INSIGHT

This Issue's Feature:

Systems Engineering Contributions from Asia-Oceania

Model Based Conceptual Development Framework Modifications

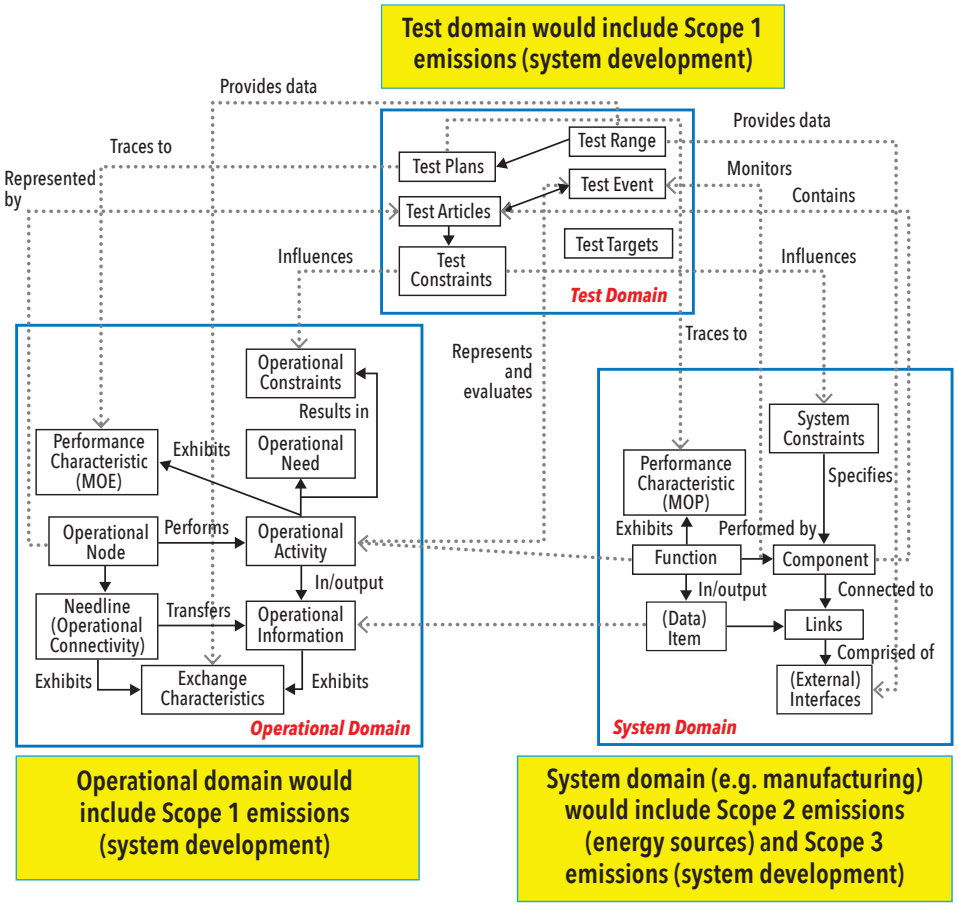


Illustration credit: from the article
Carbon Considerations for Systems Evolution
Figure 1. MBCD framework modified for carbon counting
by David Flanigan and Kevin Robinson (page 31)

APRIL 2026
VOLUME 29 / ISSUE 2

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

INCOSE

Alignment is the Advantage

Train for Skills. Align for Capabilities.

Caltech CTME designs custom learning that aligns engineers, leaders, and organizations around shared systems thinking—so teams move faster, smarter, together.

NEW CERTIFICATE PROGRAMS

- Transitioning Models to SysML v2
- MBSE Fundamentals Using SysML v2
- AI-Assisted MBSE

Caltech Center for Technology & Management Education

CTME.CALTECH.EDU

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

APRIL 2026 VOLUME 29/ ISSUE 2

Inside this issue

About This Publication

INFORMATION ABOUT INCOSE

INCOSE's membership extends to over 27,000 members and CAB associates and more than 200 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)
INSIGHT is the magazine of the International Council on Systems Engineering. It is published six times per year and

OVERVIEW

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, digital engineering, artificial intelligence, 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.

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

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

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

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 and 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 and 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 and Permissions.
If you have any questions about the permitted uses of a specific article, please contact us.

Permissions Department – UK
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 – US
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.net

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

- June 2026 issue – 1 March 2026
- August 2026 issue – 1 May 2026
- October 2026 issue – 1 July 2026
- December 2026 issue – 1 September 2026
- February 2027 issue – 1 November 2026
- April 2027 issue – 2 January 2027

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

 ISSN 2156-485X; (print) ISSN 2156-4868 (online)

ADVERTISE

Readership
INSIGHT reaches over 27,000 members and CAB associates and uncounted employees and students of more than 130 CAB organizations worldwide. Readership includes engineers, manufacturers/purchasers, scientists, research and development professionals, presidents and chief executive officers, students, and other professionals in systems engineering.

Issuance	Circulation
2026, Vol 29, 6 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 and Australia corporatesalesaustralia@wiley.com
- Europe, Middle East and 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: Marcom@incose.net

CONTACT

Questions or comments concerning:

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

Advertising—*please contact:*
Marcom@incose.net

Member Services – *please contact:* info@incose.org

ADVERTISER INDEX

April Volume 29-2

CORPORATE ADVISORY BOARD — MEMBER COMPANIES

3DSE Management Consultants
Advanced Systems Engineering, LLC
Aerospace Corporation, The
Airbus
Albers Aerospace
AM General LLC
Analog Devices, Inc.
ANSYS, Inc
Arcfield
Auburn University
Australian National University
AVIAGE SYSTEMS
Aviation Industry Corporation of China, LTD
BAE Systems
Bechtel
Becton Dickinson
Belcan Engineering Group LLC
BlueHalo Labs An AV Company
BMT Canada
Boeing Company, The
Booz Allen Hamilton Inc.
Boston Scientific Corporation
BTS Software Solutions
California State University Dominguez Hills
Caltech California Institute of Technology
Capgemini Engineering
Carnegie Mellon University Software Engineering Institute
Change Vision, Inc.
Colorado State University Systems Engineering Programs
Commercial Aircraft Corporation of China, Ltd (COMAC)
Cornell University
Cranfield University
C.S. Draper Laboratory, Inc.
CT Engineering Group
Cubic Corporation
Cummins, Inc.
Dassault Systèmes
Deloitte Consulting, LLC
Denso Create Inc
DENTSU SOKEN INC
DigiFlight , Inc.
Drexel University
Eaton
EMBRAER
FAMU-FSU College of Engineering
Federal Aviation Administration (U.S.)
Florida Institute of Technology
Ford Motor Company
GE Aerospace
General Dynamics
General Motors
George Mason University
Georgia Institute of Technology
Hitachi Energy
Honeywell Aerospace Technologies
Huawei Technologies Co. Ltd
Idaho National Laboratory
IQNOX, LLC
ISAE – Supaero
ISDEFE
IVECO Group
Jama Software
Jet Propulsion Laboratory
John Deere & Company
Johns Hopkins University
KBR, Inc.
KEIO University
L3Harris Technologies
Lawrence Livermore National Laboratory
Leidos
LEONARDO
Lockheed Martin Corporation
Los Alamos National Laboratory
Loyola Marymount University
Magna
ManTech International Corporation
Marquette University
Massachusetts Institute of Technology
MBDA (UK) Ltd
Medtronic
MetaTech Consulting Inc.
Missouri University of Science & Technology
MITRE Corporation, The
Mitsubishi Electric Corporation
Mitsubishi Heavy Industries, Ltd
Modern Technology Solutions Inc
National Aeronautics and Space Administration (NASA)
National Reconnaissance Office (NRO)
Naval Postgraduate School
Nissan Motor Co, Ltd
Northrop Grumman Corporation
ONE-SYS Srl
Pacific Northwest National Laboratory
PARAMETRIC TECHNOLOGY GMBH PTC
Pennsylvania State University
Petronas International Corporation Limited
Prime Solutions Group, Inc
Project Performance International (PPI)
Purdue University
RealmOne
Redwire Space
Rolls-Royce
RTX
Saab AB
SAFRAN
SAIC
Sandia National Laboratories
Saudi Railway Company
Shanghai Formal-Tech Information Technology Co., Ltd
Shell
Siemens
Sierra Nevada Corporation
Singapore Institute of Technology
Southern Methodist University
Space Dynamics Laboratory
SPEC Innovations
Stevens Institute of Technology
Strategic Technical Services LLC
Studio SE, Ltd.
Swedish Defence Materiel Administration (FMV)
Systems Planning and Analysis
System Strategy, Inc (SSI)
Taiwan Space Agency
Tata Consultancy Services
Terumo BCT Inc
Thales
The George Washington University
The University of Arizona
The University of Texas at Arlington
The University of Utah
Torch Technologies
TOSHIBA Corporation
Trane Technologies
Tsinghua University
UK MoD
UNCOMN
Universidade Federal De Minas Gerais
University of Alabama in Huntsville
University of Arkansas
University of California San Diego
University of Connecticut
University of Maryland
University of Maryland, Baltimore County
University of Maryland Global Campus
University of Michigan, Ann Arbor
University of New South Wales, The, Canberra
University of South Alabama
University of South-Eastern Norway (USN)
University of Texas at Austin
University of Texas at El Paso (UTEP)
US Department of Defense
Vector Informatik GmbH
Veoneer US Safety Systems, LLC
Virginia Tech
Volvo Cars Corporation
Volvo Construction Equipment
Wabtec Corporation
Warfighting Acquisition University
Wayne State University
Weber State University
Wichita State University College of Engineering
Woodward Inc
Worcester Polytechnic Institute (WPI)
Woven by Toyota, Inc.
Yulista Services , Inc.
Zuken, Inc

FROM THE EDITOR-IN-CHIEF

William Miller, insight@incose.net

We are pleased to publish the April 2026 issue of *INSIGHT* published in cooperation with John Wiley & Sons as a magazine for systems engineering practitioners. The *INSIGHT* mission is to provide informative articles for advancing the state of the practice of systems engineering. The intent is 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.

The April issue of *INSIGHT* celebrates industry contributions across Asia-Oceania advancing the practice of systems engineering. We have selected a sample of Asia-Oceania-sourced articles addressing the following themes:

- Digital Engineering
- Model-Based Test & Evaluation
- Systems Engineering for Sustainability
- Unique Abilities of the Systems Engineer
- Uncertainty in the Engineering of Systems.

In addition, we include a follow-up article from the February *2026 INSIGHT*:

- Functional Outcomes Driven Tailoring (FODT).

These themes span the five enumerated categories in the *Systems Engineering Vision 2035* (SEV2035): applications, practices, tools & environment, research, and competencies. The selected Asia-Oceania authors/co-authors and their topics are as follows:

1. Kevin Robinson and Tommie Liddy, "Acquirer Driven Digital Engineering Transformation"
2. David Flanigan and Kevin Robinson, "Exploring the Test and Evaluation Space using Model-Based Conceptual Design (MBCD) Techniques"
3. Ramakrishnan Raman, Nikhil Gupta, and Yogananda Jeppu, "Framework for Formal Verification of Machine Learning Based Complex System-of-Systems"
4. David Flanigan and Kevin Robinson, "Carbon Considerations for Systems Evolution"
5. Sunkil Yun, Shashank Alai, Yongdae Kim, Jaehun Jo, Tae Kook Kim, Dahyeon Lee, Lokesh Gorantla and Michael Baloh, "Applying a System of Systems Perspective to Hyundai-Kia's Virtual Tire Development"
6. Aswin Sukumaran Nair, "Systems Engineers – Value Added Product Owners"
7. Enayat A. Moallemi, Sondoss Elsawah, and Michael J. Ryan, "Informing the Delineation of Input Uncertainty Space in Exploratory Modelling Using a Heuristic Approach".

The follow-up FODT article is:

8. Barend Botha, "Proving the Path: Validating Functional-Outcomes-Driven Tailoring (FODT) Through Practical Applications".

We thank the contributing authors. We hope you find *INSIGHT*, the practitioners' magazine for systems engineers, informative and relevant. ■

BROADWAHE EBATORRN-CHIEF
APRIL 2026
VOLUME 29/ISSUE 2

Acquirer Driven Digital Engineering Transformation

Kevin Robinson, kevin.robinson@shoalgroup.com and **Tommie Liddy,** tommie.liddy@shoalgroup.com

■ **ABSTRACT**
Digital Engineering, like any initiative, should have a clear purpose and direction. For organisations that sit as Acquirers/Clients the Digital Transformation may not seem immediately valuable. However, for this group Digital Engineering can provide opportunities to tightly engage with their supplier network, understand trade-offs through design and upgraded lifecycles and enhanced supplier outcomes.

This paper discusses the benefits and challenges of adopting Digital Engineering in the Concept Phase, where most of the project costs are committed. It highlights that the greatest return on investment for Digital Engineering is during Concept Phase as the flow of authoritative information permeates the remainder of the lifecycle. This presents the case for acquisition agencies to both drive the application of Digital Engineering within their industry and lead by example, through Digital Engineering adoption.

■ **KEYWORDS:** digital engineering, model-based systems engineering, acquisition

INTRODUCTION

Digital Engineering, like any initiative, should have a clear purpose and direction. For organisations that sit as Acquirers/Clients the Digital Transformation may not seem immediately valuable. However, for this group implementing Digital Engineering can provide opportunities to tightly engage with their supplier network, understand trade-offs throughout design and upgrade lifecycles and support enhanced/immersive reviews.

The broad adoption of Digital Engineering is accelerating as industry sees the commercial advantages; however, this adoption is still relatively slow. Acquisition agencies, such as Government, have influenced this adoption through policy and related strategies (e.g., USA DoD 2018), but have also been slow to adopt the practice themselves. In Australia, agencies such as the Department of Defence and Transport for New South Wales have employed Model-Based Systems Engineering (MBSE) practices on a project-by-project basis, and where stakeholders have identified that a rigorous, data-centric approach can help mitigate risk. This level of adoption across multiple organisations show an initial step towards Digital Engineering but is far from an industry level Digital Transformation.

This paper makes the case that for industry to be more successfully in adopting Digital Engineering, acquisition agencies must both adopt and drive the application of Digital Engineering. It explores the literature and provides an argument for a more comprehensive adoption of Digital Engineering by acquisition agencies.

SCOPE OF DIGITAL ENGINEERING

The United States of America Department of Defence defines Digital Engineering as "an integrated digital approach that uses authoritative sources of system data and models as a continuum across disciplines" (USA DoD 2018). Digital Engineering spans the system life cycle, this includes the initial concept phase, through design cycles, production, ongoing support, and monitoring, and onto disposal or upcycle of the system. This scope remains the same for a production line systems or one-off designs. For organisations that conduct their engineering across the entire span of the system's life cycle it is possible to exercise control over the implementation of Digital Engineering. For acquisition agencies, whose focus is on the initial concept phase and ongoing support and maintenance (or operation of the system) there poses a significant challenge in implementing Digital Engineering.

Systems Engineering as part of Digital Engineering

The scope of systems engineering (described in the SEBoK (SEBoK 2021) as shown in Figure 1) spans the conception, design, development, production, and operation of physical systems. systems engineering holistically integrates the engineering disciplines being utilised to design a 'System' and interrelates with areas such as Project Management and Product Im-

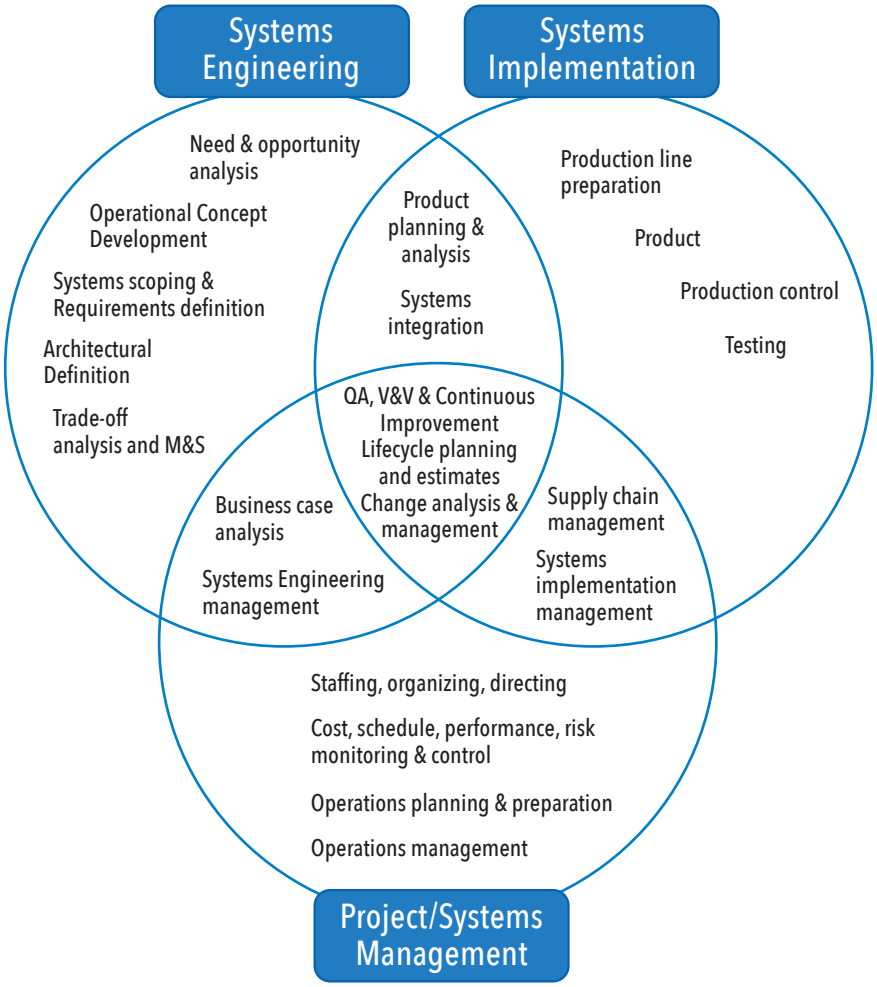


Figure 1. System boundaries of systems engineering, systems implementation, and project/systems management. Redrawn from SEBoK (SEBoK 2021)

plementation to realise this system. As such systems engineering forms the backbone of the engineering undertaken in relation to the System, throughout its life cycle. In this context, MBSE (which is a data-centric approach to systems engineering), is a key element of Digital Engineering.

In an acquisition agency context, the scope of systems engineering is no different, though part of that scope will lie within supplier agency or agencies. Within the acquisition context, systems engineering provides the comprehensive approach to analyse and combine contributions and balance trade-offs among cost, schedule, and performance while maintaining an acceptable level of risk covering the entire life cycle of a system (paraphrased from DAU 2021).

Systems engineering is making the transformation to MBSE, INCOSE Vision 2025 (Friedenthal et al 2014) states that "Model-Based systems engineering will become the "norm" for systems engineering execution, with specific focus placed on integrated modelling environments". Systems engineers adopt digital technologies that are becoming more readily available, supported by the training and education needed. This brings the discipline of systems engineering in line with other areas of engineering and design such as civil and electronic engineering that employ their model-based tools such as Computer Aided Design (CAD) and circuit board design tools, respectively.

This drive towards digital technologies, and more broadly Digital Engineering, is starting to make a real difference to complex and cross discipline projects that require a "...fully integrated engineering environment..." that provides the Systems Engineer with the "...data integration, search, and reasoning, and communication technologies to support collaboration." (Friedenthal et al 2014). The creation of digital models, both analytical and descriptive, and the integration of these models, provides more efficient and effective support to all the systems engineering activities for the design, development, manufacture, and operation of systems and as a result, mistakes are minimised, design decisions are more effective, and this increases the long-term success of the project.

Benefits of Digital Engineering

Much has been written in the literature on the benefits of Digital Engineering, including those attributed to MBSE. Unfortunately, the published benefits of Digital Engineering are based on non-empirical data, gathered through a 'lessons learned' process following project completion. The decision for employing MBSE in a project is based on the perceived benefits it offers (Henderson & Salado 2020). Once a project is complete, whether deemed a success or failure, one elicits the effectiveness of MBSE \via interview with the people involved in the projects and is therefore subjective nature. Henderson & Salado (Henderson & Salado 2020) do summarise a list of potential benefits, with better communication and information, increased traceability, reduced errors, improved consistency, better accessibility of information and others consistent across the measured, observed, perceived, and referenced categories of benefits.

So, what for the acquirer? Successful projects spend a much higher proportion of their budget on mission definition (Why do we need it? How will we use it?) and requirements engineering than less successful projects (Cook & Wilson 2018). If Digital Engineering, and specifically MBSE is to undergo adoption by the systems engineering profession, then surely the greatest return will be realised in the initial stages of the project lifecycle where MBSE can be implemented to capture systems engineering information in the form of a digital model. Undertaking the Concept Definition, through a Digital Engineering approach, should lead to better communication and information, increased traceability, reduced errors, improved consistency, and better accessibility of information. Whilst this will provide immediate benefits to the acquirer, these benefits should flow downstream to the supplier. Communicating the problem space, the needs, and the requirements of the stakeholders, through a model-based approach, will benefit the supplier before any design of the systems begins.

In 1992 the UK's National Audit Office (NAO 1992) published their findings of an examination of the Ministry of Defence's past and future initiatives on life-cycle costing. They concluded that "as much as 90% of lifecycle costs may be determined by the decisions made before production of a new weapon system begins..." This obviously places a high importance on improving the quality of the outputs from the Concept Definition phase (Figure 2) and places a high degree of responsibility for a successful project on the acquirer agency.

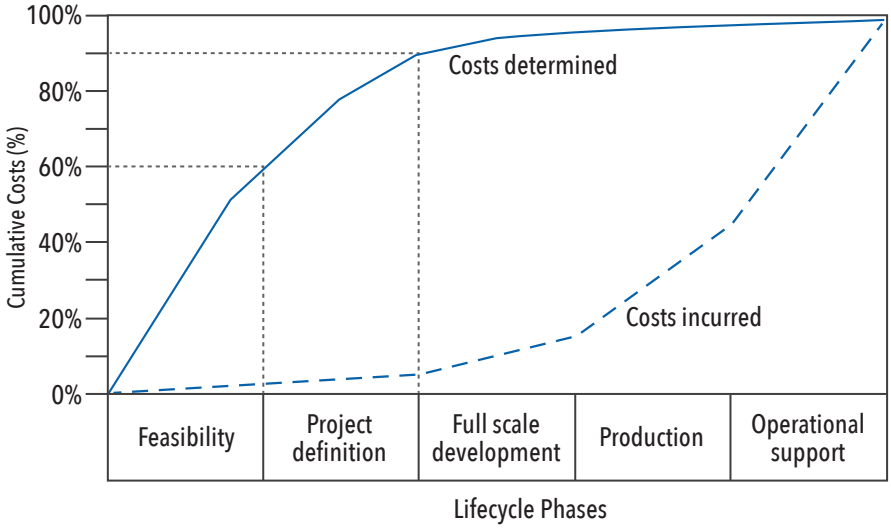


Figure 2. Lifecycle costs - commitment and expenditure. Redrawn from National Audit Office report (NAO 1992)

Decisions made by the acquirer commit up to 90% of the lifecycle costs, so reducing the risk of poor decisions by realising the benefits of Digital Engineering in acquisition agencies, will offer the greatest return of invest in improving systems engineering through a Digital Transformation.

Developing the artefacts from the Concept Phase is an acquirer responsibility. If the acquirer employs an integrated data-centric modelling approach, that provides traceability from strategic guidance to operational scenarios to user needs and system requirements, the acquirer can provide this data to the supplier. This MBSE concept definition will aid the systems designer to better understand the operational and environmental context in which the system is situated and enhance design decision making (supplier responsibility). A Digital Engineering approach allows the supplier to have a clearer and richer understanding of the complex problem, as well as having a more consistent understanding with the acquirer.

