past 20 years in a clear, structured, and comprehensive way.

Format: Hardcover

Publisher: Oxford University Press (December 4, 2018)

ISBN:

ISBN-10: 9780198821939

ISBN-13: 978-0198821939

More Information

8. EDUCATION AND ACADEMIA

8.1 University of Technology Sydney Centre for Autonomous Systems

The University of Technology Sydney (UTS) Centre for Autonomous Systems (CAS) is an internationally acclaimed robots research group. The Centre specializes in robotics research that creates positive change for government, industry, and the wider community. Researchers undertake a comprehensive program of fundamental, applied and translational research, and form key industry partnerships based on real-world applications.

From 2003 to 2010, CAS was part of the ARC Centre of Excellence for Autonomous Systems, one of the largest robotics research groups in the world. Since 2010, CAS has continued at UTS as an independent research centre.

The Centre has a growing reputation in both academia and industry for developing innovative enabling technologies that seek to:

- Improve worker health and safety,
- Increase workplace productivity and output quality across a range of sectors, and
- Assist people with health conditions and disabilities to engage more fully with life.

Vision

The CAS vision is to become the Australian leader for research, development, and commercial application of human-centred robotics.

Mission

- Undertake fundamental research on the nature of perception, design, learning and control of mechatronic devices that coexist and interact with humans.

- Undertake persuasive practical demonstrations of human-centred robotics and to lay the technical groundwork for commercial application of such systems.
- Be the research partner of choice for key industries to enable the effective translation of research outcomes to commercial application.
- Attract, train, retain and deliver high quality research students to industry and academia.
- Bring research to the undergraduate classroom to reinforce the practice-based approach in education at UTS.
- Achieve sustainability in funding, research talent, capacity and delivery of world-class outputs

CAS researchers work extensively with public and private sector clients to deliver robotics solutions across six key areas:

Image Source

More Information

8.2 New Jersey Institute of Technology

Newark, New Jersey USA

Image Source

NJIT has been recognized for its commitment to academic excellence for more than 130 years. As one of the nation's leading polytechnic universities, NJIT earned national prominence by developing relevant academic programs. Over the past decade, the size of the campus has doubled, which involved investing millions of dollars in major new campus and research facilities designed to give students the edge they need in today's competitive high-tech marketplace.

NJIT conducts more than $130 million in research and has hired more than 100 new faculty members over the past six years. Sixty research centers and specialized laboratories encourage interdisciplinary collaboration in teaching and research. For example, the Cybersecurity Research Center was established to protect data from inception to end use, beginning with the production of software to manage the data to its remote storage in the cloud. Broadly, the Center identifies systemic weaknesses that make cyber systems vulnerable to attack, designs systems to make them more secure, and hardens cyber infrastructure that has already been deployed. The current student to faculty ratio is 17:1.

More Information

9. SOME SYSTEMS ENGINEERING-RELEVANT WEBSITES

The Agile Alliance

The Agile Alliance is a nonprofit member organization dedicated to promoting the concepts of Agile Software Development as outlined in the Agile Manifesto. With more than 60,000 members and subscribers around the globe, the Agile Alliance is driven by the principles of Agile methodologies and the value delivered to developers, organizations, and end users.

https://www.agilealliance.org/

The Scrum Alliance

Scrum Alliance is an organization that provides education, resources, and support to practitioners of Scrum and Agile. Scrum Alliance offers advocacy, community engagement, research, networking and a focus on organizational change that is transforming the world of work all over the globe. The Scrum Alliance is driven by the members that make up the global community, along with everyone who seeks to achieve true work/life balance.

https://www.scrumalliance.org/

GlobalSpec

One of the engineering websites targeted more towards engineers already in the field, GlobalSpec offers a useful product and supplier list for any engineering project. Product alert emails are easily configured. The website also has a community section where engineers ask questions and collaborate to help innovate engineering technologies. Also, there is an interesting editorial answering important questions from "What is the Real Cost of an Industrial Robot" to "What Chemical Lab Equipment is Most Popular."

http://www.globalspec.com/

10. STANDARDS AND GUIDES

10.1 Peak Operational Concept Description (OCD) Standard Updated

ANSI/AIAA G-043B-2018 Guide to the Preparation of Operational Concept Documents, an update to the 2012 edition, has been released.

