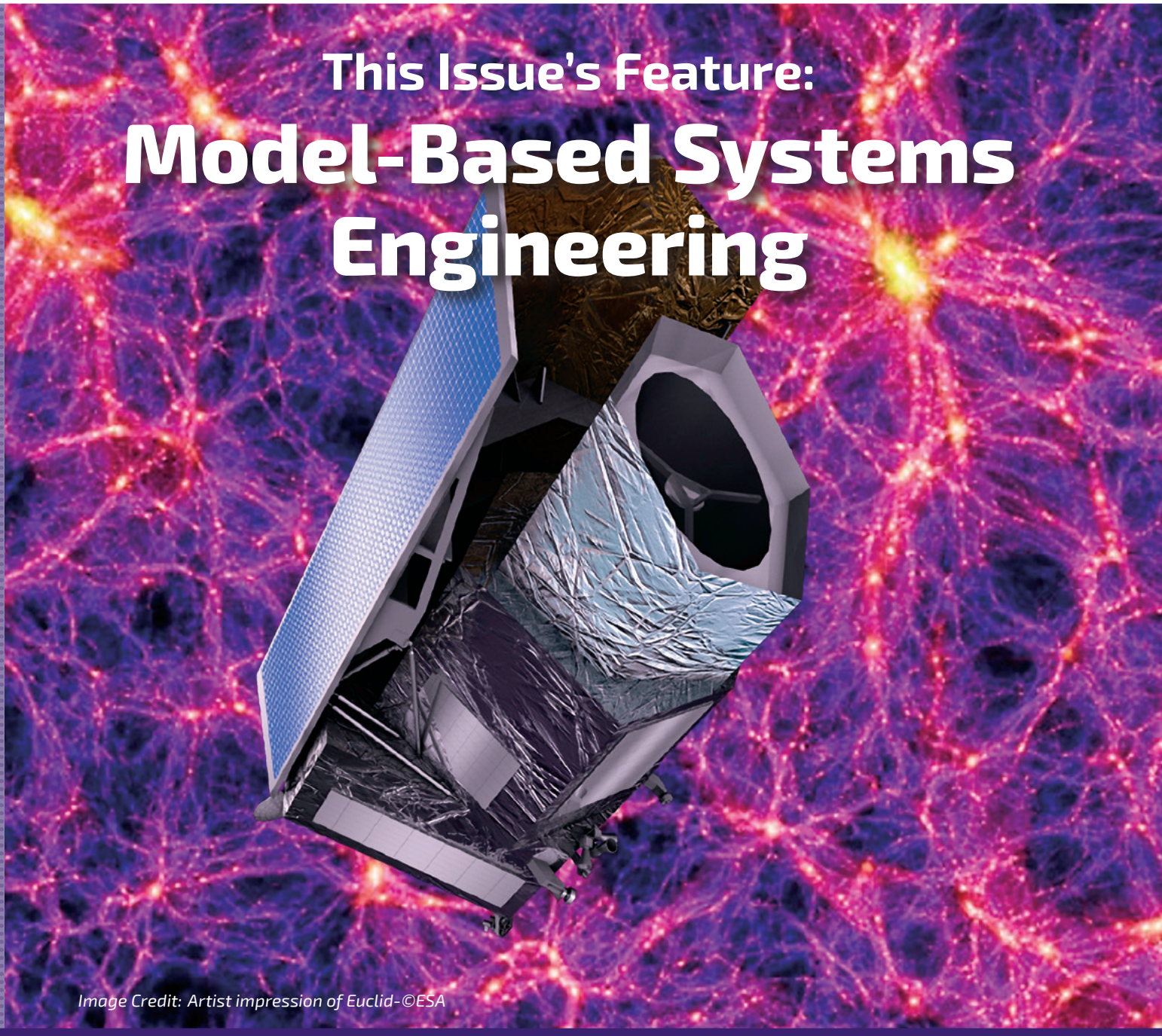
INSIGHT



This Issue's Feature:

Model-Based Systems Engineering

Image Credit: Artist impression of Euclid-©ESA

AUGUUST 2015
VOLUME 18 / ISSUE 2

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

INCOSE®

STEVENS SYSTEMS ENGINEERING

Graduate Programs Online, On Campus or On-Site

Shape the Future of Systems Engineering.

- Relevant curriculum prepares graduates to become technical leaders who successfully design, navigate and manage complex systems
- World-renowned and industry-connected faculty who are thought leaders in the field
- Home to Systems Engineering Research Center (SERC), the National Center of Excellence in Systems Engineering funded by the Department of Defense
- Located in Hoboken, New Jersey –10 minutes from NYC, the world's financial hub and home to thousands of tech companies

EXPLORE ALL SYSTEMS GRADUATE PROGRAMS–
visit stevens.edu/systems

STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY®

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

AUGUST 2015 VOLUME 18 / ISSUE 2

What's inside this issue

FROM THE EDITOR-IN-CHIEF 7

SPECIAL FEATURE 9

Model Based Systems Engineering Editorial 9

Current Modeling Trends in Systems Engineering 10

MBSE Has a Good Start; Requires More Work for Sufficient Support of Systems Engineering Activities through Models 14

MBSE in Rail Transportation – Product Families and Product Lines 19

MBSE Implementation across Diverse Domains 22

Modular Applications Functions – A Westinghouse MBSE Tool 26

Academia as an Educator in the World of MBSE 30

Polarsys: A New Collaborative Ecosystem for Open Source Solutions for Systems Engineering Driven by Major Industry Players 35

Evolving SysML and the System Modeling Environment to Support MBSE 39

Integration of MBSE Artifacts Using OSLC 42

Untangling the Digital Thread: The Challenge and Promise of Model-Based Engineering in Defense Acquisition 45

INSIGHT OP ED 56

MBSE in Perspective 56

BOOK REVIEW 58

Toward a Larger View of Negotiation 58

VISION 59

Needed: Practitioner Attention to Systems Engineering Delivery of Sustainable Value 59

About This Publication

INFORMATION ABOUT INCOSE

INCOSE's membership extends to over 10, 000 individual members and almost 100 corporations, government entities and academic institutions. Its mission is to share, promote, and advance the best of systems engineering from across the globe for the benefit of humanity and the planet. INCOSE charters chapters worldwide, includes a corporate advisory board, and is led by elected officers and directors.

For more information, click here:
The International Council on Systems Engineering
(www.incose.org)

OVERVIEW

INSIGHT is the magazine of the International Council on Systems Engineering. It is published four times per year and features informative articles dedicated to advancing the state of practice in systems engineering and to close the gap with the state of the art. ***INSIGHT*** delivers practical information on current hot topics, implementations, and best practices, written in applications-driven style. There is an emphasis on practical applications, tutorials, guides, and case studies that result in successful outcomes. Explicitly identified opinion pieces, book reviews, and technology roadmapping complement articles to stimulate advancing the state of practice. ***INSIGHT*** is dedicated to advancing the INCOSE objectives of impactful products and accelerating the transformation of systems engineering to a model-based discipline.
Topics to be covered include resilient systems, model-based systems engineering, commercial-driven transformational systems engineering, natural systems, agile security, systems of systems, and cyber-physical systems across disciplines and domains of interest to the constituent groups in the systems engineering community: industry, government, and academia. Advances in practice often come from lateral connections of information dissemination across disciplines and domains. ***INSIGHT*** will track advances in the state of the art with follow-up, practically written articles to more rapidly disseminate knowledge to stimulate practice throughout the community.

EDITORIAL BOARD AND STAFF

Editor-In-Chief William Miller
insight@incose.org +1 908-759-7110

Assistant Editor Lisa Hoverman
lisa.hoverman@incose.org

Theme Editor Joachim Fuchs
joachim.fuchs@esa.int

Senior Advertising Account Manager Joseph Tomaszewski
jtomaszews@wiley.org +1 908 514 0776

Layout and Design Chuck Eng
chuck.eng@comcast.net

Member Services INCOSE Administrative Office
info@incose.org +1 858 541-1725

2015 INCOSE BOARD OF DIRECTORS

Officers

President: David Long, *Vitech Corporation*
President-Elect: Alan D. Harding, *BAE Systems*
Secretary: Tina Srivastava, *Raytheon*
Treasurer: Jen Narkevicius, *Jenius Solutions*

At-Large Directors

Academic Matters: Art Pyster, *Stevens Institute of Technology*
Communications: Cecilia Haskins, *Norwegian University of Science and Technology*
Outreach: Paul Davies
Strategy: Jonas Andersson, *Decisionware AB*
Americas Sector: Barclay Brown, *IBM*
EMEA Sector: Jean-Claude Roussel, *Airbus Group*
Asia-Oceania Sector: Kerry Lunney, *Thales*
Chief Information Officer (CIO)**:** Bill Chown, *Mentor Graphics*
Technical Director: Paul Schreinemakers, *SEPIAdvies BV/ How2SE*
Corporate Advisory Board Chair: Max Berthold, *Swedish Defence Materiel Administration*
CAB Co-chair: Bob Swarz, *Worcester Polytechnic Institute*
Deputy Technical Director: Mike Celentano, *Roche Diagnostics*
Managing Executive: Holly Witte, *Universal Management Services, LLC*

PERMISSIONS

*** PLEASE NOTE: If the links highlighted here do not take you to those web sites, please copy and paste address in your browser.**

Permission to reproduce Wiley journal Content:
Requests to reproduce material from John Wiley & Sons publications are being handled through the RightsLink® automated permissions service.

Simply follow the steps below to obtain permission via the Rightslink® system:

- Locate the article you wish to reproduce on Wiley Online Library (http://onlinelibrary.wiley.com)
- Click on the 'Request Permissions' link, under the ‹ARTICLE TOOLS› menu on the abstract page (also available from Table of Contents or Search Results)
- Follow the online instructions and select your requirements from the drop down options and click on 'quick price' to get a quote
- Create a RightsLink® account to complete your transaction (and pay, where applicable)
- Read and accept our Terms & Conditions and download your license
- For any technical queries please contact customercare@copyright.com
- For further information and to view a Rightslink® demo please visit www.wiley.com and select Rights & Permissions.

AUTHORS – If you wish to reuse your own article (or an amended version of it) in a new publication of which you are the author, editor or co-editor, prior permission is not required (with the usual acknowledgements). However, a formal grant of license can be downloaded free of charge from RightsLink if required.

Photocopying
Teaching institutions with a current paid subscription to the journal may make multiple copies for teaching purposes without charge, provided such copies are not resold or copied. In all other cases, permission should be obtained from a reproduction rights organisation (see below) or directly from RightsLink®.

Copyright Licensing Agency (CLA)
Institutions based in the UK with a valid photocopying and/or digital license with the Copyright Licensing Agency may copy excerpts from Wiley books and journals under the terms of their license. For further information go to CLA.

Copyright Clearance Center (CCC)
Institutions based in the US with a valid photocopying and/or digital license with the Copyright Clearance Center may copy excerpts from Wiley books and journals under the terms of their license, please go to CCC.

Other Territories: Please contact your local reproduction rights organisation. For further information please visit www.wiley.com and select Rights & Permissions.
If you have any questions about the permitted uses of a specific article, please contact us.

Permissions Department
John Wiley & Sons Ltd.
The Atrium,
Southern Gate,
Chichester
West Sussex, PO19 8SQ
UK
Email: Permissions@wiley.com
Fax: 44 (0) 1243 770620
or

Permissions Department
John Wiley & Sons Inc.
111 River Street MS 4-02
Hoboken, NJ 07030-5774
USA
Email: Permissions@wiley.com
Fax: (201) 748-6008

ARTICLE SUBMISSION

INSIGHT@incose.org

Publication Schedule. ***INSIGHT*** is published four times per year. Issue and article submission deadlines are as follows:

- October 2015 issue – 5 August
- December 2015 issue – 5 October
- April 2016 issue – 5 February
- July 2016 issue – 5 June

For further information on submissions and issue themes, visit the INCOSE website: www.incose.org

© 2015 Copyright Notice.
Unless otherwise noted, the entire contents are copyrighted by INCOSE and may not be reproduced in whole or in part without written permission by INCOSE. Permission is given for use of up to three paragraphs as long as full credit is provided. The opinions expressed in ***INSIGHT*** are those of the authors and advertisers and do not necessarily reflect the positions of the editorial staff or the International Council on Systems Engineering.
ISSN 2156-485X; (print) ISSN 2156-4868 (online)

PENN STATE | ONLINE

Deandrea Thompson
Systems Engineering Graduate

Earn a Master's Degree in Engineering—Entirely Online

Systems Engineering
Software Engineering
Engineering Management

- Five- or seven-week courses over six semesters
- GREs not required for admission
- Finish in as little as two years

Recognized for the fourth year in a row by the ***U.S. News & World Report*** Best Online Programs Rankings for 2015.

BEST ONLINE PROGRAMS
U.S.News & WORLD REPORT
GRAD ENGINEERING
2015

PENNSTATE
World Campus
1855

Apply Now: worldcampus.psu.edu/INCOSE

U.Ed.OUT 15-0390/14-WC-1062bkh/smb

ADVERTISE

Readership
INSIGHT reaches over 10, 000 individual members and uncounted employees and students of almost 100 CAB organizations worldwide. Readership includes engineers, manufacturers/purchasers, scientists, research & development processionals, presidents and CEOs, students and other professionals in systems engineering.

Issuance	Circulation
2015, Vol 18, 4 Issues	100% Paid

Contact us for Advertising and Corporate Sales Services
We have a complete range of advertising and publishing solutions professionally managed within our global team. From traditional print-based solutions to cutting-edge online technology the Wiley-Blackwell corporate sales service is your connection to minds that matter. For an overview of all our services please browse our site which is located under the Resources section. Contact our corporate sales team today to discuss the range of services available:

- Print advertising for non-US journals
- Email Table of Contents Sponsorship
- Reprints
- Supplement and sponsorship opportunities
- Books
- Custom Projects
- Online advertising

Click on the option below to email your enquiry to your nearest office:

- Asia & Australia corporatesalesaustralia@wiley.com
- Europe, Middle East & Africa (EMEA) corporatesaleseurope@wiley.com
- Japan corporatesalesjapan@wiley.com
- Korea corporatesaleskorea@wiley.com

USA (also Canada, and South/Central America):

- Healthcare Advertising corporatesalesusa@wiley.com
- Science Advertising Ads_sciences@wiley.com
- Reprints Commercialreprints@wiley.com
- Supplements, Sponsorship, Books and Custom Projects busdev@wiley.com

Or please contact:
Roland Espinosa, Senior Account Manager
Print & E Media Advertising
PHONE: 201-748-6819
E-MAIL: respinosa@wiley.com

CONTACT

Questions or comments concerning:

Submissions, Editorial Policy, or Publication Management
Please contact: William Miller, Editor-in-Chief
insight@incose.org

Advertising ***Please contact:***
Roland Espinosa, Senior Account Manager
Print & E Media Advertising
PHONE: 201-748-6819

E-MAIL: respinosa@wiley.com

Member Services
Please contact: info@incose.org

CORPORATE ADVISORY BOARD — MEMBER COMPANIES

Aerospace Corporation, The
Airbus Defense and Space
Airbus Group
Alliant Techsystems
AM General LLC
Analytic Services–Applied Systems Thinking Institute
ATKINS
Autoliv
Aviation Industry Corporation of China
BAE Systems
Bechtel
Boeing Company
Booz Allen Hamilton Inc.
C.S. Draper Laboratory, Inc.
Carnegie Mellon University Software Engineering Institute
Cranfield University
Cummins, Inc.
Defense Acquisition University
Deloitte
Drexel University
Engility
Exelis
Federal Aviation Administration (U.S.)
Ford Motor Company
General Dynamics
General Electric
General Motors
George Mason University
Georgia Institute of Technology
Honeywell International
Huntington Ingalls Industries
IBM Corporation
iTiD Consulting, Ltd
Jet Propulsion Laboratory
Johns Hopkins University
KEIO University
L-3 Communications
Leidos
Lockheed Martin Corporation
Los Alamos National Laboratory
ManTech International Corporation
MAP systeme
Massachusetts Institute of Technology
Medtronic, Inc.
Missouri University of Science & Technology
MITRE Corporation, The
Nanyang Technological University
National Aeronautics and Space Administration
National Reconnaissance Office
National Security Agency Enterprise Systems
National University of Singapore
Naval Postgraduate School
Nissan Motor Co, Ltd
Northrop Grumman Corporation
PA Consulting
Pacific Northwest National Laboratory
Procter & Gamble
Project Performance International
PTC
Raytheon Corporation
Rockwell Collins, Inc.
Rolls-Royce
Saab AB
Sandia National Laboratories
Scitor Corporation
Siemens
Sotera Defense Solutions, Inc.
Stellar Solutions
Stevens Institute of Technology
Swedish Defence Materiel Administration
Systems Planning and Analysis
Tetra Pak
Thales
TNO Technical Sciences
UK MoD
United Technologies Corporation
University of Arkansas
University of Maryland
University of Maryland, Baltimore County
University of New South Wales, Canberra, The
University of Southern California
US Army ARDEC
US Army TARDEC
US Department of Defense
Vencore
Virginia Tech
Vitech Corporation
Volvo Construction Equipment
Woodward Inc
Worcester Polytechnic Institute–WPI

FROM THE EDITOR-IN-CHIEF

William Miller, insight@incose.org

INSIGHT is published in cooperation with John Wiley & Sons as a magazine for systems engineering practitioners, providing informative articles to advance the state of the practice of systems engineering. We also have the objective to accelerate the dissemination of knowledge to close the gap between the state of practice and the state of the art as captured in *Systems Engineering*, the Journal of INCOSE, also published by Wiley. News items and events are now reported in the quarterly eNewsletter published a month ahead of *INSIGHT*.

We received very positive feedback from our readers for the *INSIGHT* re-launch with the April issue on *Resilient Systems* and thank theme editor Scott Jackson and his editorial team from the Resilient Systems Working Group for their dedication and commitment to produce this work: Chair John Brtis, Paul Cleary, Katri Hakola, Azad Madni, Chandru Mirchandani, and David Yarbrough.

The August *INSIGHT* theme is *Model-Based Systems Engineering* (*MBSE*), looking at the challenges and progress since the Special Feature on *MBSE*: *The New Paradigm* in the December 2009 *INSIGHT*. The cover is an artist's impression of the Euclid spacecraft being developed by the European Space Agency (ESA) and named after the "father of geometry" to gain understanding of the geometry of dark energy and dark matter by measuring the acceleration of the universe. ESA did the requirements engineering for Euclid with SysML. This issue is also a companion to the December 2014 *INSIGHT* Special Feature on *Model-Based Conceptual Design*. We thank theme editor Joachim Fuchs and his team of authors for their contributions. We also extend beyond MBSE to Model-Based Engineering (MBE) with an additional article by Tim West and Art Pyster on "digital threads" as a framework to merge detailed physics-based design models with MBSE's conceptual and top-level architectural models, with the objective of a single authoritative representation of the system.

We introduce our first *Op Ed* column, the intent of which is to provide thoughtful, explicit opinion pieces by different columnists assessing the state of practice and identifying gaps and objectives separate from the more factually based articles reporting on systems engineering practice. The inaugural Op Ed by Jack Ring and Azad Madni challenges the systems engineering community to reflect on lessons learned to date and what changes are called for MBSE to be necessary, sufficient, and efficient. One of the significant challenges is making the case for MBSE in our work environments because of the upfront investment and training. We therefore recommend the book review by Denise Howard on *Getting to Yes with Yourself: And Other Worthy Opponents*.

We thank Rick Dove for his "call to arms" to INCOSE Working Groups to engage and contribute in next-generation agile security, soliciting contributions for the July 2016 *INSIGHT* issue on system security as a backdrop for a conversation on sustainable value. Agile security should be a resilient, emergent characteristic of systems and not an add-on to the design.

We are still in the process of establishing an editorial board to provide independent reviews of articles and an advisory board representing the constituent groups within INCOSE including Technical Operations, Corporate Advisory Board, Academic Council, Industry Outreach Board, Sectors and Fellows.

Feedback from readers is critical to the quality of *INSIGHT*. In addition to the annual Reader' Choice Survey, we encourage Letters to the Editor at insight@incose.org. Please include *Letter to the Editor* in the subject line. We hope you continue to find *INSIGHT*, the practitioners' magazine for systems engineers, informative and relevant. ■

The INCOSE Foundation Where You Make the Difference

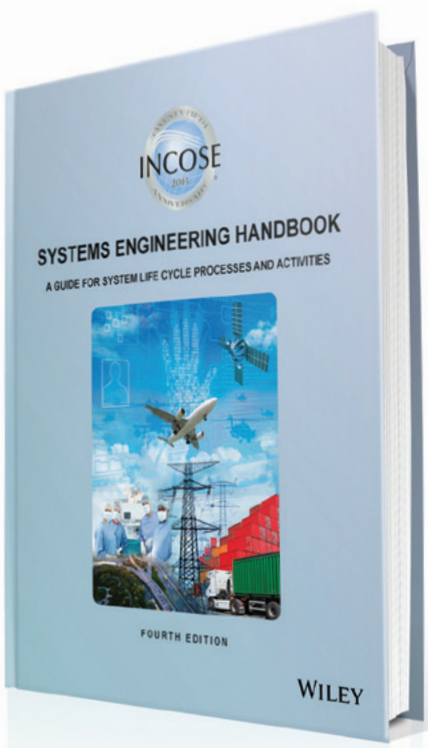
INCOSE FOUNDATION ™

Make a bright future possible through INCOSE Foundation grants and scholarships.

Contact Holly Witte, Foundation Managing Director, for more information: holly.witte@incose.org

Have you remembered the Foundation in your will? Many companies match gifts. Please ask your company to match your gift to the INCOSE Foundation. We accept all major credit cards.

Order the Latest INCOSE Handbook Now!



The objective of the *International Council on Systems Engineering (INCOSE) Systems Engineering Handbook* is to describe key process activities performed by systems engineers and other engineering professionals throughout the life cycle of a system. The book covers a wide range of fundamental system concepts that broaden the thinking of the systems engineering practitioner, such as system thinking, system science, life cycle management, specialty engineering, system of systems, and agile and iterative methods. This book also defines the discipline and practice of systems engineering for students and practicing professionals alike, providing an authoritative reference that is acknowledged worldwide.

Edited by David D. Walden, ESEP; Garry J. Roedler, ESEP; Kevin J. Forsberg, ESEP; R. Douglas Hamelin; and Thomas M. Shortell, CSEP

ISBN: 978-1-118-99940-0
304 pages • July 2015 • $80.00

INCOSE members receive
55% off the list price!
Pre-order now on
www.wiley.com
with the special
INCOSE member discount code!

To pre-order your copy now with the INCOSE member discount code, please login on the INCOSE website and go to the INCOSE Store where you will find the SE Handbook Member Discount Code Download for Pre-ordering on ww.wiley.com. The code and link will be available in your Digital Library in your Profile Home when you "purchase" the code download.

The latest edition of the *INCOSE Systems Engineering Handbook*:

- Is consistent with ISO/IEC/IEEE 15288:2015 Systems and software engineering—System life cycle processes and the Guide to the Systems Engineering Body of Knowledge (SEBoK)
- Has been updated to include the latest concepts of the INCOSE working groups
- Is the body of knowledge for the INCOSE Certification Process

This book is ideal for any engineering professional who has an interest in or needs to apply systems engineering practices. This includes the experienced systems engineer who needs a convenient reference, a product engineer or engineer in another discipline who needs to perform systems engineering, a new systems engineer, or anyone interested in learning more about systems engineering.

www.wiley.com

WILEY

AUGUST 2015
VOLUME 18 / ISSUE 2

Model Based Systems Engineering Editorial

Joachim Fuchs, joachim.fuchs@esa.int

INTRODUCTION

The *INSIGHT* edition of December 2009 (Vol. 12 Issue 4) was a Special Feature dedicated to Model-Based Systems Engineering: *The New Paradigm*. It described a number of projects and approaches, although most of them at the level of initiatives, demonstrators, and proofs of concept. Few "real-life" projects covered the full system lifecycle supported by a consistent model-based approach, but in many areas, there was the promise of expected benefits.

Then, as now, there was a healthy discussion on the value and limitations of standardization of established languages and methods (SysML) to support Model-Based Systems Engineering (MBSE).

TAKING STOCK

With the passing of five years since that publication, it seems appropriate to look at the achievements since. Systems engineers show increased understanding of MBSE in many areas, with the emphasis shifting from focusing on the problem of the "system description" to the interoperability of data and information, of managing processes, and addressing the interdisciplinary aspects as the basis underlying an industrial process. The discussion about MBSE comes back to the underlying systems engineering process, and tries to focus on the added value of a model-support for it: the models as a means towards the goal of systems engineering. Even in the INCOSE Vision 2025, MBSE is not an explicit objective any more, but the shift is towards the overall systems engineering processes, supported by an integrated, digital engineering environment.

The participation in MBSE workshops held at the INCOSE International Workshops and Symposia has nevertheless shown enormous growth, with presentations on implementation experiences in larger scale projects and processes. Initiatives establish the link between the systems engineering process supported by models, and the model and data management systems.

The articles in this issue give a diverse image of the achievements in the intervening five years. There has been clear progress towards the use of model-based support to systems engineering functions, such as described in the article by Harald Eisenmann or Ron Carson in the aerospace domain, by Marco Ferrogallini for trains, or Gary McWilliams for power plants.

The contributions clearly show the concept of MBSE expanded to more levels in systems engineering, and in addition, the interdisciplinary aspects and lifecycle aspects are better understood. However, none of the examples described yet reaches a clear status of company-wide application of specific aspects of MBSE. It also appears that it is the wrong objective to "introduce" MBSE, because there is no clearly formulated "achievement" criteria and benchmarking available. It will remain a continuing process, adapting existing processes, and therefore achieving broader acceptance.

It is certainly helpful that professional training is preparing the new generation of systems and other engineers to become familiar with the methods and possible tools of the modeling techniques, as Russell Peak, Leon McGinnis, and Chris Paredis point out in their article. The training aids in advancing systems engineering to become a model-based discipline, whilst recognizing that there remains additional work at the level of systems engineering as a stand-alone discipline. In any case, this training will bring the approach into the industries, and help pave the way for further progress.

The importance of adequate tooling is evident for all stakeholders, and the possibilities are getting better and more accessible. However, the increased availability of options of commercial as well as open source solutions (as described in the "PolarSys" contribution by Gaël Blondelle) highlight the necessity of standardization in a context requiring collaboration and exchange of information, as also addressed by Sandy Friedenthal and Roger Burkhart in their discussion on the future of SysML. Axel Reichwein and Roger Burkhart also address the need to integrate systems engineering artifacts, providing a perspective for the transition from the engineering within a single organization towards collaboration across an entire sector. They also highlight the importance of "open-world" standards and independence of tool implementations to achieve an adequate interoperability at semantics level.

OUTLOOK

None of the contributions in this issue of *INSIGHT* has demonstrated a clearly measurable return on investment (ROI) to support the "MBSE case." Nevertheless, they describe achievements in the introduction of methods and associated means into industrial processes with improvements of understanding, collaboration, and quality for complex products and in complex environments. The community needs to capitalize on these achievements and build on the gradual expansion into new areas.

Examples described in the articles still focus on the early phases of a system design, and many interdisciplinary aspects still remain to be detailed to cover full life-cycle aspects. It is also remarkable that they mostly describe approaches in "vertically integrated" organizations, meaning that all parts of the process are under common governance.

One of the challenges to be addressed in the future is therefore the extension of MBSE principles (exchange of models or other information artifacts) across organizational and contractual boundaries. Some aspects to be addressed to provide the appropriate basis for this are further progress on semantic interoperability, confidentiality (linked to data security), and governance of information and models. ■

AUGUST 2015
VOLUME 18 / ISSUE 2

Current Modeling Trends in Systems Engineering

Robert Cloutier, robert.cloutier@stevens.edu

INTRODUCTION

There have been a number of surveys and studies conducted since 2009 to investigate the adoption of modeling in the systems engineering community since 2009 (Bone and Cloutier 2010) (Cloutier and Bone 2012) (Tower 2013). In preparation for the 2015 INCOSE International Workshop, the latest survey went to a broad community via emails, social media, and discussion boards (the survey is still available if you would like to add your response, it can be found at: http://www.surveymonkey.com/r/MBSE2014). As of March 13, 2015, there were 201 responses. This article will look at some of the results from that survey.

SURVEY CONSTRUCTION

The survey was limited to eleven questions. The purpose of the first question was to collect demographic information to aid in the analysis. The next set of questions approached the use of MBSE in the participants representative companies. Then it moved on to specific uses and implementations of MBSE. Finally, the survey probed at perceived value and barriers to adoption of MBSE. The 2014 survey used many of the same questions used in the 2012 survey, and some go back to the original 2009 survey. This reuse allows us to begin to observe trends through longitudinal analysis. This paper will focus on 5 of the 11 questions. A later report will investigate the broader survey.

DEMOGRAPHICS OF PARTICIPANTS

Question 1 was about demographics. Looking at the demographics, 140 participants chose to provide demographic information. However, out of those that did answer question 1, another 37 did not list their country. Figure 1 shows the international representation we can discern from the data.

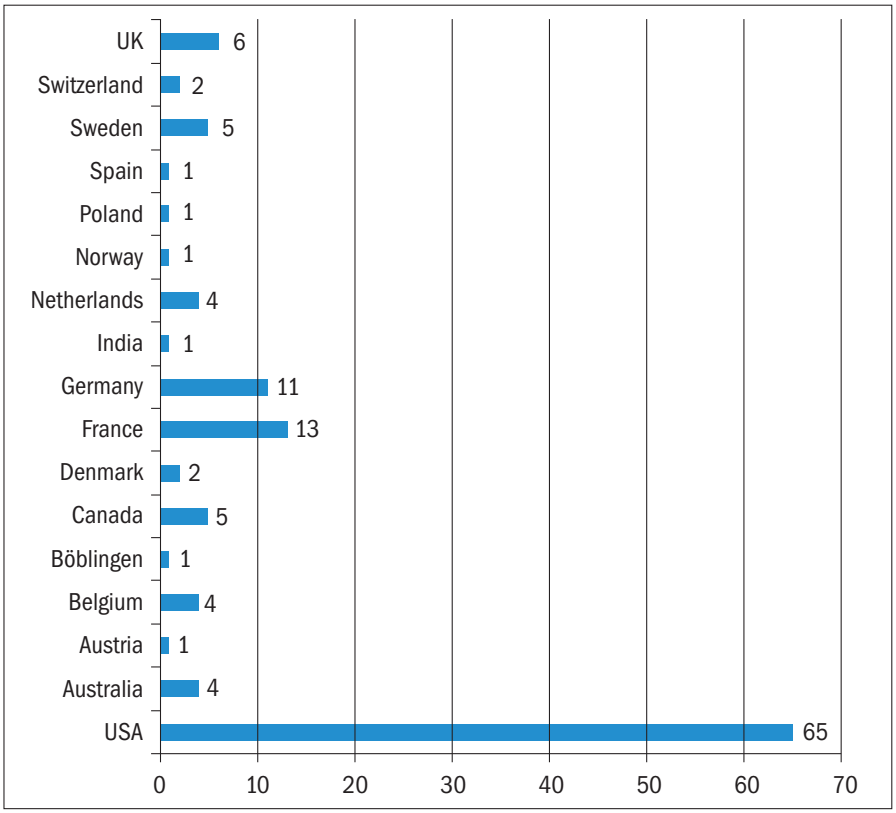


Figure 1. International Representation of Survey Responses

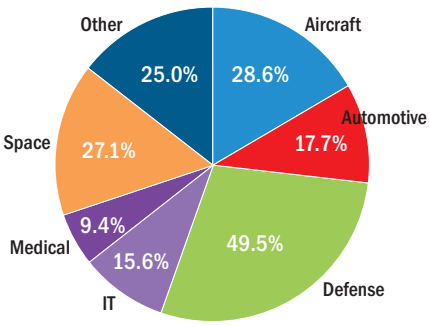


Figure 2. Expected Industries Adopting MBSE

RESULT – EXPANDING INDUSTRY ADOPTION OF MBSE

Question 3 asked, "What type(s) of industry and/or product(s) does your company represent?" There were a number of options provided, and the response to those is in Figure 2. Figure 3 shows how those responses compare with the 2012. However, the question also provided the opportunity to provide an alternate answer.

Question 3 also provided the survey participants space to provide a textual description of their industry if it did not exist as a checkbox. This "Other" response represented 25% of the survey participants. Figure 4 is a word cloud representing the

industries represented as "Other." In a word cloud, the larger the word, the more times that word appeared in the textual responses. Looking at the word cloud, some of the represented industries are not a surprise. For instance, the participants representing the MBSE tool/vendor industry were there. While variations of "academia" also appear in smaller words, the large representation of "Education" makes it clear that MBSE seems to be a primary topic in the classroom. However, there are some other interesting findings. It shows that there is a growing use/acceptance of MBSE in the non-DoD/Defense industries such as "Energy," "Rail," and "Equipment."

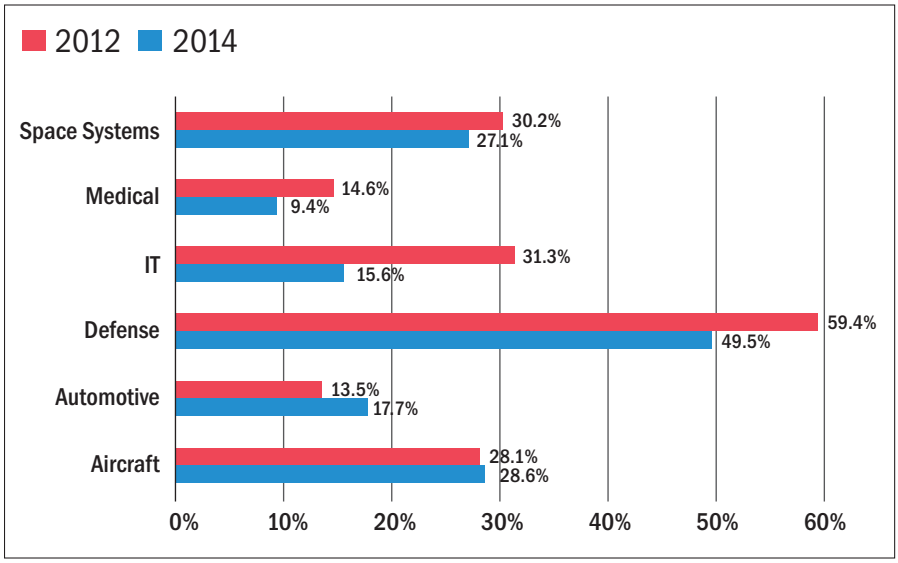


Figure 3. Expected Industries Adopting MBSE: Comparing 2012 with 2014

RESULT – INCREASING ADOPTION OF MBSE ON PROJECTS

In both surveys, the participants were to indicate on a relative scale, whether their companies were increasing awareness of MBSE, piloting MBSE, or adopting MBSE on their programs/projects. Remember, in 2012, there were 166 responses, and 201 responses in the 2014 data set (figures 5-7).

RESULT: MBSE FOCUS ON THE SYSTEMS ENGINEERING PROCESS

Another question (Question 6) asked "What is the relative focus of the MBSE effort in your organization to support each of the following?" Figure 8 (on the following page) shows there has been an increase of usage in concept design, requirements flowdown and traceability, architecture modeling, and verification planning and execution since the 2012 survey. The use of MBSE for conducting Trade Studies seems to have declined between surveys.

Question 9 asked the participants "Rank each item below in terms of the extent that it currently inhibits successful adoption of the MBSE within your organization/company." There was an option for the respondent to elaborate on their answers, and we received 33 unique responses beyond the rankings. We grouped those 33 responses into the following broad categories.