Within the INCOSE community there have been several initiatives to improve Digital Engineering, and specifically MBSE, in acquisition agencies. This includes the Model-Based Conceptual Design (MBCD) Working Group, who published an INCOSE INSIGHT Special Edition on Model-Based Conceptual Design (Robinson et al 2014) that argued the case that the aim of MBCD is "not to directly improve the quality of the individual artefacts, such as stakeholder requirements, but to enhance the design of the system concept through improving the means to derive, elicit, analyse, and record the design information as a whole. Ultimately, this aims to increase project successes, with greater overall outcomes achieved for the engineering effort invested." This, and other initiatives, have provided the knowledge base for acquisition agencies to adopt Digital Engineering, however there is a great deal of effort required to improve the maturity of Digital Engineering within acquisition agencies.

THE MATURITY OF DIGITAL ENGINEERING WITHIN ACQUIRER ORGANISATIONS

The Australian Department of Defence used MBCD on various acquisition projects. In 2008 Robinson et al (Robinson et al 2010) applied MBSE to the Ground-Based Air and Missile Defence (GBAMD) acquisition project (LAND 19/7). This project showed that an MBCD approach was "completely compatible with current mandated (document-centric) capability development processes". The approach applied MBCD in employing operational analysis to elicit user needs and derive the system requirements, producing critical acquisition documents such as the Functional and Performance Specification directly, and only from (no word processing text editing), the MBCD Model. This research demonstrated a number of benefits to the acquisition project such as enhanced access to, and communication of information, increased traceability from the project's strategic guidance to systems requirements and improved ability to identify errors and inconsistencies. The Defence Acquisition Project Lead identified that the "...approach produced a valuable project knowledge repository that will ensure continuity during future staff rotations and will allow the [document] suite to seamlessly evolve with the capability definition process" (Robinson et al. 2010).

In a 2018 paper Hallett et. al (Hallet et al 2018) found, through surveying Australian Defence personnel, that there was general agreement that while there is a marginal use of information models generated by Defence for acquisition, their development (and increased use) is not directed by the Australian Defence Organisation. Hallett also commented that those information models were only in use for acquisition, and not throughout the evaluation process or later in the system life cycle. Importantly, those interviewed agreed that the state of information model sharing across the contract boundary is non-existent.

More recently, the Systems Engineering Research Center led a collaborative research project the National Defense Industrial Association (NDIA) Systems Engineering Division, and the International Council on Systems Engineering (INCOSE) to benchmark the current state of Digital Engineering (DE) across practicing organisations (McDermott et al 2020). The survey across a broader range of industry, government and academia validated the early finding from Hallett et al (Hallett 2018) finding that most respondents scoring their level of Digital Engineering maturity as low, and specifically that "government lagged industry and academia". Decomposed survey results across survey categories of model usage and management also reflected comparable results. However, with government as the primary acquirer, the survey did note that "...government customers are mandating MBSE on programs, which is driving our digital engineering transformation," suggesting that government understands the benefits that Digital Engineering bring to an acquisition project.

ACQUIRER LED DIGITAL ENGINEERING TRANSFORMATION

For industry in general to be more successful in adopting Digital Engineering as standard practice, acquisition agencies must both adopt and drive the use of Digital Engineering within their industry. In adopting Digital Engineering approaches acquisition agencies provide two key enablers to industry: Firstly, the data-centric artefacts needed for Digital Engineering across the full life cycle of a system. Secondly, the overarching governance and control for the use and acceptance of system data developed and delivered alongside the system as well as the standards and data structures to apply.

Acquisition agencies are responsible for the early definition of the system they wish to acquire, particularly the Mission Definition. As described previously (NAO 1992), concept and system definition activities commit approximately 90% of the cost of a project prior to detailed definition and full-scale production. In discussing the return on investment for systems engineering

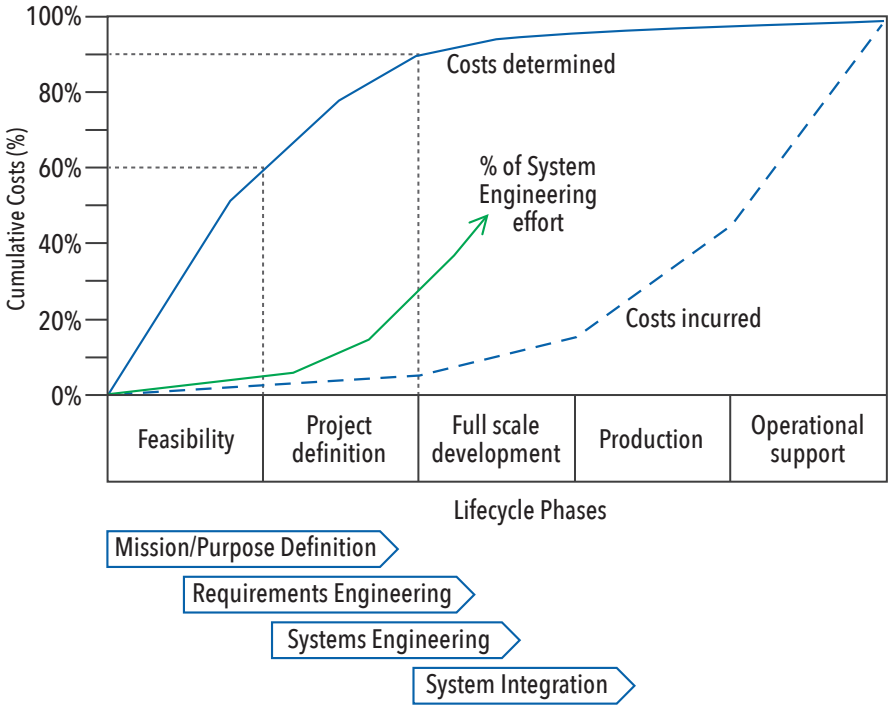


Figure 3. Systems engineering expenditure (Honour 2011) overlaid on lifecycle costs - commitment and expenditure (Redrawn from National Audit Office report (NAO 1992))

Honour (Honour 2011) showed that a 15% spend of project budget on systems engineering provides the best result on project outcomes (budget, schedule and quality) and that 40% of that spend should occur during activities such as Mission Definition, Requirements Engineering and Systems Architecting, depicted in Figure 3. During an acquisition project these activities, early in a systems life cycle, need sharing between the acquirer and supplier. In a typical project an acquirer would perform the majority of the Mission Definition, a significant amount of the Requirements Engineering and have some level of input into the System Architecture with the supplier executing the rest. To realise the benefits of Digital Engineering the acquirer agency needs to implement data centric approached to these early life cycle activities. This includes capturing system data in the form of integrated models, simulations, and other structured repositories and, most importantly, sharing those models, simulations, and repositories with supplier agencies. In doing so, supplier agencies will make more informed design decisions, and therefore increase the chance of project success. From experience working with acquisition agencies (e.g., Robinson et al 2010), the authors estimate that once the acquirer establishes a capability in MBSE, there is minimal difference in the initial level of systems engineering effort between document-centric and data-centric approaches. The only difference being that the longer-term systems engineering rework is minimised and the likelihood of projects success increased.

In addition to adopting Digital Engineering approaches within their own organisation, acquisition agencies have a role to play in driving Digital Engineering transformation within supplier organisations (potentially affecting industry wide change). Market surveys have shown that industry are late adopters of new technologies including advances in Digital Engineering technologies such as Building Information Modelling (BIM) (Walasek and Barszcz 2018). This conservative approach leads to gradual shifts across an industry. A key impact that acquisition agencies can have is in providing the leadership and governance of Digital Engineering across the system life cycle and incentivising its application. Hallett et al (Hallett et al 2018) states that supplier organisations see that "There is currently little incentive for suppliers to share models unless contractually obligated", citing a lack of appreciation of cost and management of Intellectual Property (especially with multiple suppliers involved at various stages of a systems development and sustainment) as key issues in acquisition agencies. Mandating Digital Engineering, providing a framework for controlling how data share happens between supplier organisations, defining standards and templates, appropriate data structures and interfaces, and even toolsets to use, is a role that the acquisition agency can take that will promote the adoption of Digital Engineering in supplier organisations.

Benefits to the supplier

When discussing benefits to the supplier it is worth noting that adopting a Digital Engineering approach for system design benefits the supplier organisation regardless of the approach taken by the acquisition agency. Here, benefits to the supplier organisations occurs in the context of acquisition agencies adopting Digital Engineering, providing a level of governance and direction on its use, and sharing their Digital Engineering artefacts, namely models.

Supplier organisations will see potential benefits as early as the tendering phase of a project. Cook et al (Cook et al 2014) explored an approach to model-centric information exchange across the contractual boundary for the purpose of tendering. This paper demonstrated that a tender response that utilised Digital Engineering (in this case specifically MBSE), where all system data passing between acquisition agency and supplier organisation was in the form of compatible models, allowed for the effective evaluation of the tender response. This approach, when coupled with the benefits of increased traceability, reduced errors, and improved consistency (Henderson & Salado 2020) indicates that tender responses will communicate more clearly how the response meets the tender (and importantly how it does not). This clarity should mean better alignment of stakeholder expectations at the commencement of a project and reduce the likelihood (and magnitude) of early project scope change.

Supplier organisations will also see benefits throughout the system life cycle. One of the key root causes for acquisition project failures is a lack of shared understanding between the acquirer and the supplier (Hallett et al 2018). By adopting a Digital Engineering approach that includes the acquisition agencies system data supplier organisations can generate a common understanding. The design and other engineering activities happening are directly traceable to the requirements, mission definition and other key system data. This traceability provides context to the design, opportunities to share information in forms tailored to meet specific stakeholder needs and can provide opportunities for early Verification and Validation (McDermott et al 2020). All of which aim to reduce project risk and improve the relationship between supplier and acquirer.

Challenges for the Acquirer

As discussed above, Digital Engineering is "an integrated digital approach that uses

authoritative sources of system data and models as a continuum across disciplines" (DoD Strategy 2018). This is a valid goal; however, it brings its challenges for the acquirer, and even for an acquisition agency with a mature Digital Engineering capability. The "authoritative sources of system data and models" asks for a single source of system data and models that would be shared across suppliers from different organisations. This raises three key challenges; data protection; data standards and the maturity of the data-centric tools.

Cook et al (Cook et al 2014) identifies that a key challenge for acquisition agencies is information management and controlling the flow of that information between organisations. The acquisition of a capability is a competitive environment, which "mandates careful control of information flow in both directions: the acquisition agency is required to adhere to strict probity requirements and the supplying organisations need to contain proliferation of their differentiating intellectual property that provides them their competitive edge" (Cook et al 2014). More recently, the Aerospace Industries Association white paper provided recommendations that "revise(d) regulations required to provide the government appropriate data rights" and that "intellectual property rights where early phases of mission planning and CONOPs development between industry and government still allow for protection of competing solutions" (AIA 2016). Both papers argue for data protection, whichever direction the data flows, in a digitally enabled authoritative source of system data and models.

Realising the seamless flow of authoritative sources of system data requires standards for that system data and models. As highlighted by Williams, Nallon, and Mendo, (Williams et al 2020) there are many different data interoperability standards from many different standard bodies and consortia that need to evolve and aligning across the industry. Williams, Nallon and Mendo predict that there is a four-year time horizon for that evolution and alignment, but even that seems optimistic to the authors.

Complementing the challenges with data standards is the maturity of the tools. Digital Engineering requires a diversity of tools across the various lifecycle phases and disciplines that can share the data and deliver an authoritative sources of system data and models. For the acquirer, having the right tool, to deliver the Digital Engineering environment is a challenge. There are a significant variety in MBSE tools, with different approaches to the modelling and the environment to deliver it. Despite the age of the Aerospace Industries Association white paper their recommendation to provide "...a government-industry collaborative, secure MBSE framework to support diverse tool sets and controlled data exchange to develop stable, clear, affordable, non-conflicting program requirements" (AIA 2016) is still true today.

SUMMARY

Digital Engineering delivers the opportunity for tremendous benefits, especially when applied early in the life cycle of the engineered system. Adopting Digital Engineering in the Concept Phase enhances that phase and provides benefits to the rest of the life cycle as the flow of authoritative sources of system data from the Concept Phase permeates through the latter phases. The greatest return on investment for Digital Engineering is in the hands of acquisition agencies, as most project costs are committed under the responsibility of the acquirer. Both the acquirer and the supplier have a personal stake ensuring that those committed costs have been robustly determined before any solution design decisions by the supplier.

Acquisition agencies have two roles to play in driving Digital Transformation, leadership, and adoption. Digital transformation influence by the leadership and governance of acquisition agencies must occur, and they must lead by example, through the adoption of Digital Engineering. ■

REFERENCES

- AIA (Aerospace Industries Association) 2016. "Lifecycle Benefits of Collaborative MBSE Use for Early Requirements Development." White Paper, AIA.
- Blackburn M. et al 2019. "Transforming Systems Engineering through Model-Centric Engineering, A013 Final Technical Report SERC-2019-TR-005. Hoboken, US-NJ: Systems Engineering Research Center (SERC).
- Cook S. C., Q. Do, K. Robinson, M. Lay, and M. Neidbala. 2014. "Progress on using MBSE Models as Key Information Artifacts in Project Tendering." Paper presented at the Systems Engineering Test and Evaluation Conference, Adelaide, AU, 28-30 April.
- Cook S., and S. Wilson. 2018. "The Case for Investment in Systems Engineering in the Initial Stages of Projects and Programs. Paper presented at the Systems Engineering Test and Evaluation Conference, Sydney, AU, 30 April – 2 May. https://www.shoalgroup.com/wp-content/uploads/2018/05/Cook-Wilson-2018-Case-for-early-stage-investment-in-SE-SETE-2018.pdf
- DAU (Defense Acquisition University). 2021. *Defense Acquisition Guidebook*. Fort Belvoir, US-VA: DAU.
- Hallet J., B. Hocking, and M. Vella. 2018. "Modelling Across the Contractual Boundary." Paper presented at the Systems Engineering Test and Evaluation Conference, Sydney, AU, 30 April – 2 May. https://www.shoalgroup.com/wp-content/uploads/2018/05/Hallett-Hocking-Vella-2018-Modelling-across-the-contractual-boundary-SETE-2018.pdf
- Henderson K., and A. Salado. 2020. "Value and Benefits of Model-Based Systems Engineering (MBSE): Evidence from the Literature." *Systems Engineering* 24 (1): 51–66.
- Honour, E. 2011. "Sizing Systems Engineering Activities to Optimize Return on Investment." Paper presented at the 21st Annual International Symposium of INCOSE, Denver, US-CO, 20-23 June.
- INCOSE (International Council on Systems Engineering). 2014. *A World in Motion Systems Engineering Vision 2025*. San Diego, US-CA: INCOSE.
- INCOSE Requirements Working Group. 2018. "Integrated Data as a Foundation of Systems Engineering." White Paper, INCOSE Requirements Working Group. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:rwg_data_as_a_foundation_of_se_draft_5_050218_review_copy.pdf
- McDermott T. A., N. Hutchison, M. Clifford, E. Van Aken, A. Salado, and K. Henderson. 2020. Benchmarking the Benefits and Current Maturity of Model-Based Systems Engineering across the Enterprise. Results of the MBSE Maturity Survey Part 1: Executive Summary. Technical Report SERC-2020-SR-00. Hoboken, US-NJ: Systems Engineering Research Center (SERC).
- McDermott et al. 2020. *Application of Digital Engineering Measures*. WRT-1040, Systems Engineering Research Center. 12th Annual SERC Sponsor Research Review, virtual event, November 18.
- NAO (National Audit Office (UK)). 1992. "Ministry of Defence: Planning for Lifecycle Costs." White Paper, House of Commons.

- Robinson K., D. Tramoundanis, D. Harvey, M. Jones, and S. Wilson. 2010. "Demonstrating Model-Based Systems Engineering for Specifying Complex Capability." Paper presented at the Systems Engineering Test and Evaluation Conference, Adelaide, AU, 9-11 March.
- Robinson K, M. Waite, and Q. Do. 2014. Introduction to the Model-Based Conceptual Design Special Issue. *INSIGHT* 17 (4).
- SEBoK, www.sebokwiki.org, accessed 11 Dec 2021
- US DoD (United States Department of Defense). 2018. *Digital Engineering Strategy*. White Paper, Office of the Deputy Assistant Secretary of Defense for Systems Engineering, Department of Defense. Washington, US-DC: US DoD.
- Walasek D., and A. Barszcz. 2017. Analysis of the Adoption Rate of Building Information Modeling [BIM] and its Return on Investment [ROI]. *Procedia Engineering* 172 (2017): 1227-1234. https://doi.org/10.1016/j.proeng.2017.02.144
- Williams M., J. Nallon, and W.C. Mendo. 2021. "Where is your Roadmap for implementing MBSE Data Standards?" Paper presented by the INCOSE Tools Integration & Model Lifecycle Management Working Group at the 2021INCOSE International Workshop, Virtual, 29-31 January.

ABOUT THE AUTHORS

[Editor: Author biographies were current when the paper was initially published in 2022.]

Kevin Robinson is the Head of Innovation and Chief Engineer at Shoal Group, with a distinguished career in the field of Guided Weapons in both the UK's Ministry of Defence and Australia's Department of Defence. He has made significant contributions to the development of advanced guided weapons through modelling and analysis, research, and leadership of large cross discipline teams. Throughout his career, Kevin has taken a leadership role in advancing the field of Model-Based systems engineering (MBSE) via his publications and contributions to the systems engineering community. He initiated and chaired Australia's first annual MBSE Symposium, formed and chaired INCOSE's Model-Based Conceptual Design Working Group, delivered a keynote address to INCOSE's international symposium in 2016, and has made contributions to INCOSE's Systems Engineering Handbook and related standards.

Tommie Liddy is a Senior Systems Engineer who specialises in Model-Based Systems Engineering (MBSE) techniques on a wide range of large and complex capability design projects. He is currently employed by Shoal Engineering as the systems engineering Practice Lead and is also a member of the Engineering Leadership Team. Tommie has been the technical and project lead for defence and transport capability design projects, run teams of engineering specialists conducting detailed technical studies and worked on Systems of Systems (SoS) problems to understanding program-level capability. He holds bachelor's degrees in Mechatronic Engineering as well as Mathematical and Computer Science, is an INCOSE CSEP and an Adjunct Lecturer at The University of Adelaide.

Exploring the Test and Evaluation Space using Model Based Conceptual Design (MBCD) Techniques

David Flanigan, david.flanigan@jhuapl.edu; and **Kevin Robinson,** kevin.robinson@shoalgroup.com

ABSTRACT
During the initial concept development phase, systems engineers focus on defining the problem space and system functions to explore candidate concepts that may address the systems engineers' problems. Model-based conceptual design (MBCD) techniques may be used to assist the customer and other stakeholders develop a greater understanding of the system concept, as well as identifying areas in the system that are affected by changes in requirements. This approach has generally been documented for describing the system concept in the early stages in the lifecycle, without significant focus on the test and evaluation (T&E) space that would be needed to evaluate these concepts, or identifying where the T&E space would be affected with a change in requirements. Our hypothesis is that decision makers would equally gain insight into the T&E considerations as well as system space considerations using MBCD techniques. An approach is offered to extend the previously published MBCD methodology to better consider the T&E space.

APPROACH / OUTLINE

Developing a system concept requires defining the problem space and required capabilities, functionality, and interfaces of the system concept space. A Model-based conceptual design (MBCD) technique can describe the linkage of these problems and potential solution space in order to visualize the impact of changes from the problem to the solution space, and vice versa. The MBCD approach is conveyed via a structured entity-relationship *descriptive* model instead of a traditional static document, which may promote rapid understanding of the causality of changes and may encourage quicker decision making and become informed of the problem space.

This paper offers an additional emphasis to the existing MBCD process by extending it to integrate test and evaluation (T&E) artifacts and interfaces more thoroughly to the operational domain, system domain, and analysis domain as previously described by Robinson et al. (2010). The paper starts by describing the motivation in incorporating the T&E domain to the existing methodology, and how the test domain artifacts can be modeled and analyzed, to quantify the impacts of changes to the other domains. For clarity of reading, an illustrative example is offered to explore the modified technique and offers examples of what these metrics may look like to provide insight to decision makers.

INTRODUCTION / MOTIVATION

The MBCD technique has been introduced to aid in understanding the problem space. Like existing model-based systems engineering (MBSE) techniques employed later in the lifecycle, it helps to visualize and structure systems engineering information. It allows for a richer visual picture to structure how changes in the capabilities/requirements may result from changes in the problem definition space, ultimately influencing the system capability space, concept of operations, or interfaces needed to successfully complete the mission. This approach can be helpful in the initial conceptual phase but does not currently consider in detail the T&E phase of the project during development of a conceptual system design. By including the information that more fully describes the T&E activities of the project, additional insight into the full system design may be considered, to include the requirements / capabilities to be tested as well as the complex test ranges and equipment to verify these requirements and changes in the requirements. Decision makers may receive equal insight into the entire system concept by incorporating the T&E elements into the MBCD process, as well as considering operating and system concepts.

LITERATURE REVIEW

MBCD is implemented through a series of models to provide communication between the various system development elements (developers, stakeholders, users, etc.)

and is described by Wylie et al. (2016), and Aluwihare et al. (2014) from which this paper takes motivation to extend the current methodology. Cook et al. (2015) uses the MBCD approach to assess the technical risk of concepts using modeling and the understanding of interdependencies between the different models, which can inspire the use of modeling to conduct analysis on the various concepts. Do et al. (2014) have used MBCD to explore the interactions that are needed when exchanging information and insights while executing contracts between the acquirer and supplier. Tetlow et al. (2013) utilize the MBCD approach to further explore the requirements and to assess the mission success of the conceptual system using a model-based approach. Do and Tetlow's descriptions detail the linkage between the user needs and system modeling, to develop a credible and valid system model for further analysis of the needs. Other uses of models to inform system simulations have been produced (Yaroker et al., 2013) that utilize a similar methodology as the MBCD approach.

It can be concluded from the literature review that MBCD has a wide range of applications and user communities, which provides motivation to incorporate the T&E community in this methodology.

DESCRIPTION OF THE APPROACH

The modified MBCD approach is described in five separate segments. The first defines how MBCD is used for system concept development and discusses the relevant artifacts, actors, and information. The second describes the proposed T&E extension to the MBCD technique. The third segment describes the linkage between the test domain and the other MBCD domains (notably the operational, system, and analysis). The fourth segment offers additional considerations to evaluate the entire system model. The last segment offers an approach to evaluate the new linkages and to visualize the insight gained when one domain causes changes to the other domains.

First Segment: MBCD Usage

MBCD is used to structure and link information about the understanding of a problem to possible solutions. Wylie et al. (2016) describe the usage of MBCD using *descriptive* models to describe the problem space, what the system is comprised of, and how the system interfaces are described. In their approach, they provide a logical design-based process to define the traceability, and therefore design rationale, between strategic guidance, operational activities, operational needs, functions, functional requirements, refined requirements, and software components. Through use of the descriptive models, the software developers are then able to develop their model of the system, and how it traces to the previously described artifacts. This traceability visualization can then aid the software developers and decision makers to appreciate where changes in the modified artifacts could affect the current software development plans. This level of insight can assist the decision makers to address the right problem and assist the developers to focus on the right solution set. Figure 1 provides an example of this traceability between domains through an abstraction of the schema employed to structure the model.