A recognized systems engineering best practice is the early development of operational concepts that describe the intended use of a system, in terms of intended users, uses, how it is intended to be used, and the external conditions expected or intended during use of the system. Those operational concepts are documented in one or more operational concept documents or descriptions. This American Institute of Aeronautics and Astronautics (AIAA) Guide describes which types of information are most relevant, their purpose, and who should participate in the operational concept development effort. It also provides advice regarding procedures for generation of the information and how to document it.

Alternative names for an OCD are CONEMP (a very thin and early OCD), CONUSE, OpsCon, Statement of Operating Intent (SoI), and in the regulated healthcare sector, just “intended use”. The term CONOPS was also misused as a synonym for OCD for a period due to an accident that occurred in 1998. This misnaming has now been mostly cleaned up in the various systems engineering standards and handbooks.

Access the standard here.

10.2 Object Management Group Chairs Build Momentum for Technology Standards

Chairs of the Object Management Group® (OMG®) Task Forces (TFs) and Special Interest Groups (SIGs) presented updates about their standards work during a plenary session on the last day of the OMG quarterly membership meeting, which took place from March 18-22, 2019 in Reston, Virginia.

The following Chairs reported on their groups' accomplishments:

- Claude Baudoin, co-chair of the Business Modeling & Integration Domain Task Force (BMI DTF) and Owner and Principal consultant at cébé IT and Knowledge Management: “The Business Modeling & Integration BMI DTF was pleased to hear that the three submitter teams that responded to the Business Architecture Core Metamodel (BACM) RFP have agreed to work together toward a single joint submission. In addition, an RFP for a Standard Business Report Model (SBRM) is under preparation.”

- Sanford Friedenthal, chair of the Systems Engineering Domain Special Interest Group (SE DSIG) and Consultant at SAF Consulting: “The SE DSIG included a presentation and demonstration by the SysML® v2 Submission Team (SST) co-leads on their progress since the last SE DSIG meetings, along with a series of presentations from Boeing, Ford, BAE Systems, and IBM on their application of model-based practices. OMG hosted the Meet & Greet: SysML Update on Monday evening, which included short presentations on SysML v1.6, and plans for SysML v1.7 and SysML v2, and an opportunity to meet with SysML end users and vendors."

- Bart McGlothin, chair of the Retail Domain Task Force and Solution Architect at Cisco: "Great discussions with great people, issued a new RFP for Digital Receipt, and updated our roadmap with new projects in: security, cloud, and location."

- Robert Martin, chair of the Structured Assurance Case Metamodel™ (SACM™) 2.0 Finalization Task Force and Senior Principal Engineer at MITRE: "The SACM™ 2.1 RTF report, which includes a set of graphical notations that represent the full set of argumentation concepts in SACM, was approved by the Architecture Board."

- Dr. Jeffrey E. Smith, co-chair of the Analysis and Design Platform Task Force (ADTF) and Chief Systems Engineer at the Multi Agency Collaboration Environment: "The ADTF passed motions on revising the API4KB, MVF, Safety and Reliability Profile, IMM™ and MARTE 2.0 RFI submissions as well as had presentations about: Onto State Machine Modeling, SysML V2 Submission Overview and Status, (Onto) Behavior Verification, UML® 3.0 - Is it time to refactor the specification, Review of Current SMIF Submission and Process Presentation and on the Thales MARTE 2.0 RFI responses."

- Charlotte Wales, co-chair of the Middleware and Related Services (MARS) PTF and Lead Software Engineer at The MITRE Corporation: "MARS had a very active meeting with several processes reporting progress, excellent collaboration with other OMG groups (e.g., Finance DTF and Blockchain PSIG), and initial interactions with The Open Group's FACE™ Consortium."

- Matt Wilson, co-chair of Consultation, Command, Control, Communications and Intelligence Domain Task Force (C4I DTF) and Vice President and Principal Engineer at SimVentions: "The C4I DTF is working on a FACE Profile for UAF®, submissions for DDS™ Monitoring and C2INav, and drafting requirements for other C4I technologies."