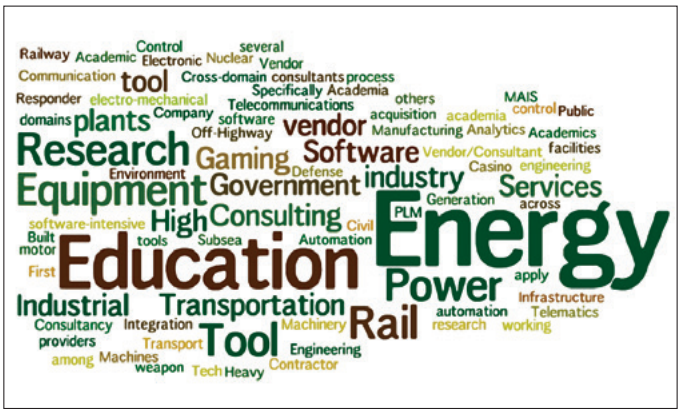


Figure 4. "Other" Industries Represented in this Survey

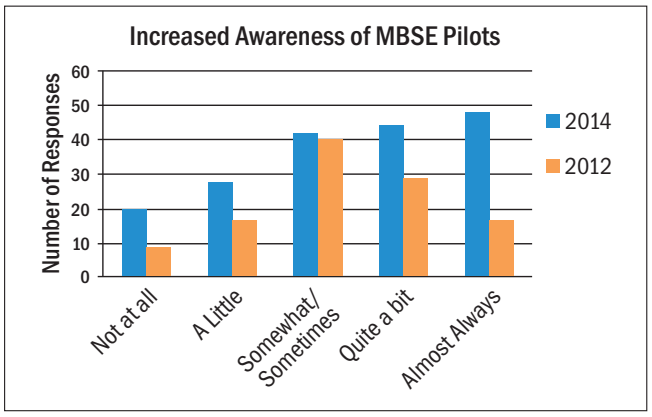


Figure 6. Increased number of MBSE Pilots

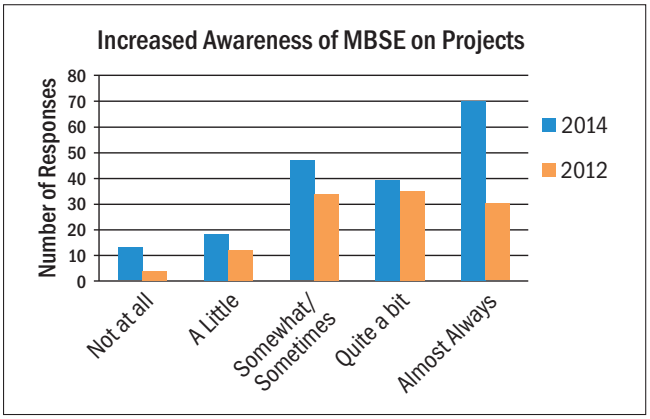


Figure 5. Increasing Awareness of MBSE on Projects

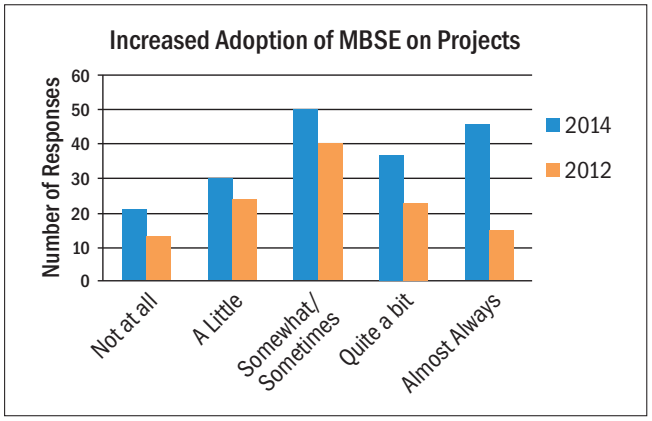


Figure 7. Increased Adoption of MBSE on Projects

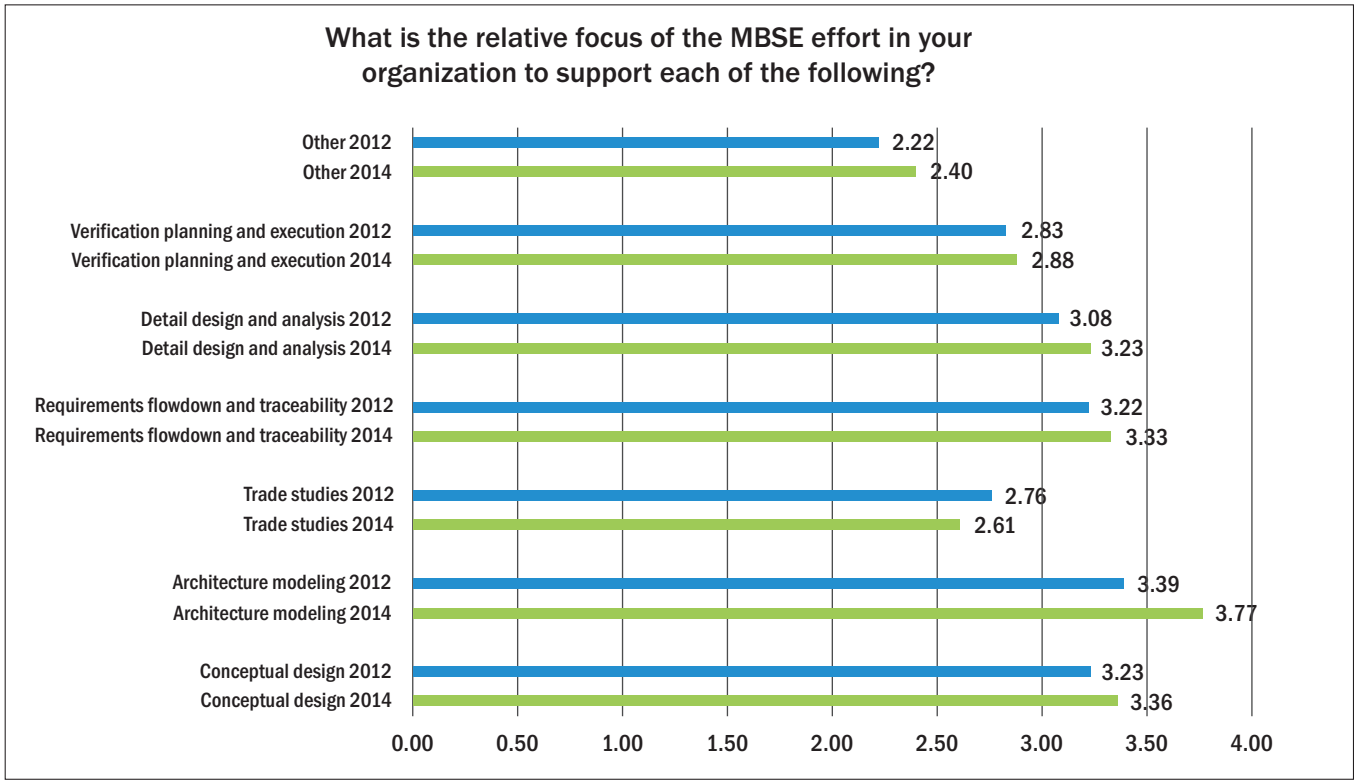


Figure 8 - Relative Focus of MBSE Efforts

- Perceived Management Issues.
- Cost/Return on Investment (ROI) Issues.
- Legacy Issues.
- Broader Systems Engineering Problems.
- Tools & Methodologies Issues.
- Lack of Skilled Practitioners.

We provide most of the individual comments below that provide some insights into the perceptions of those taking the survey (some are not provided as they were inappropriate for this article). This is simply a reporting of the responses without attribution, and does not reflect the opinion of the author or INCOSE.

PERCEIVED MANAGEMENT ISSUES

- Most management does not understand the value of MBSE – therefore support is lacking. Those that do – are all on board and pushing.
- The perception of ROI from MBSE is low against the resources. Sometimes resources for conducting MBSE activities are under estimated.
- Management stubbornness to fully engage with the workforce.
- Very little interest here in model based system engineering; management sees little or no value in it.
- They are used to working one way and do not like change; it is only when they start to see the improvements in quality and cost reduction that we talk about that they start to take notice; however it takes a long time for them to see this, as their work force is generally as resistant to change as management, and if they have not been exposed to these ideas early, it is hard work to get them to understand how to do MBSE.
- Management support is an imperative before adoption; very few projects can deploy MBSE without explicit management support. Most projects do not have management support for MBSE, and the reason usually given for not having management support is risk.
- Most organizations are so buried in the issues of managing their current systems, they do not understand the value of formal models (largely because systems were often built without them, and even if there were models, they have fallen by the wayside in favor of existing code base); we think the only real hope is extracting models from code, because existing systems is where the bulk of the effort goes.
- Not enough autonomy given to the MBSE team; leadership, with no real MBSE knowledge, mandates "terms and conditions" that can inhibit MBSE from reaching its true potential on a program or slow its adoption, against the recommendations of the MBSE champions.
- Biggest issues are that management and project leaders do not understand MBSE or the value it brings.

COST / RETURN ON INVESTMENT (ROI) ISSUES

- MBSE costs; time to implement MBSE before starting the projects.
- Defense ship builder; priority 1 is construction and hardware design, lower priority is systems engineering approach; very low priority is MBSE.
- Until it becomes mainstream, or someone demonstrates value at no cost to them, many will resist.
- Good tools are too costly (IBM Rational, Mathworks); with a tool such as SPARX EA it remains cumbersome to maintain the models.
- Many engineering projects are under tight budget and schedule constraints and staff return to what has worked for them in the past to get the job done; techniques such as learning agile are considered to have a more direct impact on the project than learning an overarching systems modeling technique.

LEGACY ISSUES

- Legacy product data – so much legacy product data in document-centric approach, with no inexpensive/simple translation process from the documents to newer model-centric approach; project

management perceives that getting over this hurdle is too high a price; most projects are "one-off" variations of prior products, so modeling the changes from a prior project is perceived as too expensive.
- Document-based regulatory system.

BROADER SYSTEMS ENGINEERING PROBLEMS

- Inconsistent definitions of systems engineering.
- Selling systems engineering is hard enough; to sell MBSE on top of that makes it even harder; we have learned to just sell systems engineering; MBSE is the way we do it; if the tools are available and people on the project have the available skills, they know the methods, then they should just take a modeling approach and not ask.
- Still there is a gap between systems engineering, certainly MBSE, and the software design and methods; for you as a software developer there is still a big room for interpretation out of the systems engineers' models; how do you verify a model?

TOOLS & METHODOLOGIES ISSUES

- Methodologies – object-oriented systems engineering methodology (OOSEM), for example, not well integrated with tools with exceptions.
- Tools are perceived as not really mature at least for SysML and language is found too complex so people prefer using MS Visio for some time; when they can be convinced of benefits then they are ready to learn new tools with more choices to do than Visio but it takes time to reach this point.
- Tool and method maturity is the only inhibitor in that we have applied to a variety of different projects and customers with great success, each of them leading to valuable features, but are always looking for more in an effort to evolve CertSAFE.
- The fact that MBSE models cannot convert directly into simulations really stinks; I was generating high-level models that could be compiled directly into simulations with timing back in the early 1980's; IBM was way ahead of the current toolset back then. Also, MBSE was created by software engineers; software engineers are not systems engineers, no matter what they think of themselves; Also, trying to force-fit a tool across all other disciplines has baffled me about MBSE.
- Lack of applications with intelligent technologies and easily executable models in simulation.
- MBSE effort in collaboration with customer/prime contractors/subcontractors are lacking; no model transfers,... no common models; project time scales inhibit the effective implementation (1 year for implementation).
- Seasoned experience in MBSE is a problem even when tools and training are available; projects also seem to create their own "style" in how 'they' decide to utilize MBSE on a project – "right fit" perspective.

LACK OF SKILLED PRACTITIONERS

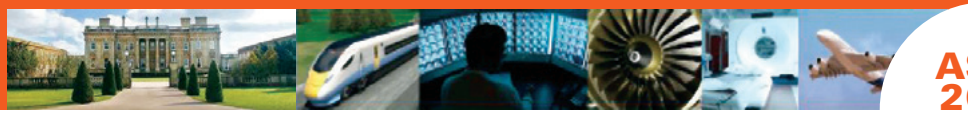
- Another major inhibiting factor is the availability of skilled practitioners; not just to actually execute MBSE-based projects, but the advocacy and energy of our most talented modelers is now spent in projects (a good thing) without much time left over to educate management or practitioners in other parts of the adoption curve than early adopters (not such a good thing).
- Lack of resources.
- The number of people with MBSE skills is rising rapidly and this is improving the situation; many of these people are being used as consultants and helping change companies by showing the internal workforce how to do MBSE and acting as champions/evangelists for MBSE within the company.
- For those that do have management support, biggest inhibitor is lack of skilled practitioners... learning curve contributes to lack of skills; cultural resistance is less important than it used to be ... perceived risk is much more important.

CONCLUSIONS

The application of modeling to systems engineering continues to advance. There are more people modeling, and more companies supporting the trend. New industries are beginning to investigate or adopt MBSE – it is no longer limited to the Defense domain. However, there are still challenges. Systems engineering is still in the early adopter phase of MBSE. Many still perceive the cultural resistance to change a major hurdle to jump. In some companies, management is hesitant in many corners to risk early adoption. The tools are getting better, but are still not mature enough. Finally, there are not enough systems engineering practitioners that are competent systems modelers. Of course, this is also an opportunity for those that are willing to take up the challenge. ■

REFERENCES

- Bone, M. and R. Cloutier. 2010. "The Current State of Model Based Systems Engineering: Results from the OMG™ SysML Request for Information 2009." Paper presented at the 8th Conference on Systems Engineering Research, Hoboken, US-NJ, March 17-19.
- Cloutier, R. and M. Bone. 2012. "MBSE Survey 2." Paper presented at the INCOSE International Workshop, Jacksonville, US-FL, January 21-22.
- Tower, J. 2013. "Model Based Systems Engineering 'The State of the Nation'." Paper presented at the INCOSE UK Annual Systems Engineering Conference (ASEC), UK-Oxfordshire, November 12-13.



"Systems Engineering comes of age"

This year marks the 21st anniversary of the formation of INCOSE UK, and in the spirit of "coming of age," the conference sub-themes are:
- What works and why it works.
- What's new and why should we be interested?
- How do you do your Systems Engineering?
- Impact of Systems Engineering outside of the workplace.

INCOSE UK Annual Systems Engineering Conference

Crown Plaza, Heythrop Park
Oxford

17th–18th November 2015

INCOSEUK

AUGUST 2015
VOLUME 18 / ISSUE 2

MBSE Has a Good Start; Requires More Work for Sufficient Support of Systems Engineering Activities through Models

Harald Eisenmann, harald.eisenmann@airbus.com

FOREWORD

The discussion on MBSE started in the late nineteen nineties, when the transition towards model-driven engineering was rapidly progressing for other engineering disciplines. However, at that time consideration of MBSE was theoretic, abstract, or even not needed at all. After 15+ years of development to realize MBSE, more elements are available, ready for use, and with that came industrial adoption. As of now, MBSE is the focus when it comes to process efficiency improvement. MBSE is a key piece for obtaining an end-to-end support of engineering processes through models, for which "Digital Engineering" seems the most appropriate term. Tightly related is the discussion on Industry 4.0, and "BIG (engineering) DATA (www.dlg.org/fileadmin/downloads/food/Expertenwissen/e_2015_5_Expertenwissen.pdf)." However, in order to complete the support of systems engineering activities sufficiently with models, systems engineers need further stimuli. This article is written from a perspective of the European Space industry, the author reflecting on the results achieved so far, and from there, he derives the need for further developments, in order to succeed with comprehensive support of the systems engineering process with models.

MBSE AS AN ENABLING ELEMENT FOR CONSISTENT DEPLOYMENT OF SYSTEMS ENGINEERING

The European Space community consolidated the systems engineering processes in the framework of ECSS (European Cooperation for Space Standardization) in the late nineteen nineties. With that, a consistent application of systems engineering in European space programs began and company specific adoptions with required refinement and tailoring occurred accordingly. The tools emerging to support systems engineering were an opportunity to consistently deploy the systems engineering process. Deployments of starting tools were mainly requirements management tools and product data management (PDM) systems.

At the same time, the discussion on model (-ing) support for systems engineering started, initially called model-driven system design (MDSD), later model-based systems engineering (MBSE). The emerging discussion on MDSD/MBSE in Europe triggered further discussion on tool support systems engineering. The following can be stated.

- **Concurrent engineering:** In particular, in early systems engineering phases, many activities happen concurrently on the same data (baseline). In this scope, systems engineers identified the need to have proper mechanisms in place for synchronizing the concurrent processes in a continuous and reliable way. A key example for that was the solution for conceptual design introduced at NASA JPL (www.lr.tudelft.nl/fileadmin/Faculteit/LR/Organisatie/Afdelingen_en_Leerstoelen/Afdeling_SpE/Space_Systems_Eng./Expertise_areas/Systems_engineering/Lectures/doc/paper13.pdf). The adoption of this approach in Europe started in the late nineteen nineties (http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:esa_-_concurrent_desigh_facility.pdf).
- **Collaborative engineering:** As for all complex systems development in European space programs, one company on one single site does not perform the activities. Rather, industrial consortiums perform the work, spread across Europe. Thus, the support request eased collaboration between locally distributed entities.
- **Modeling and simulation:** One of the key requests was to equip systems engineering with modeling and simulation functions – for system level design and verification activities. Systems engineers stressed the V-model to express the need for having modeling and simulation support throughout the complete system lifecycle.

Compared to the rapidly evolving tools in M-CAD and E-CAD, the system modeling tools were lacking maturity. Consequently, various attempts were observed where discipline models were enhanced to capture system level needs.

From the increasing awareness it was obvious that a single organization would not succeed in realizing MBSE, for getting the buy-in of the tool vendors, or enabling consistent collaboration across phases, organizations, or projects. Thus, systems engineers identified standardization as a requirement in order to manage the transition towards MBSE. From a European Space community point of view, the following activities are key for standardization.

- **Systems Modeling Language:** INCOSE identified the need of a systems modeling language, in collaboration with the Object Management Group (OMG), and the effort for SysML initiated.
- **Data Exchange for Systems Engineering Data:** The objective was to define

a common data exchange solution for systems engineering. ISO picked up this need in the framework of the STEP developments, called Application Protocol (AP) 233 (ISO 2012).

- **Engineering Database in Support of Systems Engineering:** With the consolidation of the process standards for systems engineering in ECSS, the need was to improve the systems engineering process by appropriate databases on the systems level. In general, databases were perceived as arriving (too) late in the program, being cost intensive, and not having the right fidelity, or interfaces. The goal of the ECSS – Engineering-Technical Memorandum-10-23 (E-TM-10-23) Space System Data Repository Working Group was to assess the situation and to provide a way forward. The activity was a key element to manage the transition towards MBSE, driven by common project needs. In parallel, the ESA triggered various research activities on system simulation. The focus was to obtain representative virtual models, used for system verification.

Going through all of the above-mentioned activities, the systems engineering community made the first steps towards embracing MBSE. The identified actions were in place, and managed either by global standardization organizations, or with the ECSS.

THE INITIALLY PLACED ACTIONS SHOW RESULTS – DEPLOYMENT IS ONGOING

After a few years of development, the mentioned actions did deliver, along with the objectives, and thus started the adoption of MBSE in the space programs. The status of the delivered items were the following:

- SysML: The specification of the SysML language release occurred in 09/2007 (1.0) with subsequent updates in 2009 (1.1), 2010 (1.2), and 2012 (1.3). With that, there is an established standard for a systems modeling language, implemented in different tools. The platforms for the implementation used existing Unified Modeling Language 2 (UML2) implementations, so that a SysML tool evolved from a UML tool.
- SysML – in general – plays a major role in making the case for MBSE, creating the required momentum for MBSE consideration across industrial domains. INCOSE, with the workshops on MBSE, is the facilitator of this process. There are many reported success stories (OMG 2015).
- ECSS-E-TM-10-23 "Space System Data Repository" the emerging standard for a "systems database" was published in 2008 as a draft, and formally released in 2011. The key elements in the technical memorandum (TM) were the following:
  - *Conceptual Data Model*: The semantic definition of data captures the information of a space systems engineering process, in a conceptual data model (CDM). The main purpose of the model is to serve as shared reference for the different implementations, for obtaining "semantic" interoperability. Implementation in this sense is tool customization (PLM or SysML profiling), tool development, or data exchange solutions.

    As a side effect the 10-23, CDM is a manifestation of core systems engineering knowledge, as it formally captures the systems engineering concepts in a model. *Space System Data Repository*: Various tools manage systems engineering data. Each tool relies on its own data management functions. However, to obtain a consistent set of data, required consistent application of the data management functions. This means relying on the same semantic structure, but also sharing key functions, such as version control or consistency checking. For this, systems engineers introduced the concept of the space system data repository, requesting a consistent interpretation and management of data across the different tools, operating on the shared data model. The standard is not imposing a particular architecture, (if the space system data repository is a centralized instance or a de-centralized architecture with shared services). Likewise, to SysML in the global community for system specification modeling, 10-23 provided the vision for proceeding with data management in all project phases; a key need expressed from the systems engineering community of European space projects (ECSS, 2011).

From the increasing awareness it was obvious that a single organization would not succeed in realizing MBSE, for getting the buy-in of the tool vendors, or enabling consistent collaboration across phases, organizations, or projects.

- ECSS-E-TM-10-25: "Systems Engineering – Design Model Data Exchange (Concurrent Design Facility (CDF))": In parallel to 10-23, a second standard with the designation ECSS-E-TM-10-25 was established, focusing more on early phases. For this TM the CDM of 10-23 was adopted, and specialized for the purpose of conceptual design. 10-25 was released in 2011 (ECSS 2010).
- ISO 10303-233 "Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 233: Application Protocol Systems Engineering:" After the initial work in the Systems Engineering Data Representation and Exchange Standardisation (SEDRES) project, the AP233 WG completed the work on the data exchange standard and also the integration into the ISO data exchange framework. Obviously one of the key elements was the integration with SysML, in order to support the exchange of SysML models with AP233. The AP233 standard was released in 2012 (ISO 2012).
- Functional Systems Simulation for Verification: The initial research activities, initiated and funded by ESA, demonstrated that representative simulation models on systems levels are possible. Pilot applications validated that functional simulation models can significantly reduce the cost of verification activities, due to earlier availability, increased fidelity and re-instantiation of facilities. In addition, it showed that, for many verification activities, the "executable virtual system representation" is superior due its high degree of fidelity, in terms of operation, re-configuration, and failure injection or failure investigation. The pilot applications were completed around 2005.

With the arrival of solutions realizing MBSE, the adoption and deployment of MBSE in the European space programs progressed. The following summarizes the status:

- Support for conceptual design: Facilitated through the work around the ESA CDF (ESA 2015) and ECSS-E-TM-10-25 (ECSS 2010), facilities for conceptual design are now widely adopted across the European Space community (Simoni and Roser 2012), (Denier-Gégu 2014). The particular features of those systems are to provide a platform for "concurrent" data sharing. While the initial solutions managed a "flat list" of key system properties, the 2nd generation, as realized in the Open Concurrent Design Tool (de Koning et al 2014), implementing ECSS-E-TM-10-25, allows full descriptive systems modeling, and enables re-use of models and their evolution along the life-cycle.
- SysML: The adoption of SysML in European space programs started with some delay. First, there was some reservation, due to a perceived software engineering bias in SysML, and the complexity of the language or tools. Based on many successful pilots of SysML, as reported in other domains, more and more SysML applications can be observed now, mainly for two use cases:
  - The support for process from Systems to Software, with Systems – Software co-engineering. Obviously a key interface to ensure the mission success. Various different approaches can be observed, all with the intention to have a SysML model to express the specification of the flight software.
  - The support for the "functional analysis process." Here the objective is to find the right set of system functions, modes, components, etc., which are needed in order to answer the "system requirements." The utilization of SysML for this use case is not fully consolidated yet. However, the pilots provided information on the benefits of elaboration of requirements, in terms of use cases, system functions, operational modes, and building blocks in a specification model in SysML. Also the need to connect the "specification models" to "design models" has been identified (Alvarez 2014).
- System Database: The goal of ECSS-E-TM-10-23 was to shape the future of system level data management inside projects. ESA performed some validation activities to demonstrate the technical feasibility, but also the benefit for the process – relying on the conceptual data model instantiation of 10-23 as a data-sharing platform. The most comprehensive activity was "Virtual Spacecraft Design" (VSD) (Eisenmann et. al. 2012), (ESA 2015). The demonstrator covered system design definition, integrated system database with links into the disciplines, and integrated system visualization. This validation increased the awareness for MBSE in late phases and created industrial momentum on the topic; different implementations followed (Pasquinelli et al. 2012). The Airbus Defence & Space activity RangeDB was leveraging VSD as blueprint, and did develop a system database according to those principles. It relies on the Eclipse Modeling Framework (Eisenmann and Cazenave 2012).
- Functional System Simulation: Functional system simulation to support system verification is a widely adopted practice in ESA, typically referred to as Functional Verification (FV), part of MBSE. This approach is successful in many programs, and is entirely mature in terms of simulation tools and products applied, but also with the corresponding processes enabling a smooth support of programs. With those mature solutions in place, a significant re-use of infrastructure elements, as well as models occurs. Airbus DS developed their own product line for that purpose called SimTG. It supports the complete simulator engineering process (Eisenmann, Cazenave 2010).
- The use of AP233 for data exchange has been considered, but a real operational application cannot be reported.

With that update it can be stated, that the transition to MBSE in the ESA programs is progressing. From the discussion now it seems that the programs are requesting solutions, since MBSE is considered an ongoing evolution of processes and thus a necessity to be prepared for the future.

A FULL AND SATISFACTORY SUPPORT OF THE SYSTEMS ENGINEERING PROCESS REQUIRES FURTHER ELEMENTS

With the transition started, it seems now is the right time to evaluate if the initially developed elements are sufficient to succeed and complete the transition. With the initial deployments, it seems that the systems engineering community now has to move forward and adopt MBSE as systems engineering. Tool interoperability remains one of the biggest challenges. In addition to that, emerging technologies might bring further value to MBSE. Overall, now may be the right time to consider the next step of standardization. The following identifies some elements in order to further advance the transition.

- **Complete the support of the systems engineering process through models**: Significant progress has been made in order to support the systems engineering process through models. However, the support of the entire systems engineering process is by far not given. Not covered systems engineering essential tasks are budget tracking, optimization, impact analysis, trade studies, or optimization. In general, this affects tasks where systems engineering is relying on (a subset of) data owned by another process. Closing this gap requires new interoperability functions, easing the integration, model reduction, and traceability of data.
- **Evolve systems engineering methodology:** Currently, it seems systems engineering and MBSE are treated as two different things: On one hand the systems engineering community with the process, on the other hand this …"MBSE tool." However, succeeding with MBSE requires MBSE full adoption by the systems engineering community to an extent where systems engineering = MBSE. In recognizing that systems engineering can take over responsibility for essential steps in order to further advance and manage MBSE as natural evolution of the systems engineering methodology:
  - An evolution of the systems engineering methodology to be synonymous with MBSE. Many discussions on MBSE show that the models exist, if systems engineering is clear enough on the needs. In other words, the problem now is more on the systems engineering side of MBSE. This requires a more formal capture of processes, flow of information and (model) artefacts. Systems engineers must clarify what is needed, at what points in the lifecycle, and to which level of detail. Eventually the methodology evolution will directly tie into data models (for tool customization/production) and process models for tool configuration on the flow. This is crucial, since there will be no single tool or model able to answer all systems engineering needs.
  - For the models, systems engineering has to refine vocabulary in order

to distinguish the different model types. There is no "model," as such. Rather there are models being used for elaborating on the need (specification), defining the solution (design), descriptive models, simulation/analysis models, trade models, analytical vs. verification models, history/tracking models, etc. A single reference for model taxonomy, or ontology, would do the community good.

- Impact of models on the overall process. One example is the discussion on "front-loading" the process, which means that a consistent application of models earlier in the lifecycle might help to reduce cost in later phases, but requires an increase of the effort in earlier phases.
- With all the different models arriving in the systems engineering process, the practices of "model philosophy" must evolve. Many models are in use for verification and qualification purposes. Traditionally many/all of them are hardware oriented models. Now many virtual or hybrid models are available. For the future, more and more verification activities will rely on virtual models. What is the required fidelity, representatively? What are the minimum mandatory hardware configurations needed? How do systems engineers correlate and calibrate virtual models?

- **Utilization of advanced technologies for next generation of MBSE tools:** Many of the currently available MBSE solutions rely on well-established tools and technologies, available for a number of years. For preparing systems engineering for the future, new technologies have to be considered:

  - *Evolving diagrams for model visualization*: Diagrams are essential means to express and share ideas. Therefore, it is quite important that CAD tools offer the feature for editing diagrams. Still in many cases, the tools are rather diagram editors than modeling tools. For the future, the features must evolve significantly, to overcome a situation where diagram editing is a bottleneck in modeling ("doing modeling means spending time for diagram layouting"). The improvement areas cover:
    - Support the automated layout of diagrams.
    - Provide query-based definition diagrams, retrieved from the database. This allows the definition of viewpoints of interest. Graphical view follows the database content.
    - Provide efficient cross-discipline visualization of data. Each discipline view comes with a particular notation. Systems engineering requires efficient view across disciplines.
    - Leverage 3-D visualization for systems engineering data, for visualization of complex dependencies, and more.
    - Evolution of documents into integrated model descriptions: Documents play an essential role of knowledge capturing and sharing. Documents contain background knowledge, provide guided introduction, or justify the solution given. Therefore, documents (→ complementary text) will remain an essential function in the MBSE world. For the future, documents and models require better integration. This could mean that text will be considered as a model artefact, where complementary descriptive information will have model references and queries embedded. With that model, descriptive information will remain its function, but will be managed like a model.
    - Integration of collaboration social network features for the future MBSE workbenches shall be completed with collaboration (or social network functions). With that, a firm integration of technical data with any associated collaboration related data occurs. This allows referral from discussion forums, directly into the model, and vice versa.

  - *Data management technologies*:
    - Consistent management and sharing of data is a key prerequisite for the complete adoption of MBSE. This strongly depends on the integration of the engineering tools involved in the process. Today the integration is often a considerable issue/effort, partially also lacking vendor support. For the future, MBSE must provide affordable integration, in terms of investment cost, configuration, maintenance effort, and frameworks, considering the following points:
      - Semantics will become a major issue. This requires that data models be mastered by the application community. Data management functions shall allow the customization with respect to particular data models.
      - Clear separation of process data "data to be managed," from data management related data "data required for management." While the first is driven by the process, the second is driven by tool vendors.
      - Definition of systems engineering work or data flows, allowing a tight management of data.
      - Data management functions, offering "in detail" management of data, but also with sufficient usability. One of the key improvement areas here is consistency checking, required to assess completeness and consistency.
      - Evolve modeling tools from plain databases towards more semantic expressiveness for improving the logical coherence of the descriptive models. This might require MBSE to adopt some elements from ontology systems.
      - Smart editing of data – Engineering complex systems requires many iterations within a single discipline; along the customer supply chain, "on a system level." Each iteration requires a particular update of the models, based on issues identified in the past iteration step. In order to support this the following shall be provided:
      - Detailed impact analysis of a change, identification of outdated data.
      - Identification and management of data, which is directly dependent on other data, potentially based on pre-defined rules, allowing the automated update of data.

  - *From tools to frameworks:*
    - Today, MBSE still suffers from the incompatibility of tools being used in the process. A trend is observable that frameworks are increasingly offered. Also very powerful open source technologies for model management are emerging rapidly. For the future, MBSE may need data management frameworks that provide essential shared functions, but allow the systems engineers to plug-in specific editors, simulators, visualization or advanced functions that support trades, optimization, or model transformation/reduction.

- **Evolve Standardization for data and process modeling**: In the European Space community, there is a widely shared awareness that conceptual data models are a key enabler for process and tool integration. They enable mapping systems engineering practices (concepts) and thus needs, into implementation, either tool customization or tool development. However, the "right" approaches for data modeling are still under discussion with solutions from software development, the ontology world, or the fact-based community. For further progress on MBSE, the standardization bodies need to work on the following:
  - Define and share data modeling language and methodology across the community.
  - Agree on data model architectures allowing the shared definition of key engineering principles (requirements, interfaces, building blocks, and from there allow a refinement into the different industrial domains (automotive, shipbuilding, space…) and eventually to organization/project specific tailoring.
  - However, the required standardization efforts must evolve to facilitate the transition. So far, data models are addressed in different standardization efforts, in different roles, with different emphasis, using different technologies (STEP AP 233 data exchange, SysML for graphical modeling tools, and ECSS-E-TM-10-23/25 for shared data management). Although all rely on key systems engineering principles, the resulting models are quite different.

- **Digitalization/ BIG (engineering) DATA** – The discussion on BIG data reaches into the domain of engineering. The transition towards MBSE, or more in the wider scope "digital engineering," is in the core of the digitalization effort. Once established it would allow the application of BIG DATA techniques in order to further improve systems engineering process for:
  - Improving the cost efficiency of the system design and verification process by cross relating technical data with management information.
  - Close the loop between engineering, design verification and operation in terms of robustness, reliability, and more.
  - Improve the system design process by including business and value models of the envisaged customer.
  - Gather knowledge on systems engineering process on different projects.

Based on the results of the initially triggered actions on standardization, such as SysML, MBSE is now on a good track; for the European Space community, ECSS-E-TM-10-23/25 further facilitated this with complementary items. The standards provided a shared foundation, and tools were developed accordingly. The user community benefits from mature tool implementation, which eases the way into programs. Now the next step in MBSE should be done in order to complete the transition from systems engineering to MBSE. For this, a new stimulation from standardization is needed. At the core of this effort is the requirement of the end-to-end support of systems engineering with models, including the involved disciplines and the customer supply chain. This requires some new concepts from standardization for semantics, support for process modeling, and the ease of tailoring – from common standard foundation, down to the concrete process needs. MBSE needs significant evolution for interoperability significantly for a "plug and play" of different tools. For the tools, the emerging availability of open source technologies might help evolve from cumbersome "data exchange" between different incompatible tools, to shared data management functions. With all of these issues pursued, the transition to MBSE can be completed. This transition will enable those systems engineering benefits of "BIG DATA" technologies, which will have a massive impact on industrial processes, by gathering and correlating data and models. ■

REFERENCES

- Alvarez, J.L., 2014. "Euclid Requirements Engineering supported by SysML." Paper presented at the 8th ESA Workshop on Avionics, Data, Control and Software Systems–ADCSS, Noordwijk, NL, October 27-29.
- de Koning, H.P., S. Gerené , I. Ferreira, A. Pickering, F. Beyer, J. Vennekens. 2014. "Open Concurrent Design Tool – ESA Community Open Source Ready to Go!" Paper presented at the annual Systems & Concurrent Engineering for Space Applications Conference (SECESA), Stuttgart, DE, October 8-10.
- Denier-Gégu, D. 2014. "Concurrent Engineering at Airbus Defence & Space: LessonsLearned from the SPACE CODE project." Paper presented at the annual Systems & Concurrent Engineering for Space Applications Conference (SECESA), Stuttgart, DE, October 8-10.
- ECSS (European Cooperation for Space Standardization). 2011. ECSS-E-TM-10-23A. Space engineering – Space system data repository. Noordwijk, NL: ESA-ESTEC.
- ——. 2010. ECSS-E-TM-10-25A. Space engineering – Engineering design model data exchange (CDF). Noordwijk, NL: ESA-ESTEC.
- Eisenmann H., C. Cazenave. 2012. "RangeDB in support of MBSE–Flexible Infrastructure enabling incremental application of MBSE." Paper presented at the Workshop on Simulation and EGSE for Space Programmes (SESP), Noordwijk, NL, September 25-27.
- Eisenmann H, Cazenave C. 2010. „Successful Harmonization of Simulation Infrastructures" Paper presented at the Workshop on Simulation and EGSE for Space Programmes (SESP), Noordwijk, NL, October 7-9.
- Eisenmann H., J. Fuchs, D. de Wilde, V. Basso. 2012. "ESA Virtual Spacecraft Design." Paper presented at the annual Systems & Concurrent Engineering for Space Applications Conference (SECESA), Lisbon, PT, October 17-19.
- ESA. 2015. "Concurrent Design Facility (CDF)." www.esa.int/Our_Activities/Space_Engineering_Technology/CDF.
- ESA. 2015. "Virtual Spacecraft Design (VSD)." http://vsd.esa.int.
- ISO (International Organization for Standardization). 2012. ISO 10303-233:2012. Industrial automation systems and integration—Product data representation and exchange—Part 233: Application protocol: Systems engineering. Geneva, CH: ISO.
- OMG. 2015. "SysML." www.omg.org.
- Pasquinelli, M., V. Basso, M. Marello, C.M. Paccagnini. 2012. "STEPS Results: MBSE Prototypes to support Space Exploration Projects Life-cycle." Paper presented at the annual Systems & Concurrent Engineering for Space Applications Conference (SECESA), Lisbon, PT, October 17-19.
- Simoni, L., X. Roser. 2012. "Concurrent Engineering Meta Data-Model & Multi-domain Representation." Paper presented at the annual Systems & Concurrent Engineering for Space Applications Conference (SECESA), Lisbon, PT, October 17-19.