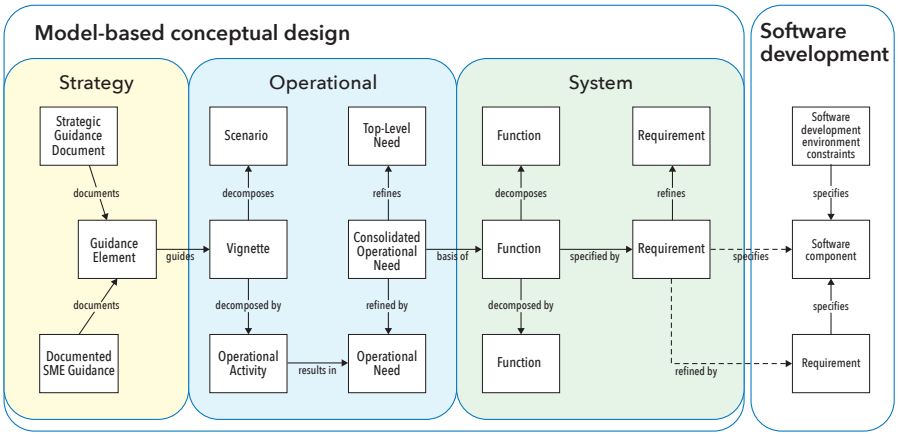


Figure 1. Traceability from conceptual design to software development (Wylie et al., 2016)

Second Segment: T&E Extension

An additional domain is proposed for inclusion into the MBCD methodology to address the T&E domain. This includes information elements that would describe the activities needed to test the requirements and functions, trace the tests to the requirements, and include the system components that would need to be tested. Proposed elements of the test domain would include test plans, test ranges, test events, test articles, test targets, and test constraints. Based on the authors' experiences across the conceptual design and T&E domains, a high-level example of the schema of this test domain is provided in Figure 2.

Third Segment: Test-Domain Linkage to Existing MBCD Domains

The newly formed test domain model may be integrated into the MBCD model through the integration of the schemas. Robinson et al. (2010) define a model-based systems engineering approach to describe a complex capability to include the enterprise context, operational domain, system domain, and the analysis domain. The strategic domain (enterprise context) focuses on the guidance. The operational domain focuses on the mission tasks, operational environment, and service requirements. The system domain focuses on the functions needed to address the mission, as well as the specific components that perform the functions. The analysis domain supports the studies and analysis to analyze the operational and system domain. Figure 3 (next page) augments the existing schema of operational domain and system domain with the test domain, thus providing the framework for developing the enhanced descriptive model, including the T&E activities.

As more domains are included with the model, the abstracted schema represented in Figure 3 increases in complexity and becomes less readable. For clarity of reading, the interfaces and directionality from Figure 3 have been converted into an interdependency matrix, shown in Table 1.

The table is intended to be read from left to right, from the source node (row) to the target node (column). A number of "1" indicates there is an interface from that specific source to target node. Note that the

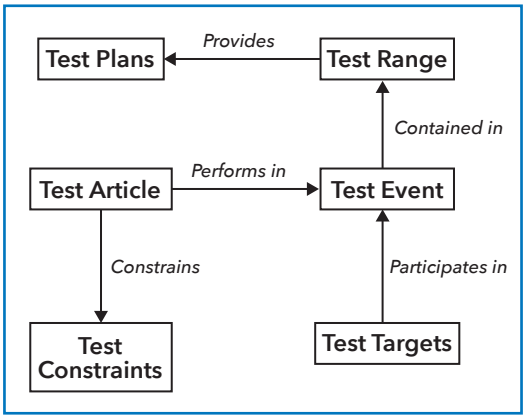


Figure 2. Test domain MBCD model

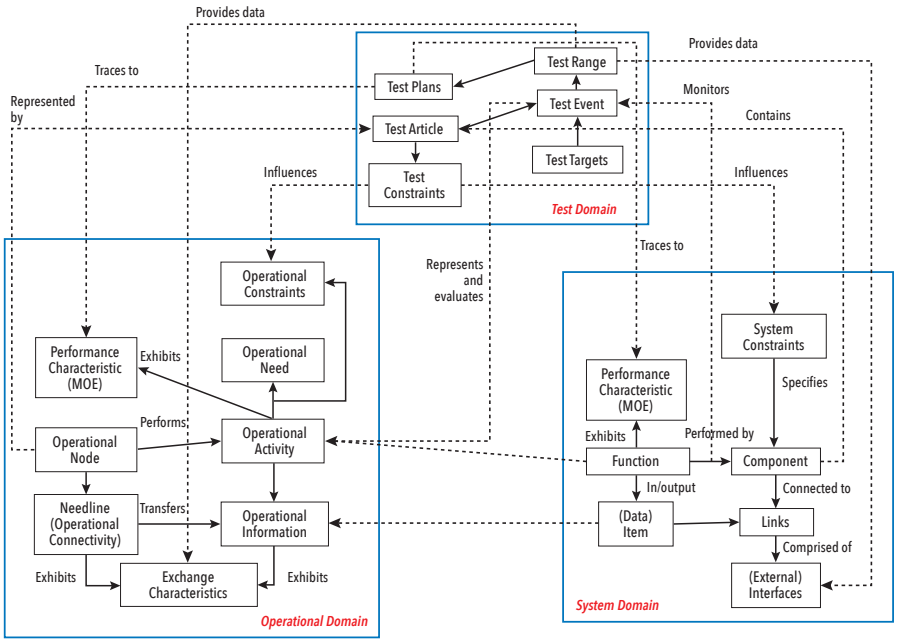


Figure 3. Modified MBCD model abstracted schema

directionality should be reflected in this matrix, as not all interfaces have a 2-way direction, although can if desired for usability and readability. This table shows the three domains (operational, system, and test), which each have three possible domain interactions (one internal and two external).

Fourth Segment: Evaluation of the New Linkage

The fourth segment evaluates how the rest of the overall descriptive model is affected when one domain element is changed. Changes may be viewed from different perspectives: the decision makers will view changes to the model as a change in capability or fielding date, which may affect their investment strategy. Developers may view changes to the model as changing their delivery dates or scheduling of efforts. Analysts may view changes to the model as updating their assessment of the system capability, which then may affect the decision maker's insight of the system's capability. Testers may

Table 1. MBCD Model interdependency

		Operational Domain								System Domain							Test Domain					
		Operational Contraints	Operational Need	Operational Activity	Operational Information	Exchange Characteristics	Performance Characteristic (MOE)	Operational Node	Needline(Operational Connectivity)	System Contraints	Component	Links	(External) Interfaces	Performance Characteristic (MOP)	Function	(Data) Item	Test Plans	Test Article	Test Constraints	Test Range	Test Event	Test Targets
Operational Domain	Operational Contraints																					
	Operational Need														1							
	Operational Activity	1	1				1															
	Operational Information					1																
	Exchange Characteristics																					
	Performance Characteristic (MOE)																					
	Operational Node			1					1									1				
	Needline(Operational Connectivity)				1	1																
System Domain	System Contraints										1											
	Component											1						1				
	Links												1									
	(External) Interfaces																					
	Performance Characteristic (MOP)																					
	Function			1							1			1		1					1	
	(Data) Item				1							1										
Test Domain	Test Plans						1							1								
	Test Article																		1		1	
	Test Constraints	1								1												
	Test Range												1				1					
	Test Event				1															1		
	Test Targets																				1	

Table 2. MBCD model interdependency primary and secondary impacts

		Operational Domain								System Domain					Test Domain					
		Operational Contraints	Operational Need	Operational Activity	Operational Information	Exchange Characteristics	Performance Characteristic (MOE)	Operational Node	Needline(Operational Connectivity)	System Contraints	Component	Links	(External) Interfaces	Performance Characteristic (MOP)	Function	(Data) Item	Test Plans	Test Article	Test Constraints	Test Range	Test Event	Test Targets
Operational Domain	Operational Contraints																				
	Operational Need														1						
	Operational Activity	1	1				1														
	Operational Information					1															
	Exchange Characteristics																				
	Performance Characteristic (MOE)																				
	Operational Node			1					1									1			
	Needline(Operational Connectivity)				1	1															
System Domain	System Contraints										1										
	Component											1						1			
	Links												1								
	(External) Interfaces																				
	Performance Characteristic (MOP)																				
	Function			1							1			1	1					1	
	(Data) Item				1							1									
Test Domain	Test Plans						1							1							
	Test Article																		1		1
	Test Constraints	1								1											
	Test Range												1				1				
	Test Event				1															1	
	Test Targets																				1

		Operational Domain								System Domain					Test Domain					
		Operational Contraints	Operational Need	Operational Activity	Operational Information	Exchange Characteristics	Performance Characteristic (MOE)	Operational Node	Needline(Operational Connectivity)	System Contraints	Component	Links	(External) Interfaces	Performance Characteristic (MOP)	Function	(Data) Item	Test Plans	Test Article	Test Constraints	Test Range	Test Event	Test Targets
Operational Domain	Operational Contraints																				
	Operational Need														1						
	Operational Activity	1	1				1														
	Operational Information					1															
	Exchange Characteristics																				
	Performance Characteristic (MOE)																				
	Operational Node			1					1									1			
	Needline(Operational Connectivity)				1	1															
System Domain	System Contraints										1										
	Component											1						1			
	Links												1								
	(External) Interfaces																				
	Performance Characteristic (MOP)																				
	Function			1							1			1	1					1	
	(Data) Item				1							1									
Test Domain	Test Plans						1							1							
	Test Article																		1		1
	Test Constraints	1								1											
	Test Range												1				1				
	Test Event				1															1	
	Test Targets																				1

view changes to the model that may affect their existing testing capabilities or future testing capabilities that need to be developed.

Changes to one domain may affect other domains described in the model. For example, if there are changes to the operational domain (such as requirements), this may affect the system development efforts if there are new capabilities needed, or if the system design approach needs to be modified. As a result of this operational requirements change, testing approaches may need to be changed, which may affect the scheduling of the test facility or modification of the test articles or targets.

Fifth Segment: Impacts of the Changes

Once the MBCD model has been modified, evaluation of the model should be conducted to ensure that the MBCD concepts are still valid, and the decision makers and other actors gain insight into the problem when changes to one domain are introduced. Several means are offered to evaluate how the linkages may be conducted. One method could be to leverage the network science community, to describe the number of nodes that are affected by a node that will change (for example, changing requirements and understanding what impacts this change would have in the other domains). Other network science metrics are size, average degree, average path length, connectedness, node centrality, and node influence.

As the triad of systems, operational, and testing domains are affected by changes in one of the domains, we may observe a change in both the primary and secondary influences that a domain has on the rest of the system. Using Table 1, changes to the test domain could affect the system and operational domain as the primary influence. However, each of these domains has their own potential influences, creating a secondary influence (system domain may affect the test and operational domain, and operational domain may affect the system and test domain). There exists a potential for the primary change in one domain to indirectly affect itself through the primary domain influenced. It may be postulate that a lesser impact will be seen through the secondary domain effect but leave this for future work to quantify the primary and secondary impacts. Table 2 provides an example of such a primary (left side) and secondary (right side) of impacts based on one modification (function from system domain). An example of a primary impact is by affecting the "function" within the system domain, will affect five elements in the system (highlighted in orange). An example of the secondary effect is that each of these elements will have their own influence on the operational, system, and test domain, moving up and down the columns (shown on the right side in blue), affecting seven elements within the system.

Illustrative Example

An illustrative example is offered to evaluate if the modified MBCD technique has merit and offers additional value to the stakeholders when changes are introduced. Here an existing example that uses MBCD to evaluate fire and emergency services (Spencer and Harvey 2014) is leveraged and simplified. This example was developed

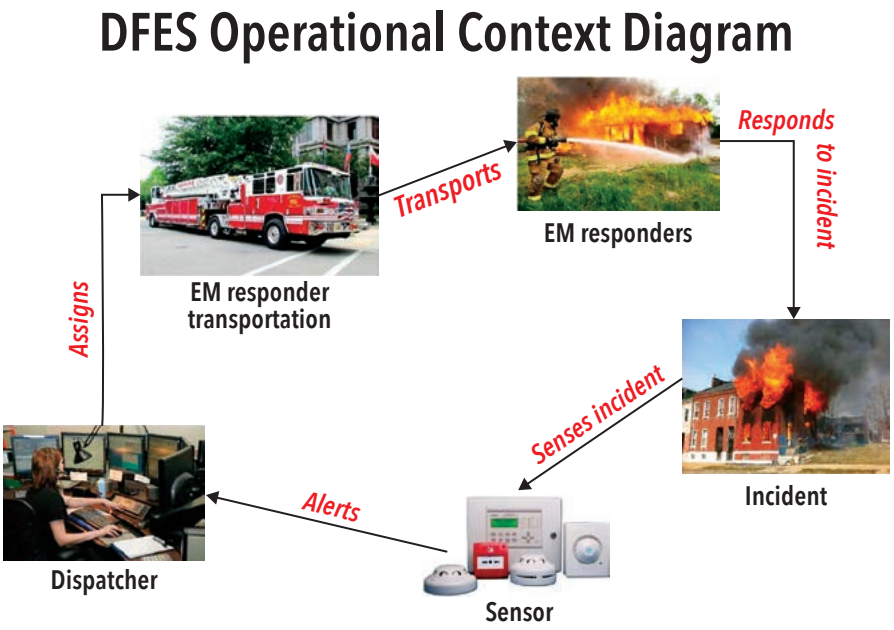


Figure 4. Modified DFES operational context diagram (OV-1)

Table 3. DFES domain elements

Functions	Systems	Actors
Sense smoke/incident	Sensor	Fire service personnel
Send alert	Telephone/radio	Rescue coordinators
Confirm incident/select action	Data Terminal	
Dispatch response units	Tanker, pumps, hoses	
Respond to incident	Transport vehicle	

Table 4. DFES test elements

Functions	Test Objective	Test Element
Sense smoke/ incident	Determine if incident is properly detected	Sensors, fire source, facility environment
Send alert	Determine if alert is sent timely	Communications (transmitter and receiver), communications environment
Confirm incident/ select action	Determine if response is correctly determined	Dispatcher, displays, dispatcher environment
Dispatch response units	Determine if dispatch is correctly executed	Response units, transportation environment
Respond to incident	Determine if response is adequately executed	Responders, fire source, facility response environment

for the Department of Fire and Emergency Services (DFES) of the Government of Western Australia. The MBCD process was followed to define the DFES mission, drivers, and capabilities, and used the capability management framework to consider during the planning, development, and execution of the capability development more thoroughly.

This example is utilized to introduce the MBCD process and the test domain. The intent of the example is to exercise the interdependency and quantification of the impact of changes when portions of the entire model are changed.

Operational Domain Description

The operational domain is defined by the DFES mission to detect, analyze, and respond to emergencies and incidents. Depicted in Figure 4 is the mission in graphical form using an OV-1. Within each of these domains, the following elements are defined in Table 3.

System Domain Description

There are numerous systems that are used in this example. These are organized by the various phases of the operation (sensing, alerting, processing, dispatching, transporting, and responding). These systems are also listed in Table 3. Note that these systems will also include the actors that will operate the systems and other aspects not included in the simplified example.

Test Domain Description

The test domain will identify several elements that would be used to test the various mission activities that are being evaluated. From our example, four capabilities would be tested, listed in Table 4 that are organized by capability test objective, and applicable test elements.

Insight and Utility of the Modified MBCD Process

The modified model can be utilized to incorporate the test domain along with the operational and system domains. While the stakeholders, development team and test team are developing their respective efforts, we would expect numerous interactions between the three teams during the capability development. Expected questions in response to a domain change should start with "how does that affect the other domains?"

The model would be developed and then verify with the three domain teams to ensure that the elements and interactions are correct. Data would be elicited through tailored interviews and workshops to determine if sufficient insight was gained by all parties during the system development.

CONCLUSIONS / NEXT STEPS

This paper has offered a modification to the existing MBCD process to incorporate the test domain into the conceptual development phase. The aim being to ensure that the testing community and capabilities are also considered during the initial development to identify long-lead capability development, or how interdependent the operational and system development teams are to affect the test capabilities.

Next steps would be to identify an example project that this approach could be applied to and gain concurrence by all three domains. A model would be developed to describe the specific domains and follow the MBCD process during the system development lifecycle. Data could be collected at relevant milestones (for example, preliminary design review, critical design review, test readiness review, etc.). If the hypothesis proves correct that insight is gained by all domain stakeholders, the project could progress to a larger and more interdependent system concept for a further proof of concept. ■

REFERENCES

- Aluwihare, Chanaka, Michael Waite, and Christopher French. 2014. "Model Based Systems Engineering: a Methodology for Collaborative Requirements Engineering." Australian Defence Engineering Conference 2014. Engineers Australia.
- Cook, S. C., et al. 2015. "Evaluation of a Model-Based Technical Risk Assessment Methodology."
- Do, Quoc, Stephen Cook, and Matthew Lay. "An Investigation of MBSE Practices Across the Contractual Boundary." *Procedia Computer Science* 28: 692-701.
- Robinson, Kevin, et al. 2010. "Demonstrating Model-Based Systems Engineering for Specifying Complex Capability." 2010 Systems Engineering Test and Evaluation Conference: SETE. Engineers Australia.
- Spencer, Daniel, and David Harvey. 2014. "A Model-Based Approach to Capability in Fire and Emergency Services." 2014 Systems Engineering Test and Evaluation Conference: SETE 2014. Engineers Australia.

> continued on page 29

Framework for Formal Verification of Machine Learning Based Complex System-of-Systems

Ramakrishnan Raman, ramakrishnan.raman@honeywell.com; **Nikhil Gupta,** nikhil.gupta4@honeywell.com; and **Yogananda Jeppu,** yogananda.jeppu@honeywell.com

■ **ABSTRACT**
A complex system is characterized by emergence of global properties which are very difficult, if not impossible, to anticipate just from complete knowledge of component behaviors. Emergence, hierarchical organization, and numerosity are some of the characteristics of complex systems. Recently, there has been an exponential increase on the adoption of various neural network-based machine learning models to govern the functionality and behavior of systems. With this increasing system complexity, achieving confidence in systems becomes even more difficult. Further, ease of interconnectivity among systems is permeating numerous system-of-systems, wherein multiple independent systems are expected to interact and collaborate to achieve unparalleled levels of functionality. Traditional verification and validation approaches are often inadequate to bring in the nuances of potential emergent behavior in a system-of-systems, which may be positive or negative. This paper describes a novel approach towards application of machine learning based classifiers and formal methods for analyzing and evaluating emergent behavior of complex system-of-systems that comprise a hybrid of constituent systems governed by conventional models and machine learning models. The proposed approach involves developing a machine learning classifier model that learns on potential negative and positive emergent behaviors, and predicts the behavior exhibited. A formal verification model is then developed to assert negative emergent behavior. The approach is illustrated through the case of a swarm of autonomous UAVs flying in a formation, and dynamically changing the shape of the formation, to support varying mission scenarios. The effectiveness and performance of the approach are quantified.

■ **KEYWORDS:** Complex System-of-Systems, Emergent Behavior, Machine Learning, Formal Verification

INTRODUCTION

A system can be considered as an integrated and interacting combination of elements and/or subsystems to accomplish a defined objective (INCOSE 2015). These elements may include hardware, software, firmware, and other support. Systems-of-systems (SoS) are systems of interest whose system elements are themselves systems (Jamshidi 2008). SoS has evinced keen interest among the systems engineering community, and there has been significant research pertaining to principles and practices on the architecture design, development, deployment, operation, and evolution of SoS (Lane 2013; Nielsen et al. 2015; INCOSE *INSIGHT* 2016; Raman and D'Souza 2018; and Raman and D'Souza 2019). Applications of SoS principles and practices span many domains, including electrical power distribution, and Internet-of-Things. SoS characteristics discussed in literature include operational/managerial independence, emergent behavior, and evolutionary development.

In a general sense, the adjective "complex" describes a system or component that by design or function or both is difficult to understand and verify. A complex system is characterized by emergence of global properties which are very difficult, if not impossible, to anticipate just from a complete knowledge of component behaviors (Aiguier et al. 2008). Emergence, hierarchical organization, and numerosity are some of the characteristics of complex systems (Ladyman et al. 2013). Specifically, for complex SoS, the "stringing" together of the constituent systems results in unique functionality and emergent behavior being exhibited at the SoS level that is very difficult to envision and predict and cannot be attributed to any of the constituent systems individually. Understanding measures of effectiveness (MOEs) (INCOSE 2015), is critical to analyze the impact of the emergent behavior at SoS level. There are different types of complexity measures dis-

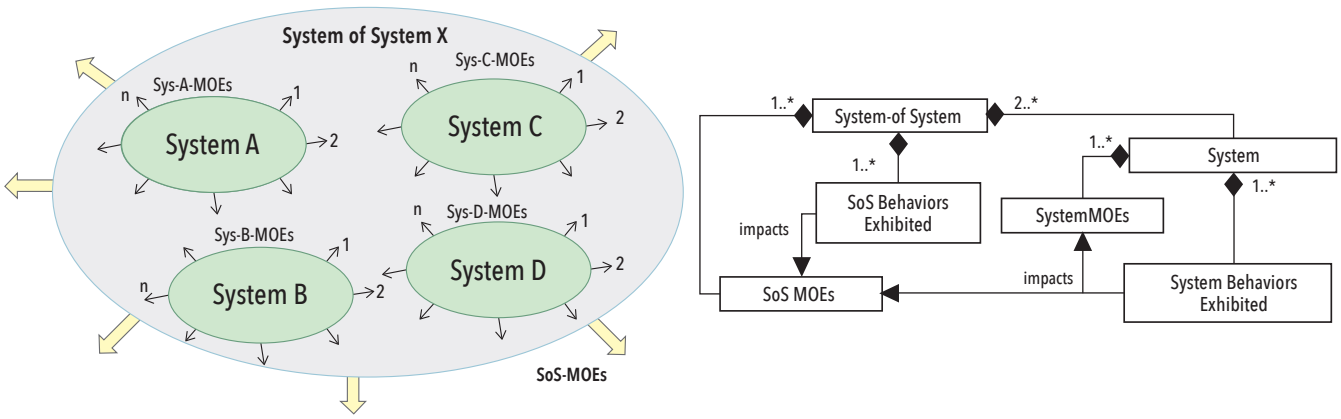


Figure 1. SoS MOEs and constituent system MOEs

cussed from different perspectives (Kinsner 2008). The perspective of complexity used in this paper is with respect to the degree of difficulty in accurately predicting the future behavior. This complexity is determined by the entity being observed, the capabilities of the observer, and the behavior that the observer is attempting to predict. This paper proposes an approach for analyzing and evaluating emergent behavior of complex SoS. In our proposed approach, the entity being observed is a complex SoS, the observer being a machine learning classifier, and the behavior being attempted to predict is the positive or negative emergent behavior of the complex SoS.

The rest of the paper is organized as follows: The next section discusses key elements pertaining to the proposed approach, namely emergent behavior, MOEs, machine learning, and formal methods. The subsequent section discusses the proposed approach and illustrates it through case of a complex SoS that comprise a hybrid of constituent systems governed by conventional and machine learning models. The case taken is a swarm of autonomous UAVs flying in a formation, and dynamically changing the shape of the formation, towards supporting different mission scenarios. Finally, benefits and limitations of the proposed approach, conclusions and future work are discussed.

EMERGENT BEHAVIOR, MOEs, MACHINE LEARNING, AND FORMAL METHODS

This section discusses some of the key elements pertaining to the proposed approach.