11. SOME DEFINITIONS TO CLOSE ON

Editor's note: These definitions are provided in our Feature Article for this month by Warren K. Vaneman.

Model Based Systems Engineering (MBSE): The formalized application of modeling (static and dynamic) to support system design and analysis, throughout all phases of the system lifecycle, through the collection of modeling languages, structures, model-based processes, and presentation frameworks used to support the discipline of systems engineering in a model-based or model-driven context. Model-Based Systems Engineering is not a new discipline designed to supersede traditional systems engineering, but a new way to address systems engineering problems such as architecting, requirements management, risk management, and analytical methods such as discrete event simulation, systems dynamics modeling, and dynamic programming.

Modeling Languages: Serve as the basis of tools, and enable the development of system models. Modeling languages are based on a logical construct (visual representation) and/or an ontology.

Ontology: An ontology is a collection of standardized, defined terms and concepts and the relationships among the terms and concepts, to capture the information that describes the physical, functional, performance, and programmatic aspects of a system. The most common format for an ontology is the Entity, Relationship, and Attribute (ERA) data schema. Each entity has a defined relationship allowing it to represent system complexity, and may include multiple attributes to capture all of the dimensions of the system.

Structure: Structure defines the relationships between the system entities, establishes concordance within the model, and allows for the emergence of system behaviors and performance characterizations within the model. The relationships between the principal entities define structure, address complexity, and ensure system traceability across the model.

Model-Based Processes: Model-Based Processes provide the analytical framework to conduct the analysis of the system virtually defined in the model. The model-based processes may be traditional systems engineering processes such as requirements management, risk management, or analytical methods such as discrete event simulation, systems dynamics modeling, and dynamic programming. Model-based processes offer different analytical approaches and are used to address the various challenges throughout the system's lifecycle.

Presentation Frameworks: Provide the framework for the logical constructs of the system data in visualization models that are appropriate for the given stakeholders. These visualization models take the form of traditional systems engineering models. These individual models are often grouped into frameworks that provide the standard views and descriptions of the models, and the standard data structure of architecture models.

Model-Based Systems Engineering Tools: General-purpose software products that use modeling languages, and support the specification, design, analysis, validation, and verification of complex system representations. Some tools are treated as synonymous with MBSE. Tools are generally popular for a period, and then are superseded by the "next best idea;" however, the MBSE concepts presented in Vaneman's article are meant to be fundamental and transcendent tools.

Concordance: The ability to represent a single entity such that data in one view, or level of abstraction, match the data in another view, or level of abstraction, when talking about the exact same thing. This singular representation of each entity allows the system to be explored from the various engineering and programmatic perspectives (viewpoints).

Viewpoint: A viewpoint describes data drawn from one or more perspectives and organized in a particular way useful to management decision-making. The compilation of viewpoints (e.g. capability, operational, system, programmatic viewpoints) represents the entire system, where the system can be explored as a whole, or from a single perspective.

MBSE Environment: The MBSE environment may consists of single or multiple tools and data repositories. Regardless, whether the environment is a single tool and data repository, or is composed of multiple tools and an integrated data repository, the four components must be implemented for the MBSE environment to be fully effective.

Attribute: An attribute is an inherent characteristic or quality that further describes an entity. The ERA approach allows for the efficient use of entities due to the attributes and relationships defined. Each entity has a defined relationship allowing it to represent system complexity, and may include multiple attributes to capture all of the dimensions of the system. System entities also have attributes such as: described by physical dimensions; satisfy capabilities; perform functions; exhibit behavior; have cost; are governed by a schedule; and, may have risks, just to name the most common. Essentially, each modeled entity should fully represent their corresponding system element. In an MBSE environment, each entity is ideally represented in the model only once.

Lifecycle Modeling Language: An example of entity, relationship, and attribute (ERA) based language. It was developed as an approach for incorporating visualization models and corresponding ontologies within the same framework.

Building Blocks: Systems consist of "building blocks" and their relationships to each other that allow them to come together in a designed form that satisfies the desired capabilities and functionality.