AUGUST 2015
VOLUME 18 / ISSUE 2

MBSE in Rail Transportation – Product Families and Product Lines

Marco Ferrogalini, *Bombardier Transportation*, marco.ferrogalini@rail.bombardier.com

INTRODUCTION

As with many other complex systems, rail transportation systems are developed using traditional document-based approaches. The exponential increase of the rail transportation systems' complexity has dramatically shown in recent years with many projects failing to meet their business objectives resulting in negative financial impacts. Increasingly, there is a better understanding of how important it is to follow correct systems engineering processes and to validate product concepts in an early stage through simulation and modeling approaches. This article reports on some aspects and outcomes based on real-life experiences from the introduction of MBSE into the rail transportation industrial sector (Chale Gongora, Ferrogalini and Moreau 2015, 239-56).

MAIN ASPECTS

The MBSE benefits are usually clear for all experts in this domain but unfortunately are often completely unknown at the top level of management.

To achieve management support and understanding of the benefits of the use of MBSE it is necessary to create a clear vision of how technical benefits will ultimately lead to business benefits. The vision push is from the MBSE experts who face the difficulties of how to create it, and how to communicate it to the management in an effective manner (two different languages: technical vs. business).

One should note here that if a correct systems engineering framework, with the key processes in place is not present inside a given organization, then it will not be possible to jump directly to MBSE because the gap to fill is too big. In addition to that, if the deployment of MBSE is to be based on the usage of SysML (one of the most used languages to describe architectures) then another layer of complexity is added since this object oriented language is on one hand complex (especially for a systems engineer) on the other hand quite abstract.

Any organization, which plans to use MBSE and SysML, will have to invest several months or even years to define the modeling methodology and the related usage of the language. One of the reasons is the fact that at the beginning the objectives and expected benefits to introducing MBSE are not well defined because the approach unknown and explored (Chale Gongora, Ferrogalini and Moreau 2015, 239-56).

Due to that, in the introduction of this new approach there is a big risk of setting wrong expectations or to misuse the language if the experts do not have the right level of competency or understanding.

Another very important factor to consider is that today the communication between the organization and the different stakeholders is document-based; we are still far from the concept of an extended enterprise. The methodology then must enable users to publish documents from the models in a format and with a content, which will be easily recognized and understood, and this can be a very big burden to accomplish.

One aspect, which shall be considered while defining the MBSE methodology, is the re-use strategy of existing products, which is present in the rail transportation sector. Normally systems engineering defines a pure top-down approach, starting from requirements, defining the architecture, and then going to the detailed design. In the railway sectors, it is quite rare to start the definition of a new product or a new product family from scratch. It is usually a mixed approach, where the existing solutions are strongly pushed up trying to fit input requirements (Chale Gongora, Ferrogalini and Moreau 2015, 239-56). Unfortunately, the traditional document-based approach doesn't support the requirements vs. existing solution analysis in an efficient manner since the completeness, consistency, and relationships between requirements, design, engineering analysis, and test information are difficult to assess because that information is spread across several documents. Here, the MBSE becomes a key enabler, solving almost all the aforementioned weaknesses of the traditional document-based approaches by providing a structured traceability framework. Already developed solutions can be stored as assets in structured catalogues accessible by every member of the organization in order to avoid to "re-inventing the hot water."

It is true that the modeling methodology, which permits systems engineers to deal with the aforementioned problems, is one level of complexity above the "entry level" methodology, which just permits systems engineers to depict a system in a model. Depending on the product and organization complexity, it might take some years to define such a "Model Based – Product Line – Systems Engineering" methodology (Chale Gongora, Ferrogalini and Moreau 2015, 239-56).

The first aspect to work on will be the process integration with a traditional top-down systems engineering approach with a bottoms-up PLE (Product Line Engineering) approach. Then the organizational aspect shall be considered (in terms of governance) and only at that point, will the MBSE framework be investigated and defined as an enabler to that global strategy. The factors which are

key to ensure a successful deployment are the support from the management at all the levels and a correct plan to lead the cultural change inside the rail transportation systems engineering community. This last piece, the correct plan for cultural change, could be the biggest and most complex breaking factor to the introduction of this new MBSE approach.

Well-structured training and correctly planned coaching and support are necessary. Depending on the level of awareness of the systems engineering community, on the cultural approach, the flexibility to the change, and on the specificities of a given context, adoption of this new approach may be very easy or very hard. A good strategy is to create a training which will remind the adopters of the fundamental steps of the technical processes of systems engineering (especially the ones related to the system design phase), explain how to perform them depicting systems and systems environment elements into objects using the SysML language and then how to do it by using a specific modeling tool (Chale Gongora, Ferrogalini and Moreau 2015, 239-56). A good training shall:

- Integrate all three main aspects: processes, methodology, and tools, showing trainees the logical link between each.
- Provide real life examples of the specific technical domain in order to allow trainees to see how a known system is analysed or described using a modeling approach.

The author believes that it is best if the trainer and future coach are colleagues of the trainees, recognised as senior experts in the company systems engineering community.

One of the most common ways to introduce MBSE is to start with a small case study to gain understanding of both the complexity and the potential benefits.

Once a case study has shown the feasibility of the approach and which benefits can be gained, it is only then that a structured roadmap is normally defined (Chale Gongora, Ferrogalini and Moreau 2015, 239-56).

REAL-LIFE EXAMPLES

As mentioned, currently in the rail transportation sector there is a strong push to migrate from a pure product development approach (for a given Customer) to a Product Line Engineering approach developing Product Families (Chale Gongora, Ferrogalini and Moreau 2015, 239-56). In this context the MBSE could be a key enabler giving a framework for structured reuse:

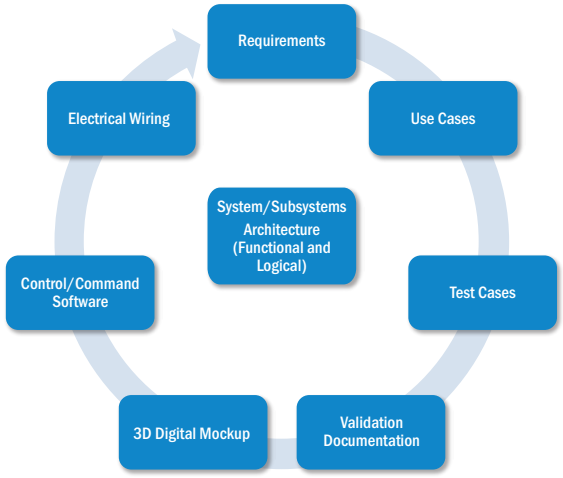


Figure 1. Model of systems engineering elements

- The Product elements will be modelled using objects of different semi-formal or formal object-oriented languages. A possible scenario is shown in Figure 1.

By defining a global traceability meta-model it will be possible to enable the consistency of the different engineering artefacts as shown in Figure 2.

Thanks to that MBSE global framework it will be possible to develop product families where the product is designed as family of products by starting with the definition of the stakeholders visible product characteristics (external variants) and internal product characteristics (internal variants) before going into the definition of all the possible detailed product performances and different alternative solutions. The product family consists of two levels (system and subsystems), associated through the traceability links declared in the engineering artefacts global meta-model and trough the product characteristic that are subject to variation (variants) at all levels. The reuse approach is thus recursive across and within the different system levels.

In responding to a new request for proposal (RFP), a gap analysis then has to be performed in order to identify the set of product characteristics (variants) of the product family that fulfill the needs stated in the RFP and the set of needs to be addressed with specific (outside of the product family) custom solutions. Once one identifies a coherent and homogenous set of family product characteristics that covers completely or, more often, partially the needs required by the RFP, then a specific product will be instantiated from the family and the specific custom solutions (if necessary) will be added.

To enable this, it is necessary to formalize the variability of the product line. A possible language that can be used to model the variability and declare the artefact dependencies is Orthogonal Variability Modeling (OVM) (Chale Gongora, Ferrogalini and Moreau 2015, 239-56) where basic notation is shown in Figure 3 below:

In parallel to the several engineering models, the Rail Transportation domain created a variability model, with the artefact dependencies (the links between engineering artefacts objects and variation points or variants). The final picture of a product family model in which engineering artefacts are described by models (or have a software representation) that can be easily

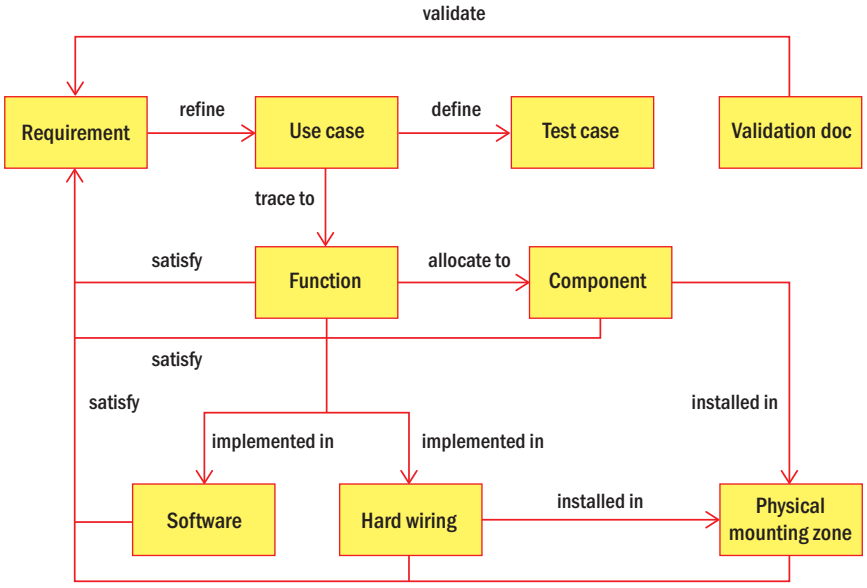


Figure 2. MBSE global meta-model

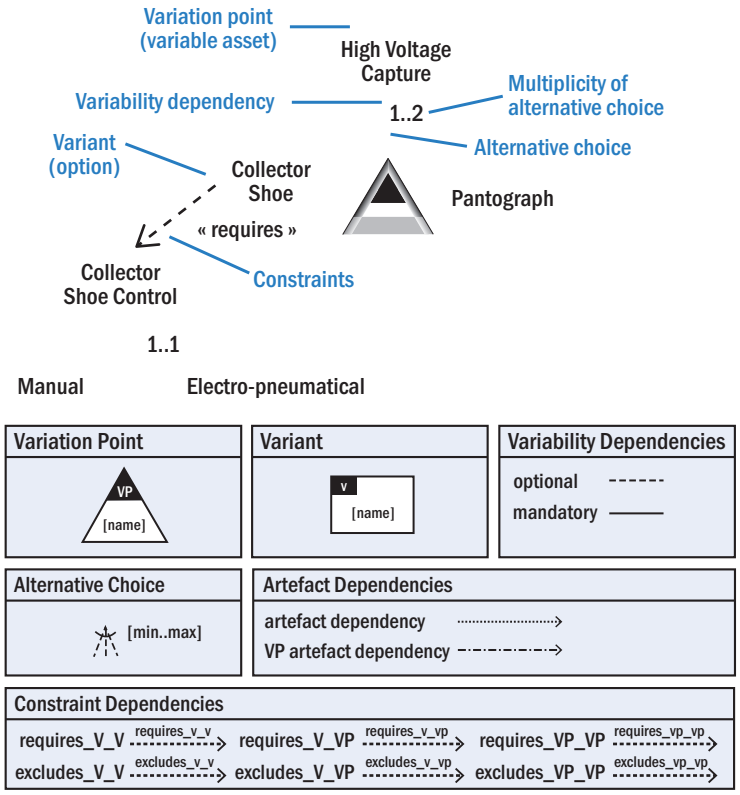


Figure 3. Basic OVM notation

reused to generate a specific product *via* an instantiation process is shown in Figure 4.

The product family variability model includes all the possible variation points and variants of the family, at all the possible system decomposition level (system, subsystem, components, and so on) and in all the different engineering domains. A key step of this approach is to create exhaustive and consistent artefact dependencies through multiple models, while respecting the inclusion and exclusion constraints between variants, as well as the relations between the objects as defined by the global traceability meta-model. The application of the described approach on new product

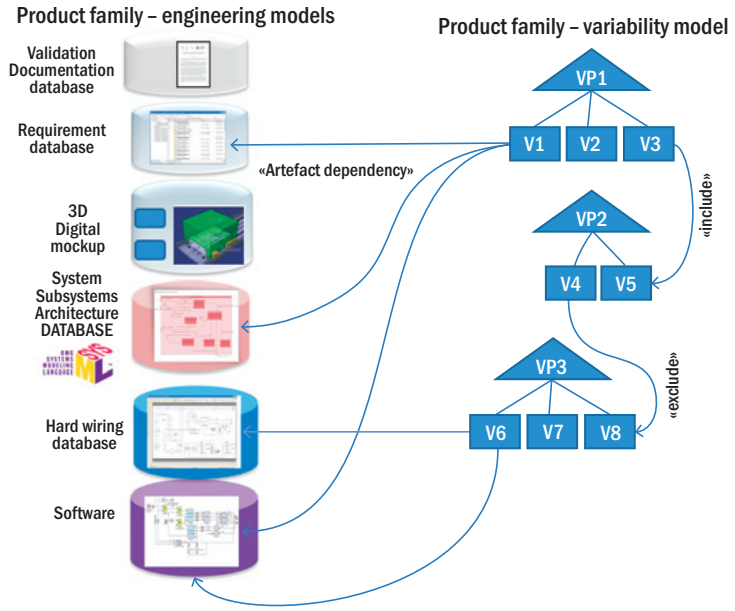


Figure 4. Overall picture of a product family model

family development has shown during the instantiation of the first derived product a dramatic reduction of the engineering effort (around 50%) in comparison with the traditional "one-shot" product development (Chale Gongora, Ferrogalini and Moreau 2015, 239-56).

CONCLUSIONS

In the actual context of the Rail Transportation domain, where MBSE has been introduced quite recently and the pilot projects phase has just ended, MBSE cannot reasonably be deployed on a running project; it will take more time to get to the same deliverables than it will using a traditional document based approach. Even if it is almost clear that the level of quality of the deliverable will not be the same (much higher with MBSE), then management normally will decide to remain in the known traditional approach which reduces the risk (according to their understanding) of not delivering on time (Chale Gongora, Ferrogalini and Moreau 2015, 239-56).

It seems then that MBSE will be deployed in such a type of context when systems engineers will demonstrate the ability to handle the new approach (methodology and tool) so well that they will be capable of delivering at least to the same schedule of the traditional document-based approach, and it is quite clear that this will take some time.

On the other hand, a "small" revolution is up coming in the Rail Transportation domain, which occurred in other sectors like automotive, and it is the introduction of the product families or the product line engineering (Chale Gongora, Ferrogalini and Moreau 2015, 239-56). It is in this new context that the introduction of MBSE seems more interesting and feasible. Thanks to the model-based approach, if the methodology is well conceived, as already mentioned, it will permit the Rail Transportation domain to create a catalogue of building blocks which will create a product family system model which will be partially instantiated in the context of a given product/project. The feasibility in terms of timing stands on the fact that the scheduling for a product family development is not particularly linked to a specific/project or customer and can be adjusted in order to permit the usage of the MBSE approach. Alternatively, an MBSE framework to support such Product Line Engineering framework is quite complex and requires some years of development and a high level of maturity in the language and in the methodology to model a railway system.

To conclude, we can state that MBSE in the Rail Transportation domain is now sorting from the exploratory research phase while having demonstrated the added value and related benefits; its deployment and adoption will be pursued in the next decade to take the place progressively of the old fashioned document-based approach (Chale Gongora, Ferrogalini and Moreau 2015, 239-56). ■

REFERENCE

- Chale Gongora, H.G., M. Ferrrogalini, and C. Moreau. 2015. "How to Boost Product Line Engineering with MBSE; A Case Study of a Rolling Stock Product Line." In *Complex Systems Design & Management; Proceedings of the Fifth International Conference on Complex Systems Design & Management CSD&M 2014*, edited by F. Boulanger, D. Krob, G. Morel and J.C. Roussel, 239-256. CH: Springer International Publishing.

AUGUST 2015
VOLUME 18 / ISSUE 2

MBSE Implementation across Diverse Domains

Ronald S. Carson, ronald.s.carson@gmail.com

(*This article is based on the 2014 INCOSE International Workshop presentation by Carson, et al.*)

MOTIVATION

As discussed in this issue and in the annual MBSE workshops, MBSE has taken root in many organizations and programs. Much of the reported positive experience to date is from single, small programs, with few analysts. The question of scalability is often raised, especially by larger organizations and those with existing product lines, for whom the transition to MBSE is not as simple as "starting fresh" on a new project with a small group of early adopters. For this article, "scalability" includes size (quantity of model elements) as well as the ability of the MBSE implementation to address widely different types of systems and projects.

IDENTIFICATION AND DEFINITION OF DIVERSE DOMAINS

Selecting an MBSE strategy in a large organization requires consideration of many factors. These include the variety of products and services, size and complexity of products and services, phases of the lifecycle, and organizational knowledge and culture regarding systems engineering, modeling, and complex, data-based-driven tools. Therefore "diversity" itself is a multi-dimensional function of organizational history (legacy products and life-cycle phases), staff history (knowledge and skills regarding systems engineering, various products, modeling, and different computer-based tools), and future plans for products, services, and organizational skills. This is made more complicated by a local organizational desire not to compromise near-term business performance for the sake of either future business outlook or other elements of the organization.

CHALLENGES FOR MBSE DEPLOYMENT ACROSS DIVERSE DOMAINS

This diversity becomes important if one is attempting to implement common approaches across diverse domains, because the common solution must be able to accommodate such diversity. In the absence of commonality, each element of the organization may be tempted to select solutions optimized for their business (lifecycle phase, product line, and customer set), organizational capability (knowledge and skills), and business outlook without regard to the other elements of the organization. Let us examine a few of these.

The ability to deploy MBSE depends, in part, on the knowledge and skills of the members of the organization in addition to process and tool infrastructure.

Business Considerations – Product Line over the Lifecycle

In aerospace as in other fields, some products designed and manufactured more than 50 years ago remain in service. The original engineering and production information is in the form of paper. While some of the information may be in electronic form, perhaps only as snap-shots in portable document format (PDF), much of the engineering analyses and drawings were not converted. Adding capabilities to existing systems via new technology insertion or reengineering unobtainable components creates an opportunity to apply MBSE in the update activity. However, the new engineering and product data must be integrated with the existing information, which itself creates data management challenges for support, training, and future technology updates.

This challenge is inverted in a transition to a new platform or system architecture using some legacy elements of the system; the larger, system architecture may be created in an MBSE environment, but needs to be integrated with legacy component design, manufacturing, and support information. Use of purchased or off-the-shelf components is a special case of important information not found in the same form as the newly created MBSE information.

Specific kinds of systems may be more or less amenable to different MBSE techniques. Software-intensive systems developed using Unified Modeling Language (UML®) may have an easier time transitioning to MBSE using SysML®. Complex aircraft and space systems that use physics-based models must be able to integrate structural and functional hardware models from a variety of disciplines into a coherent set of (if not fully integrated) models. These create challenges for MBSE methods and tools.

One must consider stakeholder needs for the engineering information. Where customers or regulatory authorities require text-based reports and deliverables, the model-based systems engineering capability must be able to faithfully export these from the models. This raises the configuration management challenge of "where is truth?" regarding the information. Is it in the deliverable, or in the model? We also need to understand and ensure satisfaction of data management, archiving, and retrieval needs over the product lifetime, which may be decades for certain products.

SCALING FOR SIZE AND COMPLEXITY

Related to product line considerations, is the need to have MBSE techniques

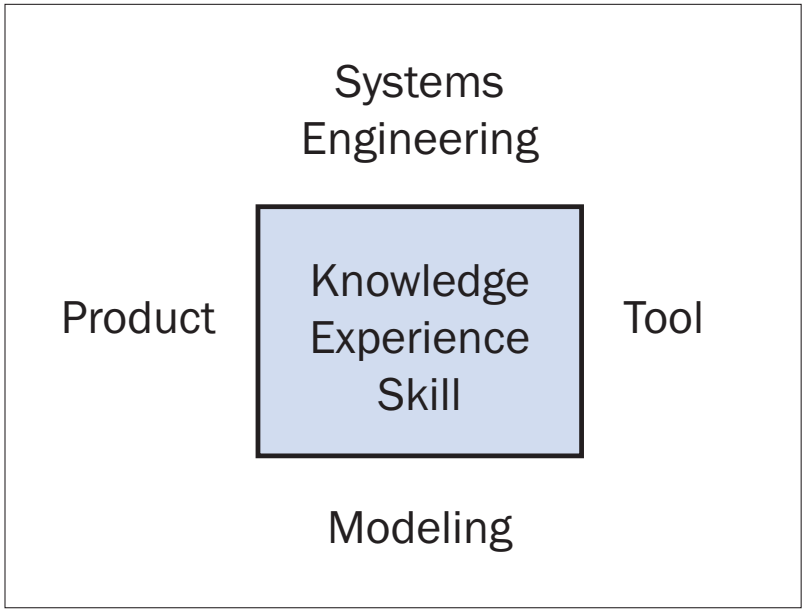


Figure 1. Four categories of skills and knowledge for MBSE

available for the diversity of both small research and development programs as well as very large and complex systems involving hundreds of organizations. Methods and tools that support a few or dozens of pieces of equipment may not fare well when confronted with gigabytes of data, hundreds of suppliers, thousands of interfaces, and millions of database objects requiring concurrent access and synchronization of content. Even standard methods are subject to local tailoring and variation to the point that data is not automatically interchangeable. Use of data interchange standards and a thoughtful program data and tool architecture are necessary to reduce the risk of federated data architectures that do not integrate. Complexity of new systems and subsystems exacerbates the size problem for new products or technology insertion. Moreover, product complexity often accompanies organizational complexity and diversity, especially for large products.

ORGANIZATIONAL CAPABILITY AND CULTURE

The ability to deploy MBSE depends, in part, on the knowledge and skills of the members of the organization in addition to process and tool infrastructure. Being able to tailor methodology to support diverse programs, products, and lifecycle phases means that common processes and tools must accommodate these consideration. Different tools may be required to support different methods (object-oriented vs. functional), but each must support the standard, common organizational processes. Tailoring of standard commercial tools and languages (SysML®) may require standardizing a subset of the overall capability for common use. For maximum effectiveness, individuals may become expert in a subset of the different tools and product areas and be able to recognize and recommend the best approach for a given product and lifecycle situation. Given the complexity of some modern tools, it may be important to have tool experts as well as systems engineering and product domain experts. This may require cadres of systems engineering experts in specific methods, tools, and product domains (Figure 1) that can deploy to help start programs or help programs make a transition to adding an MBSE capability.

Establishing an MBSE modeling capability may be one of the harder elements to teach and train. It requires systems engineering knowledge, product knowledge, skill with the modeling tools, and the judgment to model enough to solve the problem at hand without burdening a program with data that will not be used for further design or analysis. This would be an expected role for a system architect who was also proficient with particular tools.

Changing the organizational culture to support MBSE may also require recovering a more expansive view of systems engineering as more than requirements and risk management. This requires being able to relate the other systems engineering activities (architecting, verification) to program outcomes, such as requirements development and verification closure. This is easier when the organization can use the data in databases and tools to identify gaps and conflicts in design and compliance, and transition systems engineering data into downstream activities of detailed design, verification and certification, test, product support, and training. Then MBSE may sell itself, but only after some realized success.

DIVERSE SOLUTIONS

It should be apparent that in a large, diverse organization, the deployment of MBSE might not be "one size fits all." The MBSE capability requires tailoring and may therefore be different for different products, services, and lifecycle phases.

Legacy programs may apply MBSE to an incremental addition or modification as a means to introduce the new methods and tools, without disrupting an existing system and program. This may leave existing data untouched. Using common pedagogical techniques to "tell, show, do, apply" requires the MBSE deployment activity to "come along side" a program to teach, train, and support the transition to ensure adoption. If one hopes to change the organizational culture, this can be a multi-year commitment.

For new programs the challenge is to gain adoption early during the program definition phase. Short time-to-market leaves little time for learning MBSE once the program begins, yet there may be little demand to learn an MBSE approach prior to having a targeted program for application. This requires being aware of potential target programs and having just-in-time availability of mature tools, training, support, expertise, tailoring approaches, and troubleshooting prior to actually beginning the engineering. It also requires the encouragement of program leadership to recognize that the competitive environment likely requires changes to the development process for the sake of improved cost, schedule, and capability.

Organizational successes with small pilot or incremental activities on legacy programs help establish momentum to follow through on the transitions. Indeed, without continuous emphasis and resources to help make the transition to MBSE many organizations may fail to follow through because it is different and challenging. However, the persistence of the adoption generally requires support from outside a specific program to help guide the program through the challenges of adopting MBSE and improving organizational systems engineering effectiveness. ■

Stand out from the crowd become SEP certified

CTI's CSEP exam preparation course will equip you with essential understandings, tools and tips to maximize your likelihood of passing the exam.

CSEP Exam Preparation Course

This 4-Day CSEP Exam Preparation Course combines a mixture of presentations, workshops and practice examinations. In particular, the Q&A sessions ensure that any unclear topics are answered in a format that provides adequate time for the content to be mastered. All our courses are facilitated by world-class leaders, who are highly experienced and knowledgeable in dealing with all aspects of CSEP training.

CTI's 4-Day CSEP Exam Preparation Course provides you the perfect grounding to take the INCOSE CSEP examination with confidence.

CERTIFICATION TRAINING INTERNATIONAL
A PPI Company

Upcoming Deliveries

August 17 - August 20, 2015	**Austin, TX**
October 5 - October 8, 2015	**Amsterdam, the Netherlands**
November 16 - November 19, 2015	**Chantilly, VA**
December 7 - December 10, 2015	**Los Angeles, CA**
April 25 - April 28, 2015	**London, United Kingdom**

Consider an on-site course for your team

CSEP public training courses are available on-site worldwide. Take advantage of:

- huge savings on course fees
- flexible timing – take at a time best suited to your training needs
- tailoring of content to best satisfy the needs of your enterprise
- training a project team together
- fostering teamwork as a result of team knowledge
- shared understanding of principles and terminology

Find out more by visiting: www.certificationtraining-int.com

PROJECT PERFORMANCE INTERNATIONAL

Improve the results of your projects in 2015 with PPI

Helping Projects Succeed...

Learn leading edge methodologies together with sound principles that will help your organization reduce costs, meet schedules, and exceed stakeholder expectations. PPI has a worldwide reputation for providing high quality training, in major disciplines necessary to achieve project excellence.

Upcoming Project Performance Improvement Training in 2015

Systems Engineering

August 31 - September 4	Amsterdam, the Netherlands
September 14 - September 18	Munich, Germany
September 21- September 25	London, United Kingdom
October 5 - October 9	Las Vegas, NV
October 26 - October 30	La Spezia, Italy
November 23 - November 27	Eindhoven, the Netherlands
December 7 - December 11	Las Vegas, NV

Systems Engineering Management

October 12 - October 16	London, United Kingdom
November 9 - November 13	Amsterdam, the Netherlands
December 7 - December 11	Las Vegas, NV

Requirements Analysis & Specification Writing

August 31 - September 4	Amsterdam, the Netherlands
October 26 - October 30	Boston, MA
November 23 - November 27	London, United Kingdom

Requirements, OCD & CONOPS in Military Capability Development

November 30 - December 4	Munich, Germany

Human Systems Integration: The Cognitive Element

October 26 - October 30	Las Vegas, NV

"Gave fantastic details on systems engineering and a solid basis to build on"
– delegate, Australia

"The course was very thorough. It covers the most pertinent topics in improving a systems engineering organization"
– delegate, USA

"The course exceeded my expectations in meeting my objectives. Improvement of the course may be hard!"
– delegate, UK

Improve the performance of your organization and enhance your career by taking PPI's training.

www.ppi-int.com

AUGUST 2015
VOLUME 18 / ISSUE 2

Modular Applications Functions – A Westinghouse MBSE Tool

Gary B. McWilliams, gary.mcwilliams@att.net

INTRODUCTION

The Automation and Field Services (AFS) Business Segment of the Westinghouse Electric Company provides large, highly-integrated instrumentation and control (I&C) systems for nuclear power plants. These systems are applied to both new plants and I&C retrofits of existing power plants, and provide seamless, comprehensive I&C functionality. The systems are typically made up of a number of sub-systems using various product platforms driven by nuclear plant requirements. The application engineering of these I&C projects typically consists of the complete design process, including design basis, functional, system, and detailed design, and verification and validation (V&V) and licensing support.

Reports from the industry at large indicate building these highly-integrated systems efficiently is challenging, especially in effective system integration and in obtaining predictable results in terms of system performance, budget, and schedule. In some well-publicized cases, the schedule for the building of new power plants has been paced by the implementation of the associated I&C systems.

With the construction of a new generation of nuclear power plants occurring worldwide, it is critical that the issues experienced by the I&C industry at large be resolved. The model-based Modular Applications Function (MAF) process, described below, is helping Westinghouse to establish a consistent, repeatable, and efficient process for implementation of I&C systems for its AP1000® nuclear reactors (*AP1000 is a registered trademark of Westinghouse Electric Company LLC, its affiliates and/or its subsidiaries in the US, and may be registered in other countries throughout the world. All rights reserved. Unauthorized use is strictly prohibited. Other names may be trademarks of their respective owners*).

MODULAR APPLICATIONS FUNCTION (MAF) PROCESS

The MAF process supports efficient continuous systematic integration of I&C designs during the waterfall design process phases, beginning with the specification of system requirements, and proceeding through implementation. "Integration" is frequently defined as the assembling of parts of a system following their individual implementation. However, the MAF process employs the concept of "continuous integration" throughout the implementation process, including the requirements, design, manufacturing, and verification/validation phases. The goal of the MAF process is to establish an integration view throughout all steps of the implementation process so that rework of parts developed within any given phase is unnecessary.

MAF PROCESS DESCRIPTION — LEVEL 1

The appropriate point in the overall application design process to begin to address integration is during the establishment of the detailed requirements for the I&C system, before design decisions are made to achieve system implementation. By defining the functional design in modular form before implementation, the most important step is taken in the design of an integrated system. The industry is recognizing that the formalization of the system design layer and the establishment of its implementation-independent functional requirements contribute substantial benefits to the overall implementation process.

The functional design step is defined as "Level 1." The Level 1 process step captures the I&C system design functionality using a pre-defined library of platform-independent modular functional elements. These elements specify the functionality, interfaces, and configuration data required to implement the function. These specifications are independent of platform and other design decisions to the extent possible, and lend themselves to database implementation techniques, which facilitate downstream automated sub-system implementation processes. Because this level of the MAF process is independent of implementation platform, the functional design representation can include multiple subsystems without regard to subsystem boundaries.

MAF LIBRARY

The MAF library consists of the platform-independent elements to be used in the functional design definition. Each MAF element begins with a specification of the module, including its functionality, interfaces, and configuration data. The scope of the functional elements is not critical to their use in the process, but the reusability of the modules is optimized to provide the best results in terms of integration and efficiency. The individual functions are determined by those who specify the functional requirements for the systems and include related requirements such as IEC 61226 or IEEE 603 categorization requirements, definition of external interfaces, and division or separation group.

In order to generate usable MAFs easily, a tool was developed to automate the generation of a configured Visio symbol from a formatted functional MAF specification. This tool enhances standardization of MAFs and supports more efficient specification of the modular functional design.

Strategies were developed for the handling of certain functions of critical importance to integration. Among these areas are data quality signals, redundant sensor algorithms, component control, signal naming, and alarm handling. These strategies provide a coordinated direction for implementation of the function across sub-systems, and, therefore, provide the basis for generation of related MAFs.

MAF APPLICATION TOOL

In order to support the Level 1 functional capture requirements of the MAF process and the overall MAF process structure, a software-based tool was developed. A wide variety of third-party platforms were reviewed, including tools that are primarily modeling tools with some drawing support capability and tools that are primarily drawing tools with some modeling support capability. In particular, a number of Unified Modeling Language (UML) standard modeling tools are available to support the design of large, software-intensive homogeneous systems. However, the desire to configure the tool to be consistent with nuclear industry and Westinghouse standards led to the selection of Microsoft® Visio. Visio has excellent drawing support, sufficient modeling capability for the desired functions, ease of interface with Microsoft® Access, ease of adaptation for creation of industry- and Westinghouse-standard I&C diagrams, and widespread use in the engineering workforce. *Note that Microsoft is a trademark or registered trademark of its respective owner. Other names may be trademarks of their respective owners.*

MAF PROCESS DESCRIPTION — LEVEL 2

Level 2 of the MAF process consists of the I&C system design. The system design is approached by viewing the entire I&C system as a single entity from the design perspective, despite potentially being composed of multiple systems and platforms.

The Level 2 effort begins with definition of the I&C system architecture using a pre-defined library of architectural elements, analogously to the library of MAFs. The architecture definition includes the platform-specific architectures for each of the subsystems that will accept the functional definition from Level 1. This process step can be done more competently at this point, having the knowledge of the overall functional design requirements identified in Level 1. Once the overall system architecture and the architectures of the individual subsystems are defined, the Level 1 functional modules are assigned to the architectural elements using safety categorization, physical location, platform functionality, or other considerations.

For example, modules with a safety-grade categorization (IEC 61226 Cat A or IEEE 603 "1E") are assigned to architectural elements that carry a safety-grade qualification. Functional modules containing hardwired I/O may be assigned to controllers near plant elements they connect to. Special platform functions, such as alarm functionality, are assigned to the relevant architectural elements.

In addition to mapping of functionality to architectural elements, Level 2 provides the ability to consistently map module interfaces to architectural communication channels. For example, the interfaces collected between the group of modules assigned to one particular controller and those assigned to another controller may be implementable using highways, datalinks, or hardwired connections.

The Level 2 system design provides another point in the top down design process where integration and validation checks can be performed before proceeding to implementation steps.