Emergent Behavior

Emergence refers to the ability of a system to produce a highly structured collective behavior over time, from the interaction of individual subsystems (Kinsner 2008). Common examples include a flock of birds flying in a V-formation, and ants forming societies of different classes of individual ants, wherein these patterns are not induced by a central authority. For a system, emergent behavior refers to all that arises from the set of interactions among its subsystems and components. Complex systems are expressed by the emergence of global properties which are very difficult, if not impossible, to anticipate just from a complete knowledge of component or subsystem behaviors (Giammarco 2017). Emergent behavior can be characterized as positive or negative, depending on the impact on the MOEs. The challenge for complex systems is that there is inadequate knowledge on combination of events that would result in a negative emergent behavior. The intent of our proposed approach is towards learning from emergent behaviors exhibited and asserting for occurrences of negative emergent behaviors for complex SoS.

Measures of Effectiveness – MOEs

MOEs. Measures of effectiveness are the operational measures of success that are closely related to the achievement of the objective of the system of interest, in the intended operational environment under a specified set of conditions (INCOSE 2015). It reflects the overall customer and user satisfaction, and it manifests at the boundary of the system. MOEs are independent of the specific solution (INCOSE 2005). Example of MOEs include service life of a satellite, search area coverage, and survivability. Failure of the system to meet an MOE implies that the system does not meet its purpose and objectives (Smith and Clark 2006).

SoS MOEs versus System MOEs. In the context of SoS, each constituent system of the SoS has its own MOEs. The MOEs for a constituent system can be independently measured to assess its success. MOEs of the SoS are the operational measures of success for the SoS as a whole. Figure 1 illustrates SoS MOEs versus constituent system MOEs. System A can have MOEs: SysA-MOE-1, SysA-MOE-2, and SysA-MOE-3. The MOEs of System A represent the measures of success for System A as an independent system, and the MOEs for System A can be independently measured to assess the success of System A. In addition to each constituent system having its own MOEs, MOEs are also relevant at the SoS level, that is, SoSx would also have its own MOEs. The MOEs at the SoS level

		MOEs of constituent Systems								
Relative importance of each SoS MOE										
System of System MOEs	SoS MoE Weight	SysA-MoE-1	SysA-MoE-2	SysA-MoE-3	SysB-MoE-1	SysB-MoE-2	SysC-MoE-1	SysC-MoE-2	SysD-MoE-1	SysD-MoE-2
SoS MoE1	9	9	9	7					1	
SoS MoE2	7			5	7	7	7		1	
SoS MoE3	5	7	9	5					7	
SoS MoE4	1	9	9	7				7	5	5
System A is a key player in the SoS	Raw score	125	135	130	49	49	49	7	56	5
	Relative Weight	21%	22%	21%	8%	8%	8%	1%	9%	1%
	Rank	3	1	2	5	5	5	8	4	9

Relative impact of each System MOE on each SoS MOE

Figure 2. SoS – System MOE relationship matrix

represent the measures of success for the SoS as a whole. Figure 1 further illustrates the impacts on MOEs at system level and at SoS level. The MOEs of the system are impacted by the behaviors exhibited by the system. Similarly, the MOEs of the SoS are impacted by the behaviors exhibited at SoS level. Further, the behaviors exhibited at constituent system level also impacts the SoS MOEs.

As discussed earlier, one of the characteristics of SoS is that the stringing together of the constituent systems results in unique behavior and functionality that gets exhibited at the boundary of the SoS, that is, the behavior may not be attributed to any of the constituent systems functioning independently. With this being the case, the relationships between the MOEs of the SoS vis-à-vis the MOEs of the constituent systems might turn out to be complex and dynamic. There are different means to analyze the MOE relationships between the constituent systems and SoS. SoS-System MOE relationship matrix (Raman and D'Souza 2017; and Raman and D'Souza 2019) is one of the means to analyze the relationships, as indicated in Figure 2. The impact of different system MOEs on the SoS MOEs could vary. There might be scenarios where a specific constituent system might be meeting all its MOEs, but the SoS MOEs might not be met. Similar scenarios will be discussed in this paper in the next section.

Machine Learning

Machine learning can be broadly defined as computational methods using experience to improve performance or to make accurate predictions (Mohri et. al. 2012). Here, experience refers to the past information available to the learner, which typically takes the form of electronic data collected and made available for analysis. Machine learning represents the field of study that allows computer programs to learn without being explicitly programmed. The often-used definition is: "A computer program is said to learn from experience E with respect to some task T and some performance P, if its performance on T, as measured by P, improves with experience E" (Mitchell, 1997). Artificial neural networks (NN), inspired by biological neural networks in brains, comprise a collection (organized in layers) of interconnected units (nodes), with each node having the capability to receive a signal, process the signal, and transmit the processed signal to other units linked to it. NN has been used on numerous learning problems, including vision, speech recognition, social networks, board games, and medical diagnosis. In a neural network, the first layer is termed the input layer since it is connected to the external input data. The last layer is termed the output layer since it provides the outputs of the total neural network. All the other intermediate layers are termed hidden layers. Each node unit processes the signal via an activation function. Each input has a weight that can be modified. Each unit computes the activation function f of the weighted sum of its inputs.

Recently, there is an explosion in the adoption of neural network-based machine learning models in various systems and are increasingly being used to control many physical systems, such as cars and drones. A good starting point to get the context of the work discussed in this paper is a comprehensive survey paper that provides a detailed look at the field with a review of over 150 odd papers (Xiang et al. 2018) that discusses the use of neural network-based machine learning techniques in safety control systems, and the formal methods/ verification used to validate the networks. The verification of NN is a hard task as it is said to be an NP complete problem. Most of the difficulties arise from the presence of activation functions and the complex structure of the neural network. Nevertheless, neural networks-based machine learning techniques have been used in some of the safety critical systems: F-15B intelligent flight control system (William-Hayes 2005) and intelligent autopilot system (Baomar and Bentley 2017). Neural networks are however susceptible to small changes in their inputs, and therefore ensuring their correct behavior under various conditions is very important. The Reluplex algorithm, which stands of ReLU with Simplex caters to the activation function Relu using the simplex algorithm, is evaluated on a set of 45 real world NN problems (Katz 2017). In this paper, we have used MathWorks® MATLAB R2020b Statistics and Machine Learning Toolbox™.

Formal Methods

Formal methods are mathematics-based techniques for the specification, development, and verification of digital systems (RTCA 2011). The mathematical basis of formal methods consists of formal logic, discrete mathematics, and computer-readable languages. The use of formal methods is motivated by the expectation that, as in other engineering disciplines, performing appropriate mathematical analyses can contribute to establishing the correctness and robustness of a design. Formal methods can be used to model complex systems as mathematical entities. The complex system behavior is broken down into smaller units and each one of these is defined as mathematical equations. Defining systems formally enables system validation (mathematically correct behavior – mostly safety criteria) using means other than testing – like a proof of correctness. The mathematical techniques are used to prove the correctness of the assumptions and theory using property proving. There are many tools that can be used for formal methods in the systems development V-Model (Nanda et al. 2018). Another branch of formal verification is called model checking, which involves a model of the system and a way to define the property of the system. The model checking tool then explores the possible states the model can be in and checks for violations of the property. A violation of the property yields a counter example that is used for debugging the model. It may give concrete evidence of the correctness of the property and this proves that the property can never be violated for any combination of states and within the overriding assumptions. There is a possibility of the formal methods tool to provide an outcome stating that the property cannot be proved due to the limitation of the tool. This usually happens due to the large state space that is created and makes the proving impossible given the memory limitation of the computing platform. In such cases one must slice the model or limit the input space to reduce the bloat-up of the state space. In this work, we have used Simulink Design Verifier, a tool from MathWorks (SLDV 2020) that uses formal methods to generate test cases, find design errors and to prove the correctness of assertions or properties defined as Simulink blocks or MATLAB code. We have successfully demonstrated the use of SLDV in our earlier work (Raman and Jeppu 2019). In this paper, we also look at another tool called CBMC (2020). This tool is a bounded model checker that looks at properties in a small defined region and bound and argues on its correctness. CBMC works on the C code, and it has additional features like MC/DC testing, array checks, branch coverage etc. that can be used on the generated code. We explore the use of CBMC in the current problem statement to look at the NN correctness and the SoS behavior.

PROPOSED APPROACH

This section discusses the proposed framework towards application of machine learning based classifiers and formal methods for analyzing and evaluating emergent behavior of complex system-of-systems. Figure 3 provides an overview of the proposed approach. The complex SoS has a set of defined MOEs. The SoS comprises independent constituent systems, with each having their own corresponding system MOEs. The proposed approach involves

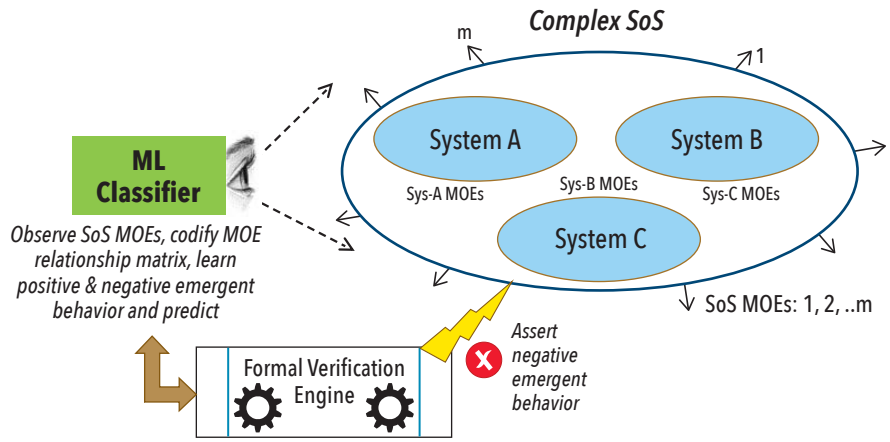


Figure 3. Overview of proposed approach

building a machine learning (ML) classifier that observes the various MOEs at SoS level and constituent system level, leverages the MOE relationship matrix (Figure 2), and learns the emergent behavior. The formal verification engine is used to assert the occurrence of negative emergent behavior. Figure 4 provides details of the proposed approach. To illustrate the proposed approach, the generic case of a swarm of UAVs is taken from publicly available literature (Tahir et al. 2019; X. Dong et al, 2019; and Hassanien and Emary 2016). This is specifically done to avoid any restrictions that may come in sharing the work with the community. The example, though generic, is sufficiently complex and is of significant relevance to the aerospace community and can further be scaled up to serve more complex use cases.

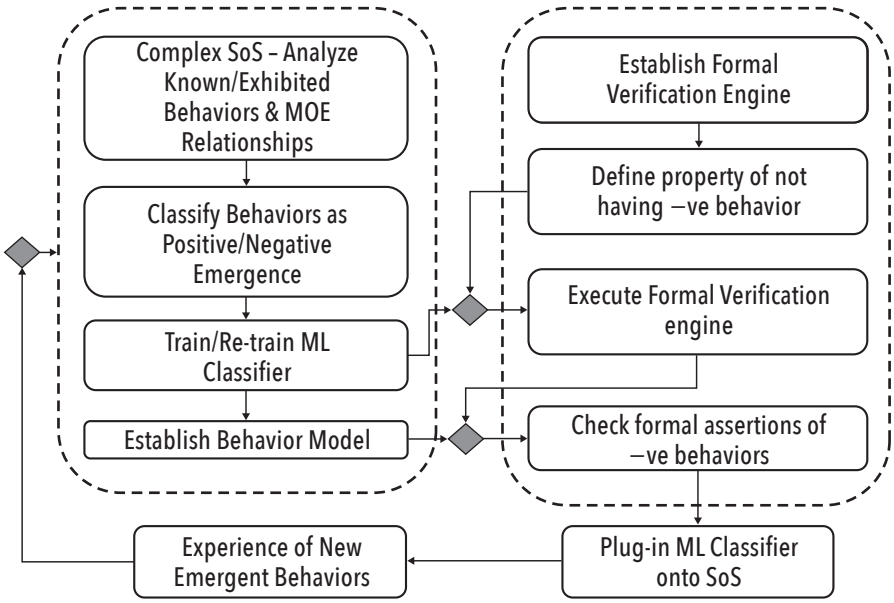


Figure 4. Proposed framework

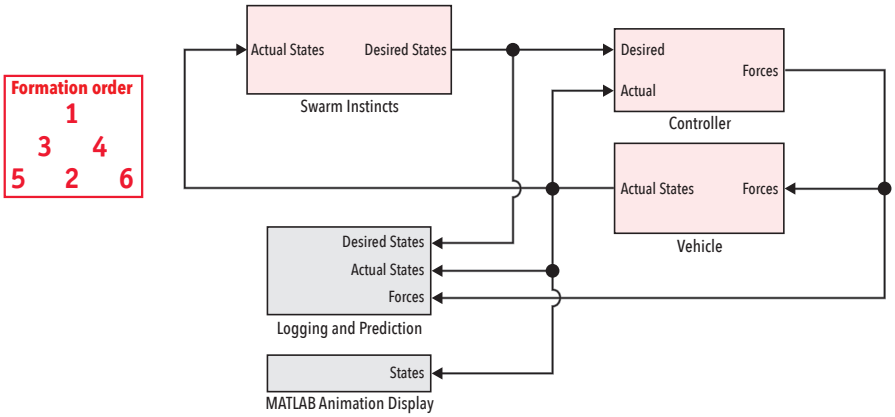


Figure 5. SoS model – with 6 constituent systems (autonomous UAVs)

Swarm Formation Flying

As discussed in the earlier section, emergence refers to the ability of a system to produce a highly structured collective behavior over time, from the interaction of individual subsystems. A typical example pertains to a flock of birds flying in a specific formation, which is supposed to give many benefits including reduced energy requirements and safety. Similar approaches have been adopted for swarm of UAVs too, wherein specific formation shapes are expected to provide specific benefits – including fuel savings and redundancy in mission coverage. The scenario being simulated is a system-of-systems comprising six autonomous UAVs as the constituent systems. The autonomous UAVs collaborate with each other during the required situations and fly in a formation to leverage the desired benefits as and when required. Various scenarios pertain to the different formation shapes as required for the mission. For an individual constituent system UAV, the MOEs would pertain to parameters such as whether the required speed constraints are adhered to or not, and whether the required space constraints with adjacent autonomous UAVs are adhered to or not. For the SoS comprising the various autonomous UAVs, MOEs would pertain to aspects such as the time duration to transition from one formation shape to another, the safety and separation constraints being adhered to by all the autonomous UAVs, and the formation shape being maintained without distortion. For experiments, simulations were done to study various scenarios that would be encountered during the autonomous UAV formation flying. MathWorks® MATLAB R2020a Aerospace & Control System Toolbox was used to build the models for the same.

The high-level SoS model is illustrated in Figure 5. The formation shape and indexing for the six autonomous UAVs is indicated in the figure (in "formation order"). For the specific formation, UAV-1 is considered as the leader of the formation. Each signal line depicted in the model represents data from all the six UAVs. The states of each individual UAV include the inertial position and velocities. The states are initialized in the vehicle block with random initial positions with respect to the leader of the formation.

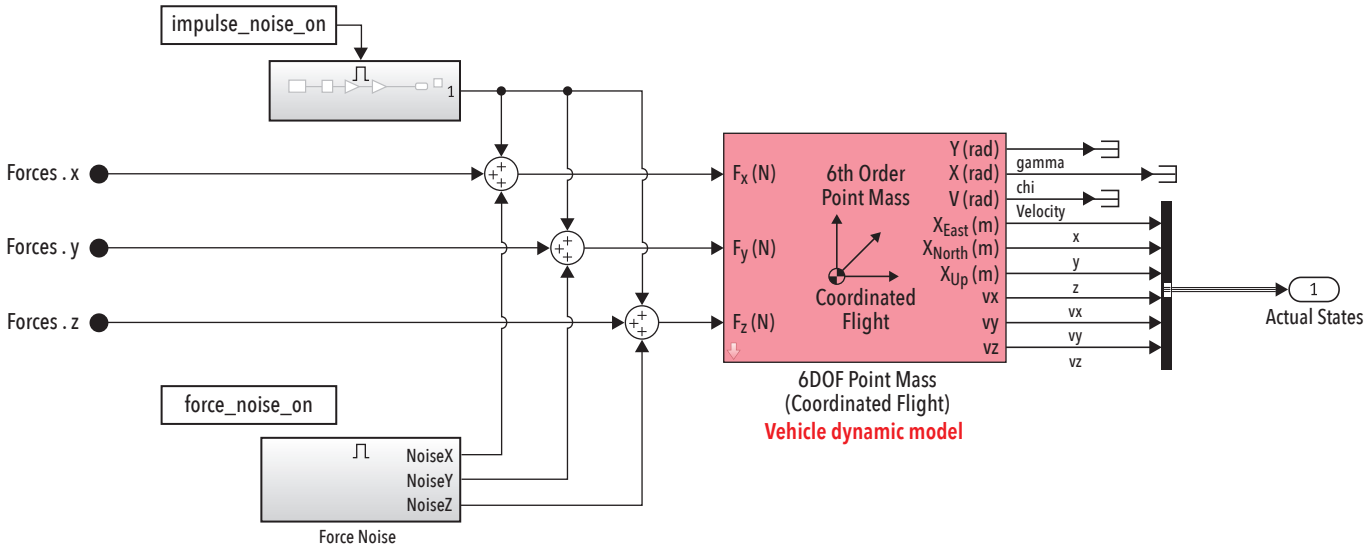


Figure 6. Constituent system model – an autonomous UAV

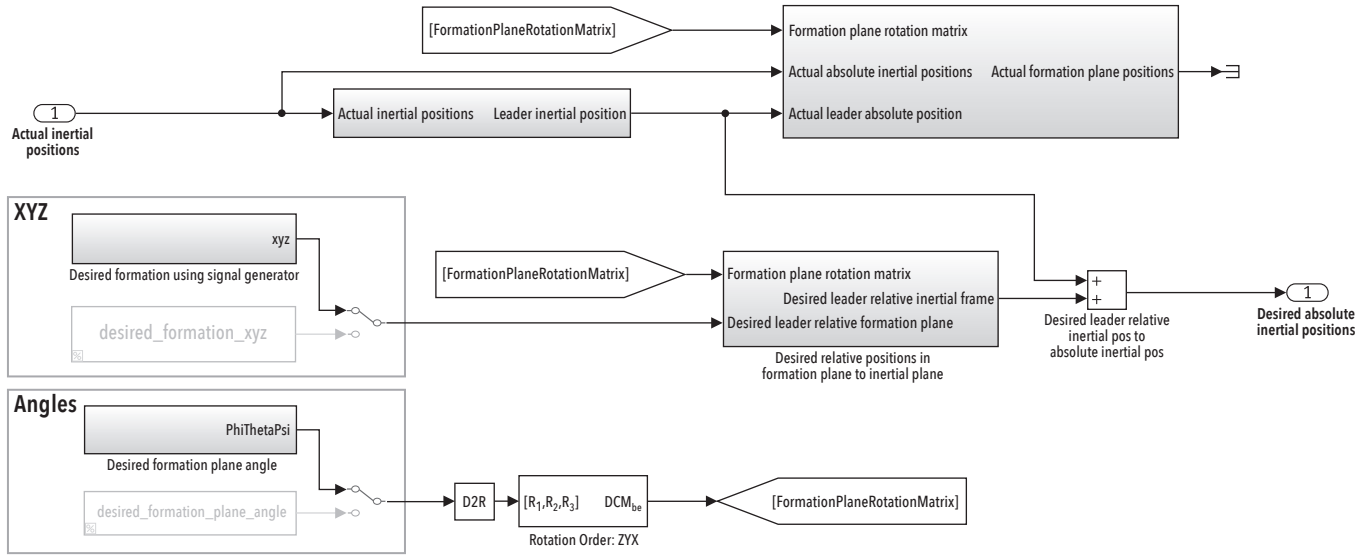


Figure 7. SoS – autonomous UAVs formation

The states are consumed by the swarm instincts block to generate desired states to maintain or change the formation. The desired states are passed onto the controller which generates required forces to be applied on the vehicle.

The required forces are then sent to the vehicle, which integrates them to get the current states. MATLAB animation display was used for visualizing the formation in real time during the various simulation runs. The constituent system autonomous UAV vehicle block is illustrated in Figure 6. The forces from the controller block are integrated within the 6-degree of freedom (6DOF) point mass block to generate the inertial position and velocities in ENU (east north up) frame. Impulse noise block introduces an impulse force at a random instant and of random total magnitude. This was used to simulate various scenarios such as wind gust. The force noise block adds a Gaussian noise to all the forces to simulate real world conditions. Figure 7 illustrates the model that pertains to the various formation shapes as part of the simulation runs. There are three frames of relevance in the model: (a) leader relative formation plane frame, in short, referred to as the formation plane (b) leader relative inertial frame, and (c) absolute inertial frame. The desired shape of formation and formation plane angles are taken as input, either from a time series file or from a signal generator. The desired formation plane angle is defined by Euler angles, which is consumed as a time series file or from a signal generator. The angles are converted to DCM (direction cosine matrix) in ZYX (axis) order. The desired positions in formation plane are transformed to the leader relative inertial frame using the DCM of formation plane. Leader relative inertial positions are converted to absolute inertial by adding leader's inertial frame position. Figure 8 illustrates the individual controller model that resides in each of the autonomous UAVs. The controller generates the desired forces to achieve the desired states from the actual current states. The cascaded proportional–integral–derivative (PID) is used to control the outer loop positions and the inner loop velocity. Saturation is added to the outputs to limit the control corrections to realistic values. Some of the formations are illustrated in Figure 9, along the rolling, pitching and yawing planes.