12. CONFERENCES AND MEETINGS

For more information on systems engineering related conferences and meetings, please go to our website.

The featured event for this edition is:

International Conference on Industrial, Production & Systems Engineering (ICIPSE)

02 June 2019, Chennai (India)

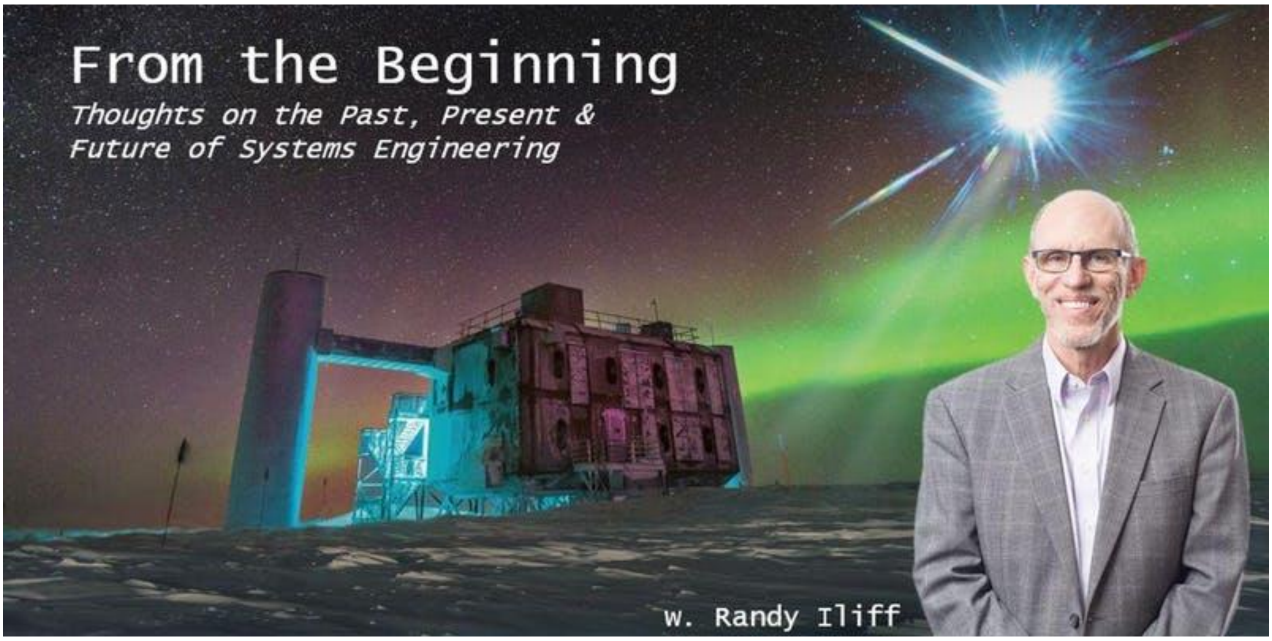
The International Conference provides a world-class platform for delegates to present and discuss all the latest research and results of scientists related Industrial, Production & Systems Engineering. This conference provides opportunities for delegates to exchange new ideas and application experiences face-to-face, to establish business or research relations and to find global partners for future collaboration. The organizing committee of the conference is pleased to invite prospective authors to submit their original manuscripts to ICIPSE 2019 at the following link: http://iraj.in/Conference2019/6/Chennai/1/ICIPSE/submission.php.

- Register for the conference here.

13. PPI AND CTI NEWS

13.1 PPI's Randall (Randy) Iliff Presents

PPI Principal Consultant Randy Iliff shared his views on the past, present and future of systems engineering with members of the Chicagoland Chapter of INCOSE on Thursday 16 May, 2019. The event was broadcast to multiple locations.



13.2 Bijan Elahi Presents Tutorial of Introduction to Medical Device Safety Risk Management in Compliance with ISO 14971

PPI's Bijan Elahi (MSEE, BS AeroE) will be delivering a tutorial on an Introduction to Medical Device Safety Risk Management in Compliance with ISO 14971. Mr. Elahi is a world-class expert in risk management and systems engineering. He is a winner of the Educator of the Year Award from the International System Safety Society for outstanding performance in system safety education.