MAF PROCESS DESCRIPTION – LEVEL 3

Once the functional and system design levels exist to the degree of completeness that supports the initiation of the implementation phase, criteria are established to allow the application requirements to be mapped and sorted to facilitate the implementation of each of the sub-system areas.

For example, functional modules that are assigned to a particular controller within a control system are identified and sorted according to the architectural elements that comprise the system to be implemented. Configuration data accompany the identified modules to establish a complete set of implementation requirements for the appurtenant sub-system.

To dovetail with the respective implementation processes, the existing sorts of implementation requirements can be mapped via pre-defined design databases to provide platform- or sub-system-specific implementation modules that correspond

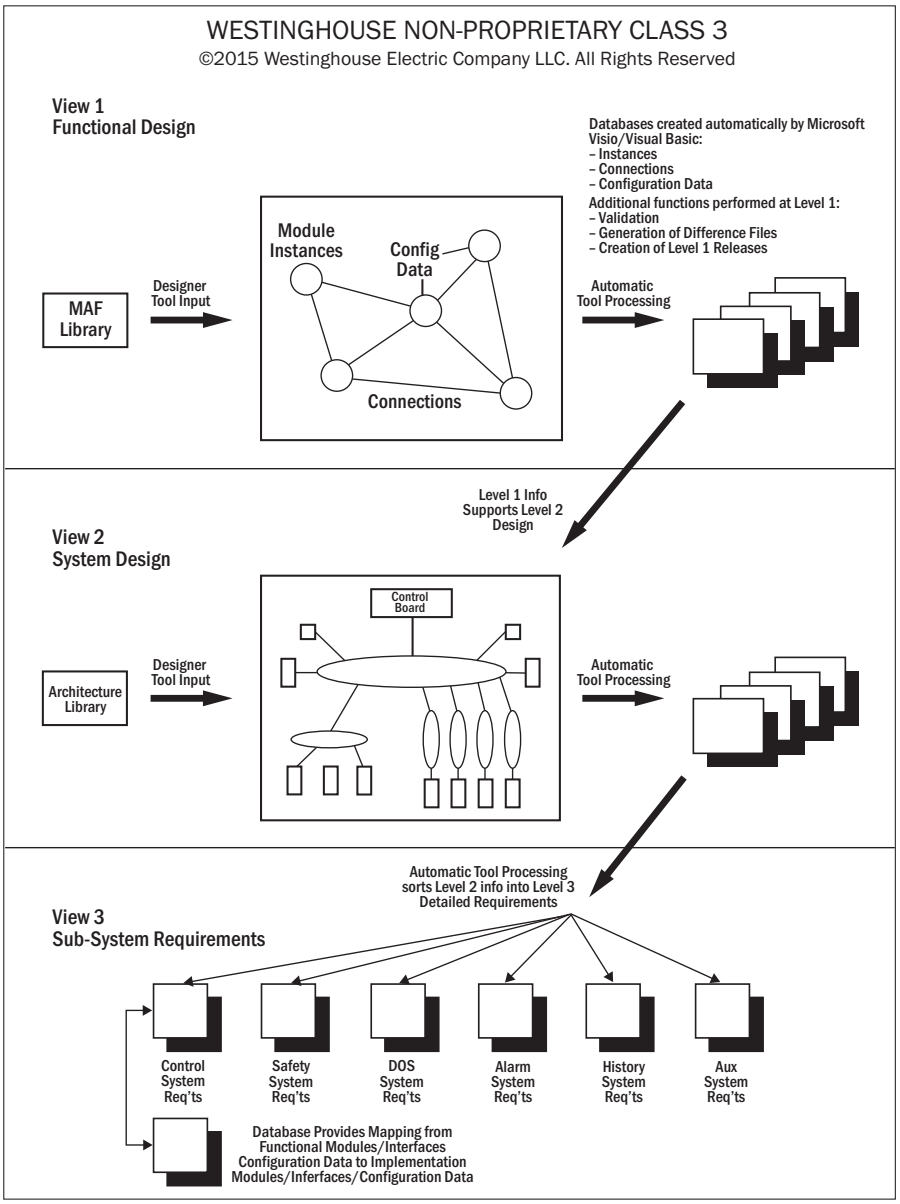


Figure 1. MAF Process

to the identified functional module. By this mechanism, the process of implementation becomes more automated, less error prone, and more efficient.

Requirements that apply only to sub-systems with special, optimized architectures, such as alarm systems, are sorted out using mechanisms specific to that architecture to facilitate its implementation.

Analogous to the other levels, the Level 3 phase provides the final point in the MAF process to perform validation checks of the implementation requirements before the implementation actually begins.

Figure 1 shows the overall MAF process including the three levels discussed.

TRAINING & INSTITUTIONALIZING

Application of the MAF process to the existing Automation and Field Services (AFS) organization required changes in the way large I&C systems are implemented. Substantial training is required to provide the necessary understanding to allow the process to be implemented properly and efficiently, and to overcome existing cultural habits. In order to assist with the conversion, the following efforts were undertaken:

- Generation of clear, formal processes that are consistent with existing processes
- Development of a tool that is intuitive and thoroughly tested
- Conversion of significant portions of an I&C design prior to use by the project application team.

The generation of clear and consistent processes is necessary to provide the basis for the project team's work. Otherwise stated, "The processes must be integrated before the products can be integrated." The application of the MAF process by the MAF team to ongoing project work provided a "proof of concept" to the project team and reduced the chaos that is likely to accompany a sudden change in work procedures. After the initial completion of an entire project, the MAF process will be ingrained in the cultural environment.

CURRENT STATUS & OPERATIONAL EXPERIENCE

The MAF process development began in early 2007 with a conceptual design of the ideas described above. The development of the process and supporting tools followed quickly in order to provide results as expediently as possible. Time was of the essence because of the need to support the safety system development for the AP1000® generation of Westinghouse nuclear reactors, which had already begun. As the Level 1 step of the MAF process was being implemented, the remaining steps of the process were being developed. MAF library development was prioritized according to project needs and executed in parallel to the tool development.

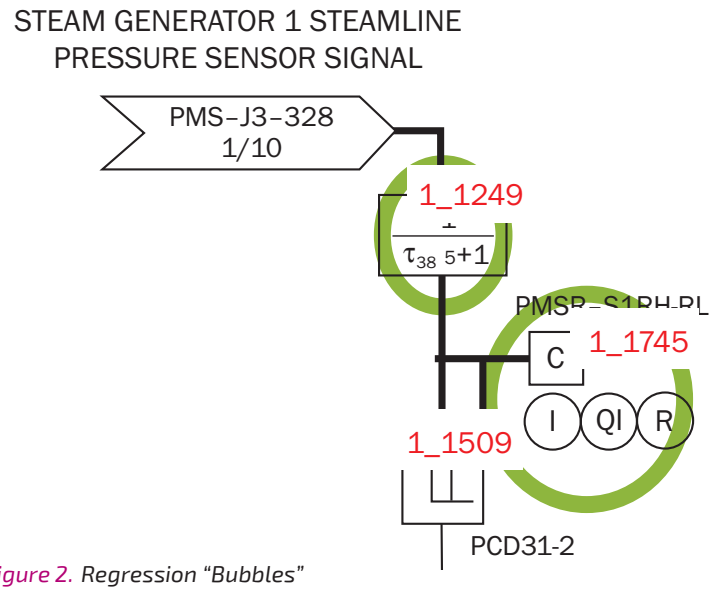


Figure 2. Regression "Bubbles"

The AP1000® plant Protection and Monitoring System has been completely implemented using the MAF process and tools. During the development, continuous improvements to the process and tools have been incorporated. For example, in order to support numerous design modification cycles of the AP1000 PMS development while keeping the overall design integrated, significant regression analysis support has been added. Among the support functions provided are the automatic generation of colored diagram "bubbles" with unique identifiers (see Figure 2) and new/old tables on data reports to identify items that changed from the previous version of the design, including functional, interface, and configuration data changes. These automated tool functions improve usability of the process and tools significantly, and facilitate generation of reports for external purposes (such as regulatory updates).

As the Level 1 process step of the MAF process was the first to be developed, it has the most operating history and has been the most evaluated step to date. According to the functional/system design group, a comparison of estimates for the AP1000® plant work performed with the MAF process shows the use of the MAF process for the Level 1 step resulted in a 30% reduction in design costs as compared with the same scope of work performed without the MAF process. This estimate is based on a comparison of the costs of the Level 1 design step and does not include auxiliary benefits from the application of the MAF process, such as reductions due to fewer required iterations of the design. It is estimated that a corresponding savings of at least 30% is being experienced due to these reductions in design iterations (included in some organizational metrics as the cost of poor quality).

The success of the MAF process is resulting in the process being institutionalized within the Westinghouse AFS business segment for large I&C systems.

CONCLUSION

The development of the MAF MBSE process and associated tools is primarily complete, although numerous refinements are still in progress. The safety system for the Westinghouse AP1000® nuclear plant I&C system was developed using the MAF process. As system engineering discussions of MBSE tools predict, there were substantial benefits seen, including elimination of integration issues, rework and efficiency improvements, and verification and validation improvements. The future application of the process to I&C projects is expected to provide substantial improvements in the technical integration of I&C systems built by Westinghouse AFS.

ACKNOWLEDGEMENTS

The author would like to acknowledge earlier efforts within Westinghouse in the area of modular I&C system design techniques, including the Sizewell B installation in the U.K., the Temelin installation in the Czech Republic, the Kozloduy installation in Bulgaria, and the Ringhals 2 installation in Sweden. ■

Lead.
Build Expertise
Solve Industry Challenges.

Master of Science in Systems Engineering

The M.S. in Systems Engineering at Kennesaw State University prepares graduates to lead engineering teams and manage complex interdisciplinary projects, solving the industry challenges of the 21st century.

Following completion of the master's degree, our graduates are prepared to move into positions of leadership in their current organizations, but are also equipped for new employment and entrepreneurial opportunities in the defense, space, transportation, energy and telecommunications industries, as well as other fields that seek their expertise in engineering and managing large and complex systems.

Program highlights:

- Courses delivered completely online.
- A graduate certificate (4 courses) and a Master of Science degree program (12 courses) are available.
- Flexibility that affords students the opportunities to select or customize a concentration sequence to meet their specific career goals and/or objectives.
- Courses taught by industry-expert faculty.

For more information about the degree or certificate programs:
GRADSYEKSU@kennesaw.edu
678-915-3060
engineering.kennesaw.edu/systems-industrial

"The most valuable aspect of the program was its direct applicability to my job. Classes were extremely relevant and carried real-world applications, as many were taught by industry experts, including some from my current employer Lockheed Martin. This is invaluable for the professional who is working full time and seeking an advanced degree."

- Brian Ehrich

Alumnus, Southern Polytechnic College of Engineering and Engineering Technology, MS in Systems Engineering, Class of 2008
SENIOR SYSTEMS ENGINEER
AT LOCKHEED MARTIN,
MARIETTA, GA.

KENNESAW STATE UNIVERSITY®
Southern Polytechnic College of
Engineering and Engineering Technology

AUGUST 2015
VOLUME 18 / ISSUE 2

Academia as an Educator in the World of MBSE

Russell Peak, russell.peak@gatech.edu; **Leon McGinnis,** leon.mcginnis@gatech.edu; and **Chris Paredis,** chris.paredis@gatech.edu

What role(s) is academia best suited to play in the world of MBSE? As with many such questions, the answers typically start with "It depends…" and there is probably not one single "right" answer. That said, in this article we provide our perspectives based on our direct involvement with MBSE and SysML since 2002 (see http://www.pslm.gatech.edu/topics/sysml/), which we hope will be helpful for others. Broadly speaking, academia probably best participates in MBSE in much the same way it best participates with any newer technology: with education (including undergraduate students, graduate students, and working professionals) and with research (including theory, demonstration applications, fostering collaboration, and analyzing trends).

This article focuses on the educational facets—both our own experiences at Georgia Tech as well as at other universities. We gave an overview to some of our early MBSE/SysML-related research in the 2009 MBSE Special Issue of *INSIGHT*. Research efforts continue and grow, and we share that work regularly in multiple venues including the recent MBSE workshop at the INCOSE International Workshop 2015 in Torrance, US-CA (see our research publications at the above pslm website, as well as http://mbsec.gatech.edu/ and others including http://www.omgwiki.org/MBSE/doku.php?id=mbse:modsim).

EDUCATION – GRADUATE DEGREE STUDENTS

In our courses, we tend to focus on SysML as the primary means to understand and implement MBSE concepts in a particular sense (as well as "systems modeling" in a broader sense). This approach is analogous to utilizing a specific computer programming language (such as python) to learn generic computer programming concepts. Given that background, the rest of this article employs the phrase "MBSE/SysML" as a shorthand way to say we are teaching MBSE concepts by using SysML as a specific implementation mechanism.

Our first attempt to teach MBSE/SysML was in the fall semester of 2007, with a graduate-level course aimed at "traditional" on-campus students. We offer the course every fall semester, with class sizes ranging from 8 to over 20 students who primarily come from the Aerospace Engineering (AE), Industrial Engineering (IE), Mechanical Engineering (ME), and Architecture departments.

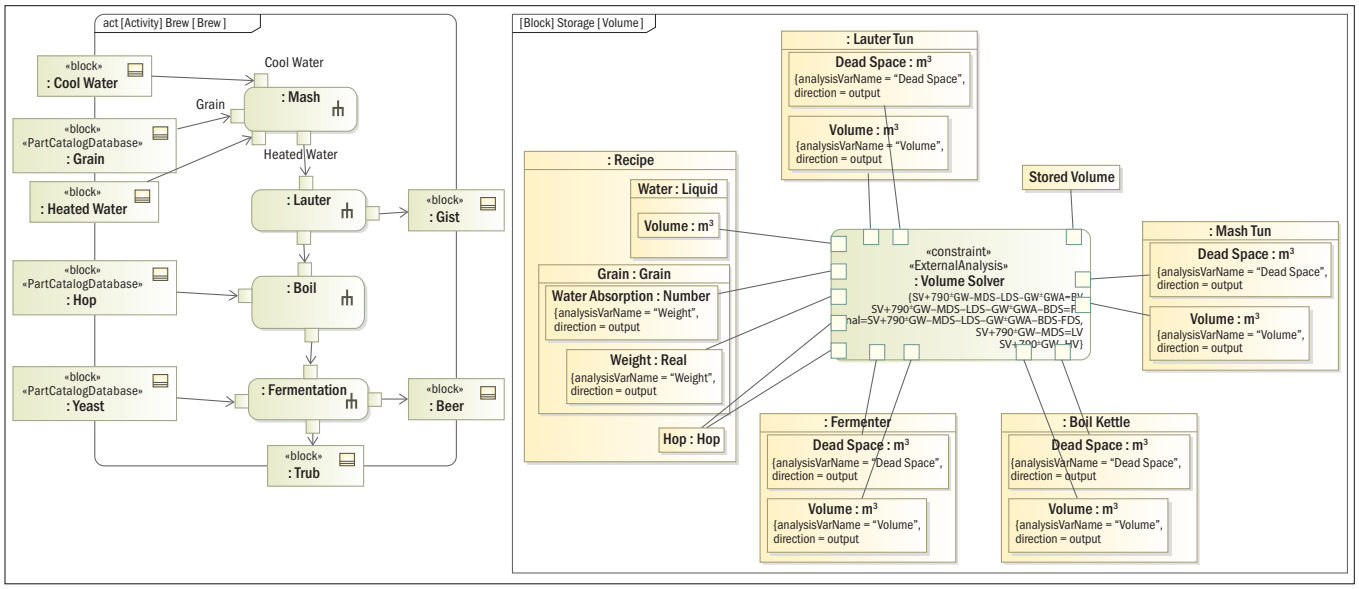
In the early offerings, the most often-asked question about SysML was "Why are we doing this?" The notion of a system model bereft of any internal analytical power did not seem useful to the students, most of whom have no experience managing large-scale complex systems development. In our most recent offering, this question never arose, because today we are able to integrate our SysML models, created using MagicDraw™, with analysis components in ModelCenter™ via Phoenix Integration's MBSEPak™, or InterCAX's ParaMagic®. In other words, SysML provides the platform for capturing a formal specification of the system of interest and for developing a range of tightly integrated analysis and synthesis models.

Of course, now that we have the tools to both create and analyze system models, we are better able to introduce students to the concept of MBSE methodology. In part because of its excellent documentation in an MBSE/SysML textbook, the methodology we present is Object Oriented Systems Engineering Methods (OOSEM) (Friedenthal, Moore, and Steiner 2015).

Our graduate level course requires a term project that uses SysML and demonstrates a system design or analysis result. As an example, one graduate student, an avid beer brewer (and over 21 years old), developed a model to specify a home brew system. His model included the desired attributes of the beer, the entire brewing process, and the parametric relationships necessary for sizing the system. Coupling the SysML model through MBSEPak to analyze components developed in Excel resulted in a "design synthesis" tool, where the least-cost hardware components could be selected to brew a desired type of beer in a desired volume.

Over the past seven years, there have been many "lessons learned" about how to teach MBSE to students in traditional degree programs. In contrast to early offerings, we now spend very little time in class teaching the specific tool, because we captured all of that in videos, which the students can watch while they work through the examples themselves. We have learned that "object thinking" is not natural for some people, and we have to spend more time guiding the students in learning that skill. Perhaps most importantly, given the modeling and analysis tools now available, an MBSE methodology is essential—otherwise every student invents their own, and this lesson carries over into the practice of MBSE as well.

Increasingly we see other universities offer similar MBSE/SysML-focused graduate courses, including Cornell, George Mason University (GMU), Loyola Marymount University (LMU), University



(a) SysML Activity Model for Brewing Process	*(b) SysML Parametrics to Calculate Kettle Volume*

Figure 1. Graduate Student Beer Brewing Project – Selected Diagrams from SysML Model

of Maryland, Missouri University of Science and Technology (MST), Rose-Hulman Institute of Technology, Southern Methodist University (SMU), and probably others we are not aware of yet. We also are beginning to see other departments within Georgia Tech experimenting with making MBSE/SysML a part of their systems classes (Civil Engineering/Building Construction/Architecture, and also Computer Science).

The other main graduate-level MBSE/SysML mechanism we offer is within our Systems Engineering Master's degree program (www.pmase.gatech.edu), developed for people who have already been working full-time for at least four years ("non-traditional" off-campus students). The program is a hybrid format where each course in the two-year program combines asynchronous distance learning with regular live webinar sessions, along with three extended face-to-face sessions. The course that focuses on learning MBSE/SysML is ASE 6005,

Table 1. Recent SysML-based Team Projects in ASE 6005.

Short Name	Project Title & System-of-Interest (SOI)
GT60	GT60 Commercial Microturbine CHP (power generation equipment – gas turbine; CHP=combined heat power)
SGS	Smart Grocery System Product Line 2013 (smart grocery system)
SPlane	Solar Plane (small manned solar plane)
DropX	DropX.1, Site-To-Store-To-You (unmanned aircraft system delivering site-to-store orders to customers)
Hybrid Boat	Eco Power Boat Product Line 2013 (plug-in hybrid leisure boat)
Fit-Er	Fit-Er Product Line 2014 (wearable computing device: fitness tracking monitor)
Jurassic Park	SmartGen's Jurassic Park(materials & supplies logistics system – food/water/meds/etc. for dinosaurs & humans)
WFAP	Wildfire Firefighter Assistant and Protector (UAV-based fire monitoring & communications system)
HEMS	Home Energy Mgt. System (residential alternative energy mgt. system - monitoring & cost optimization)
TacCOM	Tactical Cellular Communication Network (mobile battlefield communications system)
E-Pedigree	Ollie's E-Pedigree System (pharmaceutical serialization, authentication, and tracking system)
NDV	Next-Gen Delivery Vehicle (hybrid commercial vehicle for the local home/business delivery industry – e.g., UPS)
MiDi	Migraine Diary (wearable biometric device for migraine mgt.; interfaces with remote medical professionals)
EATS	Evaluating Aspects of Traveling Sustenance (food truck industry – mobile food service operations and logistics)
DeathStar	DeathStar: The Other 364 Days (military space station materials & supplies logistics system – food/equipment/etc.)

which has 25-30 students per year. After learning the basics of each main concept area in SysML, students apply their knowledge in individual projects and in team-based projects. One difference versus the above "traditional" graduate course is that all PMASE courses occur in an accelerated 7-week format (twice as fast as the normal semester schedule). Thus, students must come up-to-speed with a large volume of material and concepts in a rather short period (including building executable SysML activity models and SysML parametrics models). Students who have had some experience with object-oriented thinking (via object-oriented programming or similar exposure) tend to pick up SysML faster.

Table 1 summarizes recent team projects (and their diversity of topics), developed via an introductory MBSE process based on OOSEM. The DropX project highlighted in Figure 2 investigates what types of UAVs one needs to cover metropolitan areas of varying size and package delivery profiles. This project is particularly interesting when you consider it completed in July 2013, which was several months before Amazon and others began talking publically about their own UAV-based package delivery plans.

ASE 6005 occurs towards the end of the first year. Thereafter, most PMASE courses in the second year utilize SysML as one means to extend the students' systems engineering knowledge and experience. This approach is quite similar to how undergraduate engineering students learn calculus early in their education, and then apply calculus in most of their third- and fourth-year engineering courses.

Increasingly, some ASE 6005 students are already using SysML in their workplace before they take our courses. Others are part of organizations that have heard about the benefits and are willing to try it, often with the former student helping to lead the way. Seeing students successfully apply MBSE/SysML after graduating is particularly rewarding for us as instructors. Periodically the two worlds combine and alumni are able to apply and extend their PMASE class project topics within their work environment. Two recent examples include (a) technologies for logistics and security in the Arctic, and (b) extracorporeal membrane oxygenation (ECMO) systems (effectively a longer-term heart-lung machine) (Adams and Pihera 2013).

EDUCATION – UNDERGRADUATE DEGREE STUDENTS (AND EARLIER)

We have experimented with teaching two kinds of MBSE-related courses to undergraduates. One entitled "MBSE" focused on teaching the basics of SysML and OOSEM, with simple example systems, which require no detailed prior knowledge. It has been taught three times, with enrollments in the 20-30 range, half ME students, half IE students. The second was titled "System Design for Industrial Engineers", offered twice to similar enrollments, focused on using SysML as the platform for understanding design of "typical" IE systems, such as warehouses, factories and supply chains. We do not usually teach the concepts of context, function, architecture, and embodiment in the context of warehouses, factories, and supply chains, and this presents a teaching challenge, when combined with learning both SysML and OOSEM.

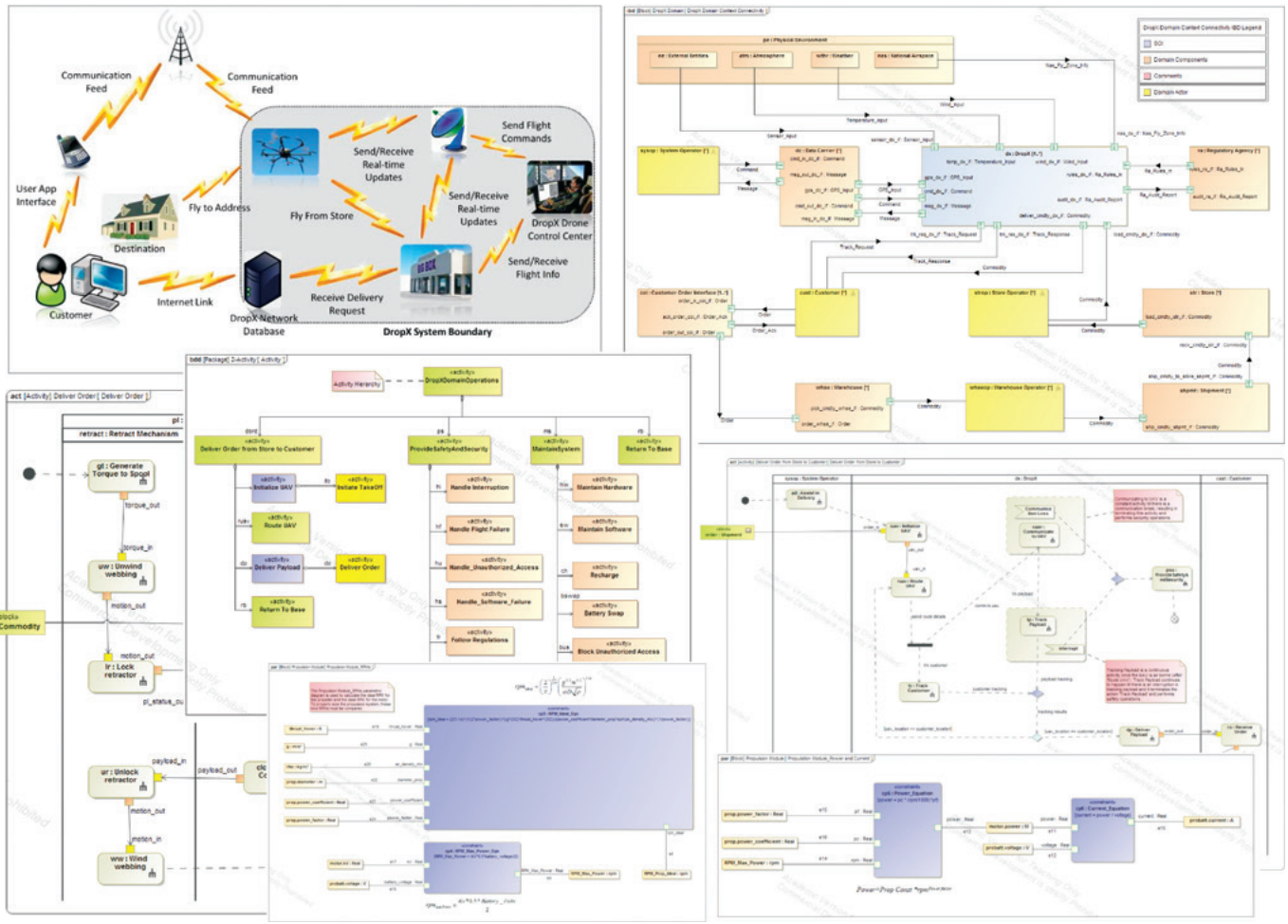


Figure 2. Team Project in ASE 6005: DropX – A UAV-based Package Delivery System (July 2013) – Selected Diagrams from SysML Model

In comparing experiences across the three cohorts (traditional graduate students, professional MS, and traditional undergraduate), the undergraduates tend to have less experience thinking about abstract concepts such as system architecture, and thus we focus on a subset of SysML capabilities with them. Today, however, the most significant hurdle is the availability of engineering faculty with the knowledge and interest to teach these topics. This represents a significant challenge to the US MBSE practitioner community, because the international community seems to be moving much more rapidly to integrate MBSE into educational curricula. In fact, there are reports that even high schools in France are teaching SysML. This is not surprising if you consider that SysML is a kind of knowledge representation language, which can be useful beyond just systems engineering applications.

EDUCATION – WORKING PROFESSIONALS

Starting in August 2008, we began offering MBSE/SysML short courses for working professionals (in collaboration with InterCAX LLC). The overall goal of these short courses is to equip participants with useful concepts and practical expertise that they can immediately apply in their jobs. While these short courses are not for academic degree credit, they do usually count for continuing education units (CEUs), as well as several professional education certificate programs.

SysML 101/201 is our introductory MBSE short course in a 3.5-day combination format. Participants learn how to interpret SysML constructs as well as how to implement those constructs utilizing a representative SysML tool. This approach is analogous to learning how to read and understand a foreign language, as well as how to write a proper document in that language. They experience utilizing their models to do useful things, including performing engineering calculations and collaboratively working on a single shared model all at the same time. One highlight is a capstone exercise (Figure 3) where the class divides into three teams who compete in developing and delivering a rover mission design (as well as a model of the rules and structure of the competition itself). The competition culminates with execution of their designs on Bluetooth-enabled mobile robots (all driven from their SysML models) and scoring against both quantitative and qualitative measures.

The relationship between our academic-degree-credit courses and these short courses is synergistic and mutually beneficial. The SysML 101/201 short course material (including hands-on step-by-step tutorials and associated starter models) initially formed core content for the academic courses. Over time the academic courses evolved and expanded, and in some cases provided the basis for additional short courses. For example, today the ASE 6005 course described above continues to utilize the SysML 101/201 material (including numerous pre-recorded lectures and tutorial demonstrations, which enable self-paced learning). Recently we started offering two new short courses based on the ASE 6005 experiences (i) for individual-sized projects (SysML 891) as well as (ii) for team-based projects (SysML 892). These new short courses are a great way for organizations to kick-start MBSE within their actual projects – they leverage a proven process that creates a useful model over an intensive 4-day period (or alternately spread out over several weeks to allow for more student-led model development time in between instructor-led sessions).

Other universities started offering similar short courses for professional education, including California Institute of Technology (Caltech) and the University of California San Diego (UCSD). Some organizations prefer a "vendor-neutral" learning approach, and universities are ideally suited to provide that. Figure 4 on the following page shows the cumulative number of participants in our Georgia Tech/ InterCAX short course program over time. Since August 2008 we have conducted 187+ short courses for 3320+ professionals in the US and internationally. Most are contract short courses (delivered onsite at the

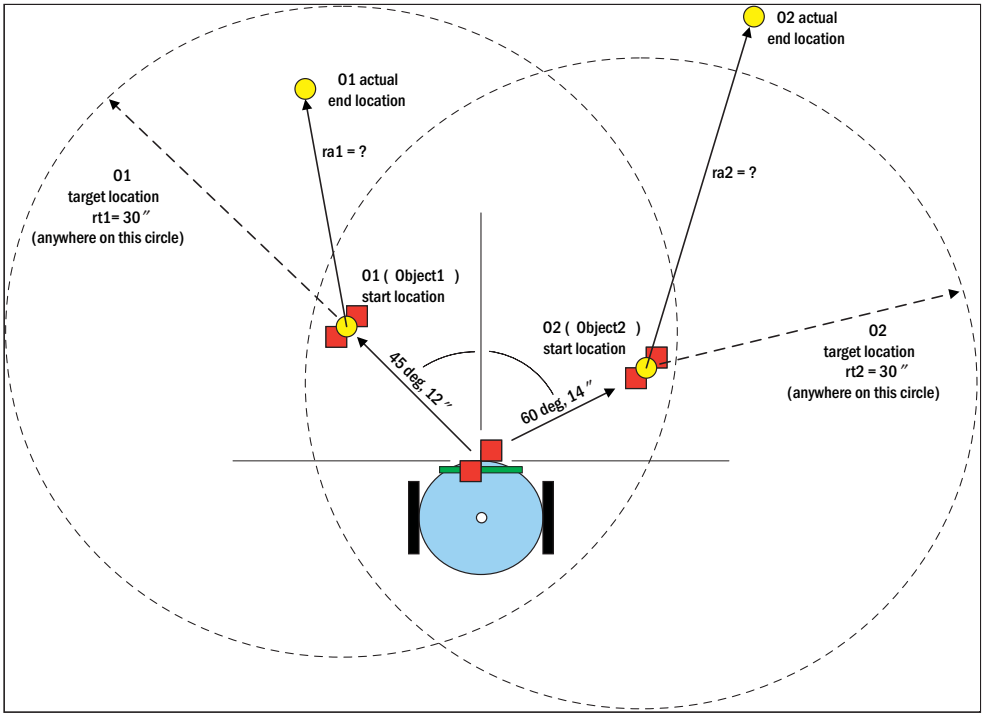


Figure 3. Capstone Exercise in SysML 101/201 Short Course – Rover Team Competition with Executable Mission Designs

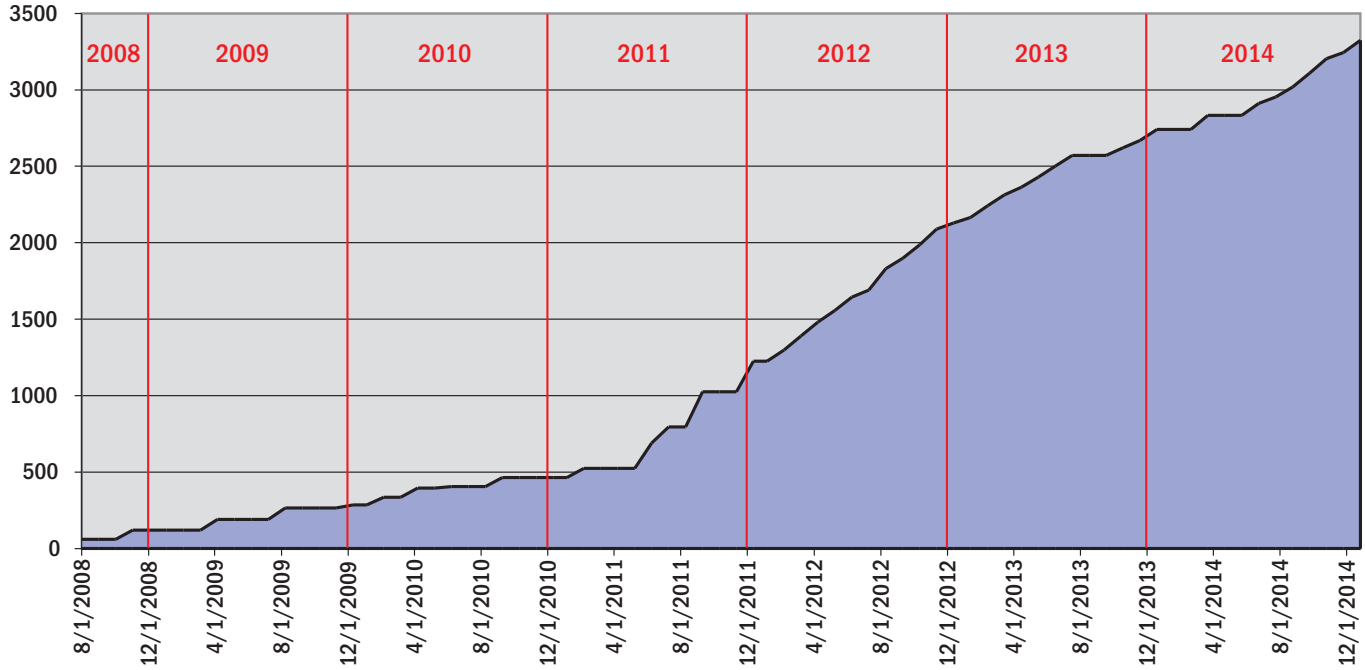


Figure 4. SysML Short Courses – Cummulative Number of Participants Over Time (as of Dec 2014)

sponsoring organization), and some are public offerings where practically anyone can register. While the majority of participants are from the aerospace and defense industry, a healthy number are from commercial industries including the automotive, biomedical, and energy sectors. The graph is useful too, as one measure of the MBSE/SysML adoption rate as a whole. Note how it follows the typical technology adoption curve such that in mid-2011 the adoption rate picked up quite a bit. Incidentally, the effect of the US federal government sequestration/budget events in 2013 and 2014 had an immediate impact on training and travel budgets, which this graph depicts. Overall, SysML-based MBSE adoption is now continuing at a steady pace.

CONCLUDING REMARKS

Organizations often come to us and ask "How can we hire more students that have learned SysML/MBSE in their university education and research experience?" One key is for industry and government organizations themselves to motivate the students to learn and utilize SysML/MBSE. How do they do that? There are two ways. One, by making it clear in their request-for-proposals and hiring advertisements that having such skills confers a significant hiring advantage. This can be further implemented by offering students summer internships where it is clear that such skills will be required (and applied to interesting topics). Another facet, which is especially important for graduate students, is research sponsorship and close collaboration with faculty. These are the methods we have seen organizations successfully apply to gain MBSE/SysML-savvy new hires. The second way to encourage the growth of SysML/MBSE-prepared graduates is to emphasize the importance of this emerging area to academic administrators, and encourage them to make SysML/MBSE a part of their degree programs.