Design of Experiments

An orthogonal array of experiments is devised to analyze the behavior of the SoS for different values of various state parameters, as indicated in Figure 10. Following

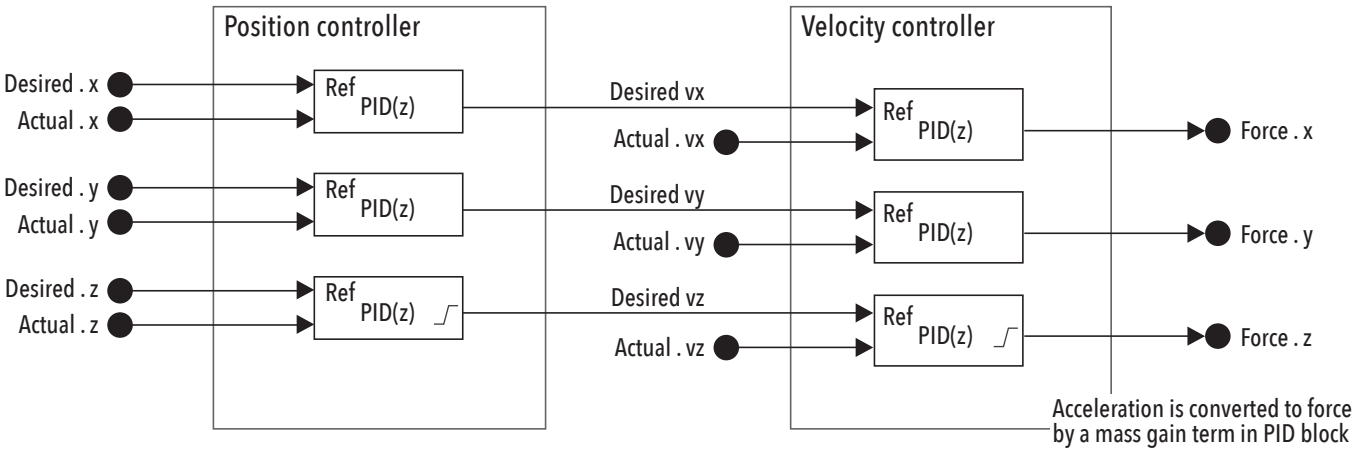


Figure 8. Constituent system: autonomous UAV – controller

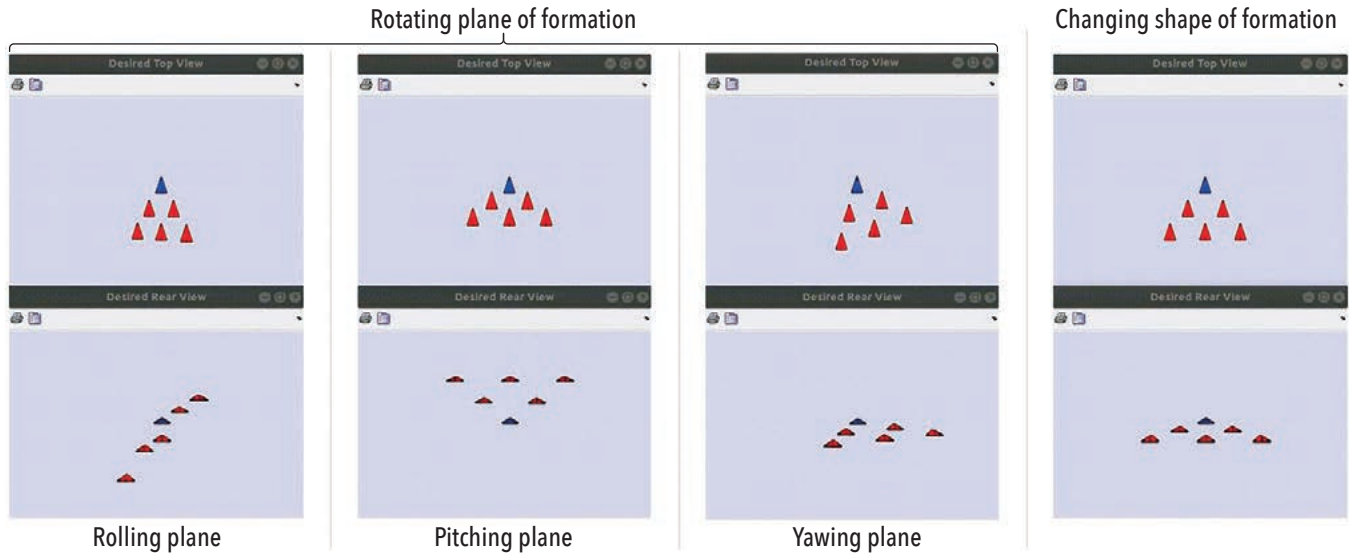


Figure 9. Various UAV formations in different planes

parameters are considered for behavior analysis: (1) initial / final formation shape, (2) angles of rotation of formation's plane of reference, as defined by Euler angles phi ϕ, theta θ, psi ψ, (3) vehicles facing wind gust, and the corresponding magnitudes on X,Y,Z axes, with/without forces.

Principal Component Analysis (PCA) and Zone Visualization

There are many factors that impact the emergent behavior of SoS, and visualization of the SoS state parameters would provide deeper insights. However, to understand the interplay of the various factors on SoS emergent behavior, a higher dimensional visualization is required which would be difficult to represent. Principal component analysis (PCA) (Martınez and Kak 2001) is used towards getting lower dimensional views. PCA comprises projection of an n-dimensional input data onto a reduced k-dimensional linear subspace, such that the reconstruction error is minimized. The lower-dimensional view is a projection of various points in the multi-dimensional space when viewed from its most informative viewpoint. PCA can be done by singular value decomposition of a data matrix, after mean centering, and normalizing the data matrix for each attribute.

Figure 11a illustrates the plot of PCA of UAV pair-wise distances (15 pairs between the five UAVs), against the PCA of UAVs pair-wise slopes (15 pairs between the five UAVs, with two slopes along XZ and YZ). The figure essentially represents the state of the SoS, reducing the multi-dimensional state parameters to lower dimensions, enabling identification of specific regions/ zones of positive (1) and negative (0) emergence. Further, the state space of a constituent system can be analyzed against the state space of the SoS, with respect to the emergent behavior and the MOEs. Figure 11b illustrates the plot of PCA of SoS versus PCA of UAV#3 MOEs. The following 3 scenarios are depicted in the figure: the zone of both SoS and UAV#3 exhibiting bad behavior (legend 0 in the plot, red); the scenario of UAV#3 meeting its own MOEs, but SoS is exhibiting negative behavior (legend 1 in the plot, yellow); and finally, the scenario of both UAV#3 and the SoS exhibiting positive behavior (legend 3 in the plot, green). Further the scenario of UAV#3 not meeting its MOEs while the SoS exhibiting positive behavior does not occur. This implies scenario wherein the constituent system is a key player in the SoS (scenario depicted in MOE relationship matrix in Figure 2). Three different supervised learning classification algorithms were tried (a) naïve Bayes classification, (b) fitted binary classification decision tree, and (c) KNN-nearest neighbor (Mitchell 1997). The decision surface of these different classification algorithms is illustrated in Figure 12, matching well with Figure 11b (legend used in Figure 12 is same as used in Figure 11b).

Machine Learning Model – ML Classifier

In the simulation runs of the complex SoS comprising autonomous UAVs, various state parameters of the formation are logged —

Experiment #	Initial Shape	Final Shape	Final psi	Final theta	Final phi	Gust UAVs	Gust X (kN)	Gust Y (kN)	Gust Z (kN)	Force on/off
1	Triangle	Triangle	-45	-45	-45	0	5	5	5	0
2	Triangle	Triangle	-45	-45	0	UAV3	2.5	2.5	2.5	1
3	Triangle	Triangle	-45	-45	-45	UAV 3 and 6	1.6	1.6	1.6	1
4	Triangle	Inv. Triangle	0	0	-45	0	5	2.5	2.5	1
5	Triangle	Inv. Triangle	0	0	0	UAV3	2.5	1.6	1.6	1
6	Triangle	Inv. Triangle	0	0	45	UAV 3 and 6	1.6	5	5	0
7	Triangle	Closed Loop	45	45	-45	0	5	1.6	1.6	1
8	Triangle	Closed Loop	45	45	0	UAV3	2.5	5	5	0
9	Triangle	Closed Loop	45	45	45	UAV 3 and 6	1.6	2.5	2.5	1
10	Inv. Triangle	Triangle	0	45	-45	UAV3	1.6	5	2.5	1
11	Inv. Triangle	Triangle	0	45	0	UAV 3 and 6	5	2.5	1.6	0
12	Inv. Triangle	Triangle	0	45	45	0	2.5	1.6	5	1
13	Inv. Triangle	Inv. Triangle	45	-45	-45	UAV3	1.6	2.5	1.6	0
14	Inv. Triangle	Inv. Triangle	45	-45	0	UAV 3 and 6	5	1.6	5	1
15	Inv. Triangle	Inv. Triangle	45	-45	45	0	2.5	5	2.5	1
16	Inv. Triangle	Closed Loop	-45	0	-45	UAV3	1.6	1.6	5	1
17	Inv. Triangle	Closed Loop	-45	0	0	UAV 3 and 6	5	5	2.5	1
18	Inv. Triangle	Closed Loop	-45	0	45	0	2.5	2.5	1.6	0
19	Closed Loop	Triangle	45	0	-45	UAV 3 and 6	2.5	5	1.6	1
20	Closed Loop	Triangle	45	0	0	0	1.6	2.5	5	1
21	Closed Loop	Triangle	45	0	45	UAV3	5	1.6	2.5	0
22	Closed Loop	Inv. Triangle	-45	45	-45	UAV 3 and 6	2.5	2.5	5	1
23	Closed Loop	Inv. Triangle	-45	45	0	0	1.6	1.6	2.5	0
24	Closed Loop	Inv. Triangle	-45	45	45	UAV3	5	5	1.6	1
25	Closed Loop	Closed Loop	0	-45	-45	UAV 3 and 6	2.5	1.6	2.5	0
26	Closed Loop	Closed Loop	0	-45	0	0	1.6	5	1.6	1
27	Closed Loop	Closed Loop	0	-45	45	UAV3	5	2.5	5	1

Figure 10. Design of experiments (DOE)

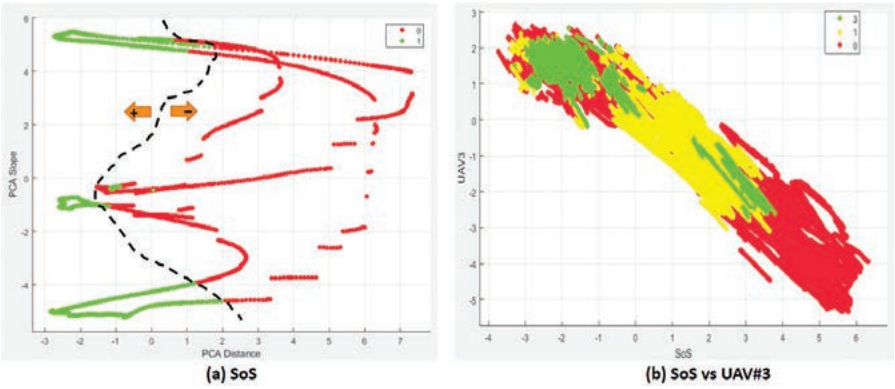


Figure 11. PCA analysis and visualization

including distances and bearings between each of the UAVs. The scenarios are labelled as "good"(1) or "bad"(0) based on the behavior seen at the SoS level. These labelled scenarios are fed into a neural network, and supervised learning algorithms were devised so that the network learns on the positive and negative emergent behaviors. Figure 13 illustrates the neural network-based ML model. The number of hidden layers, and number of units in the hidden layer defines the topology of the network. The inputs to the ML model are the various pairwise Euclidean distances, YX slopes (β) and ZX slopes (α) in the swarm, as illustrated in Figure 14. The trend of the parameters for a window of 4-time steps is provided

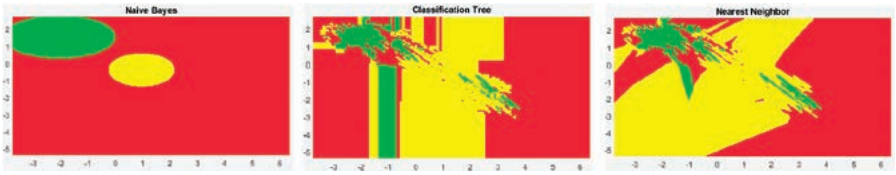


Figure 12. Classification of SoS versus UAV#3 behaviors

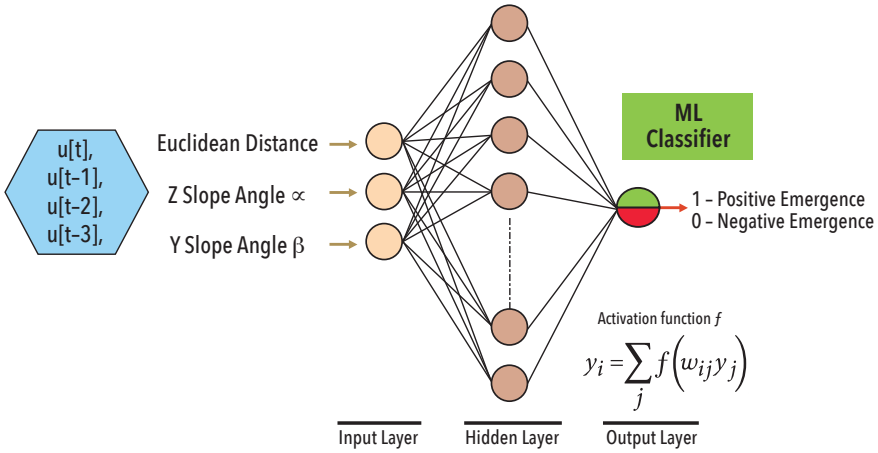


Figure 13: Machine learning ML classifier

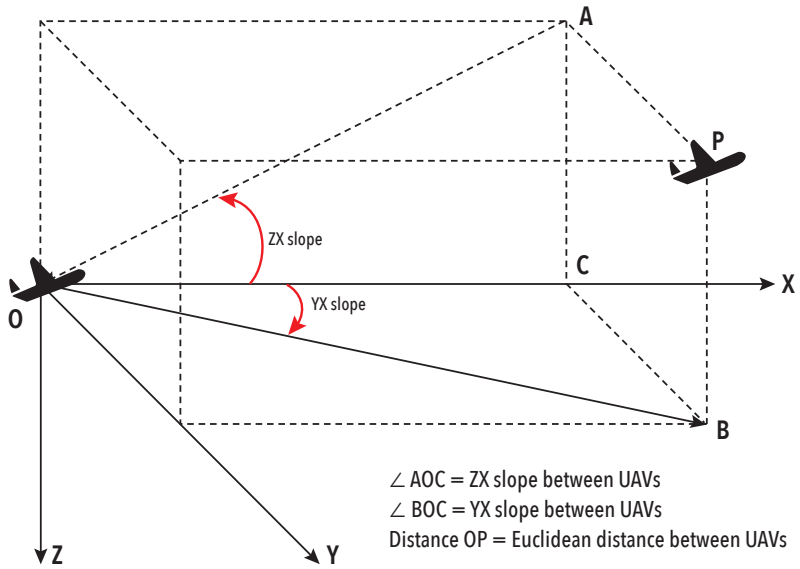


Figure 14. UAV formation – state parameters w.r.t pairs of constituent systems

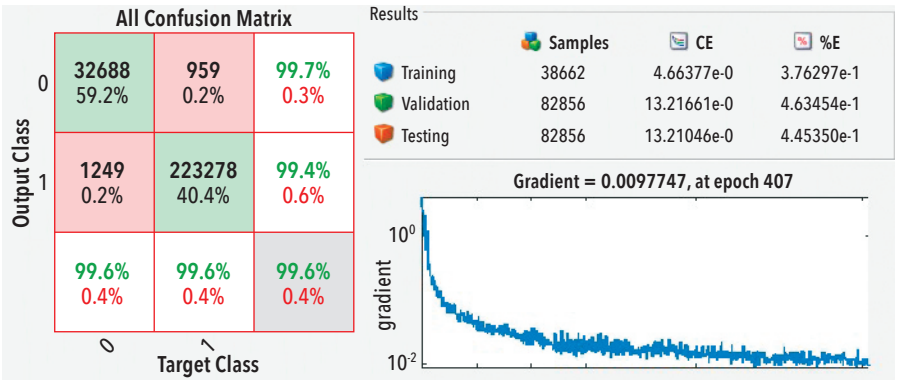


Figure 15. ML classifier – training performance

as the learning data set. The data set is further split into training set, validation set and test sets. Typically, the training set is used to fit the model, while the validation set is used to estimate prediction error for model selection. The test set is then used for assessment of the generalization error of the final chosen model. Learning algorithms are devised that can automatically tune (and learn) the weights and biases so that the output produced by the network closely matches the desired output. Mathematically, this close matching involves an associated cost function that needs to be minimized. Hence, the training process is iterative, to minimize the cost function below a threshold, with each iteration fine tuning the parameters. The iteration concludes once the cost function reaches the minima, below the expected threshold. The weights/biases thus learned by the neural network at the end of the iteration represents the parameters that can be used to directly transform the inputs to outputs. The data set comprising over half a million records is split between training (75%), testing (15%), and validation (15%) sets.

The performance of the learning is assessed in terms of cross entropy function, wherein minimizing the cross-entropy (CE) leads to better classifiers. Figure 15 illustrates the ML classifier training performance, for the scaled conjugate algorithm. The confusion matrix indicated in the figure illustrates the accurate and inaccurate classifications. The rows correspond to the predicted class (output class) and the columns correspond to the true class (target class). The diagonal cells (green color) indicate the correctly classified observations. The off-diagonal cells (light rose color) are the incorrectly classified observations. Both the number of observations and the percentage of the total number of observations are shown in each cell. The column on the far right of the plot shows the percentages of all the examples predicted to belong to each class that are correctly and incorrectly classified. These metrics are the precision (or positive predictive value) and false discovery rate, respectively. The row at the bottom of the plot shows the percentages of all the examples belonging to each class that are correctly and incorrectly classified. These metrics are often called the recall (or true positive rate) and false negative rate, respectively. The cell in the bottom right of the plot shows the overall accuracy. As seen, overall, the prediction accuracy performance achieved is 99.6%.

ML Classifier – Behavior Predictions

The trained ML classifier is used to observe the SoS behavior and predict positive and negative emergence. The ML

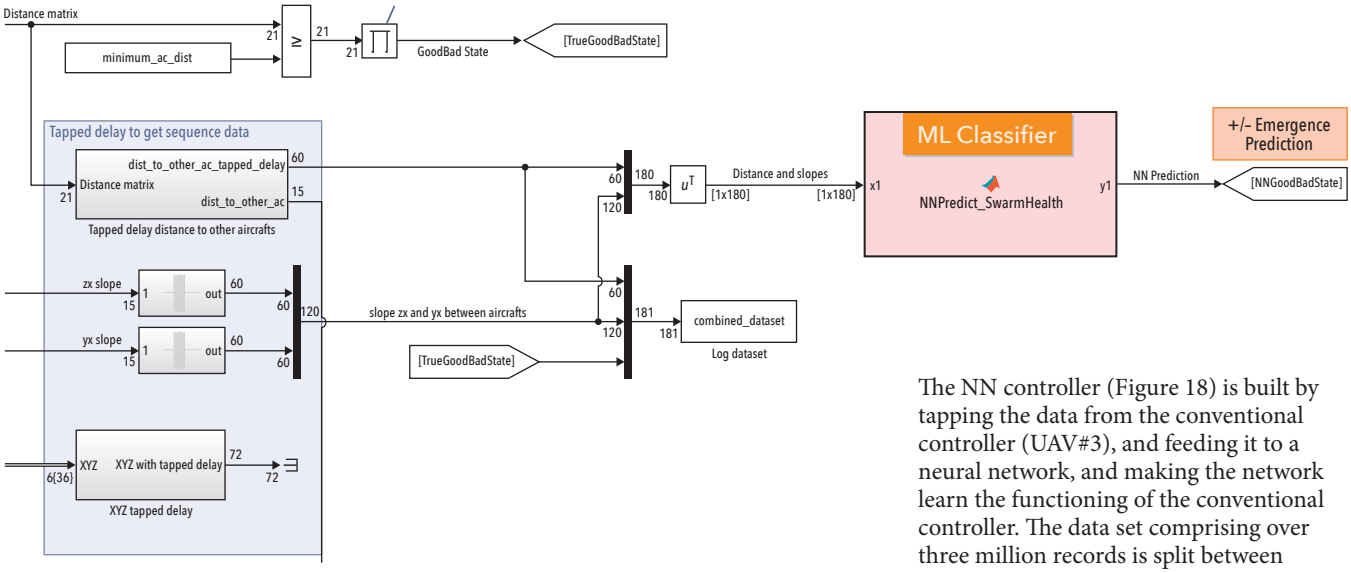


Figure 16. ML-classifier integrated into SoS model

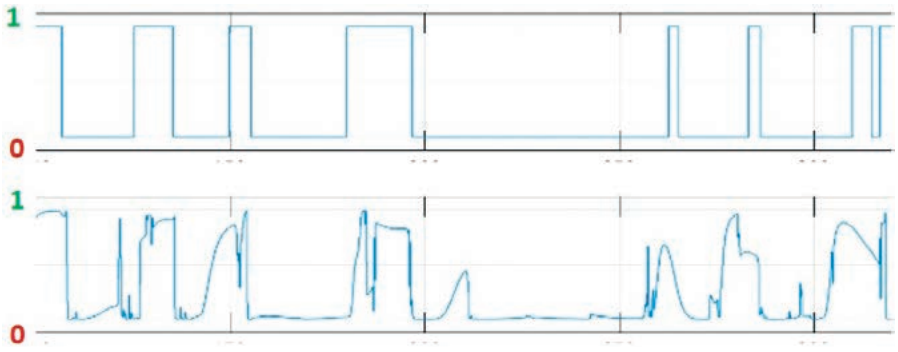


Figure 17. Machine learning model predictions of emergent behavior. (top) Sos Behavior Labelled on tiime steps as "good" (1) or "bad" (0); (bottom) ML Classifier predicted behavior probabilities, between 1("positive emergence) and 0 ("negative emergence")

classifier is plugged onto the SoS model as illustrated in Figure 16, to understand and predict the emergent behaviors as positive and negative behaviors. Figure 17 indicates snapshots of the tool scope monitor, wherein values closer to 1 indicates positive emergent behavior being exhibited, while values closer to 0 indicates negative emergent behavior.

SoS with Hybrid (conventional + machine learning) Constituent Systems

The scenario of the SoS comprising a mix of conventional constituent systems and machine-learning model-based constituent systems is then studied. Towards this, the conventional controller (Figure 8) is replaced with a machine learning based NN controller for UAV numbered #3 (Figure 5).

The NN controller (Figure 18) is built by tapping the data from the conventional controller (UAV#3), and feeding it to a neural network, and making the network learn the functioning of the conventional controller. The data set comprising over three million records is split between training (75%), testing (15%), and validation (15%) sets. The learning is stopped at epoch of 1000. The R value in this case has reached only about 0.4 after completion of 1000 epochs (a robust learning would imply an R value very close to 1). This learning is stopped to serve the purpose of dealing with a system that can exhibit negative emergent behavior at times. Figure 19 illustrates the scenario of the formation flying with the NN controller managing UAV #3 included in the formation. As indicated, UAV#3 misbehaves (that is, exhibits an oscillating behavior). This misbehavior is predicted by the ML classifier, indicating a negative emergence for the SoS.

Formal Methods

We have used formal methods for model checking in the earlier work (Raman and Jeppu 2020). We could optimize for time by changing the sampling time and abstracting the behavior of collision in the collision avoidance problem. We tried the same approach of using the Simulink design verifier on the 6 UAVs maneuvering together. The assertion was that once the maneuver occurs the UAVs will not collide while they maneuver.