This half-day takes place on the 22nd May and covers core concepts for medical device risk management, as well as familiarizing individuals with the requirements of ISO 14971. It is suitable for those who will be performing risk management, and for those who require a general understanding of risk management.

13.3 PPI sponsors Joseense Amateur Tennis Association (AJTA) tournament in Brazil

PPI had the opportunity and privilege to be part of and to sponsor the Joseense Amateur Tennis Association (AJTA) tournament in São Paulo, Brazil. AJTA is a tennis association founded in 2019 with the purpose of encouraging and expanding the practice of tennis in the outlying areas of São José dos Campos.

14. PPI AND CTI EVENTS

On-site systems engineering training is being delivered worldwide throughout the year. Below is an overview of public courses. For a full public training course schedule, please visit https://www.ppi-int.com/course-schedule/

Systems Engineering 5-Day Courses

Upcoming locations include:

- Melbourne, Australia (P006-777)

  08 Jul – 12 Jul 2019

Requirements Analysis and Specification Writing 5-Day Courses

Upcoming locations include:

- Bristol, United Kingdom (P007-492)

  01 Jul – 05 Jul 2019

Systems Engineering Management 5-Day Courses

Upcoming locations include:

- Washington, D.C., United States of America (P1135-170)

  09 Sep – 13 Sep 2019

Systems Engineering Overview 3-Day Courses

Upcoming locations include:

- Chantilly, Virginia, United States of America (P884-15)

  09 Dec – 11 Dec 2019

Requirements, OCD and CONOPS in Military Capability Development 5-Day Courses

Upcoming locations include:

- Pretoria, South Africa (P958-58)

  29 Jul – 02 Aug 2019

Engineering Successful Infrastructure Systems (ESIS5D)

Upcoming locations include:

- Amsterdam, the Netherlands (P2005-2)

  23 Sep – 27 Sep 2019

Architectural Design 5-Day Course

Upcoming locations include:

- Pretoria, South Africa (P1768-19)

  15 July – 19 July 2019

CSEP Preparation 5-Day Courses (Presented by Certification Training International, a PPI company)

Upcoming locations include:

- Austin, Texas, United States of America (C002-86)

  12 Aug – 16 Aug 2019

Medical Device Risk Management 3-Day Course

Upcoming locations include:

- San Francisco, California, United States of America (P1848-4)

  26 Aug – 28 Aug 2019

Other training courses available **on-site** only include:

- Project Risk and Opportunity Management 3-Day
- Managing Technical Projects 2-Day
- Integrated Product Teams 2-Day
- Software Engineering 5-Day

15. UPCOMING PPI PARTICIPATION IN PROFESSIONAL CONFERENCES

PPI will be participating in the following upcoming events. We support the events that we are sponsoring and look forward to meeting old friends and making new friends at the events at which we will be exhibiting.

The INCOSE International Symposium 2019

(Exhibiting)

Date: 20 – 25 July, 2019

Location: Orlando, Florida, USA

EnergyTech Conference 2019

(Exhibiting)

Date: 21 – 25 October, 2019

Location: Cleveland, Ohio, USA

The INCOSE International Symposium 2020

(Exhibiting)

Date: 18 – 23 July, 2020

Location: Cape Town, South Africa

Asia Oceania Systems Engineering Conference 2019

(Exhibiting)

Date: 17 – 18 October, 2019

Location: Bangalore, India

Kind regards from the PPI SyEN team:

Robert Halligan, Editor-in-Chief, email: rhalligan@ppi-int.com

Ralph Young, Editor, email: ryoung@ppi-int.com

René King, Managing Editor, email: rking@ppi-int.com

Project Performance International

2 Parkgate Drive, Ringwood, Vic 3134 Australia

Tel: +61 3 9876 7345

Fax: +61 3 9876 2664

Tel Brasil: +55 12 9 9780 3490

Tel UK: +44 20 3608 6754

Tel USA: +1 888 772 5174

Tel China: +86 188 5117 2867

Web: www.ppi-int.com

Email: contact@ppi-int.com

Tell us what you think of PPI SyEN. Email us at syen@ppi-int.info.