We hope the above summary is helpful (a) to understand how we have approached teaching MBSE/SysML at Georgia Tech over the past 8 years in a variety of contexts, and (b) to motivate others in their teaching, learning, and collaboration pursuits. The overwhelming response from our class participants is that having a unified language such as SysML makes the "system model" concept much more tangible and explicit. Otherwise, that concept typically evokes at worst a blank stare and at best a vague, implicit combination of loosely dependent documents (plus a few narrowly-focused models in some cases). Many of us believe that MBSE/SysML is doing for systems engineering what CAD/CAE did for mechanical engineering in the 1980s/1990s (Lee 1999). The analogies and benefits, as well as the investments, lessons-learned, and pitfalls-to-avoid are strikingly similar. ■

REFERENCES

- Adams, N. and L. D. Pihera. 2013. *Informing ECMO Therapy Improvements through Systems Engineering Principles & Practices*. Paper presented at the Conference on Systems Engineering Research (CSER), Atlanta, US-GA, March 19-22.
- Friedenthal, S., A. Moore, and R. Steiner. 2015. *A Practical Guide to SysML, Third Edition: The Systems Modeling Language*. Burlington, US-MA: Morgan Kaufmann.
- Lee, K. 1999. *Principles of CAD/CAM/CAE Systems*. Reading, US-MA: Addison-Wesley Longman Inc.

Polarsys: A New Collaborative Ecosystem for Open Source Solutions for Systems Engineering Driven by Major Industry Players

Gaël Blondelle, gael.blondelle@eclipse.org; **Francis Bordeleau,** francis.bordeleau@ericsson.com; and **Daniel Exertier,** daniel.exertier@thalesgroup.com

INTRODUCTION

PolarSys (www.polarsys.org) is the Eclipse Working Group dedicated to open source solutions for systems engineering. It was created in 2012 by major industry players at the convergence of two main trends: the maturity of open source ecosystems and the fact that the conventional commercial tools ecosystem did not fully satisfy the growing needs for extensible and long lasting MBSE tools.

Companies like Airbus, Ericsson, and Thales rely on software development technologies and tools to maximize engineers' productivity in order to reduce product development time, cost, and risks, and to get to market faster. The use of leading-edge technologies and tools is essential to maximizing capabilities to innovate and develop key product differentiators. In some domains such as the development of critical embedded systems, these tools are also key to demonstrating compliance with safety standards.

As we will see in the following briefings from Ericsson and Thales, the critical nature of this dependency and the need to resolve certain tooling issues, led these organizations to take control of their own tools destinies. They dedicated significant resources to solving issues by leveraging open source as an enabler for open innovation – hence, the creation of PolarSys.

ERICSSON SUPPORT FOR PAPYRUS

In "Model-Based Engineering: A New Era Based on Papyrus and Open Source Tooling," Francis Bordeleau, Product Manager Software Development at Ericsson, writes that after 25 years of using Model Based Engineering (MBE) tools, we observe that:

1. **UML provides the required foundation for implementing the MBE vision, but tools are still a main issue** – The adoption of UML as an industry standard has brought major benefits in terms of communication, technology reuse, and hiring of qualified personnel. It also fostered the establishment of a vibrant research community focused on developing a broad range of UML-based capabilities. However, the tools are still a major obstacle to the broader adoption of MBE.
2. **Existing commercial modeling tools only provide a subset of the required MBE capabilities** – End-users have an ever-increasing need for a broad range of capabilities related to different development aspects. Moreover, end-users need better tool usability, customization, and integrations to improve their productivity. In spite of all of the research on the different MBE aspects over the last decades, there is still no integration of the research into existing commercial tools.
3. **No single company can provide all of the required MBE capabilities** – It is simply not possible for a single company to have expertise in all aspects needed to develop the required capabilities. Any of the existing commercial tools only provide a (small) subset of the capabilities required by end-users. Therefore, in addition to the adoption of UML as a standard language, we need the MBE community (end-users, suppliers, and research/academia) to converge to a common UML tool/platform that allows contributions from different companies and research groups in an open and incremental manner.
4. **Proprietary technologies are a major obstacle to innovations and technology transfer** – The restriction on proprietary tools/platforms preventing use in an open manner by researchers force them to develop their own prototype tools to conduct research. This situation creates two major problems: it creates a difficult obstacle to the transfer of new capabilities to end-users due to technical and legal issues, and it results in a high percentage of research resources being invested in the development of prototype tools that bring no significant value. As a result, end-users get very little out of the research investments made in the different MBE aspects.

Based on these four main observations, Ericsson concluded that open source is the only way to fulfill the overall MBE vision. Consequently, in 2012, Ericsson began investing in Papyrus to provide an open source alternative to existing commercial tools.

Since then, Ericsson and other industrial partners have worked closely with CEA List and the open source modeling community to improve Papyrus to reach the required level of industrial readiness to enable large scale deployment at Ericsson.

THALES BEGAN WITH CLOSED DEVELOPMENT OF ITS MBSE SOLUTION, THEN MOVED IT TO OPEN SOURCE

This experience report comes from the interview "How Thales published their modeling tools in open source (http://www.lemagit.fr/article/Comment-Thales-a-place-ses-outils-de-modelisation-dans-lOpen-Source)," by Daniel Exertier, Manager of the Model Driven Engineering Domain at Thales. Exertier says: "At Thales, we have a strong culture of technological innovation, which is rarely shared with the market of engineering environments." He went on to explain that for Thales, modeling is a centerpiece for improved engineering productivity and systems quality, but engineers were reluctant to use existing tools from the market, because they could not meet their operational engineering needs.

So, in the late 2000's they decided to implement MBSE solutions that specifically fit with these needs. They created the Arcadia (https://www.polarsys.org/capella/publis/An_Introduction_to_Arcadia_20150115.pdf) methodology and the Capella tools and deployed them to large projects for systems engineering, particularly in Space, Avionics, and Defense.

In 2013, Thales realized, during adoption, that the closed nature of the tooling was an obstacle to larger deployment. Subcontractors and partners were seeing a risk in using Capella due to its proprietary nature. Consequently, Thales decided to share the technology under open source with the goal of establishing it as a de facto standard. They set out to grow the community by partnering with players in the entire engineering value chain, from industry and professional services and technology providers, to researchers, thereby establishing the necessary open innovation dynamics.

OEM'S AND LARGE ORGANIZATIONS ARE READY TO TAKE A BIGGER ROLE IN SHAPING THEIR SYSTEMS ENGINEERING TOOLS

These experiences show that while continuing to work with large software vendors, end-users are entering into a new phase in their approach to tools and want to have more influence on the future of the tool chains they depend on. It corresponds to a new development alternative enabled by open source and open innovation.

In the 1980's, most companies were developing their own tools (including operating systems, programming languages, compilers, debuggers, and version control systems).

During the 1990's, software development tool companies emerged allowing end-users to focus on their core businesses. In some cases, they outsourced a subset of their tools to these new software vendors.

During the 2000's, widespread consolidation occurred in the sector with several side effects: some tools disappeared, some products used a radically more aggressive commercial approach, and generally, users had to face several painful forced tool migrations.

At the same time, important open source initiatives like Eclipse emerged as relevant platforms for tools, including model-based tools with the Modeling Top Level Project (https://www.eclipse.org/modeling/emf/).

In response to the needs of large organizations for better tools to improve their productivity and ability to innovate, the need for extensible platforms for research, and the maturity of Eclipse Modeling technologies, several initiatives appeared between 2005 and 2010:

- Topcased (http://www.topcased.org/), initiated by Airbus as a systems engineering platform for safety critical embedded systems;
- Papyrus, created by CEA List to fully implement UML/SysML specifications and serve as an extensible platform for both industrial end-users and researchers; and
- Arcadia and Capella, respectively the methodology and its dedicated tool support created by Thales for systems engineering activities.

ECLIPSE AS AN ENABLER FOR OPEN INNOVATION IN OPEN SOURCE

Since its inception, the Eclipse Foundation (https://eclipse.org) has created an ecosystem to enable collaboration among competitors on platforms, and the creation of added-value products on top of the platform. Organizations leverage the Eclipse approach to implement open innovation through open source, by collaborating on open platforms and enabling the creation of extensions, open source or not, on top of the platform.

In 2009, the Eclipse Foundation initiated Working Groups (WG) (http://bit.ly/eclipse_iwg) to support industry driven communities that leverage this approach in specific areas such as IoT (Internet of Things), location aware technologies and automotive.

WGs leverage the state-of-the-art open source infrastructure and governance of the Eclipse Foundation, add a specific industry focus, and extend its governance and infrastructure.

POLARSYS, THE OPEN SOURCE COMMUNITY FOR SYSTEM ENGINEERING TOOLS

Since its creation as an industry-driven open source community, the goal of PolarSys has been the same: to create an open source ecosystem to build and maintain sustainable open source tool chains for systems engineering and for the development of embedded systems.

PolarSys is committed to open innovation with open source; better extensibility with a technology platform with no lock-in where under the Eclipse Development Process, any company, or research organization can add features and, better interoperability through access to all data formats.

As the PolarSys platforms grow, they enable faster tools development by extension and adaptation to specific domains or organizations.

To support the ecosystem, PolarSys provides new common services such as project maturity assessment (http://dashboard.polarsys.org), a common platform for testing, common marketing initiatives, and a common infrastructure for very long-term support (VLTS).

One of the basic operating rules of VLTS is the capacity to build and test a patched version of a solution in one week during the first ten years of support. After this period, it turns to lower expectations in terms of reactivity (one month can be acceptable), but with higher expectations in terms of durability, that is, it must be operational as long as the tool is in use.

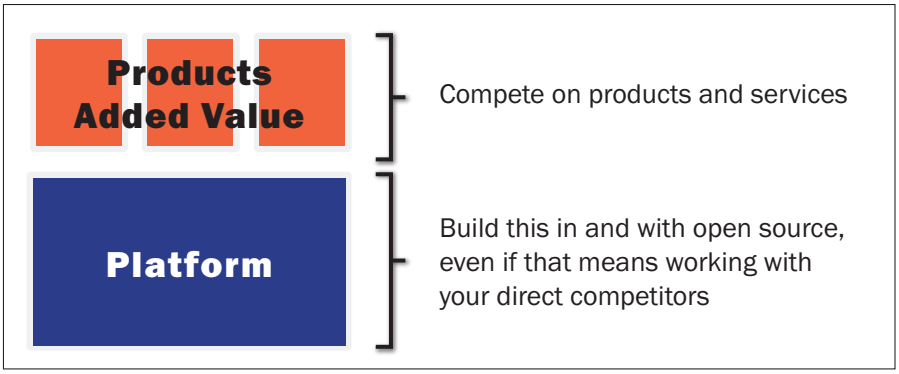


Figure 1. An extensible ecosystem

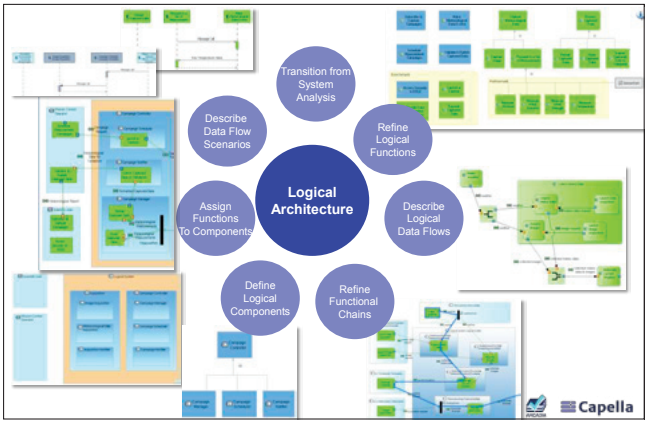


Figure 2. Examples of viewpoints for Arcadia/Capella Logical Architecture

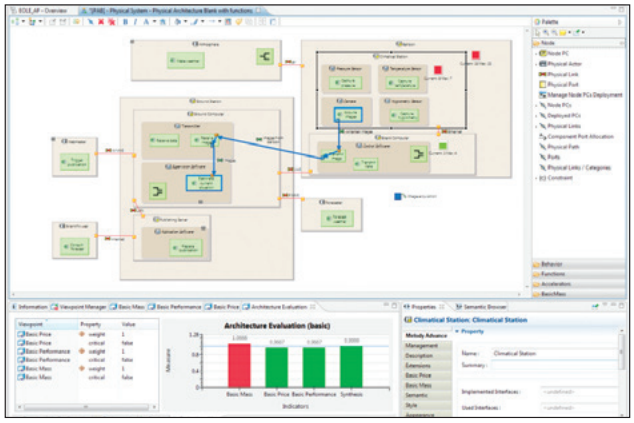


Figure 3. Architecture trade-off with Capella using non-functional viewpoints

POLARSYS SOLUTIONS BRIDGE THE GAP BETWEEN OPEN SOURCE PROJECTS AND OPEN SOLUTIONS

In 2014, PolarSys introduced the concept of "PolarSys solutions (http://www.polarsys.org/solutions)." The goal here was to bridge the gap between open source projects and open source solutions deployed by PolarSys members. This means they have passed extensive tests, provided documentation, created marketing collateral, and are associated with reliable commercial support offerings.

One of the unique advantages of PolarSys solutions is that versions of those solutions are only online when deployed by large organizations on real projects. Coupled with the maturity assessment, this is considered extensive proof that the solution is mature enough, and the professional services ecosystem is structured enough to replicate such deployments in other large organizations.

The current list of solutions includes Capella (http://www.polarsys.org/solutions/capella), Egit/EGerrit for configuration management, Papyrus (http://www.polarsys.org/solutions/papyrus), C/C++ development tools, and Titan for testing.

CAPELLA, THE SYSTEMS ENGINEERING TOOL FOR ARCADIA

Capella provides the tools necessary for engineering-wide collaboration of stakeholders. Through shared reference architectures and the ability to apply different viewpoints, users can understand, master and define complex architectures, and make documented decisions on architectural trade-offs (see Figure 2).

Capella addresses various levels of architecture including operational, system, logical and physical architectures, and resolves non-functional constraints and trade-offs using non-functional viewpoints (see Figure 3).

UNIVERSITY OF MARYLAND
MASTER OF SCIENCE
SYSTEMS ENGINEERING

EARN THE DEGREE THAT PREPARES YOU TO:
- Create complex systems and services
- Become an effective technical leader-collaborator
- Reap the rewards of a career in systems engineering

MODEL-BASED SYSTEMS ENGINEERING
IN A SYSTEMS RESEARCH ENVIRONMENT

Since 1987, the Institute for Systems Research's MSSE degree has been educating systems engineers to apply a quantitative approach to analysis, design, tradeoff, and ensuring successful system operations.

Today our program stresses model-based systems engineering and enhances that approach for complex and multi-domain systems.

go.umd.edu/iimsse

INSTITUTE FOR
SYSTEMS RESEARCH
A. JAMES CLARK SCHOOL OF ENGINEERING

PAPYRUS, SYSTEMS ENGINEERING BASED ON SYSML, MARTE, AND ADVANCED EXTENSIBILITY FEATURES

Papyrus, the PolarSys solution for SysML and UML modeling, integrates with other Eclipse and PolarSys tools to fulfill lifecycle integration needs, such as source control (Git), document generation, and reporting. Through its integrated support for SysML (1.2 and 1.4), Papyrus provides the basis for the adoption and use of MBSE (see Figure 4).

Several extensions support the expression of non-functional constraints especially with MARTE (http://www.omgmarte.org/). Moka (https://wiki.eclipse.org/Papyrus/UserGuide/ModelExecution) is another extension for model simulation and animation (see Figures 5 & 6).

Although some extensions are still in incubation, Papyrus plugins demonstrate its extension capabilities.

CONCLUSION

Capella and Papyrus are the two cornerstones of the PolarSys response to the need for open source solutions for systems engineering. These solutions address a wide range of MBSE needs and large companies use them, including Ericsson and Thales.

In addition to building an open industry-driven ecosystem around those solutions, PolarSys addresses other domains and engineering activities. These include runtimes with the project Papyrus RT (https://projects.eclipse.org/projects/modeling.papyrus-rt) that provides both a customized version of Papyrus and the corresponding C/C++ runtime; safety engineering with ESF (https://www.polarsys.org/projects/polarsys.esf); and requirements engineering and traceability with Reqcycle (https://www.polarsys.org/projects/polarsys.reqcycle). Additional projects are also coming to PolarSys in the next months in the domains of simulation and product lifecycle management (PLM). ■

REFERENCES

- Bordeleau, F. 2014. "Future of MBE/MDE/MDD in the Industry—Open Source is the Only Solution!" Keynote presentation at ACM/IEEE 17th International Conference on Model Driven Engineering Languages & Systems, MoDELS 2014, Valencia, ES, September 28–October 3. http://models2014.webs.upv.es/index2_htm_files/Keynote2.pdf.
- Bordeleau, F. 2014. "Model-Based Engineering: A New Era Based on Papyrus and Open Source Tooling." In Proceedings of the 1st Workshop on OSS4MDE co-located with ACM/IEEE 17th International Conference on Model Driven Engineering Languages & Systems, OSS4MDE@MoDELS. Valencia, ES, September 28–October 3.

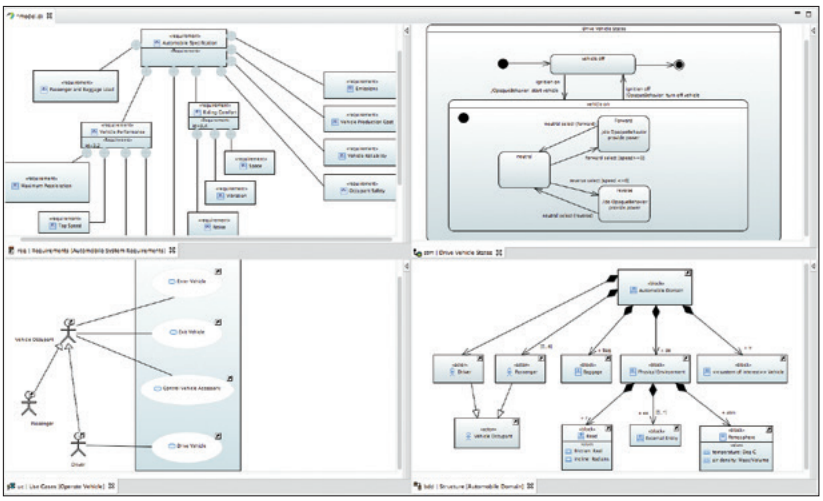


Figure 4. Papyrus SysML editors

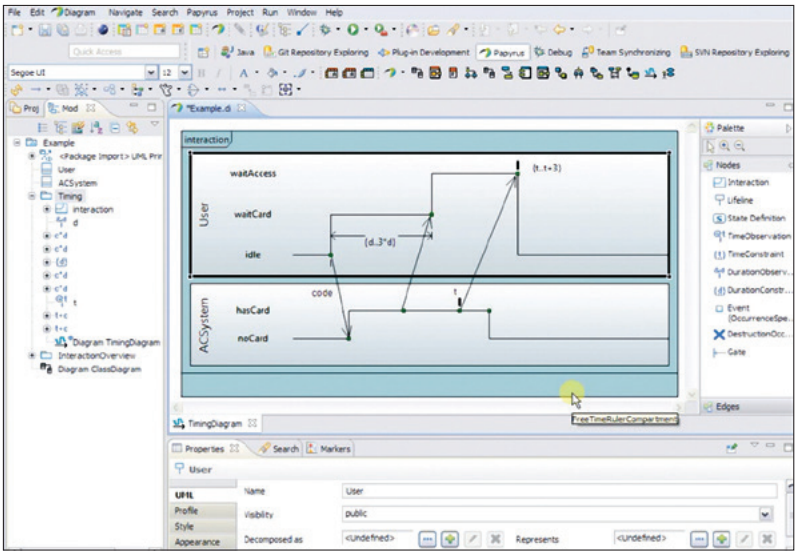


Figure 5. MARTE timing diagram

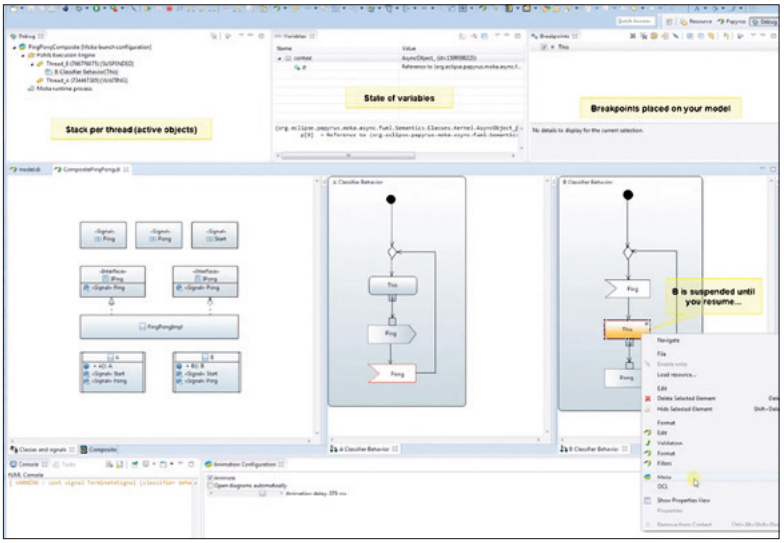


Figure 6. Moka, the Papyrus extension for model animation

AUGUST 2015
VOLUME 18 / ISSUE 2

Evolving SysML and the System Modeling Environment to Support MBSE

Sanford Friedenthal, safriedenthal@gmail.com and **Roger Burkhart,** burkhartrogerm@johndeere.com

INTRODUCTION

The transition to a MBSE approach is essential for systems engineering to meet increasing demands of system complexity, design cycles, productivity, and quality. Many other engineering disciplines, such as mechanical, electrical, and controls engineering, utilize models as integral parts of their practice. Such models have long been important for systems engineering as well, but MBSE emphasizes the need to create a coherent model of the system that helps to integrate other aspects of the design, including electrical, mechanical, and software. This **system model** provides a shared view of the system that can drive communication and coordination across the system development lifecycle. This model represents the authoritative source of information, maintained to ensure consistency and traceability across requirements, design, and verification. The model-based approach contrasts with the traditional document-based approach in which information exists across many different documents created in common applications such as Word, Visio, Excel, and PowerPoint. To take full advantage of a model-based approach, the system model maintenance is a fundamental part of the technical baseline, integrated with other engineering models and tools.

BACKGROUND AND CURRENT STATE OF SYSML

The Object Management Group (OMG) adopted the OMG Systems Modeling Language (OMG SysML™, www.omgsysml.org) in 2006 and Version 1 was available in 2007. The SysML specification resulted from a collaborative effort between INCOSE, the OMG, and the ISO STEP AP-233 Working Group (WG) to develop the requirements for the language, and then develop the system modeling language solution in response to these requirements.

SysML is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. The language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and constraints.

Since its adoption, SysML enabled broad recognition and increased adoption of model-based systems engineering practices across industry. Systems engineers learned much from this experience, in both the strengths and weaknesses of SysML as a language, and the benefits and challenges in adopting and applying MBSE.

The systems engineering community needs a standard system modeling language, such as SysML to express fundamental systems engineering concepts such as system composition, interconnections and interfaces, functional and state-based behavior, parametric aspects of a system, and relationships between requirements, design, analysis, and verification. Such a language is proving to be an essential capability to specify and architect increasingly complex systems, and to overcome the informational "Tower of Babel" in the systems engineering community. The capability to express system concepts in the form of models results in quality improvements by reducing downstream design errors and in productivity improvements through reuse of models across projects and throughout the lifecycle. Systems engineers realize other benefits, such as the ability to automate tasks such as change impact analysis, and to auto-generate reports and documentation with increased confidence that the information is valid, complete, and consistent.

A major emphasis of the SysML community has been on the integration of SysML models and tools with other engineering tools. There has been significant progress integrating system-level models with software design, especially for software developed with Unified Modeling Language (UML) models and tools, since SysML derives from UML. In addition, considerable progress has been made integrating SysML with engineering analysis and simulation using various integration methods and tools. SysML integrates extensively with requirements management tools. Some integration of SysML with product lifecycle management and hardware design tools exists, but much more remains to integrate.

A key challenge for MBSE is the learning curve to develop the modeling skills. Some of this challenge is inherent in the complexity of systems and systems engineering, such as the need to define and maintain multiple consistent views of a system. Other challenges are unique to the language and current modeling tools. The challenges of learning a more formal language, as well as the tools to build and maintain models, together with the MBSE methodologies for effective application,

all require significant investment by the organization to build an MBSE capability. Applying these capabilities on a program requires a disciplined approach to establish and maintain the system model as an integral part of the technical baseline throughout the system lifecycle.

FUTURE DIRECTIONS FOR SYSML

The OMG Systems Engineering Domain Special Interest Group (SEDSIG) chartered the System Modeling Assessment and Roadmap WG to assess how well SysML is supporting MBSE, and to develop a roadmap for SysML as part of a System Modeling Environment. The WG is beginning to identify driving requirements for the next generation of SysML and the tools that implement the language. Some of the initial capabilities and requirements are below. These are subject to further analysis, inputs, and review with the broader community.

System modelers who perform MBSE in the broader context of Model-Based Engineering (MBE) use a System Modeling Environment (SME). This environment must provide basic capabilities that impose requirements on both the modeling language and the tools. Some of the key capabilities for the SME include:

- Model construction
- Model visualization
- Model analysis
- Model management
- Model exchange and integration support for MBSE collaboration and workflow.

Some of the key effectiveness measures include:

- **Expressive**: Ability to express the system concepts
- **Precise**: Representation is unambiguous and concise
- **Presentation/communication**: Ability to effectively communicate with diverse stakeholders
- **Model construction**: Ability to efficiently and intuitively construct models
- **Interoperable**: Ability to exchange and transform data with other models and structured data
- **Manageable**: Ability to efficiently manage change to models
- **Usable**: Ability for stakeholders to efficiently and intuitively create, maintain, and use the model
- **Adaptable/Customizable**: Ability to extend models to support domain-specific concepts and terminology.

Based on the above capabilities and effectiveness measures, some of the preliminary driving requirements for the next-generation system modeling language and tools are as follows:

1. The next-generation modeling language must express the core systems engineering concepts. This requires definition of a robust data model that reflects these concepts. The requirements that drove SysML derive from the original Systems Engineering Conceptual Model, jointly developed by the INCOSE/OMG/AP233 WG requirements team. Modifications and refinements to this model will occur in light of lessons learned over the last several years, and as necessary to express the core systems engineering concepts.
2. The next-generation modeling language must include precise semantics that avoid ambiguity and enable a concise representation of the concepts. SysML currently leverages the UML metamodel for much of its semantic foundations. The language must derive from a well-specified logical formalism that can leverage the model for a broad range of analysis and model checking. This includes the ability to validate that the model is logically consistent, and the ability to answer questions such as the impact of a requirement or design change, or assess how a failure could propagate through a system. The language and tools must also integrate with a diverse range of equation solvers and execution environments that enable the capture of quantitative data.
3. The next-generation modeling language and tools must provide flexible and rich visualization and reporting capabilities to support a broad range of model users. SysML currently includes concepts for view and viewpoint. Tool vendors and end users have been able to apply this capability to query the model and provide flexible reporting capability. The next generation must extend this capability with advanced visualization techniques that include dynamic zoom, filtering, traversal of model relationships, and visualization of the dynamic behavior of a system, such as those provided by simulations. The modeling language must also support symbol libraries that extend well beyond the current SysML notations. In addition, the modeling environment must provide a simplified web interface to dynamically view the model from a diverse set of viewpoints.
4. The next-generation modeling language and tools must enable much more intuitive and efficient model construction. It often requires several clicks to capture a core concept in a model. More streamlined and efficient user interfaces could reduce the time and effort to build and maintain a model. The ability to repeat common modeling patterns with reduced user input (table-based entry) is another capability to increase modeling productivity and understanding.
5. The next-generation modeling language and tools must support MBSE in the broader context of Model-Based Engineering (MBE), where the models and tools fully integrate across discipline-specific engineering tools, including hardware and software design, analysis and simulation, and verification. All these model-based tools working together establish an environment for engineering the total system.
6. The next-generation modeling language must provide a standard application programming interface (API) to provide dynamic access to the model, while providing appropriate access controls. It should also integrate with emerging platforms for managing and integrating model-based content, such as Open Services for Lifecycle Collaboration (OSLC), which is based on linked data and semantic web technology, and the Functional Mockup Interface (FMI), which provides model exchange and co-simulation capability for executable behavior models. Model transformation is another core capability of the SME that provides the ability to translate from one modeling language to another.
7. The next-generation modeling language must be capable of management in a heterogeneous and distributed modeling environment. The ability to manage change to the model, where multiple users are collaborating on a single model, is challenging enough. This basic capability requires extensive branch and merge capability that includes effective means for evaluating and integrating changes from multiple users, while maintaining a history of all changes. These challenges increase when multiple models and tools are all part of the collaboration. The ability to integrate with Product Lifecycle Management (PLM) environments, which enable versioning, configuration, and variant management, is a fundamental SME requirement.
8. Usability must be a primary consideration for the next-generation modeling language and tools. As noted previously, the learning curve for the SysML language and tools is quite steep. The next-generation modeling language and tools must enable efficient and intuitive use by a

broad range of users with diverse skills. This imposes requirements on model precision, model construction, model visualization, model management, and several other aspects of the language and tools.

9. The next-generation modeling language and tools must be highly adaptable and customizable to multiple application domains. This implies that the modeling language must be extensible to address domain-specific concepts, and that the modeling tools provide flexible means for the user to enter, analyze, and visualize model data in ways that are meaningful to each domain. In addition, the SME must accommodate customization performed in a standard and rigorous way.
10. To protect investments made by organizations, the next-generation modeling languages must support the migration of existing models with minimum information loss. Models must also be capable of being stored in neutral formats, retained for future access.
11. The next-generation modeling language and tools must be modular and extensible to enable evolution of the above capabilities to take advantage of on-going advances in technologies, concepts, methods, and theories.

SUMMARY

During the seven-plus years since the release of SysML 1.0 as a standard systems modeling language, the language and tools have matured considerably, as has the practice of MBSE. There is now much broader recognition that systems engineering must evolve to a model-based discipline. The transition to a model-based discipline is an INCOSE strategic initiative. It is also a key practice in the INCOSE Systems Engineering Vision 2025 (38-39), which describes the projected state for MBSE in 2025 as:

> *"Formal systems modeling is standard practice for specifying, analyzing, designing, and verifying systems, and is fully integrated with other engineering models. System models are adapted to the application domain, and include a broad spectrum of models for representing all aspects of systems. The use of internet-driven knowledge representation and immersive technologies enable highly efficient and shared human understanding of systems in a virtual environment that span the full lifecycle from concept through development, manufacturing, operations, and support."*

The need for a standard systems modeling language that can be readily adapted to domain-specific applications is a key enabler of this MBSE capability. Systems engineers gained much knowledge from the use of SysML and MBSE applications. These insights can inform the development of the next-generation systems modeling language.

The path forward has begun through the OMG SE-DSIG. This must be a collaborative effort drawing upon a diverse range of end users, tool vendors, academics, and others who can provide a language that is broadly accepted by the engineering community, industry, and academia. ■

AUGUST 2015
VOLUME 18 / ISSUE 2

Integration of MBSE Artifacts Using OSLC

A. Reichwein, axel.reichwein@koneksys.com and **R. Burkhart,** burkhartrogerm@johndeere.com

INTRODUCTION

The use of computer-interpretable models for systems engineering, in contrast to paper documents, is called MBSE. Benefits of model-based systems engineering over textual approaches include a higher level of support for automation and less ambiguous system representations. As systems become more complex, more information needs to be exchanged for collaboration purposes between various engineering disciplines. This is difficult to realize because engineering data is created with different software applications, resulting in data available in different formats and accessible through different protocols. A key challenge in MBSE is to achieve integration of data of diverse engineering disciplines in order to support data traceability, data consistency, data synchronization, and data interoperability.

Most data integration solutions are currently realized through tool integration solutions, in other words through the integration of specific software applications. As an example, Product Lifecycle Management (PLM) solutions provide an integrated set of mechanical engineering software applications, whereas Application Lifecycle Management (ALM) solutions provide an integrated set of software engineering applications. Unfortunately, these solutions cannot integrate all engineering tools but only a selected few, as the total number of engineering tools is huge. In the context of MBSE in which there is an unmanageable number of diverse engineering tools, a true tool-independent data integration solution is the only viable solution. This goal can be achieved through a new disruptive data integration solution based on the world wide web.

Open Services for Lifecycle Collaboration (OSLC) is an initiative to specify a uniform architecture and open standards for data integration based on the world wide web. The web has proven to be scalable, easily extensible, simple, and reliable. The success of the web is based in part on its hardware- and software-independence. It relies on standardized resource formats (e.g., HTML) and protocols for accessing resources (e.g., HTTP) whereby anyone can access any web page using any browser on any device. OSLC has adopted a similar approach for data integration by standardizing the representation of shared resources and information transfer protocols. OSLC takes advantage of the web infrastructure and the semantic web by reusing several concepts including Linked Data, RDF/XML, HTTP protocol and RESTful web services.

LINKED DATA

OSLC is based on the web to capture links between engineering data. The web allows users to define links between web pages and to navigate from one web page to another. The web is the perfect infrastructure to capture links between documents. But the web community is going a step further. It wants to use the web to not only capture relationships between documents, but also between data.

Currently, web pages in HTML consist of text written in plain English, which is an unstructured format, and which cannot be easily parsed and understood by computers. In turn, this doesn't allow intelligent and precise queries about data on the web. An initiative called Linked Data has the goal of publishing data on the web in a structured format, which can easily be parsed and understood by machines. A famous example of Linked-data is DBPedia, the equivalent of Wikipedia in structured data format. Queries running against structured data can return more precise results than queries running against unstructured data. As an example, a query running against Linked Data intended to retrieve all capitals in Europe, would return the precise list of capitals, while a similar query in Google would only return a list of web pages.

OPEN SERVICES FOR LIFECYCLE COLLABORATION

Linked Data allows the web to capture relationships between data in general, including links between engineering data. OSLC is based on Linked Data and other web technologies such as RESTful web services. OSLC is the result of the IBM Jazz project whose purpose was to improve the integration of software engineering tools. Specifications and source code of the Jazz project were made public in 2008 as OSLC under an open-source license. Even though it was founded by IBM, OSLC is now managed since 2013 by the OASIS standardization organization, as a vendor-neutral and open standard. The OSLC standards, which were adopted before 2013, are in the process of being converted into OASIS standards. The OSLC steering committee is composed of various organizations including companies from software and aerospace engineering. The open process for developing and publishing OSLC specifications also contributes to neutrality across tool implementations.

OSLC has been adopted by many tool vendors, mostly in the software engineering domain (OSLC Software 2015). However, nothing prevents OSLC to be used in more domains. Lately, Siemens Teamcenter has announced support for OSLC (Siemens PLM 2015). European research projects are also applying OSLC for the integration of mechanical engineering and embedded system design tools (Cesar 2013; iFEST 2013; SPRINT 2013; SafeCer 2013; Crystal 2015). Open-source implementations of OSLC adapters for Simulink and MagicDraw SysML are available (Eclipse Lyo 2014b; Eclipse Lyo 2014a).

PUBLISHING ENGINEERING DATA ON THE WEB

A central idea of OSLC is to make engineering data available on the web, which isn't necessarily the public web but possibly a private company intranet. According to OSLC, data is published on the web according to four Linked Data principles.