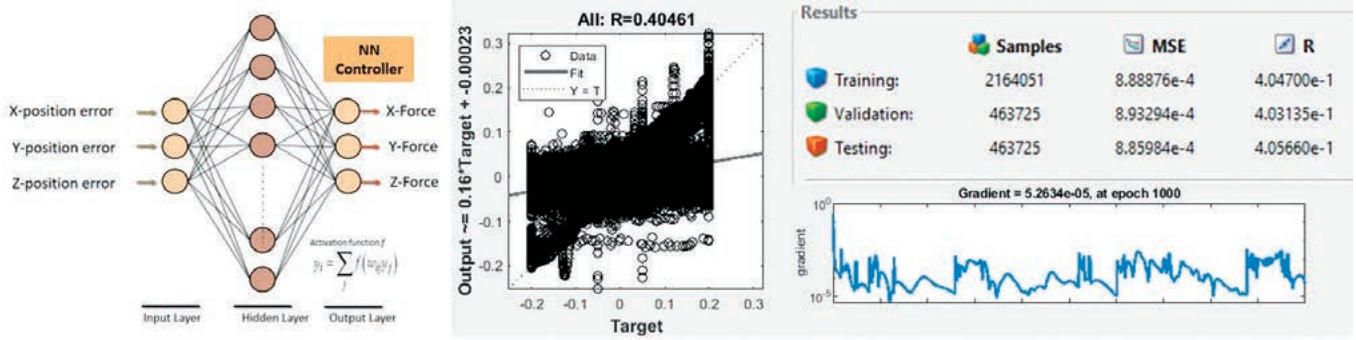


Figure 18. NN controller

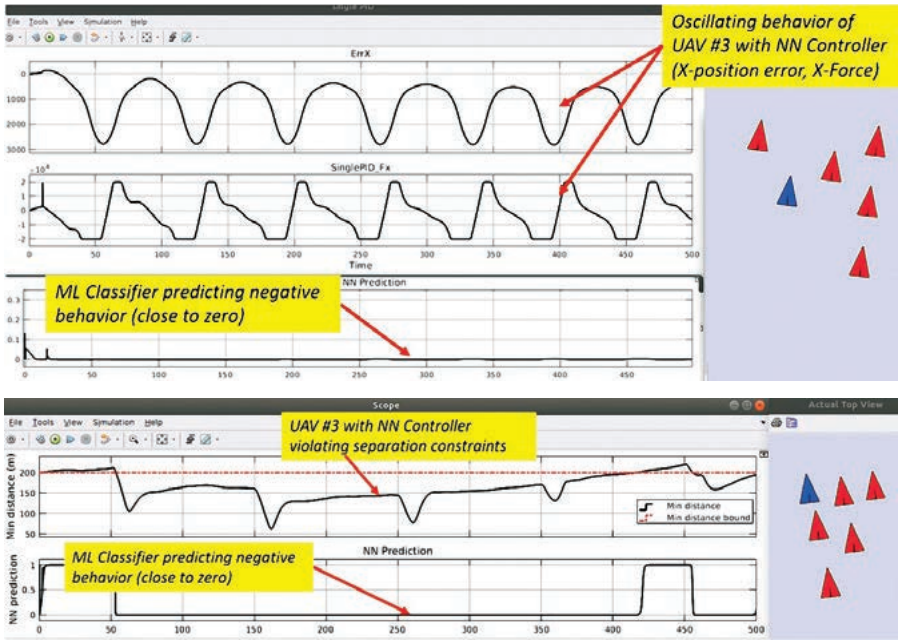


Figure 19. SoS behavior with machine learning based UAV #3

During the maneuver, a wind gust can shift location of the UAVs. The individual controller behavior was approximated to a proportional control. The sample time was increased, and the direction cosine matrix eliminated by considering the maneuver in a plane of operation. These assumptions were made based on our earlier experience of using formal methods.

The Simulink design verifier took about eight hours to indicate that there could be a collision if there was a gust. We had to look at a small zone around the gust region for the model checking. Bringing in a variable in the gust time could not provide a result as it hit the limits of the computing resources used. We explored a C bounded model checker CBMC on the C language implementation of the neural network. The 6 UAV engagement scenario was coded in C and the minimum distance computed between the various UAV as a measure of asserting the behavior that collision could not happen. CBMC is a bounded model checker, and we can look at the zone where the maneuver occurs for proving the correctness. CBMC unwinds the "while loop" for the time of execution and "for loops" for the 6 UAV. The ML classifier in loop in CBMC hit the limits of the computing resources used. Other features of CBMC like array bound checks and divide by zero checks worked well with the neural network code. Subsequently, C program simulation of UAVs was tried – the simulation of the 6 UAV in the C code is shown in Figure 20. The 6 UAVs move in a 2-dimensional space and carry of a maneuver to invert the V shape. A wind gust disturbance is applied that shifts the position of all UAVs during the maneuver. The UAV realign to the new shape as seen in the plot. The simulation works well as seen. We now need to prove that it is always the case. We define the problem statement for the formal correctness as ***Definition 1*** (see below).

CBMC can prove that the condition is satisfied but to really ensure that the behavior is correct we have changed the threshold for minimum distance to a larger value to ensure that it fails and provides a counter example. CBMC can provide us this solution with a failure. It takes about 700 seconds to solve the problem. We provide the complexity of the problem as the number of variable and clauses the CBMC generates from the UAV SoS C code. The number of variables and clauses define the complexity of the problem. CBMC converts the C code into a conjunctive normal form (CNF). The CNF is a conjunction of one or more clauses, where each clause is a disjunction of variables. The C code gets translated to approximately 2,350,198 variables which are combined into around 9,416,265 clauses. The approximation is because these number change with the change in the range looked for the time of disturbance t_d. In view of the complexity of the neural network problem and the large time taken by the formal methods tool, we have explored another way of looking at the neural network performance in terms of the PCA explained above. The PCA defines the plot of the principal components of the slope or angle between each UAV and the distance measures the range between the UAVs. It is possible to define a region in

Definition 1: The distance between UAV_i and UAV_j at time t is given by $d_t(i,j) = \sqrt[2]{(x_i - x_j)^2 + (y_i - y_j)^2}$ where $(x_i - x_j)$ and $(y_i - y_j)$ represent the coordinates of UAV_i and UAV_j respectively.

Given start time maneuver t_m, end time for maneuver t_c and a disturbance at time t_d where $t_m \le t_d \le t_c$ then property $\phi = \min_{1 \le i,\, j \le 6 \wedge i \ne j} d_t(i,j) \ge \Delta$, $\forall t: t_m \le t \le t_c$ where Δ is maximum unsafe distance, holds. We look at this property in CBMC using the assert statement and assume the time of disturbance as a variable. We look at 20 seconds after maneuver as t_c

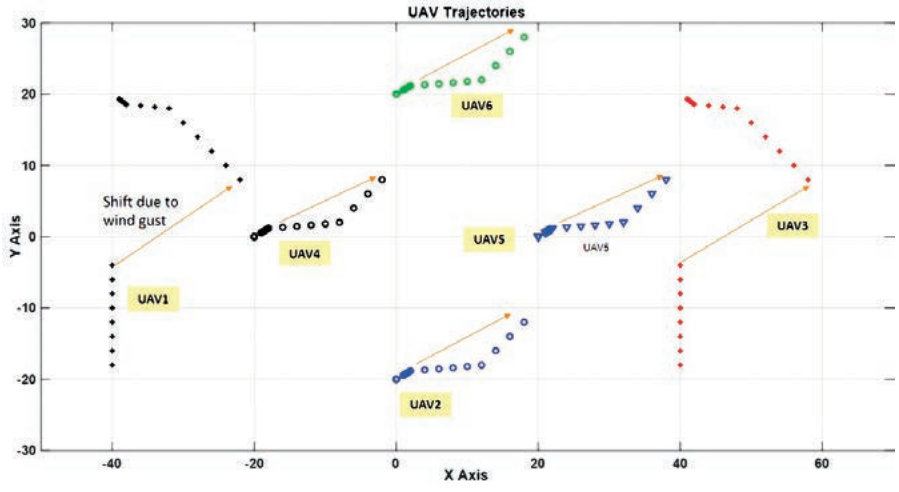


Figure 20. SoS – trajectories simulation in C programming language

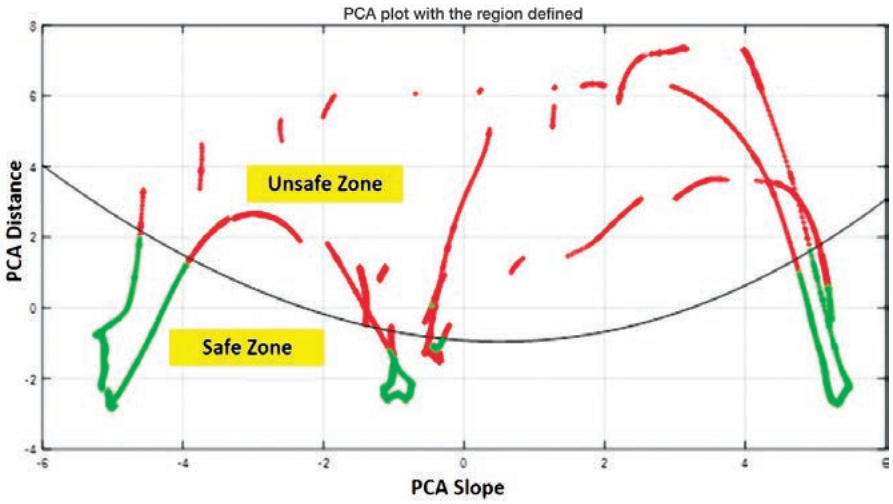


Figure 21. PCA and the region defined by the polynomial

the plot of PCA that defines a safe behavior for the swarm of 6 UAV. The region can be defined in this case a polynomial in X (the PCA slope) providing a Y (PCA distance) as shown in Figure 21. If the PCA values are in the safe zone and the neural network indicates so then we can say that neural network is correctly predicting the MOE of collision or unsafe swarm behavior. In the initial study on this given a small zone where there was a collision the formal method could provide an indication that the neural network was indeed providing an error. As the space increases, we again hit the tool and memory limitations. The results are preliminary but this, we feel, is a good direction to explore in future.

CONCLUSIONS & FUTURE WORK

This paper presented a novel approach towards application of machine learning based classifiers and formal methods for analyzing and evaluating emergent behavior of complex system-of-systems. The various elements of the framework were illustrated through a case of a swarm of autonomous UAVs flying in a formation, and dynamically changing the shape of the formation, to support varying mission scenarios. PCA based zone classification for analyzing constituent system versus SoS behaviors, machine learning classifier models for predicting positive and negative emergent behaviors, and formal verification models were presented, including their effectiveness and performance. In the future, we plan to enhance the framework to address multiple scenarios pertaining to evolution of the SoS. Over a period, different changes can happen in constituent systems, such as new functions getting added, obsolete functions getting removed, efficient means of realizing some of existing functions being incorporated and other structural changes. These changes could cause changes in the emergent behavior of the SoS. ■

REFERENCES

- Aiguier, M., P. L. Gall, and M. Mabrouki. 2008. "A Formal Definition of Complex Software." Proceedings of the 3rd International Conference on Software Engineering Advances, ICSEA, MT.
- Baomar, H., and P. J. Bentley. 2017. "Autonomous Landing and Go-Around of Airliners Under Severe Weather Conditions Using Artificial Neural Networks." Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS) Linköping, SE.
- CBMC. 2020 Bounded Model Checking for Software. https://www.cprover.org/cbmc/.
- Dong, X., Y. Li, C. Lu, G. Hu, Q. Li and Z. Ren. 2019. "Time-Varying Formation Tracking for UAV Swarm Systems With Switching Directed Topologies." *IEEE Transactions on Neural Networks and Learning Systems* 30 (12): 3674-3685.
- Giammarco, K. 2017. "Practical Modeling Concepts for Engineering Emergence in Systems of Systems." IEEE System of Systems Engineering Conference (SoSE), Waikoloa, US-HI, 18-21 June.
- Hassanien, A.E. and E. Emary. 2016, *Swarm Intelligence – Principles, Advances, Applications*. CRC Press.
- INCOSE. 2005. Technical Measurement Guide. INCOSE TP-2003-020-01.
- ———, 2015. *Systems Engineering Handbook*. 4th Edition. Hoboken, US-NJ: Wiley.
- ———, 2016. Systems of Systems, *INSIGHT 19* (3).
- Jamshidi, M. 2008. *Systems of Systems Engineering: Principles and Applications*. CRC Press.
- Kang, E., and L. Huang. 2018. "Formal Specification and Analysis of Autonomous Systems in PrCCSL/Simulink Design Verifier." Cornell University Library arXiv:1806.07702. https://arxiv.org/abs/1806.07702/.
- Katz, G., C. Barrett, D. Dill, K. Julian, and M. Kochenderfer. 2017. "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks." Cornell University Library arXiv:1702.01135.
- Kinsner, W. 2008. "Complexity and its Measures in Cognitive and Other Complex Systems." 7th IEEE International Conference on Cognitive Informatics, Stanford University, US-CA, 14-16 August.
- Ladyman, J., J. Lambert, and K. Wiesner. 2013. "What is a Complex System?" *European Journal for Philosophy of Science* 3: 33-67.
- Lane, J. A. 2013. "What is a System of Systems and Why Should I Care?" USC-CSSE-2013-001.
- Martınez, A. M., and A. C. Kak. 2001. "PCA versus LDA." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2: 228–233.
- Mitchell, T. 1997. *Machine Learning*. McGraw Hill.
- Mohri, M., A. Rostamizadeh, and A. Talwalkar. 2012. *Foundations of Machine Learning*. MIT Press, Cambridge US-MA.
- Nanda M., J. Jayanthi, and Y. Jeppu. 2018. "Formal Methods—A Need for Practical Applications." eds. M. Nanda and Y. Jeppu. *Formal Methods for Safety and Security*. Springer.
- Nielsen, C.B. et al. 2015. "SoS Engineering: Basic Concepts, Model-Based Techniques and Research Directions." ACM Computing Surveys (18).
- PVS. 2018. Specification and Verification System. http://pvs.csl.sri.com/
- Raman, R., and M. D'Souza. 2017. "Knowledge Based Decision Model for Architecting and Evolving Complex Systems-of-Systems." INCOSE International Symposium 27: 30-44.

- Raman, R., and M. D'Souza. 2018. "Learning Framework for Maturing Architecture Design Decisions For Evolving Complex SoS." 13th IEEE Conference on System of Systems Engineering (SoSE), Paris, FR, 19-22 June, pp. 350-357.
- Raman, R., and M. D'Souza. 2019. "Decision Learning Framework for Architecture Design Decisions of Complex Systems and Systems-of-Systems." *Systems Engineering* 22: 538– 560.
- Raman, R., and Y. Jeppu. 2019. "An Approach for Formal Verification of Machine Learning based Complex Systems." INCOSE International Symposium 29: 544-559.
- Raman, R., and Y. Jeppu. 2020. "Formal Validation of Emergent Behavior in a Machine Learning Based Collision Avoidance System." IEEE International Systems Conference (SysCon), CA, 24-27 June.
- RTCA. 2011. DO-333 Formal Methods Supplement to DO-178C and DO-278A.
- SLDV. 2020. Simulink Design Verifier.
- Smith, N., and T. Clark. 2006. "A Framework to Model and Measure System Effectiveness." 11th ICCRTS Coalition Command and Control in The Network Area Conference.
- Tahir, A., J. Böling, M. Haghbayan, H. T. Toivonen, and J. Plosila. 2019. "Swarms of Unmanned Aerial Vehicles — A Survey." *Journal of Industrial Information Integration* 16, https://doi.org/10.1016/j.jii.2019.100106.
- Williams-Hayes, P. 2005. "Flight Test Implementation of a Second Generation Intelligent Flight Control System." Infotech@Aerospace Conference, Arlington, US-VA, 26-29 September http://doi.org/10.2514/6.2005-6995.
- Xiang, W., P. Musau, A. W. Ayana, D. M. Lopez, N. Hamilton, X. Yang, J. Rosenfeld, and T. T. Johnson. 2018. "Verification for Machine Learning, Autonomy, and Neural Networks Survey." Cornell University Library arXiv:1810.01989 http://arxiv.org/abs/1810.01989.

ABOUT THE AUTHORS

[Editor: Author biographies were current when the paper was initially published in 2021.]

Dr. Ramakrishnan Raman received B.Tech and MS degrees from IIT Madras, and PhD from IIIT-Bangalore. He is a certified Six Sigma Black Belt and is an INCOSE certified Expert Systems Engineering Professional – ESEP. He has extensive systems and software engineering experience in domains of building/industrial automation, and aerospace. He has been the lead systems engineer and architect for the design of many complex systems globally over the years. He is currently a principal systems engineer at Honeywell Technology Solutions, Bangalore. (ORCID:0000-0002-8471-9172)

Dr. Yogananda Jeppu is a BE in electronics and communication, from Mangalore University, a post graduate in missile guidance and controls from Pune University. He has a PhD in certification of safety critical control systems using model based techniques. He has been working in the field of control system design and implementation, simulation of aerospace systems, and verification and validation for aircrafts and missiles for the past 32 years. He started his career in 1987, working on missiles and the Indian Light Combat Aircraft program with Defense R&D Organization. He is currently working in Honeywell Technology Solutions as a principal systems engineer. (ORCID: 0000-0003-1401-6348)

Nikhil Gupta completed his BTech and MTech in aerospace engineering from IIT Madras in 2018 with a specialization in guidance, navigation, and control of unmanned aerial vehicles. He has participated in several aeromodelling competitions and loves piloting RC aircraft and multirotors. He holds a patent to his name and is certified as a SolidWorks Associate and in Six Sigma Green Belt for software. Currently, he is working in Honeywell Technology Solutions as a senior embedded engineer in the field of alternate guidance and navigation methods primarily using computer vision and neural networks.

Flanigan and Robinson continued from page 17

- Tetlow, M. et al. 2013. "Modelling Requirements for Mission Success Prediction." MODSIM2013, Adelaide, AU.
- Wylie, Matthew, David Harvey, and Tommie Liddy. 2016. "Model-Based Conceptual Design Through to System Implementation-Lessons from a Structured Yet Agile Approach." 2016 Systems Engineering Test and Evaluation Conference: SETE 2016. Engineers Australia.
- Yaroker, Yevgeny, Valeriya Perelman, and Dov Dori. 2013. "An OPM Conceptual Model-Based Executable Simulation Environment: Implementation and Evaluation." *Systems Engineering* 16 (4): 381-390.

ABOUT THE AUTHORS

[Editor: Author biographies were current when the paper was initially published in 2019.]

David Flanigan is a member of the principal professional staff for The Johns Hopkins University Applied Physics Laboratory, providing systems engineering services to various Department of Defense and Department of Homeland Security clients, and has 20 years of active duty and reserve service with the US Navy. A graduate of the University of Arizona, he holds a MS in information systems and technology, a MS in systems engineering from the Johns Hopkins University, and a PhD in systems engineering and operations research from George Mason University. Dr. Flanigan is a member of INCOSE, INFORMS, and MORS.

Kevin Robinson is the chief engineer at Shoal Engineering with a distinguished career in the field of guided weapons in both the UK Ministry of Defence and Australia's Department of Defence. He has made significant contributions to the development of advanced guided weapons through modelling and analysis, research, and leadership of large cross discipline teams. Throughout his career, Kevin has taken a leadership role in advancing the field of model-based systems engineering (MBSE) via his publications and contributions to the systems engineering community. He initiated and chaired Australia's first annual MBSE symposium, formed and chaired INCOSE's model-based conceptual design working group, delivered a keynote address to INCOSE's international symposium in 2016, and has made contributions to INCOSE's Systems Engineering Handbook and related standards. He has joined INCOSE's future of systems engineering (FuSE) initiative core team.

APRIL 2026
VOLUME 29/ISSUE 2

Carbon Considerations for Systems Evolution

David Flanigan, David.Flanigan@jhuapl.edu; and **Kevin Robinson,** kevin.robinson@incose.net

■ **ABSTRACT**
In the early stages of systems development, systems engineers will typically evaluate alternatives based on performance, cost, risk, and schedule to evaluate the solution space of alternatives. While these criteria have proven to be successful, there is growing interest in the analysis of carbon costs as well to contribute to the decision making. These decision criteria are very good to help the decision maker select the best alternative within the solution space in which to develop a system concept. We offer another criterion for consideration to account for carbon expenditure throughout the systems engineering lifecycle. We believe that including this dimension can influence decision makers to evaluate a richer portion of the solution space. This approach is developed and exercised with a notional example.

INTRODUCTION

Systems concepts during the conceptual development phase are often evaluated in terms of performance, cost, and schedule, particularly when evaluating different alternatives for moving to the development phase. With the growing interest and concern of emissions to affect the entire planet, this paper takes motivation to introduce a new aspect for considering system concepts, with a carbon count. There are several ways to examine counting carbon, be it at the manufacturing, or operations level. We introduce a new perspective for systems engineers and program analysts to consider when developing system concepts that would be often considered within an analysis of alternatives (AoA). We perform a literature review on the ways to consider carbon emissions as well as integrating into an AoA-type format for decision makers.

LITERATURE REVIEW

Our first part of the literature review explores a systems engineering analysis and decision-making framework that can support the consideration of carbon emissions early during the systems engineering lifecycle.

INCOSE's model-based systems engineering (MBSE) initiative (INCOSE 2022) has been driving toward implementing MBSE practices across the whole lifecycle, including in the concept stage. The benefits of MBSE are well published, with Henderson & Salado (2020) identifying a list of measured, observed and perceived benefits, including better communication and information, increased traceability, and better accessibility of information. It is an information framework, realized in a schema, that supports the MBSE benefits Henderson & Salado identify.

In 2008 Robinson et al. (2010) explored the application of MBSE to a real acquisition project in the concept stage, within the Australian Department of Defence. This exploration showed that a model-based conceptual design (MBCD) approach was "completely compatible with current mandated (document-centric) capability development processes." Key to the success of this project, was the clear definition of the schema underpinning the MBCD approach. This early definition of the concept stage schema has been expanded since to include information classes such as risk (Cook et al. 2015) and the test domain (Flanigan and Robinson 2019). Flanigan and Robinson (2020) demonstrated how a MBCD approach, with a robust schema, can be employed to better consider resilience in the concept stage with alignment to an analysis of alternatives (AoA) approach defined in the AoA Handbook (US Air Force 2017). The MBCD schema, described by Flanigan and Robinson (2020), potentially provides the information framework for introducing a new carbon-counting perspective for systems engineers and program analysts, to better inform decision makers within the Analysis of Alternatives approach.

Our second part of the literature review looks at the different government organizations concerns on carbon emissions reductions and goals. These may assist the analysis of our case study.

The United States Department of Defense (DoD) (September 2021) considers climate change in their climate adaption plan, particularly how systems are developed. Specific US military services have published their strategies and goals on how to address climate change and reduction in the carbon footprint goals (DoD September 2021, DoD October 2021, Department of the Air Force 2022, Department of the Army 2022, Department of the Navy 2022). The United Nations and other countries also have identified carbon reduction goals (UN 2015, UN 2022, Commonwealth of Australia 2022) which range from improving vehicle efficiency to reducing certain elements during system development to implementing activities to reduce the overall global average temperature.

Other research has been performed to calculate the carbon footprint of components (Wang 2022, Yung et al. 2018, Gupta et al. 2022, Williams et al. 2002) and add these calculations along with other

aspects of the systems engineering lifecycle to account for the overall carbon count. Müller et al. (2020) considers raw material manufacturing, component production, system integration, system testing, system usage, system maintenance, and system retirement as some of the key activities to consider when accounting for carbon. Mathers et al. (2014) provides estimates of carbon footprint per mile depending on the mode of transportation and weight of material carried, some of the modalities include air, ship, rail, and truck. As there are numerous options to construct the system, the analyst should be cognizant to understand the scope of analysis when considering carbon counting: do we count all the way back to when extracting materials out of the ground, when the components are already manufactured, or when the final system has been integrated?

When asked to evaluate the solution space between multiple system alternatives, systems engineers will typically perform an analysis of alternatives (AoA). Typical criteria are performance, cost, schedule, and risk. We find that the decision analysis approach can be applied to multiple parts of the systems engineering lifecycle (technical planning, technical assessment, stakeholder requirements, requirements analysis, and architecture design) (DoD 2022). NASA (2020) also has a similar decision analysis approach to identify parts of the lifecycle (mission concept, system requirements, mission definition, system definition, preliminary design review, critical design review, production readiness review). The US Air Force (USAF 2017) has an Analysis of Alternatives (AoA) Handbook that describes several integrated product teams (IPT) to focus on alternatives, effectiveness analysis, cost analysis, and risk assessment, as well as comparing alternatives, evaluating differing costs, capabilities, and risks. It is here where we can draw some motivation to include carbon footprint as well. Systems engineering textbooks (Kossiakoff et al. 2020; Buede and Miller 2016) contain additional information regarding lifecycle models as well as decision criteria.

APPROACH

We will revisit the systems engineering lifecycle and model based conceptual development (MBCD) framework, performed in previous work (Flanigan and Robinson 2019, 2020, 2021, 2022) to evaluate how conceptual design systems engineering can be applied to the problem in the operational, systems, and testing domains. We believe that the MBCD framework is still valid for this approach, with some modifications applied to the framework.