The first Linked Data principle states that data published on the web must have a Uniformed Resource Identifier (URI). So for example, the requirement defined in a specific engineering tool would be published on the web with a specific URI, in the same way as a web page is identified on the web through a URL. Data published on the web is called a web resource. Data published on the web according to OSLC is called an OSLC resource.

The second Linked Data principle states that a resource must have an HTTP URI, such that regular HTTP clients, such as a web browser, can use the HTTP protocol to dereference the URI and get a representation of the resource. The representation returned to the client could be in any format, such as HTML, PDF or JSON.

The third Linked Data principle states that a resource must be available in a machine-readable format. The W3C standard for publishing documents on the web is HTML and the equivalent W3C standard for publishing structured data on the web is Resource Description Framework (RDF). So data published on the web can be available in different formats, but it must at least be available in RDF in order to comply with the Linked Data initiative.

The fourth Linked Data principle states that a resource should be published on the web with links to related resources, in order to better understand the context of a resource. Information is typically only meaningful within a certain context. This also applies for engineering information. For example, a requirement may have relationships with many other engineering artifacts, for example with components which satisfy the requirement, or with use cases which refine the requirement, or with derived requirements. If a client looks up a requirement on the web, it is helpful if he can easily navigate to related resources in order to get a better understanding of the context of this requirement.

RETRIEVING ENGINEERING DATA ON THE WEB

Once engineering data has been published on the web according to OSLC, a client, which can be an engineer, or a software application, can look up OSLC resources on the web using the HTTP protocol. A client can choose to retrieve a resource in different formats through the HTTP Accept header field.

FORMAT OF ENGINEERING DATA ON THE WEB

RDF has features that facilitate merging of data even if their underlying schemas differ, and specifically supports the evolution of schemas over time without requiring all the consumers or providers of data to be changed. For example, within OSLC-based tool integration, RDF resources describing change requests could be linked with RDF resources describing tests.

LINKING ENGINEERING DATA ON THE WEB

OSLC adapters are web applications responsible for publishing engineering data on the web. Once data, possibly created by different tools, is published on the web, it is easy to connect it through links, just as it is easy to connect web pages through hyperlinks. A link between OSLC resources is simply an RDF statement indicating a linked resource through the subject URI, the link type through the predicate URI, and the other linked resource through the object URI. A link is always uni-directional, so a "bi-directional" link between two OSLC resources could be implemented through two uni-directional links.

A link can be between any two OSLC resources if the link predicate does not impose any constraints on linked resource types. For example the "description" predicate of the Dublin Core vocabulary (DCMI 2015) can be used between any two resources (no RDFs:range nor RDFs:domain properties), as any resource can provide a description of any other resource. In contrast, the "spatial" predicate of the same vocabulary describing spatial characteristics of a resource imposes the linked resource to be of type "Location," a resource type defined in the same vocabulary. Most OSLC RDF vocabularies do not impose constraints on predicates and their linked resource types in order to facilitate the connectivity of resources. Organizations can reuse unconstrained predicates of OSLC RDF vocabularies and if necessary extend them to add custom constraints.

DELEGATED DIALOGS

Links between OSLC resources can only be defined by knowing the URIs of the OSLC resources at each end of the link. The URIs of OSLC resources are very often not known in advance by engineers. OSLC has defined a protocol for facilitating the preview, search, selection, and creation of OSLC resources through a standardized HTML interface. This allows engineers to create, read, and select OSLC resources owned by another tool through an HTML delegated dialog without having to switch to the other tool.

LINK STORAGE LOCATION

A link between OSLC resources is simply an RDF statement, which can be saved in the RDF description of an OSLC resource, or in a separate RDF database, or in both. The advantage of storing a link in an RDF database, also called a triplestore, is that it will be indexed and query engines will find it easily. The advantage of storing a link in the RDF description of an OSLC resource is that the OSLC resource will contain additional contextual information in accordance with the fourth Linked Data principle. Ideally, a link should be saved both in the resource description and in an indexed RDF container, such as a triplestore. However, it isn't always possible to keep separate descriptions of the same link in sync. As an example, a link between OSLC resources may be defined with a third-party application which may not have the authorization to include the link in the RDF representation of the corresponding OSLC resource, provided by an OSLC adapter, due to access rights. The link storage location will be chosen depending on access rights of solutions intended to manage links between OSLC resources.

LINKING VERSUS DUPLICATING DATA

The possibility of easily defining links between engineering data through OSLC has an impact on integration workflows. Until recently, it was common to duplicate data across multiple tools in order to reuse data in tool-specific contexts. For example, a requirement may have been copied into a PLM tool in order to define additional relationships between this requirement and other artifacts managed by the PLM tool. Such workflow can be avoided with OSLC as a requirement described as an OSLC resource can be linked with another OSLC resource representing any artifact of any tool. By adopting OSLC, many systems engineering artifacts such as requirements and test cases no longer need to be duplicated. However, mechanical engineering artifacts such as the center of gravity or the mass parameter values will still need to be duplicated across mechanical engineering models in order for the models to remain consistent and meaningful.

OSLC IN STANDARDIZATION EFFORTS

In order to support interoperability between OSLC adapters, and in order to enable efficient querying capabilities for OSLC resources, it is important to be able to identify OSLC resources of the same type. For example, it is important that all OSLC resources describing requirements be of the same type "Requirement." Furthermore, it is important for OSLC resources

of the same type to have a common set of properties. Based on a common resource type definition, an OSLC adapter can export a resource that can subsequently be imported by another OSLC adapter.

The common resource definition is described in an RDF vocabulary that describes resource types and resource properties related to a specific domain. For example, the OSLC Requirements Management Vocabulary specifies a resource type named "Requirement" and resource properties such as "validatedBy" and "elaboratedBy". Based on OSLC resources conforming to a common vocabulary, a systems engineering tool can for example export a requirement, which can then be imported by a PLM tool, and an engineer performing a query to find requirements can find both requirements of a systems engineering tool as well as a PLM tool in his query results.

OSLC RESOURCE SHAPES

Even though RDF vocabularies specify resource types and resource properties, they do not impose well-formedness constraints on OSLC resources. For example, the OSLC Requirements Management Vocabulary does not prevent an OSLC resource having two "ID" properties even though it should only have one. The OSLC community has defined a new RDF Vocabulary to describe well-formedness constraints for OSLC resources. This new vocabulary is called the OSLC Core vocabulary. Constraints expressed in RDF according to this vocabulary are called OSLC Resource Shapes, and specify the multiplicity and allowed values of OSLC resource properties.

OSLC SPECIFICATION

The combination of an RDF vocabulary and OSLC resource shapes for a specific domain constitute an OSLC specification. The OSLC specifications cover typical Application Lifecycle Management (ALM) domains such as requirements management and change management. They have been defined through a community-based effort in working groups and the definition of new OSLC specifications is an ongoing activity.

OSLC resources published by OSLC adapters should conform as much as possible to OSLC specifications. However, OSLC specifications cannot cover all imaginable resource types, resource properties, and constraints. OSLC resources can therefore contain non-standardized properties, such as tool-specific properties, in addition to standardized properties defined in OSLC specifications. The list of OSLC adapters and their conformance to specific OSLC specifications can be found online (OSLC Software 2015).

OSLC STANDARDIZATION EFFORTS FOR MBSE

Systems engineers need to collaborate with engineers from many disciplines as systems engineers are responsible for defining the main technical aspects of a system and for managing the project progress to meet budget and time constraints. Subsequently, a large amount of information is exchanged between system engineers and discipline-specific engineers, and a lot of relationships exist between systemss engineering and discipline-specific engineering artifacts. As systems engineering artifacts play a central role in system design, it is crucial to have OSLC specifications covering systems engineering concepts.

EXISTING AND FUTURE OSLC SPECIFICATIONS

Existing OSLC specifications do not cover concepts related to system architecture such as blocks, ports, and connections between ports. Furthermore, the existing OSLC specifications have not been defined in alignment with existing mainstream standards such as the Systems Modeling Language (SysML) or the Requirements Interchange Format (ReqIF) of the Object Management Group (OMG). The process of defining a new OSLC specification and the process of getting a new OSLC specification adopted by tool vendors, can be fast tracked if it is based on an existing widely adopted standard. It is therefore likely that additional standardization efforts will take place in order to convert traditional non-OSLC based standards into OSLC specifications. As an example, the OMG OSLC4MBSE working group (OSLC4MBSE 2015a) has worked on converting the SysML standard into corresponding OSLC specifications. A SysML RDF vocabulary, which is not an official standard but a draft, is available online (OSLC4MBSE 2015b).

OPEN WORLD STANDARDS

An OSLC specification does not try to specify all the possible concepts to which relationships may exist. It is considered an "open world" standard meant to be used flexibly in combination with other arbitrary resources. As an example, the "validatedBy" relationship of the OSLC Requirements Management (RM) Vocabulary can exist between any two resources, for example between a requirement and a test case, or between a requirement and an analysis result. In contrast, SysML specifies the "satisfiedBy" relationship which can only exist between a requirement and another SysML NamedElement. SysML is therefore considered a "closed world" standard.

In the context of MBSE, in which it is impossible to predict in advance all the relationships between engineering data that can exist, "open world" standards such as OSLC specification provide more flexibility. Furthermore, the small size of OSLC specifications has multiple advantages: faster standard development, easier maintainability, and faster adoption by tool vendors. This does not mean that all standards need to be "open world" standards. There are many scenarios in which all types of data relationships need to be defined in advance in order to guarantee data consistency and data integrity.

CONVERTING TRADITIONAL STANDARDS INTO OSLC SPECIFICATIONS

A large standard such as SysML should be converted into multiple, small, and manageable OSLC specifications rather than into a single large OSLC specification. A traditional systems engineering standard such as SysML is large in size as it specifies all the concepts and relationships that can exist in the language. On the other hand, an OSLC specification is typically very small in size by only specifying resource types and resource properties of a very specific domain. However, the risk of defining redundant concepts over multiple OSLC specifications is higher than within a single specification. The unambiguous and consistent definition of MBSE concepts represents to this date a challenge as a multitude of different MBSE standards with often similar or completely redundant concepts have been defined. An overarching working group will need to supervise the various OSLC specifications related to MBSE in order to avoid the definition of redundant concepts. An overarching working group will need to supervise the various OSLC specifications related to MBSE in order to avoid the definition of redundant concepts.

MAXIMIZING THE VALUE OF MBSE

As soon as multiple OSLC adapters publish OSLC resources, it is practical for engineers to have a single point of entry to all OSLC resources, in a very similar way as Google is used as a single point of entry to web pages. The single point of entry to OSLC resources allows engineers to query OSLC resources, define and visualize relationships between OSLC resources, and define workflows to synchronize OSLC resources. Such solutions already exist and are very useful to get the most value of an OSLC-based ecosystem.

SUMMARY

The web is scalable and based on open technologies. OSLC is an approach to reuse the web for data integration. OSLC thus allows organizations to define relationships between engineering artifacts without being limited to a tool-specific data integration solution. We therefore >

Untangling the Digital Thread: The Challenge and Promise of Model-Based Engineering in Defense Acquisition

Timothy D. West, timothy.west.2@us.af.mil and **Art Pyster,** art.pyster@stevens.edu

Abstract. Although MBSE has made great strides in offering better ways to perform systems engineering, it has not yet incorporated the myriad physics-based models generated by the discipline engineers during the detailed design process. The US Air Force is proposing a related approach, the Digital Thread, as a framework to merge these detailed design models with MBSE's conceptual and top-level architectural models, resulting in a single authoritative representation of the system. Could this approach be the solution to streamlining system development, accelerating production, simplifying sustainment, and lowering the total system lifecycle cost? This paper introduces the Digital Thread concept, provides an aspirational scenario describing how Digital Thread may be employed in a future combat aircraft acquisition, and discusses a number of the challenges that currently separate this aspiration from reality.

INTRODUCTION

During a typical U.S. Department of Defense (DoD) weapon development program, literally hundreds of digital models simulate the behavior of various components of the system at various points along the development timeline. Systems engineers maintain and update very few of these models as the system design evolves, and even fewer undergo Verification, Validation, and Accreditation (VV&A). These models generally fall within one of two groups: the conceptual and top-level architectural models associated with MBSE, and the myriad physics-based and mathematical models generated by the discipline engineers during detailed design. As part of DoD's push for more robust use of digital system models, the U.S. Air Force is advocating a new Digital Thread concept that offers a potential approach to closing that gap between MBSE and the physics-based models. The goal of this initiative is to finally achieve the dramatic development cycle time reduction that advocates of model-based weapons development within the Air

Reichwein *continued from previous page*
expect that the next-generation of PLM and ALM solutions will most likely be based on Linked Data, as well as OSLC, to achieve unprecedented levels of data connectivity in engineering. ■

REFERENCES

- Cesar. 2013. "Cost-Efficient Methods and Processes for Safety Relevant Embedded Systems." http://www.cesarproject.eu/.
- Crystal. 2015. "Crystal." http://www.crystal-artemis.eu/.
- DCMI. 2015. "DCMI Metadata Terms." http://purl.org/dc/terms/.
- Eclipse Lyo. 2014a. "OSLC MagicDraw Adapter." https://wiki.eclipse.org/Lyo/MagicDraw.
- ——. 2014b. "OSLC Simulink Adapter." https://wiki.eclipse.org/Lyo/Simulink.
- iFEST. 2013. "Industrial Framework for Embedded Systems Tools." http://www.artemis-ifest.eu/.
- OSLC Software. 2015. "Software – Open Services for Lifecycle Collaboration." http://open-services.net/software/.
- OSLC4MBSE. 2015a. "OMG OSLC4MBSE Working Group." http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-oslc:oslc4mbse_working_group.
- ——. 2015b. "SysML RDF Vocabulary." http://www.omg.org/techprocess/experimental-rdf/SysML/1.3/.
- SafeCer. 2013. "Safety Certification of Software-Intensive Systems with Reusable Components." http://www.safecer.eu.
- Siemens PLM. 2015. "What's New in Teamcenter Service Pack 10.1.2." https://community.plm.automation.siemens.com/siemensplm/attachments/siemensplm/Teamcenter-Blog/14/1/Siemens-PLM-Whats-New-in-Teamcenter-Service-Pack-10-1-2-fs-Y16.pdf.
- SPRINT. 2013. "Software Platform for Integration of Engineering and Things." http://www.sprint-iot.eu/.

Force have been espousing since the 1990s (Kraft, et al. 2015). After providing a brief overview of MBSE, this paper describes Digital Thread – the "single authoritative digital surrogate" (Kraft, CREATE-AV and the Air Force Digital Thread 2014) of the system under development – along with a similar concept, Digital Twin – "the 'as-built' piece of the Digital Thread (Kraft, et al. 2015)." Subsequent sections describe the initiatives underway in this area, as well as the potential challenges this concept is likely to face. The paper concludes with a fictional scenario of how Digital Thread could help develop and sustain a future U.S. Air Force fighter aircraft.

MBSE 101

Over the last several years, MBSE has become a ubiquitous term within the realm of systems engineering, yet there is no well-accepted definition. The International Council on Systems Engineering (INCOSE) *Systems Engineering Handbook* (INCOSE 2011) uses the term several times, but fails to precisely define it. (The INCOSE Systems Engineering Handbook does discuss the Object-Oriented Systems Engineering Method (OOSEM), whose objectives include the "[i]ntegration of model-based systems engineering (MBSE) methods with object-oriented software, hardware, and other engineering methods.") The same is true for four commonly used textbooks on systems engineering (Blanchard & Fabrycky 2006) (Buede 2000) (Kossiakoff and Sweet 2003) (Larson, et al. 2009). The *Guide to the Systems Engineering Body of Knowledge (SEBoK)* (BKCASE Editorial Board 2014) describes MBSE as "[t]he formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases"—a definition it shares with INCOSE's *Systems Engineering Vision 2020* (INCOSE 2007). Friedenthal, Moore, & Steiner (2015) expand upon the INCOSE definition by emphasizing MBSE as a mechanism "to perform the systems engineering activities that have traditionally been performed using the document-based approach," ideally resulting in a "coherent model of the system" and "enhanced specification and design quality." Long and Scott (2011) describe MBSE as "a 'layered' process of analyzing and solving systems design problems," where the design problems are handled "in layers of increasing granularity in order to converge strategically on the solution;" they also emphasize MBSE's "use [of] a consistent language to describe the problem and the solution, produce a coherently designed solution, and comprehensively and verifiably answer all the system requirements posed by the problem." In other words, MBSE provides an integrated framework upon which a systems engineer can assemble a core model of the system; as the design matures, systems engineers can add layers of detail and complexity to create a more robust and representative model of the final design. A clear benefit of this approach is the transition "from controlling the documentation about the system to controlling the model of the system (Friedenthal, Moore, & Steiner 2015)."

A commonly used language for MBSE is the Systems Modeling Language (SysML), developed by INCOSE and the Object Management Group (OMG) to support "the specification, analysis, design, verification, and validation of a broad range of complex systems," including "hardware, software, information, processes, personnel, and facilities" (OMG 2012). Software developers such as Vitech Corporation and Sparx Systems have leveraged the power and simplicity of SysML to create conceptual modeling products such as CORE and Enterprise Architect, respectively. These products enable system designers to capture the requirements of the system in a digital database, track the status of those requirements through the verification and validation process, and even build architectural models of the system to simulate its conceptual behavior.

However, MBSE is still very different from "One Model to Rule Them All." The typical MBSE implementation tends to focus on what Buede (2000) described as "three categories of qualitative modeling approaches": (1) *data models* (Entity Relationship Diagrams and IDEF1 Diagrams), which address relationships between inputs and outputs of the system; (2) *process models* (Data Flow Diagrams, IDEF0 Diagrams, and N^2 Charts), which decompose the system functionally and depict the flow of inputs and outputs between these functional subsystems; and (3) *behavior models* (Function Flow Block Diagrams, Behavior Diagrams, and State Transition Diagrams), which define "the control, activation, and termination of system functions needed to meet the performance requirements of the system." Additionally, MBSE implementations frequently cross the gap from conceptual design into physical architecture, but fall short of incorporating the myriad mathematical models created by the discipline engineers during the detailed design process for a complex system. In other words, MBSE, although an immensely valuable part of the systems engineering process, tends to be conceptual and qualitative, whereas Digital Thread adds additional layers of quantitative and analytical rigor to model-based design.

Currently, mathematical models are more the domain of tools such as Mathworks' ubiquitous MATLAB/Simulink, the ANSYS family of simulation tools, and 3D CAD-based design packages such as Dassault Systems' SolidWorks and PTC Creo Parametric's Pro/ENGINEER. MATLAB/Simulink is traditionally used to model signal processing intensive systems, whereas the 3D CAD programs model the geometry associated with the physical systems. The richness of both of these CAD programs comes from the many add-ons that automate the discipline engineering analysis required for the product, such as Pro/ENGINEER's ability to model the non-Newtonian flow into the 3-D shape to help the engineer virtually identify and fix areas where a plastic injection mold may fail to fill properly. However, few of these tools provide the "real-time" simulation capability needed to fully leverage the Digital Thread concept.

Systems engineers attempted to add quantitative rigor to MBSE models. Buede (2000) describes the software development community's use of Object-Oriented modeling to complement its data, process, and behavior models with additional technical depth. Likewise, Vanderperren and Dehaene (2006) cite examples where Unified Modeling Language (UML, a precursor to SysML used in software development) and Simulink models were loosely coupled in what they referred to as "co-simulation," where separate models run in parallel and exchange information real-time during simulation execution. Unfortunately, this approach requires the creation and maintenance of two separate models using separate tools. The alternative more frequently employed is to painstakingly code the model from scratch using Java, C++, or a similar programming language.

WHAT IS A DIGITAL THREAD?

Notionally, Digital Thread represents a framework for merging the conceptual models of the system (the traditional focus of MBSE) with the discipline-specific engineering models of various system elements. However, the concept is new enough that a robust, widely accepted definition is still emerging. In the 2013 Global Horizons: United States Air Force Global Science and Technology Vision, the Chief Scientist of the Air Force described Digital Thread as a digital linkage between materials, design, processing, and manufacturing that "will be the game-changer that provides the agility and tailorability needed for rapid [weapon system] development and deployment, while also reducing risk (U.S. Air Force 2013)." Later that year at a National Insti-

tute of Standards and Technology (NIST) symposium, Kraft (2013) referred to it as a "cross-domain, common digital surrogate of a materiel system" that "provides the analytical framework for organizing output from high-fidelity, physics-based models across the entire lifecycle." During a National Defense Industrial Association (NDIA) conference in 2014, Christian (2014) described Digital Thread as "an extensible, configurable and Agency enterprise-level analytical framework that seamlessly expedites the controlled interplay of authoritative data, information, knowledge, and computer software in the enterprise data-information-knowledge systems, based on the Digital System Model template, to inform decision makers throughout a system's life cycle by providing the capability to access, integrate and transform disparate data into actionable information." In a prior presentation at the same conference, Zimmerman (2014) offered the same definition, less the reference to "computer software." At the American Institute of Aeronautics and Astronautics (AIAA) SciTech 2015 conference, Walker (Kraft, et al. 2015) offered a slightly leaner definition: "an extensible, configurable, and enterprise level framework that seamlessly expedites the controlled interplay of authoritative data, information, and knowledge to inform decisions during a system's life cycle by providing the capability to access, integrate, and transform disparate data into actionable information." In addition to deleting "Agency," "analytical," and "computer software," this version omits "the enterprise data-information-knowledge systems, based on the Digital System Model template," where the Digital System Model (DSM) is defined by Christian (2014) and Zimmerman (2014) as "a digital representation of a weapon system, generated by all stakeholders, that integrates the authoritative data, information, algorithms, and systems engineering processes which define all aspects of the system for the specific activities throughout the system lifecycle." To clarify the distinction between Digital Thread and DSM, Kraft (2013) and Christian (2014) describe Digital Thread as "the physics-based modeling instantiation" of the DSM concept. Zimmerman (2014) describes it more succinctly as a "framework for DSM use."

Despite their subtle differences, these definitions share a common theme of "authoritative data." These two words deserve special attention because they communicate that the vision for Digital Thread is not *just* a model, or a collection of models. Rather, its ultimate goal is to be *the* definitive repository of *authoritative* information containing *everything* we know about the system at that instant of time. To be the authoritative source of information, it has to be both current and complete. To be useful to decision makers, someone must organize this information in a manner that facilitates data mining in a thorough but timely manner. This indeed is a lofty goal. Regarding the subtle differences between DSM and Digital Thread, this paper will sidestep that issue by taking the viewpoint that they are complementary concepts with similar goals and similar challenges; the fact that multiple organizations within DoD are pursuing similar concepts may increase the likelihood that the concept will become an accepted practice for complex system development across the Department.

Defense contractor Lockheed Martin is credited with coining the term Digital Thread to describe its approach to using the 3D solid models generated during the engineering design process to support the manufacturing process for both the computer numerically controlled (CNC) machining of metal parts and the composite programming system (CPS) approach to fiber placement in composite parts: "the unbroken data link [between the engineering CAD model and the models loaded in the CNC or CPS machine] is the digital thread" (Newton 2014). Lockheed Martin has reported a number of successes in employing this approach to improve part fit and reduce rework on mechanical parts for the F-35 fighter (Kinard 2010). Lockheed is also using these same models to generate the graphics for training documents and maintenance manuals, extending the Digital Thread even further through the system lifecycle preventing configuration mismatches between these documents and the actual aircraft.

This direct CAD-to-CNC prototyping merely scratches the proverbial surface in terms of potential uses for Digital Thread. DoD, and particularly the Air Force, shows considerable interest in the concept, and is generating new ideas on how to integrate the concept into the entire lifecycle of a weapon system. The Air Force Test Center (AFTC) is championing an expansion of Digital Thread into the Test and Evaluation (T&E) arena. Since 1995, Kraft has been proposing the use of physics-based computational models to augment and streamline the traditional regimen of ground and flight testing required to prepare a new military aircraft for production. For instance, he has proposed the coupling of airplane aerodynamics, propulsion, heat transfer and structural response models to give a more complete picture of how an entire aircraft system might respond to a given flight condition, and how those responses would accumulate over the life of the aircraft. Early in the design process, such models could help developers analyze alternatives and make initial design trades, and to identify potential design flaws while "sunk" development cost is low. Later in the development process, the model could be used to guide testing; that is, test where the model indicates potential problems, where the model uncertainty is large, and/or where the consequences of the model being wrong are too high (Kraft 2014).

Kraft's concept for Digital Thread entails the merger of the probabilistic output from various high-fidelity mathematical models into a reduced-order response surface, which systems engineers frequently update throughout the development process, resulting in a "single authoritative digital surrogate (Kraft 2014)" of the system under development. Because it is a reduced-order model, the computational time required to execute the model is "fractions of a second instead of [the] hours or days" associated with executing a large computational fluid dynamics (CFD) based solution, yet the reduced-order nature of the model "does not necessarily translate into reduced accuracy" (Kraft 2014). The rapid execution of the model enables designers and acquirers alike to quickly run various "what if" scenarios with the model, enabling more insight into system performance and more efficient design optimization. Because the model uses a standard framework, users can incorporate it easily into larger wargaming engagement models to see how various iterations of the system design affect the larger air battle. Replacing the simplistic models used in wargames today will enable more accurate trade studies on the effects of various aircraft performance characteristics (maximum acceleration, turn rate, climb rate, radio-frequency signature, and more) at the "system of systems" level, and a more accurate assessment of interoperability between legacy systems. Because the model updates throughout the system lifecycle, it remains the definitive, authoritative representation of the system that decision makers can use to better inform billion-dollar acquisition decisions.

The Air Force Research Laboratory (AFRL) has offered an additional spin on the concept, proposing the creation of a "Digital Twin" of each individual airframe and engine. Walker (Kraft et al. 2015) described the Digital Twin as "an integrated multi-physics, multi-scale probabilistic simulation of an as-built system that uses the best available physical models, sensor information, and input data from the Digital Thread and a Digital System Model to mirror the life of its corresponding physical twin," and elsewhere as "the 'as-built' piece of the

Table 1. Comparison of systems modeling approaches

"TRADITIONAL" MBSE	DIGITAL THREAD	DIGITAL TWIN
Focuses on top half of Systems Engineering "Vee" Diagram	Addresses complete Systems Engineering "Vee" Diagram	Additional emphasis on system O&M
Maturing approach with several demonstrated successes	Fledgling approach with a few successes	Notional concept, no demonstrated success
Acceptable operation on consumer-grade computers	Slow to moderate operation on today's fastest supercomputers, depending on the application	Computational power for full implementation currently unavailable
Wide variety of commercial software available today	Prototype software tools in development	Tool development awaiting required computer power
	Promises accelerated system development, better manufacturing quality & consistency, and lower lifecycle cost	Promises further lifecycle cost reductions through enhanced failure prediction and true condition-based maintenance

Digital Thread." According to this concept, each time the Air Force takes delivery of a new aircraft, it also receives a "[trillion] degree-of-freedom" digital model of the as-built aircraft that is "is ultrarealistic in geometric detail, including manufacturing anomalies, and in material detail, including the statistical microstructure level, specific to this aircraft tail number; " the model would accept "probabilistic input of loads, environmental, and usage factors, and it also tightly couples to an outer-mold-line, as-built, CFD model" of that aircraft tail number (Tuegel et al. 2011). After each sortie, data logs from that specific sortie could be downloaded from the physical aircraft and "flown" on the Digital Twin to ensure the condition of Digital Twin reflects the same amount of stress and strain as seen on the real aircraft. With sufficient computing power, AFRL believes they can predict where, when, and how a given component will fail, enabling more effective, prediction-based maintenance of the aircraft. Table 1 compares "traditional" MBSE with Digital Thread and Digital Twin.

CHALLENGES TO THE DIGITAL THREAD CONCEPT

The Digital Thread concept described above potentially offers significantly more utility than simply leveraging existing 3D CAD drawings to manufacture a single part or a simple subassembly. A description of potential uses is in Figure 1. When properly employed, Digital Thread has the potential to dramatically shorten the product development timeline, enhance product quality and manufacturing consistency, improve sustainability, and reduce the total lifecycle cost of the system. However, to achieve these benefits, it must first overcome several technical, programmatic, cultural, and bureaucratic challenges, as listed in Figure 2. The following section describes those challenges.

- Exploration of System Trade Space
- Model-Based Analysis of Alternatives
- System Performance Predictions
- Sensitivity Analysis of Performance Results
- Direct CAD-to-CNC & 3D Printer Prototyping
- System Documentation Development
- Accelerated Test & Evaluation
- Digital Interoperability Assessments
- System/Component Life Estimation

Figure 1: Potential applications of Digital Thread

- Design Tool Compatibility
- Protection of Intellectual Property
- Adequate Computer Processing Power
- Enabling Software
- Verification, Validation & Accreditation
- Model Maintenance
- Bureaucracy and Cultural Inertia

Figure 2: Challenges to the Digital Thread concept

Design Tool Compatibility

3D CAD-based design packages such as Dassault Systems' SolidWorks and PTC Creo Parametric's Pro/ENGINEER output CNC-friendly data files, essentially automating the current capabilities of Digital Thread. As robust as they are, however, these tools lack the sophistication to accurately model the myriad physical aspects associated with an aircraft design, in addition to the requirements management features of Core or Enterprise Architect. As an example, consider the aerothermoelastic behavior of an F-16 aircraft wing. In flight, the aircraft wing bends because of aerodynamic loading created by increased pressure on the leading edge and the underside of the wing. Because air temperature generally decreases with altitude, this change in thermal environment also affects the shape of the wing. As the shape of the wing changes, so do its structural behavior and its aerodynamic performance. In certain flight conditions, the wing and/or the weapons, sensor pods, and associated suspension equipment may encounter flutter or other dynamic oscillations. This interaction between aerodynamics, thermodynamics, and wing structural elasticity is a very complex phenomena, not readily simulated by toolsets today. Given their niche applications, the financial incentive for Dassault or PTC to expand their software to incorporate this behavior is low. ANSYS and aircraft primes offer rudimentary tools for modeling this behavior, but not at the level of fidelity needed for Digital Thread. To compensate for this lack of fidelity, conservative developers tend to over-design the structure, increasing aircraft weight, which forces a decrease in payload capacity and/or fuel volume.

Developers should incorporate these phenomena into the Digital Thread to make

it more viable. Thus, a common language, framework, and/or interface to enable these models to run in a simultaneous or iterative fashion are requirements such that the output of one module becomes the input for the next iteration of a different module. This drives tough questions about what the appropriate framework should be for this process, how loosely or tightly to couple the individual modules within the framework, and what data exchange must occur in order to make such a framework function.

Further complicating the development of such a framework is the fact that many of the mathematical models are incompatible with each other. DoD is attempting to address this issue through its Computational Research and Engineering Acquisition Tools and Environments (CREATE) program, established in 2008 to enhance the development and acquisition of new weapon systems by closing capability gaps in weapon conceptual design, design verification, and the design environment. The CREATE program has already delivered several "multi-disciplinary, physics-based simulation software products developed to enable full-vehicle design analysis and testing via high-fidelity simulation" – including one for rotorcraft (*Helios*) and one for fixed-wing aircraft (*Kestrel*) – which share a "common scalable infrastructure" that enables software developers to share modules across both aircraft-type simulators (US Air Force 2015). Additionally, both *Helios* and *Kestrel* are compatible with *Firebolt*, the CREATE tool for modeling propulsion systems. CREATE has developed a similar toolset for the Navy (*CREATE-Ship*) to accelerate ship design (Mackenna 2012). According to Dr. Robert Meakin, the program manager for the Air Vehicles portion of the CREATE program, tools such as *Helios* and *Kestrel* will enable DoD to produce "physics-based, optimized conceptual designs," quantify the performance of those designs, and assess performance sensitivity to new technologies or proposed design enhancements in "days to weeks" vs. the multi-month process associated with many of DoD's current Analysis of Alternatives (AoA) studies, resulting in better-informed acquisition decisions earlier in the acquisition process, before large portions of the system lifecycle costs are locked in (Meakin 2011). Since CREATE is a DoD-funded development, the software is not proprietary; it is also designed to "be used by an average CFD practitioner with a basic level of training," unlike many proprietary tools, which require extremely knowledgeable personnel with extensive training to properly interpret the results (Meakin 2011).

To realize the full potential of such a tool, the DoD must either convince its defense contractors to provide CREATE-compatible models, or build its own model. For example, a robust model of the U.S. Air Force's F-22 fighter would incorporate the aircraft's nose and forward fuselage (built by Lockheed Martin), wings and aft fuselage (built by Boeing), and the F-119 turbine engines (built by Pratt and Whitney). Using tools such as CREATE, Air Force engineers could reverse-engineer these models from information provided by the contractors and/or attained through performance testing. For instance, they could easily generate a basic rigid-body model of the F-22 using only the aircraft's Outer Mold Line (OML) geometry. Expanding that model to include the aircraft's aero*elastic* behavior, however, requires substantially more information, such as details of the structural design underneath the aircraft's skin. Expanding into the aero*thermo*elastic regime requires an understanding of how various elements of that substructure dissipate heat, to where those elements dissipate the heat, how those structures change dimensionally as they are heated and cooled, and how the system's aerodynamic performance changes as a result of this heating and cooling. Yet, the model would still be incomplete without addressing how the aerothermoelastic behavior changes in response to pilot inputs, flap scheduling, and other system stimuli that causes the control surfaces to deflect. Building a model with this level of fidelity would require enough proprietary information from the prime contractors that the resulting model, although built by Government personnel, would itself likely be proprietary. Clearly, the process would be more efficient if the contractors developed and provided the model. This would also avoid the cost and configuration control challenges associated with building and maintaining two sets of models. Additionally, it would resolve the dilemma of determining which model was authoritative. This would not necessarily mean the contractor has to use the CREATE toolset; proprietary tools capable of generating a CREATE-compatible output file would suffice.

One potential solution would be for the DoD to develop its own models to determine the system requirements, then mandate model submission during the subsequent source selection process, followed by regular model updates throughout the duration of the development effort. This would enable the source selection team to perform their own system performance comparisons during the selection process, and to conduct sensitivity analyses on performance results; as an example, the team may assess how the maximum range of the aircraft varies with changes in the airflow boundary-layer and other drag-inducing effects on the wings. Having a common model framework would also facilitate the addition of these models into the larger, operational-level wargaming simulations, where the battlespace-wide effects of an incremental increase in a given performance parameter could be better assessed. This larger environment could also provide a playground for assessing interoperability between the proposed systems and those already in operation. Indeed, the DoD Engineered Resilient Systems (ERS) Community of Interest has already demonstrated the feasibility of this "try before you buy" concept in its 2013 analysis of three candidate design concepts for the next generation cargo transport aircraft (the C-X program) using tools similar to those available in CREATE; the team utilized a low-fidelity tool capable of quickly estimating performance trends associated with changes in a variety of parameters including empty weight, maximum takeoff weight, maximum thrust, maximum payload range, average fuel burn rate, takeoff distance, landing distance, lifecycle costs, and cost per flight hour, and injected these results into an operational engagement model to better understand the capabilities and design tradeoffs inherent in each of the three designs (Kraft 2014).

Makenna (2012) proposed a similar requirements-definition approach for the Navy that utilizes "set-based" design principles, where the user defines and gradually narrows down broad sets of design parameters as tradeoff information becomes available until the user identifies a best solution. Makenna posits that set-based design approach produces a design in the same amount of time as the traditional point-design methodology, while avoiding the pitfall of later including premature design decisions, designs near or beyond the edge of feasibility, and costly repairs to correct design deficiencies discovered late in the developmental process.

Protection of Intellectual Property

Another challenge associated with building a fully robust, layered model of the system will be addressing the proprietary nature of some of those layers. Pushing Digital Thread to its full potential will require striking an optimum balance between complete insight into the design approach, manufacturing processes, physical properties, and more of the system with protection of the "secret sauce" that gives one company a competitive edge over another. In the aerospace industry, for instance, each company has intellectual property, such as unique design tools, development processes, manufacturing techniques,

and more, that it carefully guards as proprietary information or trade secrets.

The Economic Espionage Act of 1996 (EEA), which criminalizes the theft or unauthorized disclosure of trade secrets, defines the term as follows:

"[T]he term "trade secret" means all forms and types of financial, business, scientific, technical, economic, or engineering information, including patterns, plans, compilations, program devices, formulas, designs, prototypes, methods, techniques, processes, procedures, programs, or codes, whether tangible or intangible, and whether or how stored, compiled, or memorialized physically, electronically, graphically, photographically, or in writing if (A) the owner thereof has taken reasonable measures to keep such information secret; and (B) the information derives independent economic value, actual or potential, from not being generally known to, and not being readily ascertainable through proper means by, the public (United States 1996).

In spite of the EEA and the potential penalties it establishes, economic espionage does occasionally occur. Even the most reputable of companies are not immune from the misdeeds of a few bad actors within its ranks. In 2003, for instance, the US Air Force cancelled over $1 billion worth of space-related contracts with a major defense contractor for the latter's inappropriate acquisition of "thousands of pages of trade secrets" from an industry arch rival (Pae 2003). That same year, a start-up company in the space business alleged that a defense contractor "'blatantly' stole [its] trade secrets while the two companies were discussing a potential venture" (Pae 2003).

Meanwhile, the Federal Bureau of Investigation reports that foreign espionage efforts continue to grow, with U.S. companies having suffered financial losses in excess of $13 billion during fiscal year 2012 (Dilanian 2012). This reportedly includes "top secret" information on 25 of America's most advanced weapon systems, including the F-35 Joint Strike Fighter, the V-22 Osprey, the Aegis Ballistic Missile Defense System, and the Black Hawk helicopter, among others (Millar and Knight 2013). In the case of the F-35, this hacking likely slowed down the F-35 fielding, since the developer probably had to rewrite software code and redesign communications system to overcome vulnerabilities exposed through the security breach (Reed 2012).

Clearly, potential industrial espionage and international hacking into defense contractors' computer systems will complicate any effort to build a consolidated end-to-end Digital Thread model for any of these weapon systems, since all are composed of components and subsystems from myriad companies, and most of these components incorporate the trade secrets from the component developer. This means the component vendor must not only come to trust the system integrator and/or model builder, but also the latter's mechanisms for protecting intellectual property. The F-22 example cited above would require integration of Lockheed Martin's proprietary information regarding the nose and forward fuselage, with Boeing's proprietary information on the wings and the aft fuselage, and Pratt and Whitney's proprietary F-119 engine information, not to mention the proprietary data from countless sub-tier vendors who supplied the three main contractors. In this example, each company would have to trust the others with their intellectual property, in spite of ongoing competition and the risk of theft of trade secrets. Furthermore, each would have to trust the others' ability to protect their digital data from external theft, in spite of growing cyber espionage from potential adversaries.

Under such a concept, vulnerability in one participant's network puts the intellectual property of all the other players at risk too. When weapon systems are involved, this is not just a financial or intellectual property risk; it is a national security risk; directly to the pilots flying those airplanes and then to the people and property that they protect. Like most risks, these have mitigation strategies, but mitigation often translates into additional scope of work and, potentially, cost and schedule growth. Just like the commercial-sector counterpart, a DoD program manager must make smart decisions about how to manage risk without crippling the efficiency of the development program. In the hypothetical case of our F-22 model, it could be stored on a computer located in a vault at Lockheed, with no external connectivity. This might work for Lockheed, but would be challenging for Boeing and Pratt & Whitney – and for the Air Force engineers who need access to the model. Even at Lockheed, it would likely drive extra overhead costs associated with more stringent security measures, additional personnel screening, vault maintenance, etc. A less expensive but perhaps riskier approach would be to establish encrypted connections between the developers and the Air Force. Further cost reduction occurs by only encrypting the sensitive information and sharing the non-sensitive information on an unclassified server, but is the risk of doing this acceptable? In true academic fashion, the answer is "it depends." It depends upon how sensitive the information really is, how much damage will be done if the information is compromised, which stakeholders would be impacted by the damage and to what degree, and so on. Each program is unique, and each program manager must strike a balance between program efficiency and information protection that is acceptable to everyone whose information is at risk.

Kraft's approach to Digital Thread sidesteps this issue altogether by enabling contractors to provide reduced-order inputs to the single authoritative digital surrogate; this enables contractors to use their own proprietary design tools to generate their respective inputs, while also protecting their proprietary processes and technologies (Kraft 2014). The challenge with this approach is that it forces the Government to assume each contractor's model is a detailed, accurate, and technically-rigorous representation of the system elements—at least until test data are available to validate this assumption. A potential solution to this problem would be to have the contractor provide enough information about the system to enable the Government to generate its own moderate-fidelity model with which it could perform an order-of-magnitude verification of the contractor's model. However, Kraft's approach also poses a more fundamental quandary: how can a reduced-order model also be authoritative? Should not the model with the higher fidelity be authoritative?

Adequate Computer Processing Power

A key enabler in the progress to date with Digital Thread has been the steady increase in available computational power. Computational power has historically grown at an exponential rate, doubling every 18 months, in accordance with Moore's law (Kaku 2011), while the cost per transistor has decreased by an exponential rate, declining by half in approximately the same 18 month timeframe (Kurzweil 2005). That exponential growth is not yet sufficient to allow real-time modeling of complex aerodynamic flow patterns. Sophisticated CFD models often take days to weeks to run, even on supercomputers today. The computational demands for Digital Twin are even greater: for this concept to work, we must be able to simulate an hour's worth of flight time on the digital model within an hour of real time. To put this processing power into perspective, consider China's Tianhe-2, currently the world's fastest computer, which operates at 33.86 petaflops (Knapp 2014). AFRL's scientists predict that a full Digital Twin model will require "exaflop" computer horsepower (100 times faster than Tianhe-2); as Figure 3 depicts, such computational capability is not projected to exist until 2022, and without it, "the power of the Digital Twin for life prediction and

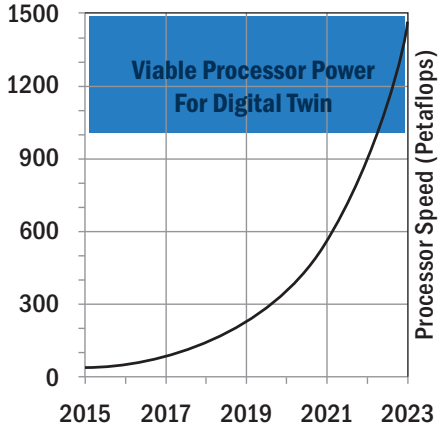


Figure 3: Predicted growth of computational power

decision making is lost" (Tuegel et al. 2011). This 2022 timeframe assumes that Moore's Law of exponential computational power increase remains true, which many doubt – including Gordon Moore himself. When asked in 2007 about the future of his namesake law, he predicted that it would end in "ten to fifteen years" (Kaku 2011).

Although the full utility of Digital Twin's life prediction capability may remain a decade in the future, engineers are already exploiting the potential of Digital Thread through such efforts as described above for the C-X program, and through a related study that uses historical wind tunnel test data from the F-22 against which to compare simulation results from the CREATE. The intent of this study is to determine the flight regimes where the simulation data are accurate, employ a statistically-based Design of Experiments approach to select the minimum number of test points required in these regimes, and assess the loss in model fidelity associated with using the reduced data set; the study team believes it can utilize this approach to reduce the four-year wind tunnel test campaign traditionally associated with a fighter aircraft development program by as much as 25 percent (Kraft 2014).

The Air Force Seek Eagle Office, which certifies weapons, tanks, and external pods for carriage and release on Air Force fighter aircraft, is likewise using computationally-intensive, analytical tools to simulate the aerodynamics associated with weapons separation and the aircraft flying qualities before and after weapon separation. Although Seek Eagle's approach utilizes a variety of tools instead of CREATE, the objective is similar: to verify conditions where adverse flying qualities or unacceptable flight control behavior are likely to occur. This information would enable Seek Eagle to minimize the flight test matrix in areas where anomalies are unlikely, and to increase test safety where one expects anomalies by creating a pre-flight prediction of how severe one anticipates the anomalies to be. Seek Eagle is even going one step beyond basic aerothermoelastic modeling by incorporating human-in-the-loop flight control inputs and the associated digital processing that occurs between the pilot's hand and the F-16's control surfaces to enhance the realism of the simulation output. Simulation and flight test results compared favorably for the F-16, although "millions of CPU hours" were required to reach this level of fidelity. To accomplish this simulation, Seek Eagle used an Army supercomputer with 1,600 processor cores; Seek Eagle estimates that 4,000 to 8,000 processor cores will be required within two years to handle the growth in model complexity and the demand for faster computational turn-times: "[t]o be a viable M&S [Modeling and Simulation] option for the Test and Evaluation (T&E) community, jobs must be submitted and results analyzed in days or weeks rather than months" (Clifton, Ratcliff, and Bodkin 2014).

The office tasked with delivering the massive computational horsepower required by Seek Eagle and others in the near term and with meeting AFRL's Digital Twin processing requirements in the longer term is the DoD High Performance Computing Modernization Program (HPCMP). Providing over 2.5 billion processor hours per year, HPCMP consists of five supercomputing centers; its 2014 acquisition of two supercomputers worth $65 million more than doubled its available computing power from 8.1 to 16.5 petaflops (Hintson 2014). To realize the full potential of Digital Thread and Digital Twin, HPCMP must continue this level of investment for another decade.

Software that Leverages that Computer Horsepower

Newer and faster computer hardware alone is insufficient to close this gap in computational power. Doing so will also require new software applications that more optimally exploit these faster speeds. As Kaku indicates, "brute computational power does not equal intelligence," or in this case, intelligent modeling (Kaku 2011). Maccaba, who claims that the overall corporate return on investment in information technology has declined in spite of exponential growth in hardware capability, points to size and complexity (vs. capability) as the software features that follow Moore's Law as a primary reason for this decline, noting that the growth in "the size and complexity of operating systems has largely offset much of the increase in raw computing power (Maccaba 2014)." He also blames the proliferation of bloated operating systems for accelerating obsolesce of computer hardware and exacerbating maintenance requirements for legacy software. Clearly, to fully realize the potential of the Digital Twin requires a healthy dose of software engineering, in addition to hardware enhancements.

Software development investments must continue in order to reach the full potential of Digital Thread. A growth in processor core count without companion modifications in the software to utilize those cores results in untapped processor potential. Because these tools need to be open-source and/or nonproprietary in order to reach beyond corporate boundaries, the U.S. Government will likely remain the primary source of that funding. The tools generated by the DoD-funded CREATE program fit this profile. The fact that the CREATE program is run by the same office tasked with delivering the massive computational horsepower required by Seek Eagle and others is a positive sign that the DoD understands the importance of investing in companion software for its supercomputing hardware. Whether the DoD continues to fund its HPCMP despite looming budget challenges will be a telltale sign of just how committed it is to achieving the Digital Thread/ Digital Twin concept.

Model Verification, Validation, and Accreditation

Verification, Validation, and Accreditation (VV&A) are key elements in the process of assuring the model is accurate. DoD Instruction 500.61, which mandates VV&A for DoD models, simulations, and associated data, offers the following definitions for these terms (US DoD 2009):

Verification*. The process of determining that a model or simulation implementation and its associated data accurately represent the developer's conceptual description and specifications.*

Validation*. The process of determining the degree to which a model or simulation and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model.*

Accreditation*. The official certification that a model or simulation and its associated data are acceptable for use for a specific purpose.*

To paraphrase Peter Drucker (2010), verification is the process of ensuring we "built the model right" by determining whether it meets the associated design-to or build-to specifications; validation is the process of ensuring we "built the right model" by determining whether it satisfies the needs of the stakeholders; and accreditation is

the process of officially certifying that the model is acceptable for its intended use. In the case of a Digital Thread/Digital Twin, none of these should be a one-time, Big Bang event, and all rely on mature software development processes that includes clear requirements definition, rigorous design documentation, a formal configuration control process, and a thorough test and evaluation program. A novel approach to the latter is the Test-Driven Development (TDD) process recommended by the Extreme Programming (XP) software development methodology; in this approach, software developers write the test code for each block of software *before* writing the software itself, forcing the developer to articulate module requirements, associated inputs and outputs, and potential ways that a user might break the code before the first line of software is written (Larman 2005).

Formal VV&A of a model can take longer to accomplish than building the model itself, and can account for up to 30 percent of the total cost of developing the model (Farr 2007). This is especially true when decisions made based on the output of that model potentially affect life and limb. For instance, when an inspection reveals micro-cracking in a wing spar, but the Digital Twin model says we can fly the jet for another 100 hours, do we trust the model? Do we waive the inspections altogether until the model says they are needed? For this concept to work, we must develop an affordable mechanism for assessing model accuracy, understanding inherent assumptions and limitations, and determining where we can and cannot consider its output as definitive.

Model Maintenance

Assuming we do succeed in attaining an accurate model and in getting the model accredited, the accuracy of the Digital Thread will quickly degrade if not kept up to date with each design change. In the case of the Digital Twin concept, it must remain current with the replacement of each part, even when the replacement has the same form, fit, and function as the original. Perhaps a different CNC machine made the new part or it was subject to a different head treatment. For myriad reasons, it may possess slightly different electrical, thermal, or mechanical properties, taken into consideration by such predictive models. Additionally, if one desires to include the effects of wear and usage on the Digital Twin, a separate digital instantiation would be required for each part or subcomponent in the inventory, even if those parts are initially identical in every way. For instance, assume the Digital Twin of the F-119 engine is a composite model built from Digital Twin models of ten major components, and that the fleet requires 500 engines. Neglecting the spares associated with those ten components, this would require the developer to maintain 5,500 Digital Twins – one for each of the major components for each of the engines, in addition to the 500 full-up engine models. Following each flight, the models associated with the 20 components installed in that particular pair of engines would have to be aged to reflect the stresses and strains generated by that flight, and the composite Digital Twin for each of the two engines updated to reflect the current condition of its constituent components. Never mind that the actual F-119 has significantly more than ten components and each of those has subcomponents requiring modeling to accurately predict the likelihood of one of those parts failing on the next sortie. The effort required to maintain accurate Digital Twins of the aircraft engine inventory, let alone the entire aircraft fleet, is a significant effort and would only be feasible if the approach to maintaining the model and its accreditation is simple, fast, and affordable – adjectives that are not traditionally associated with model accreditation. To receive buy-in across the acquisition community, the requirement is real data to show that the model maintenance is actually cheaper than the cost associated with scrap, rework, unnecessary maintenance actions, and more.

Bureaucracy and Cultural Inertia

Just as MBSE adoption did not occur overnight, neither will Digital Thread. Although the CAD-to-CNC variant of Digital Thread is receiving mainstream acceptance, the more revolutionary aspects of this concept will likely face significant cultural inertia. The test community will have to be convinced that the digital model is accurate, and that systems engineers can properly calibrate it with a much smaller test matrix. The model accreditors will have to be convinced to minimize, and in some cases eliminate, actions needed to reaccredit the model each time it changes; Digital Thread is, after all, intended to be a living model, implying myriad changes over the lifecycle of the system. Maintainers will have to be convinced that Digital Thread and Digital Twin offer a safe and more effective approach to system maintenance than the traditional time-based and usage-based preventative maintenance strategies. The acquisition community will have to be convinced that the return on investment in Digital Thread justifies the added up-front cost to the system. Naysayers will exist in each of these tribes, especially at the early stages of implementation. Digital Thread may even encounter inertia from the systems engineering community, where MBSE has already proven to be fit for its intended purpose. Additionally, the DoD's acquisition system, contractual language, and other bureaucratic processes will require modification to better enable Digital Thread. Strong leadership and demonstrated results will be the keys to overcoming this inertia, results such as those described in the following scenario.

PROSPECTIVE DIGITAL THREAD SCENARIO: F-X DEVELOPMENT

Digital Thread and Digital Twin are both intriguing concepts that promise significant improvement over the traditional MBSE approach in terms of reducing technical and programmatic risk, accelerating the development process, and minimizing the overall lifecycle cost. To illustrate how to realize these potential benefits, consider the following hypothetical scenario. In October 2017, the U.S. Air Force received authorization from Congress to conduct an analysis of alternatives to finalize requirements for the next-generation fighter aircraft, designated F-X. A team of DoD experts developed five notional concepts, using the performance characteristics of the F-22 as their initial baseline. Utilizing a set-based design approach and the *DaVinci* tool created under the CREATE program to perform parametric analyses of aircraft concepts, they evaluate each of these concepts at various altitudes, airspeeds, g-loads, and angles of attack, within a system of systems realistic threat environment. A team of seasoned combat pilots reviewed the results of the simulation to provide face validity, assess the trade space associated with each design, and selected a high-performance, twin-engine, tailless aircraft with extended range capability. The *DaVinci* results for this configuration defined the system-level specifications for the aircraft.

Following approval of the specifications and formal authorization to initiate procurement, the Air Force released two Requests for Proposals (RFPs). These RFPs were unique in that they were the first to require offerors to submit *Kestrel*-compatible models of aircraft design concepts and *Firebolt*-compatible models of engine designs. The Air Force received three engine and three aircraft proposals; one of the engine proposals was "nonresponsive" – the eliminated offeror failed to include a *Firebolt* model with its proposal. The reviewers performed a six-model virtual "fly-off," comparing each aircraft/engine

combination. The fly-off revealed that one of the three candidate aircraft designs was inferior and failed to meet critical performance parameters; that revelation eliminated an offeror from the competition. The reviewers compared the simulations resulting from the four remaining configurations along with more traditional source selection criteria (development cost, unit procurement cost, estimated lifecycle cost, proposal risk) to choose the winning aircraft and engine design. The new contracts required the winning vendors to maintain model currency throughout the developmental program and to deliver updated models to each other and to the Government's F-X Program Office at least quarterly. The prime contractors signed Nondisclosure Agreements (NDAs) with each other that assessed stiff penalties for intellectual property violations and for negligent handling of proprietary information. The NDAs also authorized the prime contractors to inspect each other's information protection protocols. The primes established similar NDAs with any subcontractors with access to proprietary information.

By the time the Engineering and Manufacturing Development phase started in 2019, the HPCMP possessed 200-petaflop computational capability, reducing the required run time for a given condition by 85 percent. Additionally, the HPCMP had released a new version of *Kestrel* with a more refined mesh generation logic that automatically varied the grid density with the complexity of the aircraft geometry and the complexity of the airflow over the aircraft surface. For instance, the grid was smaller near the leading and training edges of the wing than in the middle of the wing. As the angle of attack increases, and with it the propensity for turbulence and flow separation on the wing surface, *Kestrel* automatically increases grid density. This dynamic mesh generation capability eliminated the need to develop a worst-case grid, reducing the number of calculations required for a given condition, yielding another 15 percent reduction in run time.

The dramatic reduction in computational run time generated by the new supercomputers and the updated *Kestrel* software enabled the test team to generate a high-fidelity prediction of aircraft performance throughout the flight envelope. The test team leveraged the *Kestrel* predictions, in conjunction with a Design of Experiments-based approach, to develop a wind tunnel test matrix that was 20 percent smaller than used on the F-22, despite the larger flight envelope of the F-X. This was achieved by eliminating most heart-of-the-envelope test points, where *Kestrel* had proven to be most accurate. A similar approach optimized the test matrix for the ground testing of an engine prototype. The results of both tests updated the Digital Thread representation of F-X. Additionally, the HPCMP funded the Air Force Research Laboratory to investigate areas where the *Kestrel* predictions were incorrect and to update the aerodynamic algorithms to reduce these errors on future simulations.

As a result of a solid initial design, a stable funding profile, and a fixed requirements set, the first F-X prototype was delivered in 2025 – a full year ahead of schedule. The key to early delivery was design stability and accurate performance predictability throughout the design process. Test pilots at the Air Force Test Center conducted a performance, stability, and control flight test matrix that was 15 percent smaller than that proposed for the F-22 twenty-five years earlier; unlike the explosive growth in test matrix size seen once the F-22 began flying, the F-X matrix only grew by three percent over the two-year performance, stability, and control test phase.

The one major setback to the F-X development program occurred during avionics systems testing. The CREATE program's early emphasis on aerodynamics, thermal performance, and structural loading resulted in minimal investment in simulating the in-flight behavior of advanced electronics. The four-year avionics test program stretched to five because of electromagnetic interference from the new radar, which caused spurious output on the aircraft datalink system. The F-X program reached Initial Operational Capability (IOC) in late 2032, following the fielding of three squadrons at Langley Air Force Base in Virginia.

Throughout the test program, the onboard aircraft instrumentation recorded detailed information about every second of flight. In 2025, shortly after the HPCMP reached 2,000-petaflop computational capability, the engine vendor implemented a Digital Twin trial. The Air Force installed two new, heavily pedigreed engines in the F-X airframe. After each sortie and each engine run, engineers exposed the Digital Twin for each engine to the same conditions seen by the real engines. At first, each hour of simulated aging required an hour of supercomputer runtime per engine; at IOC, the HPCMP had 20,000 petaflops (or 20 exaflops) of computational capability, enabling an hour of simulated aging on both engines in less than five minutes.

In spite of the massive processing capability available in 2032, this was still insufficient to model every failure mode in every component in every system on the aircraft's Digital Twin. Additionally, the Air Force determined that the time, expense, and opportunity cost to do so did not justify modeling every system, even if the computational resources were available. Instead, engineers modeled only the engine, airframe, and key structural components at IOC; for three squadrons of 18 planes each, even this demanded a large chunk of the HPCMP resources. Initially, the Air Force employed a traditional flight-time based maintenance program. For instance, the Air Force initially removed and thoroughly inspected engines after every 2,000 hours of operation. At each inspection, engineers compared each engine to its Digital Twin, and updated the latter to reflect the inspection results. Within two years, as the Digital Twin became more accurate, the inspection interval was expanded to 5,000 hours or projected failure. It took another 10 years to convince the Air Force maintenance community to trust the Digital Twin projections for the full 10,000-hour engine life. By that time, maintainers used commercially available tablet computers to perform post-flight aging.

The above scenario is full of aspirational optimism, but it clearly illustrates the potential technical benefits of Digital Thread and Digital Twin as an instantiation of the DSM concept. These benefits include an earlier understanding of how the proposed requirements translate into projected system performance, a better understanding of the requirements trade space and the system impact associated with those trades, earlier opportunity for user buy-in based on more realistic models, earlier identification and resolutions of design issues, more accurate failure prediction, and more effective lifecycle sustainment. The scenario also highlights some important issues, such as the need for stable requirements. Developers cannot stabilize the design if the requirements are in constant flux; no matter how much computational power they have at their disposal. Likewise, when the budget changes dramatically, the requirements must also change to balance the books; hence, the need for predictable funding. The scenario also highlighted the need for the customer (in this case, the Air Force) to drive the implementation through such measures as contractual clauses. Additionally, in order to make the digital fly-off fair, all vendors were required to submit models that were compatible with the Government's CREATE toolset, which also enabled the Government analysts to make adjustments for differing assumptions in the various models. In addition to demonstrating the

importance of having the right software, the scenario also demonstrated how an imbalanced emphasis on one design area (e.g., air vehicle performance) could cause the design team to overlook other crucial aspects of the design – in this case, the electromagnetic interference, an area where many recent fighter aircraft development programs have struggled (The CREATE program is currently developing a radio-frequency module to help avoid such electromagnetic issues). Finally, it acknowledges that just because we *can* model every component in the system does not necessarily mean we *should*.

This scenario might not be as far-fetched as it appears. During his AIAA presentation, Walker (Kraft 2015) indicated the Air Force has already identified four "pathfinder" acquisition programs for application of the Digital Thread/Digital Twin concept: the Joint Surveillance and Target Attack Radar System (JSTARS) Recapitalization Program, the Air Force Next Generation Trainer (T-X) Program, the Global Positioning System (GPS) Operational Control Segment (OCX) Upgrade, and the Long Range Standoff (LRSO) Cruise Missile Program. These four pathfinders represent a diverse sample set to validate this new approach: T-X will likely be a commercial-off-the-shelf (COTS) aircraft, LRSO a from-scratch missile design, JSTARS a COTS airframe with specialized equipment, and GPS OCX a ground-based satellite control system. Walker emphasized the need for Digital Thread on LRSO, where accurate system models, including manufacturing and production models, will be required up front in order to ensure the proposed design solution can meet the stringent cost and capability requirements needed to ensure viability of the program.

TWO AUTHORITATIVE DIGITAL SURROGATES?

Assume for a moment that Digital Thread is able to overcome all of the above challenges. That still leaves a major elephant in the room: Digital Thread and MBSE appear to be on two separate and distinct paths, with the Systems Engineering community supporting MBSE and those such as Kraft and Walker, whose backgrounds are rich in discipline engineering, advocating the Digital Thread framework. If not addressed, this schism could lead to a two-model scenario. Although this may initially seem to be a minor problem, it results in the development, maintenance, and accreditation of two separate models, which drives additional cost and scope into the program. It also creates the potential for model mismatches caused by configuration control issues. For instance, an update in one model was never added into the other model, or perhaps it was incorrectly incorporated. To avoid this problem, a common language and/or interface needs to be established that integrates (or, as a minimum, federates) these two systems into a unified whole. Without this bridge, the potential benefit of the Digital Thread concept will significantly diminish.

THE WAY AHEAD

Clearly, we are very far from the "One Authoritative Model" utopia. However, that fact should not dissuade us from contemplating and perhaps taking steps in that direction. Indeed, we require action on multiple fronts to be successful. Mechanisms, such as new contract language, could demand delivery of these models with the weapon system. We need improvements to better protect digital data from hacking and espionage. The DoD could champion the effort to develop the framework and the interface to interconnect the various modules; the CREATE tools described above are a step in this direction. Each of the Services could make a requirements definition using the set-based design process and source selection via virtual fly-off essential elements in its acquisition strategy for all major programs; Walker's identification of five Air Force pathfinder programs is a major step in this direction. The DoD could also look for creative ways to employ the Digital Thread throughout the lifecycle, and invest in those opportunities when they arise. This is currently happening in Kraft's study to compare *Kestrel* aerodynamic predictions for the F-22 with actual wind-tunnel data in order to determine where to reduce tunnel testing on future aircraft developments, and to determine where to enhance the *Kestrel* algorithms to better reflect aerodynamic reality. However, none of these activities will yield the full potential of Digital Thread without continued investment in people, processes, and processor power. Each of the Services must remain committed to restoring the technical expertise they possessed 20 years ago, and to growing a new generation of in-house scientists and engineers capable of fully utilizing the power of the CREATE toolset and the Digital Thread concept. Through continuing efforts such as this, the model-based engineering community could make progress towards the vision of "One Authoritative Model." ■

REFERENCES

- BKCASE (Body of Knowledge and Curriculum to Advance Systems Engineering) Editorial Board. 2014. *The Guide to the Systems Engineering Body of Knowledge*. Version 1.3.1. R. D. Adcock, ed. Hoboken, US-NJ: The Trustees of the Stevens Institute of Technology.
- Blanchard, B. S., and W. J. Fabrycky. 2006. *Systems Engineering and Analysis*. 4th ed. Upper Saddle River, US-NJ: Pearson Education, Inc.
- Buede, D. M. 2000. *The Engineering Design of Systems: Models and Methods*. Hoboken, US-NJ: John Wiley & Sons, Inc.
- Christian, T. 2014. "Determining the Contents of the Digital System Model." Paper presented at the NDIA 17th Annual Conference on Systems Engineering. 30 October. Springfield, US-VA: National Defense Industrial Association. http://www.dtic.mil/ndia/2014system/16968ThursdayTrack4Christian.pdf/.
- Clifton, J. D., C. J. Ratcliff, and D. J. Bodkin. 2014. *Stability and Control Test and Evaluation Process Improvements through Judicious Use of HPC Simulations*. Eglin Air Force Base, US-FL: US Air Force Seek Eagle Office. http://www.hpc.mil/images/hpc-mpo_images/featured/clifton_18jun13.pdf.
- Dilanian, K. 2012. "Foreign Spying Against U.S. Companies on the Rise, FBI Says." *Los Angeles Times*. 29 June. http://articles.latimes.com/2012/jun/29/business/la-fi-economic-espionage-20120629/.
- Dilanian, K., and C. Parsons. 2013. "White House Adopts New Strategy to Safeguard Intellectual Property." *Los Angeles Times*. 20 February. http://articles.latimes.com/2013/feb/20/business/la-fi-obama-cyber-20130221/.
- Drucker, P. F. 2010. *The Drucker Lectures: Essential Lessons on Management, Society and Economy*. New York, US-NY: McGraw-Hill.
- Farr, J. V. 2007. *Simulation of Complex Systems and Enterprises*. Hoboken, US-NJ: Stevens Institute of Technology.
- Friedenthal, S., A. Moore, and R.Steiner. 2015. *A Practical Guide to SysML - The Systems Modeling Language*. 3rd ed. Waltham, US-MA: Elsevier, Inc.
- Haskins, C., ed. 2011. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 3.2.2. Revised by K. Forsberg, M. Krueger, D. Walden, and R. D. Hamelin. San Diego, US-CA: INCOSE.

- Hintson, R. 2014. "High Performance Computing Modernization Program Doubles Capabilities." 25 January. Lorton, US-VA: US Department of Defense. http://www.hpc.mil/images/hpcdocs/newsroom/HPCMP_Doubles_Capabilities.pdf/.
- INCOSE Technical Operations. 2007. *Systems Engineering Vision 2020*, version 2.03. Seattle, US-WA: International Council on Systems Engineering.
- Kaku, M. 2011. *Physics of the Future: How Science Will Shape Human Destiny and Our Daily Lives by the Year 2100*. New York, US-NY: Doubleday.
- Kinard, D. 2010. "The Digital Thread – Key to F-35 Joint Strike Fighter Afforadability." *Aerospace Manufacturing and Design*. 1 September. http://www.onlineamd.com/amd-080910-f-35-joint-strike-fighter-digital-thread.aspx#.VXwx78vbKcw/.
- Knapp, A. 2014. "China Once Again Boasts The World's Fastest Supercomputer." *Forbes*. November 17. http://www.forbes.com/sites/alexknapp/2014/11/17/china-once-again-boasts-the-worlds-fastest-supercomputer/.
- Kossiakoff, A., and W. N. Sweet. 2003. *Systems Engineering: Principles and Practice*. Hoboken, US-NJ: John Wiley & Sons, Inc.
- Kraft, E. M. 2014. "CREATE-AV and the Air Force Digital Thread." Paper presented at the 53rd AIAA Aerospace Sciences Meeting, Kissimmee, US-FL. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2015-0042.
- Kraft, E. M. 2013. "Expanding the Digital Thread to Impact Total Ownership Cost." Paper presented at the 2013 NIST Model-Based Enterprise Summit, Gaithersburg, US-MD. December 18. National Institute of Standards and Technology. http://www.nist.gov/el/msid/upload/1Kraft_DigitalThread.pdf/.
- Kraft, E. M., K. Baldwin, J. P. Holland, and D. E. Walker. 2015. "The Digital System Model: The New Frontier in Aerospace & Defense Acquisition." Paper presented at the AIAA SCITECH 2015. 12 January. Kissimmee, US-FL. American Institute of Aeronautics and Astronautics. http://jamesdrewjournalist.com/tag/scitech-2015/.
- Kurzweil, R. 2005. *The Singularity Is Near: When Humans Transcend Biology*. New York US-NY: Penguin.
- Larman, C. 2005. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd ed. Westford, US-MA: Pearson Education.
- Larson, W. J., D. Kirkpatrick, J. J. Sellers, L. D. Thomas, and D. Verma. 2009. *Applied Space Systems Engineering*. Boston, US-MA: McGraw-Hill Companies, Inc.
- Long, D., and Z. Scott. 2011. *A Primer For Model-Based Systems Engineering*. 2nd ed. Blacksburg, US-VA: Vitech Corporation.
- Maccaba, B. 2014. "Why Software Doesn't Follow Moore's Law." Forbes. 19 May. http://www.forbes.com/sites/ciocentral/2014/05/19/why-software-doesnt-follow-moores-law/.
- Mackenna, A. 2012. "Using CREATE's Rapid Ship Design Environment to Perform Design Space Exploration for a Ship Design." Paper presented at the NDIA Conference on Physics-Based Modeling in Design & Development for US Defense. 7 November. Denver, US-CO: National Defense Industrial Association. http://www.dtic.mil/ndia/2012physics/Wednesday14965_Mackenna.pdf/.
- Meakin, R. L. 2011. "Multi-Disciplinary, Physics-Based Simulation Software Products of the CREATE-AV Project." Paper presented at the NDIA SE Division M&S Committee. 15 February. Washington, US-DC: National Defense Industrial Association. http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Documents/Committees/M_S%20Committee/2011/February/NDIA-systems engineering-MS_2011-02-15_Meakin.pdf.
- Millar, L., and Knight, B. 2013. "Chinese Spies Hacked Secret US Weapons Systems Including F-35 Joint Strike Fighter." *ABC News Online*. 28 May. http://www.abc.net.au/news/2013-05-29/reports-chinese-hackers-targeted-us/4719352/.
- Newton, R. 2011. "Manufacturing With a Digital Thread." GraphicSpeak. 21 April. http://gfxspeak.com/2011/04/21/manufacturing-with-a-digital-thread/.
- Open Source Specification Project. 2012. *OMG Systems Modeling Language (SysML)*. Version 1.3. Needham, US-MA: Object Management Group. http://www.sysml.org/docs/specs/OMGSysML-v1.3-12-06-02.pdf/.
- Pae, P. 2003. "$1 Billion in Contracts Yanked From Boeing." *Los Angeles Times*. 25 July. http://articles.latimes.com/2003/jul/25/business/fi-boeing25/.
- ––. 2003. "Boeing Satellite Unit Stole Trade Secrets, Suit Says." *Los Angeles Times*. 28 May. http://articles.latimes.com/2003/may/28/business/fi-boeing28/.
- Reed, J. 2012. "Did Chinese Espionage Lead to F-35 Delays?" *DefenseTech*. 6 February. http://defensetech.org/2012/02/06/did-chinese-espionage-lead-to-f-35-delays/.
- Stackley, S. J. 2012. "Use of In-House Engineering and Technical Resources." Washington, US-DC: Assistant Secretary of the Navy. http://www.secnav.navy.mil/rda/workforce/Documents/ASNRDAUseofInHouseEngineeringTechResourcesFeb20121.pdf.
- Tuegel, E. J., A. R. Ingraffea, T. G. Eason, and S.Spottswood. 2011. "Reengineering Aircraft Structural Life Prediction Using a Digital Twin." *International Journal of Aerospace Engineering*. January. New York, US-NY: Hindawi Publishing Corp. http://www.hindawi.com/journals/ijae/2011/154798/.
- US Department of Defense. 2009. *DoD Instruction 5000.61: DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)*. http://dtic.mil/whs/directives/corres/pdf/500061p.pdf/.
- United States. 1996. Public Law 104-294: *Economic Espionage Act of 1996*. http://www.gpo.gov/fdsys/pkg/PLAW-104publ294/pdf/PLAW-104publ294.pdf/.
- US Air Force. 2013. *Global Horizons: United States Air Force Global Science and Technology Vision*. Washington, US-DC: US Air Force Office of the Chief Scientist. http://www.defenseinnovationmarketplace.mil/resources/GlobalHorizonsFINALREPORT6-26-13.pdf/.
- US Air Force. 2015. *Statement of Objectives (SOO) for CREATE AV Assessment & Analysis Technology Development Broad Area Announcement (BAA) Call #3*. Wright-Patterson Air Force Base, US-OH: US Air Force Lifecycle Management Center. https://www.fbo.gov/index?tab=documents&tabmode=form&subtab=core&tabid=3d53de10d4bf5be5cedee2b3adb6b8ef/.
- Vanderperren, Y., and W. Dehaene. 2006. "From UML/SysML to Matlab/Simulink: Current State and Future Perspectives." Paper presented at the Design, Automation and Test in Europe (DATE) Conference. 6-10 March. Munich, DE. http://www.omgsysml.org/from_umlsysml_to_simulink.pdf/.
- Zimmerman, P. 2014.. "Digital System Model Development and Technical Data." Paper presented at the 17th Annual NDIA Systems Engineering Conference. 30 October. Springfield, US-VA: National Defense Industrial Association. http://www.acq.osd.mil/se/briefs/16969-2014_10_30-NDIA-SEC-Zimmerman-DSM-vF.pdf/.