Systems Engineering Lifecycle

The systems engineer works across the entire lifecycle of the system being engineered. From the concept stage to the retirement of the system, the systems engineer must coordinate the lifecycle activities to ensure that risks are managed, opportunities are explored and ultimately a successful system in delivered, deployed and retired (Walden et al. 2015). However, this lifecycle, and the changing nature of systems, is being questioned by the challenges that society needs to address. As we are all painfully aware, climate change and exploitation of earth's finite resources are one of our most serious challenges that needs solving. Systems engineers must meet that challenge.

Highlighted in the United Nations sustainable development goals (SDGs) (United Nations 2022), the climate action goal calls for us to "Take urgent action to combat climate change and its impacts". To address this, INCOSE responded in its Systems Engineering Vision 2035 (INCOSE 2022) to challenge systems engineers to "help bring about informed overarching system solutions to climate change which include changes in public policy with coordinated and actionable mitigation steps that influence societal, corporate, and individual behaviors." This paper aims to contribute to that cause by considering how the systems engineering lifecycle can be improved to better act on climate change.

If we are to successfully reduce the carbon footprint of the systems we deliver, deploy, and retire, then the "carbon cost" of each stage of the lifecycle must be challenged. As we know, the projects that spend the greatest proportion of their resources in the concept stage are likely to be the most successful (Honour 2011). If the future systems to be engineered are to have a reduced carbon footprint, and reduce their impact on the environment, then understanding what can be achieved in the concept stages is surely going to lead to the greatest success.

A well-executed concept stage should provide immediate opportunities for reducing the carbon footprint of the system being engineered, with those opportunities flowing downstream through the remaining stages of the lifecycle. Within the systems engineering body of knowledge there have been many advances in the delivery of the concept stage (Robinson, Waite, Do 2014), but there are none known to the authors that have explicitly focused on developing opportunities that lead to reducing the carbon footprint. The concept stage is

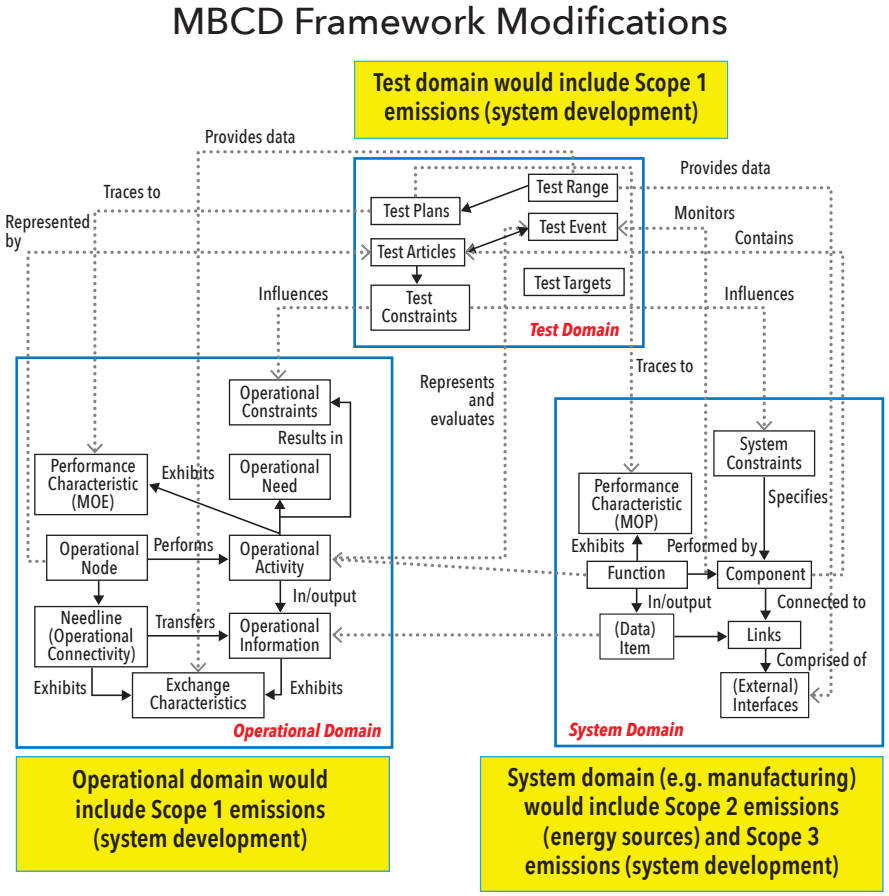


Figure 1. MBCD framework modified for carbon counting

focused on developing the stakeholder needs and requirements, reducing risks, pursuing opportunities, and defining the concepts of use (Walden et al. 2015). All these artifacts should set the project vision for reducing the carbon footprint of the system when considering delivery, deployment, and retirement. For this paper, the authors have chosen to explore the development and definition of the concepts of use, through model-based conceptual design approaches (Robinson, Waite, Do 2014).

MBCD Framework, Revisited

We revisit the MBCD framework as described to evaluate the changes to consider carbon factors. Figure 1 provides an updated view of the framework with discussion on how it could be modified for our analysis purposes.

Examining the MBCD schema, Figure 1, we can see that many of the systems engineering information classes captured in the schema can be considered to influence the carbon cost of a system being delivered, deployed, and retired. Key is the systems usage, as identified by Sparrevik and Utstøl (2020), as Scope 1 emissions, compared to Scope 2 (as a result from purchased energy sources), and Scope 3 emissions (development of systems). In the MBCD schema the 'operational activity' is the key information class to understanding the usage of a system, and therefore the highest cost of carbon pollution from energy sources.

The system engineer can make design choices between different solution options, based on the carbon cost to develop the component, however, to estimate the carbon cost across the full lifecycle carbon cost, the concept of use must be analysed for alternative solutions. This is described through the 'operational activity' information class in the model. For example, a single component may have a high individual development carbon cost (Scope 3), however if it provides an enhanced usage performance then it may pollute less carbon due to its efficient use (Scope 1). Conversely a low development carbon cost component (Scope 3) may have high pollution usage cost due to its poor performance and therefore inefficient use (Scope 1).

In this paper we explore the AoA approach to make carbon-cost design choices during the conceptual design of the operational activities, that are then realized through system concepts.

NOTIONAL EXAMPLE

We will describe our approach towards a notional example of surveillance of forest fires, such as in a national park, to support the command and control of bushfire response options. The size of the park is so large that the concept may require a multi-layered approach for adequate and timely detection and monitoring of the fires. Several alternatives are identified to produce notional analysis for the decision makers to consider.

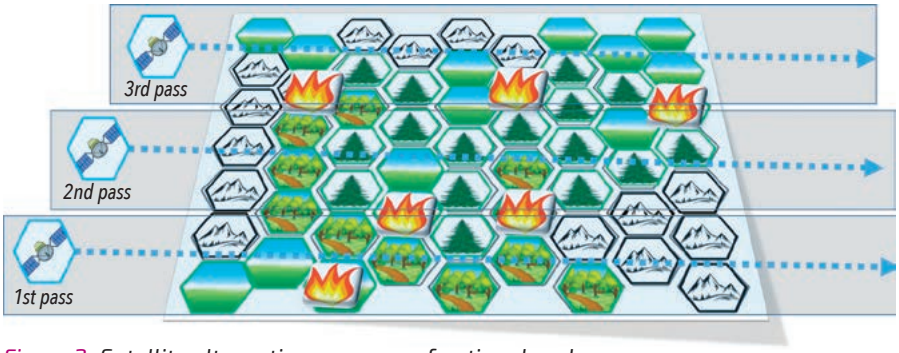


Figure 2. Satellite alternative coverage of national park

System Concept Alternatives

For this example, we have three separate alternatives. The first is a satellite-based system that can provide a frequent revisit surveillance capability on the park and has broad coverage. The satellite orbit is not easily or quickly changeable, so the park coverage is based on the orbit revisit rate. The satellite will not have great resolution into a specific area of the park. The satellite is (relative to other alternatives) very expensive to launch, operate, and maintain. This sensor is normally operated as a single system for the park example. Figure 2 provides an example of the satellite coverage.

The second alternative is an airborne system, such as a crewed aircraft or uncrewed aerial system (UAS). This provides a mobile surveillance capability that can be rapidly reassigned to different parts of the park and can provide a detailed view into specific areas. The airborne system is slower (relative to the satellite) and will take some time to build a complete picture of the park and the fire status. The airborne system is moderately expensive to operate and maintain. This sensor is normally operated in a single system configuration, with several other airborne systems available for maintaining a higher availability. Figure 3 provides an example of the airborne system coverage.

The third alternative is a series of ground-based sensors to provide a very detailed view of the ground conditions. This alternative is fixed (so cannot expand any coverage of the park on its own) and may be relocatable by personnel. These sensors are normally pre-established in some form of pattern across multiple geographically dispersed locations to provide some indications and warning of the fire condition. This sensor is the least expensive to operate and maintain. Figure 4 provides an example of the ground sensor coverage.

For the purpose of this example, we develop a set of utility functions for the three alternatives for specific criteria to consider. These are: wide area surveillance, detailed surveillance, relocatability, manpower needed for operations, cost,

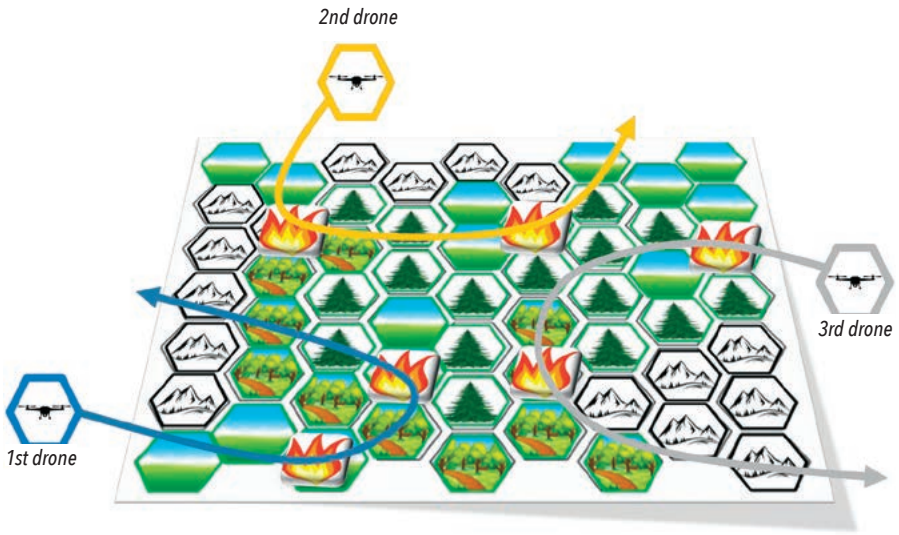


Figure 3. Airborne alternative coverage of national park

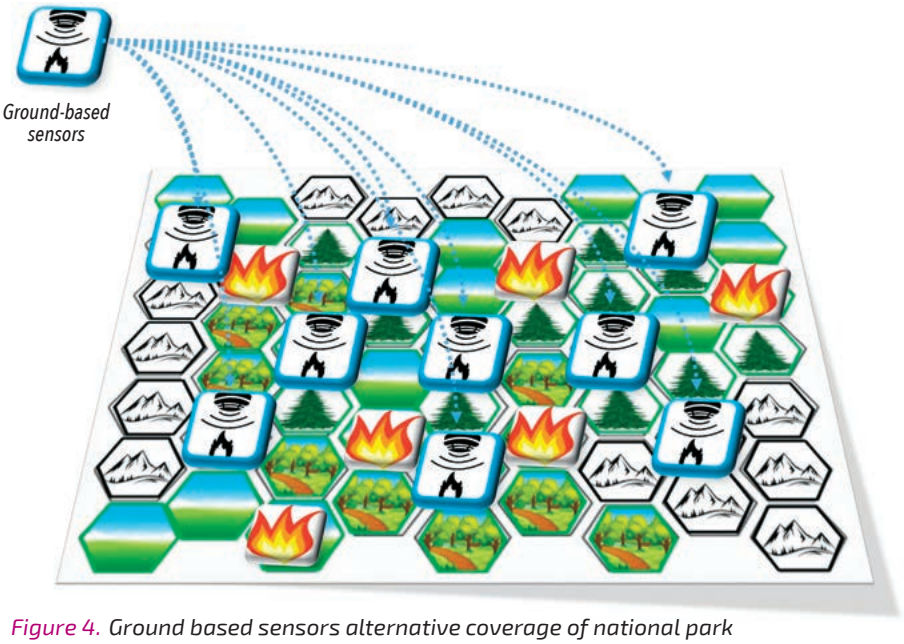


Figure 4. Ground based sensors alternative coverage of national park

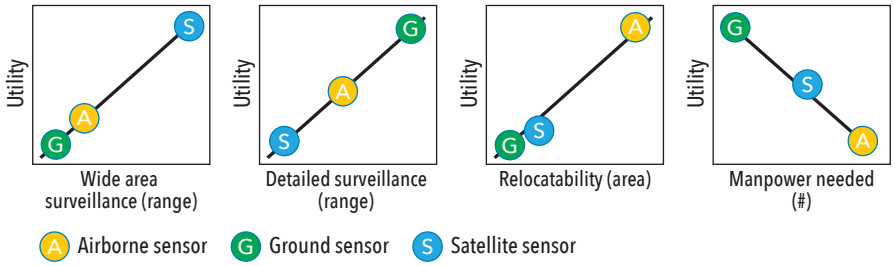


Figure 5. Fire sensing alternatives utility functions

and carbon emissions. Figure 5 provides an example of the first four of these utility functions and the systems.

Using these utility functions and analysis principles, we can develop notional performance values for each of the alternatives, assuming that each of the criteria are weighted equally. We will also utilize the calculations given in Mathers et al. for estimating the carbon costs of the alternatives, and satellite estimates from Segert (2021). We may be able to describe these alternatives to decision makers as found in Figure 6, with performance in the bottom right quadrant. Note that each of these values are the average of a high and low alternative. The other three plots look to evaluate the criteria of performance, cost, and carbon cost.

Alternatives Analysis

This can describe other areas to analyze the alternative solution space, as shown in Figure 7. The figure on the left can indicate the optimal path given a starting point (open triangle) and to visit each fire location, given the nearest location, and subsequently finding the next nearest location. If there are constraints on airborne system endurance, then a fleet of airborne systems would be required, as the figure on the right indicates to show 3 systems are needed (if 300 minutes are the limit for each system).

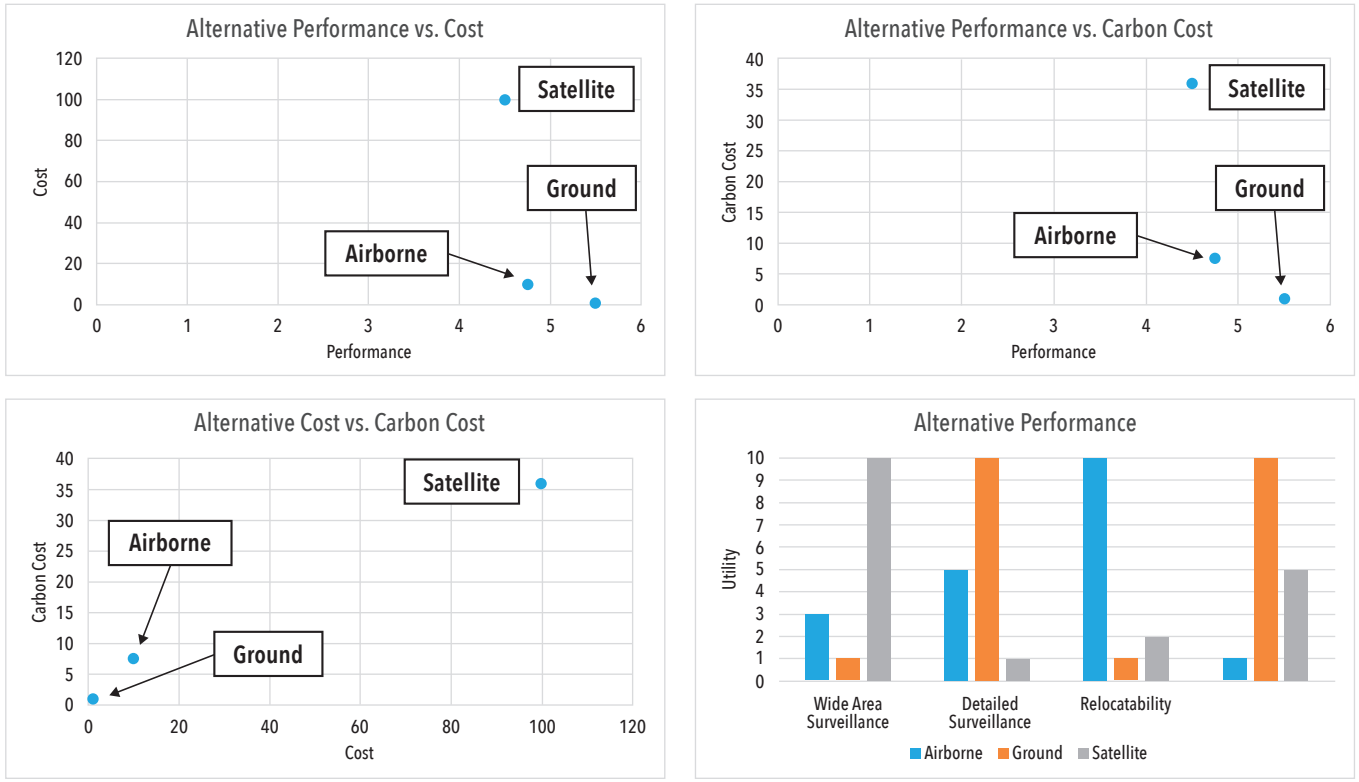


Figure 6. Alternative performance charts

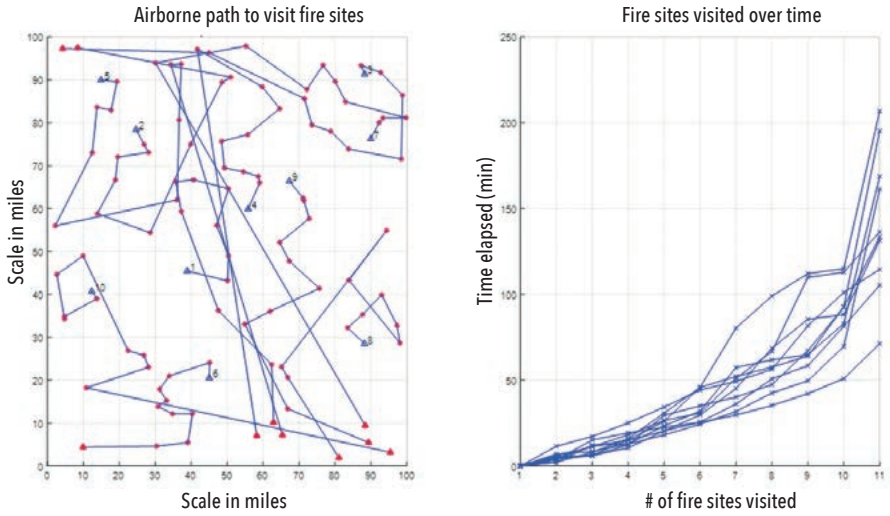


Figure 7. Airborne-based flight path and fire detections

During a trade study, we may consider different aircraft patterns (random vs. ladder-search pattern vs. other search patterns) to see if there are differences in fire detection performance, cost, and carbon cost. We may also consider varying the number of aircraft and then evaluate the metrics. Although we increase the number the aircraft to shorten the time to detect all fires, we expect to increase the cost and carbon costs, which would be a tradeoff by the stakeholders.

Ground sites are distributed throughout the park and have an effective range. If there are constraints on the number of ground sites, or variability in detection range, then this performance may change. See Figure 8 for an example visual of the fire locations and sensor layout, as well as the cumulative detections. This figure provides two examples of ground sensor deployment – the figure on the left is a fixed pattern, while the figure in the middle is a random pattern. During a trade study, we may consider different placements of the ground sites to include random, perimeter, or checkerboard patterns, and determine how well each of the options may vary in terms of fire detection performance, cost, and carbon cost. Like the aircraft example, the employment of the sensors would be a consideration in the tradeoff decisions by the stakeholders.

Satellites have an effective "block" or "swath" of sensor range. The number of fires can then be collected within each of the blocks, visualized in a horizontal orientation. In this example, Figure 9 shows a visual of the 3 horizontal blocks from 3 notional satellite passes, and the cumulative detections within each block. Like the other alternatives, we would expect the larger number of satellites would improve performance at the detriment of cost and carbon cost and would be a consideration in the tradeoff decisions by the stakeholders.

To help categorize the analysis of alternatives, we can consider a simple 8-unit cube. Depicted in Figure 10, the traditional metrics of performance and cost are on the horizontal axes, with carbon cost on

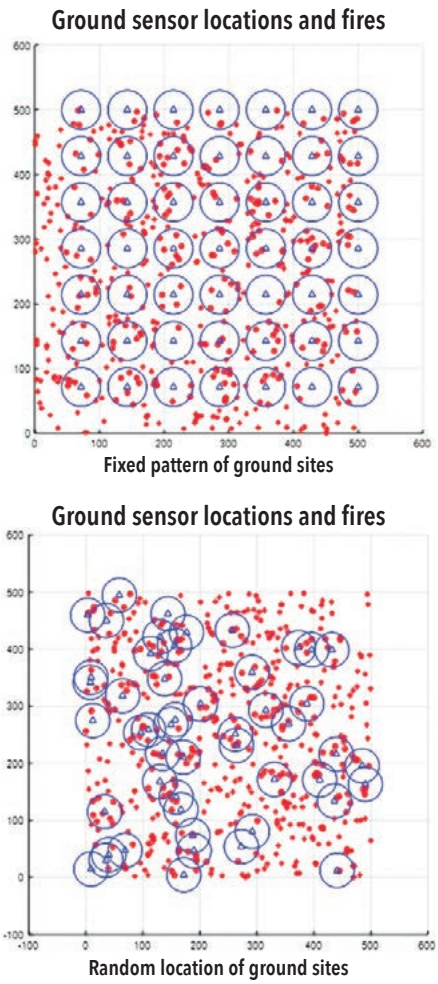


Figure 8. Ground-based sensor locations and fire detections

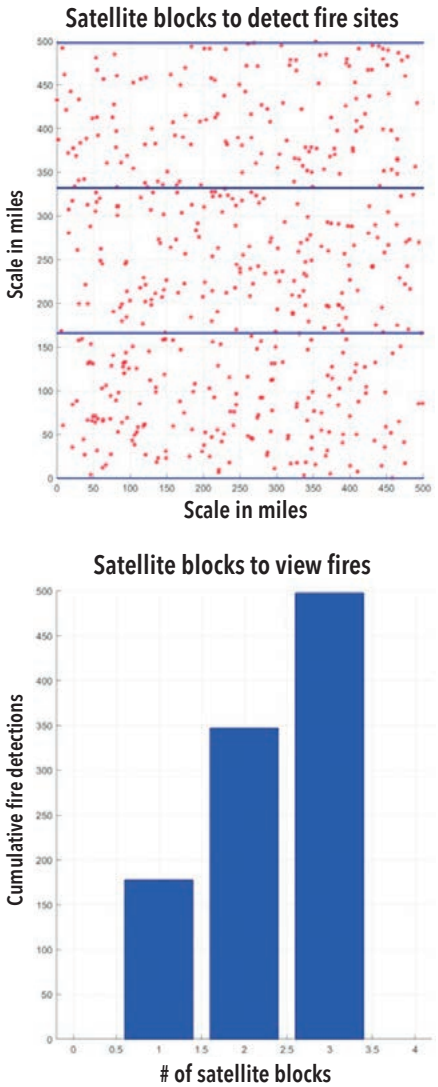


Figure 9. Satellite-based sensor blocks and fire detections

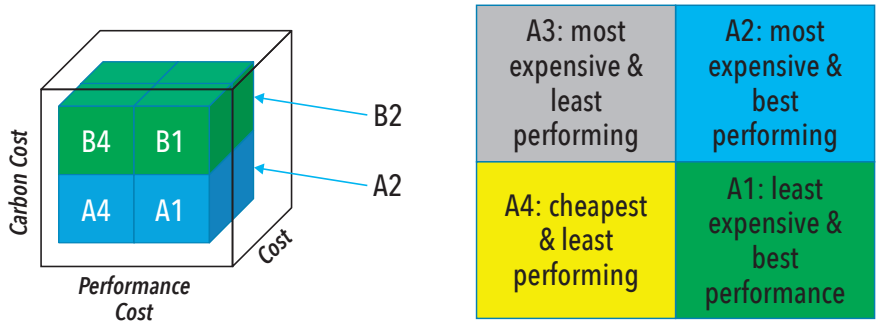


Figure 10. Quadrants in the carbon cube

the vertical. The goal being to develop the concept of use for each solution options such that they best system appear in the A1 quadrant, being the least expensive, best performing, and lowest carbon cost. The carbon costs are intended to be in two levels, the A-level (bottom) has the lower carbon cost, and the B-level (top) has high carbon values.