AUGUST 2015
VOLUME 18 / ISSUE 2

MBSE in Perspective

Jack Ring, jack@ontopilot.com and **Azad M. Madni,** azad.madni@usc.edu

In more than a decade of MBSE-related activity several INCOSE participants said and did much. This article suggests that it is time to pause and reflect on what lessons we have learned to date. Ideally such reflection should reveal MBSE status relative to at least three factors:

a) What MBSE users and their systems customers really need, how well both are being served, and which users are qualified to engage in systems work?
b) What kinds of MBSE methods, processes, and tools are necessary, sufficient, and efficient for systems engineering practitioners in the future? In particular, what kind of MBSE best enables the modeling of autonomous systems capable of performing in non-deterministic environments, and then enables follow-up with advice and consent through subsequent phases of system realization, activation, and value generation?
c) What kinds of MBSE techniques are necessary, sufficient, and efficient for modeling sociotechnical systems that fall under the rubric of complex systems engineering?

What Users and Their Customers Really Need

Having seen descriptive modeling of problem situations and prescriptive modeling of envisioned situations in the context of "you mean we are playing for money?," we are becoming increasingly convinced that an effective MBSE approach must enable the creation of models that clearly address:

- **Effects**: Describes the influence an existing system has on its context and prescribes the impact an envisioned system is intended to have on that existing system, "Model-based Conceptual Design," (INCOSE 2014). Effects are characterized as changes that the system causes or influences, or are supposed to cause or influence on system gradients, structure, or content, "Systems of the Third Kind: Distinctions, Implications, and Initiatives" (Ring 2012). This includes not only stimulus-response scenarios but also Do No Harm constraints, an important facet, ineffectively addressed in MBSE discussions to date.

- **When and While needed**: Also called Operational Availability, or Service Continuity. While this notion has been around for decades, MBSE has yet to address anticipating or ensuring Operational Availability in examples to date. It is important to note that system resilience and adaptability are sub-factors or enablers of Operational Availability or Service Continuity.

- **Conveyed and Confirmed**: Systems engineers must communicate the model to those expected to evolve the model into an actual, physically realized, deployed system. Specifically, the model must pass 'incoming inspection' conducted by appropriate stakeholders. Further, the continual verification of the model as being accurate and adequate as the system continues to evolve and improve is a requirement.

How Well "Systems People" are Being Served

Reflection is in order when it comes to assessing achievement of Requisite Variety in the user's system, Requisite Saliency in the sociotechnical system that delivers the user's system, and Requisite Parsimony in the above. After Action Reviews or post hoc surveys of actual users foster and facilitate reflection, particularly from those engaged in marginally successful projects. A useful reference in this regard is *The Reflective Practitioner: how professionals think in action* (Schön D. A.1983).

Requisite Qualification to Engage in Systems Work

This includes: MBSE self-checking capabilities that detect design deficiencies; assessment of correlation, if any, between INCOSE Certification and MBSE usage, particularly MBSE effects on productivity and innovation, as well as scalability limits on MBSE usage (project size, geographic dispersion, rate of change of intended Effects, Operational Availability or Conveyance) and, finally, practitioners' attitude toward MBSE.

What Kinds of MBSE Are Necessary, Sufficient and Efficient for Systems Engineering Practitioners in the Near Future?

In 2007, a reflection session involving 20+ experienced personnel reported in "Strategy for a Whole Systems Modeling Capability (Ring 2009)" and concluded that there was a pressing need for a system modeling language by 2015 that was an order of magnitude more semantically expressive than extant examples, e.g., SysML. Further, 10-year old kids today are using SIMCity and similar software to devise self-adaptive, goal-seeking simulations of envisioned systems. Does the MBSE evolution roadmap anticipate and accommodate these kinds of unconventional designers expected to enter the work force by 2025?

What Enhancements to MBSE will be Necessary, Sufficient and Efficient for Modeling Sociotechnical Systems that will be Under the Purview of Systems Engineering?

We expect that upon due reflection, it will become apparent that most system projects conducted in accord with current standards, guides, and handbooks do not explicitly include practitioner learning as an integral facet of systems engineering activity. Further, we also expect the conclusion that this gap is due to a process view of work, rather than a results or outcome-oriented view. When the view of systems engineering begins as a story or scenario of decisions made, and artifacts produced (Madni, 2015), then appropriate reflection can identify the necessary, sufficient and efficient capabilities that the systems engineering project-level workforce must have. Further appropriate reflection will identify the arrangement of functions and features that the project-level workforce must exhibit, and the scenarios of fallibility tests that such workforce must experience to produce artifacts that users can trust. This advance entails new kinds of

thinking, particularly about implicit aspects of systems as foretold by Escher's Drawing Hands (http://www.wikiart.org/en/m-c-escher/drawing-hands). This is because each decision made regarding the intended operational system changes the context faced by systems engineering practitioners the next day. The latter influences the content, structure, and behavior required of the resulting engineered system.

There is little doubt that due reflection will lead to new and superior formulations of MBSE principles, theory, process, architecture, and tools. As importantly, it will lead to faster improvement of systems engineering workforce competencies. Every project will then be seen not only as a way of delivering value to stakeholders but also as a learning environment for systems engineering practitioners. Frederick Herzberg, University of Utah, who proposed six levels of growth in developing character, notably, recommended a pattern for such learning: a) knowing more; b) understanding more; c) creativity – qualitative leap of insight; d) effectiveness in ambiguity; e) individuation – some unique talent that sets you apart; and f) real growth – the ability to pursue character accomplishments and self respect separate and distinct from personality adjustments for status and security.

Herzberg notes that dialog with others can help with the first three. However, the last three come from self-discovery, i.e., reflection about reflection. Hopefully, the future MBSE vision will encourage every systems engineering practitioner to be simultaneously both a producer and a learner. ■

REFERENCES

- INCOSE (International Council on Systems Engineering). 2014. "Model-based Conceptual Design." *INSIGHT* 17(4).
- Madni, A. M. 2015. "Expanding Stakeholder Participation in Upfront Systems Engineering Through Storytelling in Virtual Worlds." *Systems Engineering* 18(1): 16-27.
- Ring, J. 2009. "Strategy for a Whole Systems Modeling Capability." Paper presented at the 3rd annual IEEE Systems Conference, Vancouver, CA-BC: 279-284.
- Ring, J. 2012. "Systems of the Third Kind: Distinctions, Implications, and Initiatives." *INSIGHT* 15(2): 9-12.
- Schön, D. A. 1983. *The Reflective Practitioner: how professionals think in action*. Ithaca, US-NY: Sage Publications.

Systems Engineering: The Journal of The International Council on Systems Engineering

Call for Papers

Systems Engineering journal is intended to be a primary source of multidisciplinary information for the systems engineering and management of products and services, and processes of all types.

Systems Engineering is the archival journal of, and exists to serve the following objectives of, the International Council on Systems Engineering (INCOSE):

- To provide a focal point for dissemination of systems engineering knowledge
- To promote collaboration in systems engineering education and research
- To encourage and assure establishment of professional standards for integrity in the practice of systems engineering
- To improve the professional status of all those engaged in the practice of systems engineering
- To encourage governmental and industrial support for research and educational programs that will improve the systems engineering process and its practice

The journal supports these goals by providing a continuing, respected publication of peer-reviewed results from research and development in the area of systems engineering. Systems engineering is defined broadly in this context as an interdisciplinary approach and means to enable the realization of successful systems that are of high quality, cost-effective, and trustworthy in meeting customer requirements.

The journal will accept and review submissions in English from any author, in any global locality, whether or not the author is an INCOSE member. A body of international peers will review all submissions, and the reviewers will suggest potential revisions to the author.

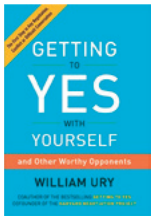
Editorial selection of works for publication will be made based on content, without regard to the stature of the authors. Selections will include a wide variety of international works, recognizing and supporting the essential breadth and universality of the field. Final selection of papers for publication, and the form of publication, shall rest with the editor.

Submission of quality papers for review is strongly encouraged. The review process is estimated to take three months, occasionally longer for hard-copy manuscript.

Systems Engineering operates an online submission and peer review system that allows authors to submit articles online and track their progress, throughout the peer-review process, via a web interface. All papers submitted to *Systems Engineering*, including revisions or resubmissions of prior manuscripts, must be made through the online system. Contributions sent through regular mail on paper or emails with attachments will not be reviewed or acknowledged.

All manuscripts must be submitted online to *Systems Engineering* at ScholarOne Manuscripts, located at: http://mc.manuscriptcentral.com/SYS

Full instructions and support are available on the site, and a user ID and password can be obtained on the first visit.



Toward a Larger View of Negotiation

Getting to Yes with Yourself: and Other Worthy Opponents

By William Ury
New York, US-NY: HarperCollins, HarperOne, 2015 (ISBN: 978-0-06-236338-1)

Reviewed by Denise Howard, SDenise.Howard@incose.org

Negotiation is not just an occasional and optional activity in engineering projects, but a frequent and critical activity, often significantly affecting client satisfaction, perhaps the most important initial measure of project success. Negotiating skill may determine whether creative responses to problems are encouraged or bypassed, greatly affecting project outcomes. Expert engineers have long been aware of the importance of good negotiating practice. Barry Boehm's Theory W and his emphasis on win-win negotiation is a familiar example from software engineering. In addition, from an architecting perspective, there may be a simple reason supporting win-win: anything less builds in mismatches, which have the potential to become difficulties sometime, somewhere.

The work of the Harvard Negotiation Project has been the gold standard for win-win negotiation since the 1981 publication of Getting to Yes, by cofounders Roger Fisher and William Ury. Lately, attention has partially turned from those sitting across the table, to us, opening something of a frontier. It is this frontier, which William Ury traverses, in his latest book.

Systems engineers will probably feel at home with the six-step graphical map provided by the author, a clockwise directed circle, with three main categories, "Yes" to Self, Life, and Others; pairs of successive chapter level topics are associated with each category. The addition of Life to Self and Others represents an important expansion of thinking, which now extends to include the web of things affected by the negotiation. In this larger view, Self, Life, and Others becomes the basis for win-win-win.

A brief description from the first step in *Getting to Yes with Yourself* may convey some sense of the book: Negotiators are often urged to put themselves in the shoes of the other. However, in a turnabout, Ury urges us to put ourselves in our own shoes. One reason to put on our own shoes is that people often do not allow themselves the understanding they readily accord to others. Their needs then become difficult for them to understand and their ability to understand the needs of others becomes limited as well. Another reason: "As Carl Rogers, one of the founders of humanistic psychology once noted: 'The curious paradox is that when I accept myself as I am, then I can change."

In the discussion, illustrated by professional and personal stories, systems thinkers will find familiar motifs, including virtuous and vicious circles, paradox, and interconnectedness. These descriptions seem to be emerging, not yet fully connected or consciously integrated, but intriguing. Perhaps these concepts, so familiar in systems, provide the best articulation of some underlying structure, yet only partially apparent. It seems too soon to know, and too important to hurry.

Getting to Yes with Yourself is short and perhaps deceptively simple. The ideas are clear, but mastery will likely be beyond a lifetime of dedication. The most basic start though, should bring improvements. Heuristics, as used in systems architecting, may help overcome a problem that the author notes, which is that it is difficult to remember what to do as a demanding situation unfolds. Table 1 suggests some candidate heuristics; readers will likely be able to shape better ones to share.

Since how we negotiate embeds itself in our architecture, and who we are fundamentally affects how we negotiate, it is inescapable: *Who we are is what we build*. Ury's new book is therefore a serious step toward better architecting and better building, for us, for others, and for life itself. ■

Table 1. Some provisional negotiating heuristics

Name	Heuristic	Chapter
Acceptance	"Put yourself in *your* shoes."	1
Balcony	When the going gets tough, "go to the balcony."	1
Inner BATNA	"Own your needs:… We can make a strong unconditional commitment to ourselves to take care of our deepest needs, *no matter what* other people do or don't do."	2 (55)
Connection to Life	"Remember your connection to life: … We are not separate at all, as Einstein points out, but rather inextricably woven into a larger web of human and other living beings."	3 (70-71)
Present Opportunity	"[T]he key is to look for the present opportunity. In most situations, … there is an opening if we are attentive enough to see it."	4 (93)
Trust	Choose trust over fear, where trust is defined as "the confidence that you will be able to deal with the challenges that come your way."	4 (104)
Respect	Basic respect is always a requirement.	5
Triple Win	Create win-win-win outcomes.	6
Encouraging Generosity	Create value relentlessly, for Self, Others, and Life.	6

AUGUST 2015
VOLUME 18 / ISSUE 2

Needed: Practitioner Attention to Systems Engineering Delivery of Sustainable Value

Rick Dove, dove@parshift.com

As systems engineers we all want the fruits of our labors to be successful — to be valuable to our customers. Value comes from systems that deliver needed and usable functionality at least long enough to provide a respectable return on investment. We know how to do the business math and calculate the operational life required to justify the investment in an expected and defined system employment environment.

Before a return on investment goes positive, denial or erosion of value can occur. If the systems we use every day to get our jobs done cease to work, that's value denied. If they become less productive, that's value eroded. Value denied can happen for many reasons, which we expect the system designers have anticipated with mitigation preparations. A simple example is the expectation that systems will break occasionally for whatever reason and require service, and the anticipation and provision of the service needs are part of the greater system. All breakage is not complete denial of functionality, as some manifest as reduced effectiveness. We all use email systems, for instance, and have found their functionality and ability to support high productivity degrading over time – more spam to deal with, more restrictions on communication imposed by increasing security concerns.

Systems security is clearly a contribution to sustainable value. Resistance to attack is necessary to prevent denial of value. Resilience under attack is necessary to prevent erosion of value. The challenge is that systems security is not a capability delivered in isolation. It is an integral part of the design and delivery of systems. This article is going to focus on the threat to sustainable value posed by intelligent and determined system adversaries, and suggest that each of our working groups has relevant expertise that needs to come forward.

Three things this article wants to accomplish: 1) show what is necessary to have a better use of systems, 2) show where INCOSE responsibility lies to define and deliver sustainable value, and 3) challenge each working group to contribute in their area of expertise and interest. The July 2016 issue of *INSIGHT* will use system security as the backdrop for a discussion of sustainable value. This does not require security expertise, but it does require the application of relevant knowledge in each working group. The July 2016 issue of *INSIGHT* will offer a way to make that contribution from a practitioner's point of view.

To put the problem in context, the adversary is agile – leading in innovation and adaptability (Dove and Shirey 2010). The only effective countermeasure is an equally agile system security capability. The challenge of *agility* in both threats and safeguards is major, yet remains ignored, or hampered by incompatible systems engineering.

In systems engineering, security is closeted as a specialty engineering process, too easily avoided as a prime functional need. Functionality builds on a presumed foundation of sustainable operation, but security issues undermine that foundation. Sustainable system operational capability is the first functional need. Maslow's hierarchy of needs for organisms recognizes this at the lowest levels as sustenance (food, air, etc.) and safety (Maslow 1943). Only after assurance of existence are higher forms of functional performance considered.

From (Maslow 1943) "*The safety needs*. — If the physiological needs are relatively well gratified, there then emerges a new set of needs, which we may categorize roughly as the safety needs.… The organism may equally well be wholly dominated by them. They may serve as the almost exclusive organizers of behavior, recruiting all the capacities of the organism in their service, and we may then fairly describe the whole organism as a safety-seeking mechanism. Again we may say of the receptors, the effectors, of the intellect and the other capacities that they are primarily safety-seeking tools. … Practically everything looks less important than safety."

The Systems Security Engineering Working Group (WG) by necessity focuses on general system security engineering issues and standards, and cannot bring to bear the focused practitioner's expertise resident in the diversity of INCOSE WGs. All WGs have a responsibility to help address security issues in volatile and evolving systems engineering and systems operational environments.

Some compelling examples follow, and then some guidance for bringing WG expertise to bear on the issues. The intention is to show new needs, show where WG contribution is needed, and sound a call to arms. All systems are finding themselves in new and different environments, vulnerable to unprecedented attack for which our engineered systems are unprepared. If we do not guide the preparation, who will?

RESILIENT SYSTEMS WG

Resilient systems are inherently more secure, yet the fundamentals of resilient

systems have nothing to do with security devices, security software, or security engineering expertise. A resilient system recovers from whatever impedes its functionality regardless of source. System resilience as a security concept is recognized strongly in security circles, but weakly in systems engineering circles. Resilient-systems engineering does not have to be deeply versed in the methods of attackers, but does have to recognize that the malicious intender is succeeding in degrading system performance that manifest as only an intelligent and determined attacker can cause. Resilient design is necessary to build security into the system instead of trying to test it out. This WG has an important opportunity to put some focus on malicious and intelligent system-degradation intent.

ARCHITECTURE WG

Only an architectural design concept that enables both reactive resilience and proactive composability will enable agile security, providing the possibility of at least maneuverable parity with the agile adversary. The Agile Systems and Systems Engineering WG outlined fundamental architectural concepts (Dove 2013; Dove and LaBarge 2014). It is time for the Architecture WG to carry this forward, recognizing that architecture enables or inhibits system security, and providing systems engineering guidance for effective security integration with architecture.

SYSTEMS OF SYSTEMS WG

Systems of Systems (SoS) brings new forms of security concerns. Constituent systems change asynchronously and independently. The concepts of trust need considerable rethinking. Vision 2025 (INCOSE 2014) suggests, "The Internet of Things extends the SoS challenge beyond interconnected computers and users, to include increasingly interconnected systems and devices that monitor and control everything from household appliances to automobiles. … Techniques for analyzing interactions among independent systems and understanding emergent behaviors in SoS must mature and become commonplace."

COMPLEX SYSTEMS WG

The adversarial community is a complex adaptive and self-organizing system, leading in attack innovation by indiscriminately sharing methods, targets, vulnerabilities, attack tools, and resources. The protection community has problems in functioning like an effective community system, and needs a complex systems engineering understanding and approach to solve the requirements and impediments of community-sharing systems.

HUMAN SYSTEM INTEGRATION WG

This is a big one. A major problem with security is that it often has a user interface that impedes personal productivity and organizational goal accomplishment. For instance, everybody's pet peeve: change your password every XX days, make it long and unmemorable, but don't write it down. Therefore, what happens is that people and organizations ignore or otherwise skirt the security requirements when the perception is that these are impediments. It appears that security engineers think more about protective functionality than the realities of usability. Systems engineering must address this with guidance provided by appropriate expertise, but ultimately by reducing reliance on human compliance as a human system integration necessity. The problem in human-system security integration is that we have too much integration and not enough usability. Security expertise needs complimenting guidance from human-system integration expertise.

REQUIREMENTS WG

System security requirements are stuck on compliance with standards and best practices, and need to look more at responsibility, intent, and operational adaptability for future unknown threat vectors rather than inadequate bets on check-off boxes for yesterday's conformance. Here, we need a new approach to requirements, one that speaks to intent and objectives rather than conformance, one that speaks to an environment constantly evolving into unexpected territory. This moves into an area of requirements language that addresses sustainable objectives, recognizing competent best efforts over duplication of best practices. While this WG's Charter scope states "discipline-specific considerations of related areas are avoided," the expansion to dealing with evolving system requirements for evolving environments is not a security-specific need – all system requirements need to consider unpredictable future needs if sustainable functionality is valued. We need to find a way to describe security requirements in the context of time-invariant objectives that define sustainable value.

RISK MANAGEMENT WG

Risk management has a history based on statistical profiles of the past. The risks faced in security keep changing in innovative ways that one cannot evaluate statically or objectively with mathematics at project inception. We need to evaluate systems security design characteristics in the context of risk and not compliance. Systems engineering requires a new fundamental approach for the evolving risk environments our systems cope with today. A new approach should not focus on security alone, as the system's environmental evolution affects all aspect of system risk; but the nature of evolving security risk offers a capability target for establishing new risk assessment requirements, objectives, and methods. Collier (2014) provides the argument supporting a need for new thinking.

CRITICAL INFRASTRUCTURE PROTECTION AND RECOVERY WG

Critical infrastructure is threatened by multiple vectors, from physical destruction to cyber-control intervention. Stuxnet was the wakeup call (INCOSE 2015). Control systems are high-leverage targets of opportunity, from industrial to banking, transportation to healthcare, and defense to offense. To date, INCOSE has no WG focused on control systems. Industrial control systems, for critical infrastructure in particular, are in need of systems engineering perspectives and attention. This WG cannot ignore the Achilles heel of critical infrastructure.

ENTERPRISE SYSTEMS WG

Sony's recent blackmail (Huddleston 2014) and subsequent economic effect appears to have had a nation-state behind it. This event appears to have shaken the security-priority apathy out of many corporate boardrooms. Enterprise IT security strategy is clearly inadequate, and only systems engineering responsibility and attention can change this on a meaningful scale.

INDUSTRY WG

The industry working groups of Automotive and Healthcare should have particular concern in the systems security area. Current automotive security approaches have demonstrated cyber-vulnerabilities, and now come self-driving cars and car-collaborative communications. Security-in-healthcare shortfall is becoming a regular news item, with medical device security gaining increased attention. These problem areas need attention in systems engineering development, to enable security engineers to exercise their expertise.

AUTONOMOUS SYSTEM TEST AND EVALUATION WG

Now here is an area of critical security concern. Autonomous systems with cyber-controls are highly tempting targets of attack. How can you test an autonomous device for something not yet revealed as a threat vector? The testing need is beyond what current test and evaluation best practice and procedure encompasses. Autonomous systems are likely to manifest

emergent behavior. Emergent behavior is necessary, but some will be bad. Without the capability for emergent behavior to be identified and evaluated, such systems will be inadequate in a world of unpredictable situations . We need new test and evaluation approaches for non-deterministic systems.

SYSTEMS ENGINEERING EFFECTIVENESS WG

This WG's Charter states that its "scope is to address the collection, analysis, and distribution of quantitative evidence of the value of SE throughout the system life cycle." Embedding systems engineering responsibility for system security and showing the value proposition doesn't require security expertise. Systems engineering effectiveness is hard to imagine without system-engineered sustainability.

THINGS TO THINK ABOUT WHEN CONSIDERING WG CONTRIBUTIONS

Agile security is addressed here as a system-engineered capability which enables agile systemic behavior. Agile security is both a resilient (reactive) and composable (proactive) capability to effectively address the unpredictable and evolving environment of adversarial attack – a systems-engineered capability enabled by systems architecture and design. This definition is agnostic to the type of system, and can encompass information systems, cyber physical systems, physical systems, enterprise systems, social systems, infrastructure systems, and military systems.

The author feels the considerations listed below are necessary for next-generation agile security, but unlikely sufficient. The intention below is to express these considerations in minimal but concise words, leaving large latitude on how to achieve the intent in guidance contributions from WG expertise:

- Holistic Systems Engineering – Need: Effective security reconfiguration/augmentation/evolution. Intent: Holistic systems thinking, security embedded in system design as part of system functionality, security system architecture structured and designed for adaptability and evolution.
- Collective Intelligence – Need: Shared knowledge base. Intent: Knowledge and experience shared and assimilated individually for organic thought and action. See Boyd (1987).
- Harmony – Need: Embraceable rather than enforceable security. Intent: Support rather than inhibit human and organizational productivity and goal achievement.
- Self-Organization – Need: Response capability at cyber-speed. Intent: Response decision and action automated or human-enabled at the point of maximum knowledge.
- Consistency – Need: Eliminate undependable security functionality. Intent: Systemically automated security devoid of reliance on human compliance.
- Distrust – Need: Safely employ people and components that change asynchronously over time. Intent: Component-level self-protection that distrusts all interaction. See Kott (2014) for acknowledgement of the need for mixed trust systems.
- Shape Shifting – Need: Static system architecture that one can observe and probe over time to discover vulnerabilities. Intent: Moving target defense (change functional methods) and offense (evolve capability and functional methods). See Horowitz (2012, pages 23-25) for defense.
- Component Conscience – Need: Component self-awareness and evaluation of behavior. Intent: Self monitoring internal conscience as an embedded independent agent. See Dove (2012) and Horowitz (2015, pages 73-105).
- System Conscience – Need: Detection of unpredictable emergent system and system-of-systems behavior. Intent: System-wide emergent behavior monitoring by embedded independent agents.
- Peer Behavior Judgment – Need: Detection of aberrant component operational behavior caused by design flaws, system malfunction, human error, and malevolent control penetration. Intent: Peer-peer behavior monitoring. See Dove (2009).

A CALL TO ARMS

The July 2016 *INSIGHT* issue will deal with Agile Security, as a joint leadership effort between the Systems Security and the Agile Systems WGs – but these two WGs do not intend to dominate the contributions. We instead want to engage many WGs in agile-security contributions relevant to their interests and expertise. WGs can contribute by helping to identify the issues relative to their knowledge, and should not try to offer security solutions, but rather sketch what appropriate solutions need to consider from the WG expertise point of view. This can be a systems engineering beginning for the guidance of next generation security. Contact rick.dove@parshift.com with your intent to contribute before the end of September 2015. First draft 2,000 word essays will be due end of November 2015 for open review at an International Workshop 2016 collaborative workshop. ■

REFERENCES

- Boyd, John, 1987. Organic design for command and control. One of the briefings in *A discourse on winning and losing*. Maxwell Air Force Base, AL: Air University Library Document No. M-U 43947. http://dnipogo.org/john-r-boyd
- Collier, Zachary A. Igor Linkov, Daniel DiMase, Steve Walters, Mark (Mohammad) Tehranipoor, James H. Lambert. 2014. Cybersecurity Standards: Managing Risk and Creating Resilience. *Computer*, IEEE Computer Society, September. www.parshift.com/Files/PsiDocs/Paths&MethodsForPeerBehaviorMonitoringAmongUnmannedAutonomousSystems.pdf
- Dove, Rick. 2009. Paths for Peer Behavior Monitoring Among Unmanned Autonomous Systems. ITEA Journal 30: 401–408, September. Best publication of the year award. www.parshift.com/Files/PsiDocs/Pap090901IteaJ-PathsForPeerBehaviorMonitoringAmongUAS.pdf
- Dove, Rick and Laura Shirey. 2010. "On Discovery and Display of Agile Security Patterns." Presented at the 8th Conference on Systems Engineering Research. Hoboken, US-NJ, March 17-19. www.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf
- Dove, Rick. 2012. Righteousness and Conscience as a Path to Socially Acceptable Autonomous Behavior. *INSIGHT* 15 (2): 32-34. International Council on Systems Engineering, July. www.parshift.com/s/120701Insight-RighteousnessAndConscience.pdf
- Dove, Rick, 2013. Sustainable Agile Security Enabled by Systems Engineering Architecture. *INSIGHT* 16 (2): 30-33. International Council on Systems Engineering, July. www.parshift.com/s/130701Insight-EnablingSustainableAgileSecurity.pdf
- Dove, Rick and Ralph LaBarge. 2014. "Fundamentals of Agile Systems Engineering – Part 1." International Council on Systems Engineering International Symposium 2014, Las Vegas, US-NV, 30Jun–03Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1.pdf
- Horowitz, Barry (PI). 2012. Security Engineering Project. A013–Final Technical Report SERC-2012-TR-028-2, 23-25. October 24. www.dtic.mil/dtic/tr/fulltext/u2/a582703.pdf

AUGUST 2015
VOLUME 18 / ISSUE 2

- Horowitz, Barry (PI). 2015. System Aware Cyber Security for an Autonomous Surveillance System On Board an Unmanned Arial Vehicle. Systems Engineering Research Center. Technical Report SERC-2015-TR-036-4 Part 1a, 73-105. January 31. http://serc2cdn1.mannadesignworks.netdna-cdn.com/wp-content/uploads/2014/11/SERC-RT-115-Security-Engineering-Pilot-Final-Report-SERC-2013-TR-036-4-Parts-1a-1b-3-4-20150131.pdf
- Huddleston, Tom, Jr. 2014. The Sony hack should make cyber security a hot boardroom topic. *Fortune Magazine*, December 23. http://fortune.com/2014/12/23/sony-hack-security-boardroom/
- INCOSE. 2014. *A World in Motion – Systems Engineering Vision 2025*. International Council on Systems Engineering. June. www.incose.org/ProductsPubs/products/sevision2025.aspx
- INCOSE. 2015. Cyber Security Considerations in Systems Engineering. *Systems Engineering Handbook 4th Edition*, Section 3.6.4. John Wiley and Sons.
- Kott, Alexander, Ananthram Swami, Patrick McDaniel. 2014. Security Outlook: Six Cyber Game Changers for the Next 15 Years. *Computer*, IEEE Computer Society, 47(12):104-106, December.
- Maslow, A.H. 1943. A Theory of Human Motivation. *Psychological Review*, 50, 370-396. http://psychclassics.yorku.ca/Maslow/motivation.htm

INCOSE®

INCOSE Certification

See why the top companies are seeking out INCOSE Certified Systems Engineering Professionals.

Are you ready to advance your career in systems engineering? Then look into INCOSE certification and set yourself apart. We offer three levels of certification for professionals who are ready to take charge of their career success.

Apply for INCOSE Certification Today!

ASSOCIATE INCOSE ASEP SYSTEMS ENGINEERING PROFESSIONAL

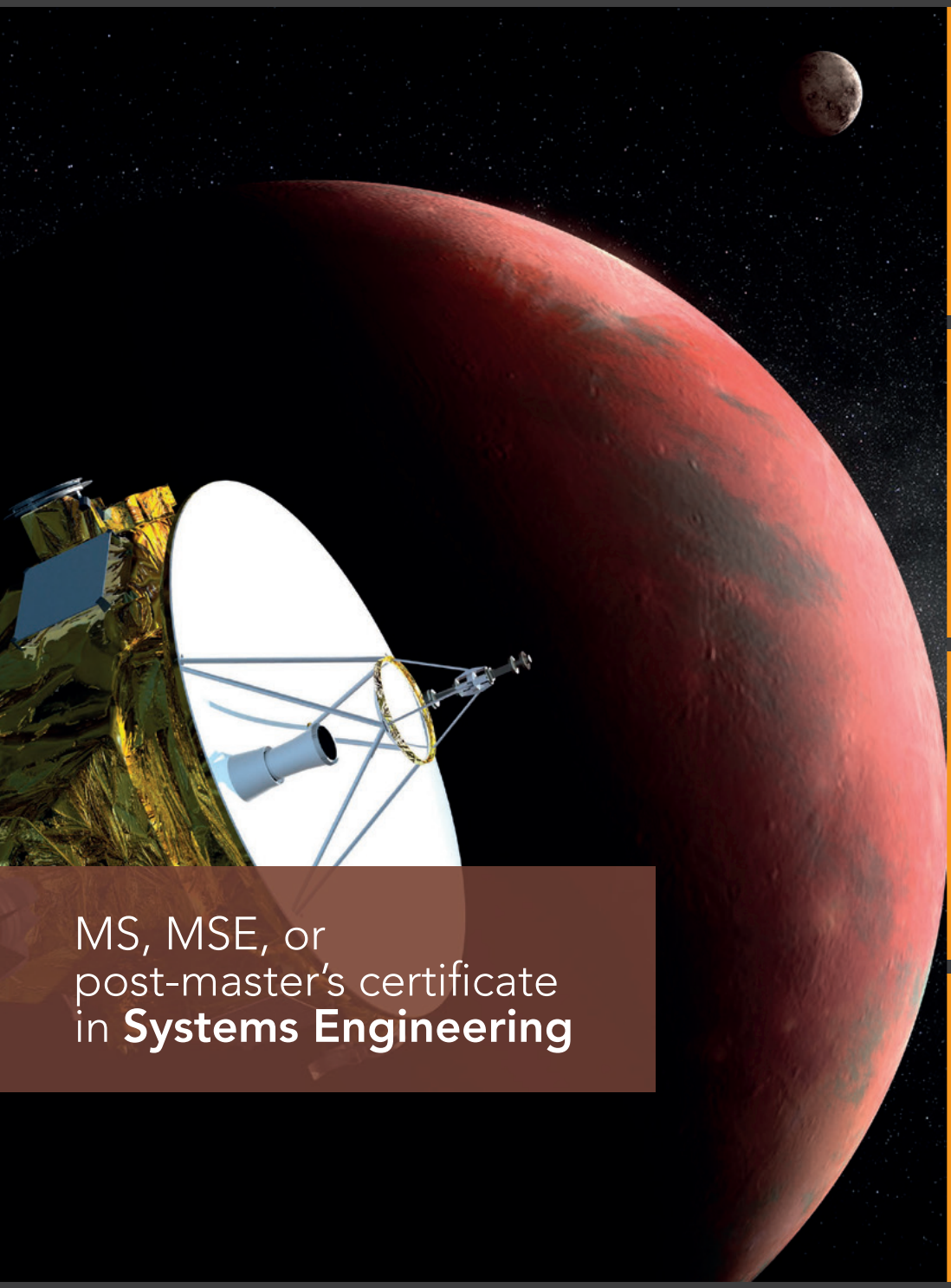
CERTIFIED INCOSE CSEP SYSTEMS ENGINEERING PROFESSIONAL

EXPERT INCOSE ESEP SYSTEMS ENGINEERING PROFESSIONAL

Visit www.incose.org or call 800.366.1164

Johns Hopkins Engineering

Online and On-Site Part-Time Graduate Programs



MS, MSE, or
post-master's certificate
in **Systems Engineering**

LARGEST SE MASTER'S PROGRAM IN THE WORLD

GOOD LEADERSHIP MEETS SOLID TECHNICAL SKILLS

1 of **3**

ABET-ACCREDITED SYSTEMS ENGINEERING PROGRAMS IN THE NATION

LEARN MORE AT EP.JHU.EDU/SE

800-548-3647

We currently are not accepting applications to the online Post-Baccalaureate Certificate or M.S. programs in Systems Engineering from students who reside in Arkansas or Kansas, or to the online Post-Master's Certificate or M.S.E. programs in Systems Engineering from students who reside in Kansas. For more information, please contact an admissions representative. Students should be aware of additional state-specific information for online programs.

For additional online education state authorization information, visit ep.jhu.edu/oesa.

JOHNS HOPKINS
WHITING SCHOOL *of* ENGINEERING

WPI | Corporate and Professional Education

SYSTEMS ENGINEERING

Join Us!

INCOSE International Symposium, Seattle, WA
July 13 to July 16, 2015
Booth #B3/B4

Graduate Programs & Initiatives

- New Ph.D. in Systems Engineering
- New Professional Development Seminar Series in Program Protection Planning and Systems Security Engineering
- New Graduate Certificate in Systems Thinking
- Systems Engineering Beyond Aerospace and Defense
- Systems Engineering Can Fuel Innovation and Business Growth for Medical Device Companies

Visit Us Online

cpe.wpi.edu/+syseng

100 Institute Road | Worcester, MA 01609 | 508.831.5517 | cpe@wpi.edu | cpe.wpi.edu