To analyze the alternatives, we do a simple experiment with varying the numbers of nodes for each solution class, and therefore their concept of use. For the aircraft we increase numbers from 1 to 10 to surveil the entire park. The cost and carbon costs increase linearly with each added aircraft, while the performance shows an increased value with each additional aircraft in the problem.

Figure 10 provides a view of the performance vs. cost vs. carbon cost with increasing aircraft. We can see that it shifts rapidly from the A4 quadrant to the A3 quadrant with increasing aircraft and could extrapolate based on the points that it would eventually reach the A2 quadrant with more aircraft. For the satellites, it rapidly shifts from the A4 to A2 quadrant, but at a high cost (vertical axis). For the ground sensors, it starts slowly in the A4 quadrant, and will eventually reach the A2 quadrant with a larger number of systems. Given this image of performance vs. cost vs. carbon costs, decision makers may opt for a mix of systems and their capabilities to reach the performance, cost, and carbon cost goals. By plotting the goals into the quadrants, as shown in Figure 11, we can identify where future improvements can be made.

SUMMARY

We can demonstrate from our notional example that analyzing the carbon cost due to the concept of use can play a significant impact in the selection of alternatives, and through the analysis of the problem, can identify where certain alternatives may or may not meet our respective goals. Prioritization of the criteria can be performance with decision analysis techniques such as analytic hierarchy process (AHP) or rank ordered centroid (ROC), with weighted sums to gather performance and charts to show cost-effectiveness (or now carbon-effectiveness or carbon-cost) comparisons can be made.

The solution classes and concept of use were captured as operational nodes and operational activities, respectively, in the MBCD schema. The employment of the MBCD schema has been kept at the simplistic level to demonstrate the viability of such approach in developing a conceptual design of the solution system. Complexity of the model-base representation should be increased in a full analysis of alternatives to explore the full mission, logistical support, and other such operational activities required to deploy a capability.

This simplistic study helps start the transition towards systems engineers better considering the impact of carbon emissions, and ultimately environmental impact in the design of new capabilities and delivering the environmental goals of Systems Engineering Vision 2035 (INCOSE 2022) that INCOSE, and society more broadly, desperately needs.

NEXT STEPS

We limited our research to explore the carbon cost during the use of the capability (Scope 2), and only for a small excerpt of a mission scenario. We recommended that the next steps should expand the scenario to be more holistic and representative and

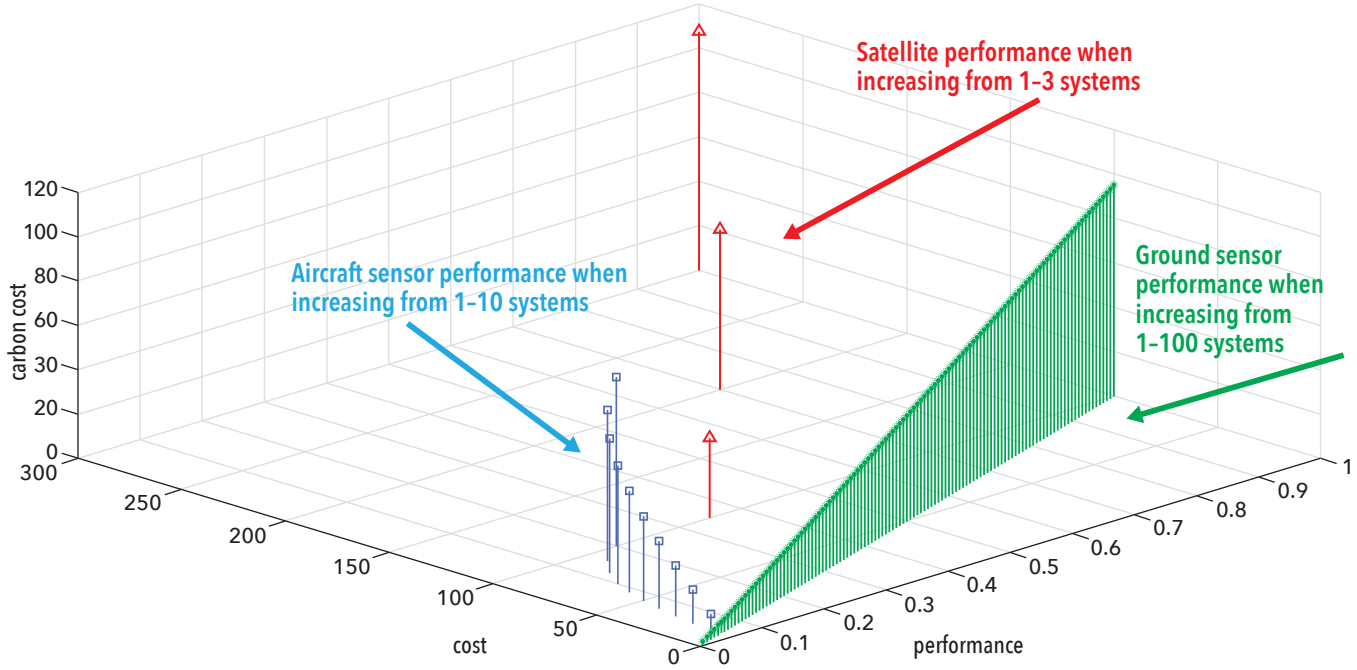


Figure 11. Example performance vs. cost vs. carbon cost plots

consider the carbon cost across the remain two emission scopes of development (Scope 3) and direct energy (Scope 2). This would then give a clearer, more definitive conclusions made during the concept design phase and the analysis of alternatives study.

As we've shown the first steps towards considering different criteria for consideration, we could also look at different fidelities of simulation to consider carbon costs more accurately, particularly as operations get more complex and detailed. We may also consider different parts of the lifecycle to evaluate the creation of the systems vs. system operations vs. system retirements. Another approach could be considering the recycling costs and impacts to the system performance.

The MBCD schema provided the framework to focus the approach of assessing the carbon cost of alternative solution options. We hypothesis that further research would increase clarity on how carbon cost could be better represented in the schema, beyond a measure associated with the operational node and activity in the schema, to aide decision makers. For example, the risk of environmental impact through carbon emissions could it be identified in the schema as a risk. Risk identified in the concept stage is traditionally only consider as technical, performance, cost, or schedule risk. Alternately, given the financial evaluation of carbon emissions becoming prevalent in society, the cost of carbon emission could form part of the mission costs. ■

REFERENCES

- Buede, Dennis M., and William D. Miller. 2016. *The Engineering Design of Systems: Models and Methods*. Hoboken, US-NJ: Wiley.
- Commonwealth of Australia. 2015. "Australia's 2030 Emissions Reduction Target" https://www.agriculture.gov.au/sites/default/files/documents/summary-australias-2030emissions-reduction-target.pdf, last accessed 19 September 2022.
- Department of Defense. 2021. "Statement by Secretary of Defense Lloyd J. Austin III on Tackling the Climate Crisis at Home and Abroad," 27 January. https://www.defense.gov/News/Releases/Release/Article/2484504/statement-by-secretaryof-defense-lloyd-j-austin-iii-on-tackling-the-climate-cr/, last accessed 19 September 2022.
- Department of Defense. 2021. "Department of Defense Climate Adaptation Plan", 1 September. https://www.sustainability.gov/pdfs/dod-2021-cap.pdf, last accessed 19 September 2022.
- Department of Defense. 2021. "Department of Defense Climate Risk Analysis", October. https://media.defense.gov/2021/Oct/21/2002877353/-1/-1/0/Dod-Climate-Riskanalysis-Final.pdf, last accessed 19 September 2022.
- Department of the Air Force. 2022. "Department of the Air Force Climate Action Plan." October. Washington, US-DC.
- Department of the Army, Office of the Assistant Secretary of the Army for Installations, Energy and Environment. 2022. "United States Army Climate Strategy." February. Washington, US-DC.
- Department of the Navy.n.d. "Climate Action 2030." https://www.navy.mil/Portals/1/Documents/Department%20of%20the%20Navy%20Climat e%20Action%202030.pdf, last accessed 19 September 2022.
- Department of Defense.2022. "Systems Engineering Guidebook." February.
- Flanigan, D. and K. Robinson. 2019. "Exploring the Test and Evaluation Space using Model Based Conceptual Design (MBCD) Techniques." In INCOSE International Symposium (Vol. 29, No. 1, pp. 1072-1083), Orlando, US-FL, 20-25 July.
- Flanigan, D. and K. Robinson. 2020. "Employing Model Based Conceptual Design to Identify Test Range Resources Required to Validate the Delivered Solution." In INCOSE International Symposium (Vol. 30, No. 1, pp. 65-77), Virtual Event, 18-23 July.
- Flanigan, D., and K. Robinson. 2021. Employing a Model Based Conceptual Design Approach to Design for Resilience. In INCOSE International Symposium (Vol. 31, No. 1, pp. 537-550), Virtual Event, 17-22 July.
- Flanigan, D., and K. Robinson. 2022. "Conceptual Design for Resilience." In INCOSE International Symposium (Vol. 32, No. 1, pp. 262-276), Detroit, US-MI, 25-30 June.
- Gupta, Udit, et al. 2022. "Chasing carbon: The elusive environmental footprint of computing." IEEE Micro 42.4: 37-47, Chicago, US-IL, 1-5 October.
- Henderson and Salado. 2020. "Value and Benefits of Model-Based Systems Engineering (MBSE)." *Systems Engineering* 24:51–66.
- Honour, E. 2011. "Sizing Systems Engineering Activities to Optimize Return on Investment." INCOSE International Symposium, Denver, US-CO, 20-23 June.
- INCOSE. 2022. *Systems Engineering Vision 2035*, https://www.incose.org/about-systems-engineering/se-vision-2035, last accessed 20 November 2022.
- INCOSE. n.d. "MBSE Initiative." https://www.incose.org/incose-member-resources/workinggroups/transformational/mbse-initiative, last accessed 28 November 2022.
- Kossiakoff, Alexander, et al. 2020. *Systems Engineering Principles and Practice* Third Edition. Hoboken, US-NJ: John Wiley & Sons.
- Mathers, Jason, et al. 2014. "The Green Freight Handbook." Environmental Defense Fund.
- Müller, Leonard Jan, et al. 2020. "A Guideline for Life Cycle Assessment of Carbon Capture and Utilization." *Frontiers in Energy Research* 8: 15.
- National Aeronautics and Space Administration. 2020. "NASA Systems Engineering Handbook." NASA SP-206-6105 Rev 2, 27 January.
- Robinson, K., M. Waite, and Q. Do. 2014. "Introduction to the Model-Based Conceptual Design." *INSIGHT* 17(4):7.
- Segurt, T. n.d. "The Carbon Footprint of a Satellite." https://www.linkedin.com/pulse/carbon-footprintsatellite-tom-segert, last accessed 20 November 2022.
- Sparrevik, M., and S. Utstøl. 2020. "Assessing Life Cycle Greenhouse Gas Emissions in the Norwegian Defence Sector for Climate Change Mitigation." *Journal of Cleaner Production* 248:119196. United Nations, "The 17 Goals." https://sdgs.un.org/goals, last accessed 20 November 2022.
- United Nations.n.d. "Sustainable Development Goals, Goal 13: Take Urgent Action to Combat Climate Change and its Impacts." https://www.un.org/sustainabledevelopment/climate-change/, last accessed 19 September 2022.
- US Air Force. 2017. "Analysis of Alternatives (AoA) Handbook." 4 August. https://afacpo.com/AQDocs/AoAHandbook.pdf, last accessed 19 September 2022.
- Walden, D. D., G. J. Roedler, and K. Forsberg. 2015. INCOSE Systems Engineering Handbook Version 4: Updating the Reference for Practitioners. In INCOSE International Symposium (Vol. 25, No. 1, pp. 678-686), Seattle, US-WA, 13-16 July.

> continued on page 50

Applying a System of Systems Perspective to Hyundai-Kia's Virtual Tire Development

Sunkil Yun, piesun@hyundai.com; **Shashank Alai,** shashank.alai@siemens.com; **Yongdae Kim,** ydkim200@hyundai.com; **Tae Kook Kim,** kook.kim@siemens.com; **Jaehun Jo,** jaehunjo@hyundai.com; **Dahyeon Lee,** dahyeon.lee@siemens.com; **Lokesh Gorantla,** lokesh.gorantla@siemens.com; and **Michael Baloh,** michael.baloh@siemens.com

ABSTRACT
Systems engineering has become important in almost every complex product manufacturing industry, especially automotive. Emerging trends like vehicle electrification and autonomous driving now pose a system of systems (SoS) engineering challenge to automotive OEMs. This paper presents a proof-of-concept (PoC) that applies a top-down SoS perspective to Hyundai-Kia Motor Corporation's (HKMC) virtual product development process to develop a performance-critical component of the vehicle, the tire. The PoC demonstrates using the Arcadia MBSE method to develop a consistent, layered, vehicle architecture model starting from the SoS operational context down to the lowest level of system decomposition in the physical architecture thereby capturing top-down knowledge traceability. Using the concept of functional chains, several vehicle performance views are captured that serve as the basis for architecture verification orchestration across engineering domains using a cross-domain orchestration platform thereby validating key vehicle/tire performance metrics that influence the tire design parameters. Preliminary results of the study show that applying a method-based modeling approach could provide several benefits to HKMC's current product development approach such as reduced time to model, SoS knowledge capture and reusability, parameter/requirement traceability, early performance verification, and effective systems engineering collaboration between the OEM, tire design supplier, and tire manufacturers.

INTRODUCTION

With recent developments in electric vehicle and driver assistance technologies, the automotive product development landscape has changed significantly over the past decade. The modern car as we know now is a highly complex system that comes with somewhere between 70 and 100 electronic control units (ECUs) that control most of the vehicle's functions and over 100 million lines of code that make up all the vehicle's software (Mihailovici 2021). An addition to this trend has been the dramatically changing context of the vehicle's operating environment. With increasing emphasis on limiting the environmental impact as well as enhanced driver safety, the need for ramping up the civil infrastructure to meet the vehicles' operational demands is evident. Smart energy grids, 5G-enabled communication networks, smart parking systems that will require vehicle-to-grid (V2G), vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technology interfaces are just few of the many operational environment constraints that will drive automotive product development in the coming decades (Varanasi 2022, Litman 2022, Bhatti et al. 2021). Compared to the traditional systems engineering viewpoint, where product development is seen as a system engineering problem, the approach must evolve where developing a vehicle must be seen as part of a bigger mission that consists of disparate systems that interact with each other to deliver a common mission or a capability. Particularly, developing an electric vehicle with advanced driver assistance systems can no longer be looked at as an isolated product development problem. Add to that the regulatory complexity that will arise from urban mobility systems which can easily render current product development approaches unscalable (Freemark et al. 2022, Eugensson et al. 2013).

Looking at the other side of the problem, the vehicle performance needs are equally, if not more important to provide a safe and sustainable quality product to future consumers. With changing user needs, every

new vehicle program presents numerous challenges such as managing significantly higher variations in the product lines, minimizing development risks and costs, reduced cycle times, to name a few, while ensuring that the product meets its optimum performance targets. In the case of autonomous driving, occupant safety is one of the most crucial drivers in technology development. To that effect, tire manufacturers are already analyzing the emergent impact of smart, intelligent tires to realizing safer autonomous mobility solutions with highly promising possibilities (Continental 2022). This implies that every changing user need may impact operational mission needs which cascade down to the individual subsystem and component requirements including their performance and design constraints. Not only systems, but the way their constituent subsystems and components are designed and managed in an enterprise needs to be more efficient to meet current product development timelines. Automotive manufacturers have become wary of this trend and have begun investing heavily into evolving their product development approaches, moving from a document-based to a model-based paradigm. HKMC acknowledges this trend and has committed to developing next-generation engineering and product data management environments for digital mobility transformation (Siemens 2021). From a system engineering standpoint, this clearly presents a 'system of systems (SoS)' challenge. According to the INCOSE SoS primer, a SoS is a collection of independent systems, integrated into a larger system that delivers unique capabilities. The independent constituent systems collaborate to produce global behavior that they cannot produce alone (INCOSE 2018). Following this definition, in this proof-of-concept (PoC), we apply a SoS perspective to HKMC's virtual tire development process. The scope of the project is to develop a purpose-built vehicle concept architecture that can:

1. Provide a descriptive reference of the SoS context, its constituent vehicle's functions, structure, and interfaces,
2. Enable early vehicle/tire performance verification based on predefined metrics and,
3. Enable system-to-subsystem collaboration with downstream subsystem and component designers/architects through a cross-domain collaboration platform.

Right from the beginning of architecture development, the SoS perspective is applied to the virtual tire development process such that the development of the independent constituent system, the purpose built vehicle (PBV), which is the electric vehicle is seen as a key contributor to achieving the higher capability of the SoS. Once the SoS problem domain is established, we present a methodological approach to develop the constituent PBV using a modular vehicle architecture concept, iteratively and recursively decomposing down to the tire component, the design of which is the goal of the study.

(a) (b)

Figure 1. Purpose built vehicle concept (Hyundai Motor Group TECH)

Purpose Built Vehicle (PBV)

Purpose built vehicle (PBV) is Hyundai Motor Group's eco-friendly, multi-purpose mobility vehicle solution that can be provided at low cost and is intended to meet customer's business purpose and needs (Hyundai 2021). It is a modular device with a simple structure whose design can be adapted to changing customer demands and business requirements. An important aspect of the PBV mission is to provide solutions to increase customer business value and maximize the efficiency of business operations. PBV devices can span from 3m to a maximum of 6m in length based on an expandable architecture (skateboard platform) and can respond quickly to various business and customer UXs in mobility, logistics, living space, etc. In addition, if combined with autonomous driving technology in the future, it can be used as a robot taxi or unmanned cargo transportation. Figure 1 (a) and (b) show examples of the PBV concept. Applying the MBSE approach to PBV development is expected to capture PBV knowledge in a layered modular architecture at different levels starting from the SoS context down to the physical component description, with traceability to different levels of requirements and parameters. The architecture shall include descriptive performance knowledge that captures key vehicle performance concerns such as ride and handling, durability, mileage, NVH (noise, vibration, and harshness) and low rolling resistance co-efficient. After several iterations of architecture definition and verification, the PBV architecture is expected to capture critical performance details including the functionality specification with performance metrics and optimum component design parameters along with requirement traceability. This knowledge once captured is expected to be reusable across projects and programs potentially leading to significant process improvements.

Virtual Tire Development

In response to growing environmental concerns, the United Nations Economic Commission for Europe (UNECE) enacted the worldwide harmonised light vehicles test procedure (WLTP), a vehicle performance measurement standard, and continues to discuss environmental and energy-related vehicle regulations. CO2 emission regulations greatly influence the overall automotive industry towards which it is developing sustainable solutions (UNECE 2014). It has been reported that a 10% reduction in a vehicle's rolling resistance can reduce CO2 emissions by 1.5-2.0% (Riemersma and Mock 2012) and provide about 1% improvement in fuel economy (Barrand and Bokar 2008). Consequently, automotive suppliers have been developing low rolling resistance tires as a practical solution to improving vehicle fuel efficiency and reducing emissions. Also, vehicle tire characteristics are known to have a strong correlation with the various vehicle performances as tires provide the four main contact points between the vehicle and the driving surface. When tires are developed to objectively reduce rolling resistance to improve fuel efficiency, other performances of the vehicle such as ride comfort, handling, and noise, tend to deteriorate, especially for EVs, making it harder to deliver a quality product to the market.

Conventional tire development processes pose several limitations to developing newer vehicles like electric PBVs. For instance, batteries, an essential element of electric PBVs, increase the overall weight of the vehicle, and the electric motors generate high torque. This poses significant challenges to electric PBV tire development to satisfy the target performance requirements in harsh driving

environments relative to tires designed for non-EVs. In the case of autonomous PBV, ride comfort performance needs may outweigh low tire rolling resistance needs. In other words, there are characteristics of newer electric vehicle parts and customer requirements that may go against the fuel efficiency targets (Koengkan et al. 2022, Weiss et al. 2020).

The traditional tire development process at HKMC was not suitable to address the modern PBV development challenges. Tire performance is optimized by focusing on vehicle-level performance targets. However, the multi-attribute performance goals cannot be satisfied simultaneously as they conflict with each other at the tire level. In the existing development process, tire performance development is performed mainly on real tires which makes it difficult to harmonize performance targets at the tire level and the vehicle level in the early development stages. In addition, tire is currently modeled as a black-box item at late development stages which is reused from previous vehicle programs resulting in a need to frontload tire engineering for new PBV during early vehicle performance development. To address these needs, a virtual tire development process using a virtual tire model has been established, that focuses on:

1. Adjusting individual performance targets at vehicle level
2. Dividing vehicle-level individual performance targets into tire level
3. Coordinating individual performance targets at the tire level.

As shown in Figure 2, in Phase 1, the vehicle level performance requirements (targets) defined during Architecture Gate 1 (AG1) are divided down to the tire level performance targets in Architecture Gate 2 (AG2). Virtual tire models (MF, MF-Swift, etc.) are used to determine optimum specification that can maximize the individual performances of tires in AG2. In the case of a tire, a particularly weak performance of tire is identified based on trade-offs. A standard tire specification is selected. Suppliers design tires that maximize individual performances based on the standard specifications. In this process, the functional tire characteristics (FTC) of the tire are evaluated using a virtual tire model based on the tire design parameter (TDP), and the harmonization performance of the tire is verified. By repeating this process in Architecture Gate 3 (AG3), a tire model that lacks performance compared to the standard specification is defined and vehicle level performance is verified. Finally, the real tire design is derived in Architecture Gate 4 (AG4). Currently, this architecture development approach to virtual tires uses multiphysics tire models that are developed in silos and are usually disconnected. Moreover, each individual performance engineer usually communicates separately with the tire simulation/design engineers that are usually from HKMC partner companies leading to inefficiencies in the virtual tire development process. There is an identified need for a robust and secure approach to enable efficient communication among development teams. A descriptive vehicle architecture that captures a consistent representation of the vehicle performances can provide a common source of vehicle performance knowledge that can be shared among the vehicle performance development teams, tire designers and eventually manufacturers in the later stages of the development life cycle that are usually located across different organizations. The paper describes an approach at HKMC that attempts to address these challenges.

The remainder of the paper is organized as follows: The second section provides a review of the literature around MBSE and SoS followed by an overview of the MBSE approach applied in this PoC study

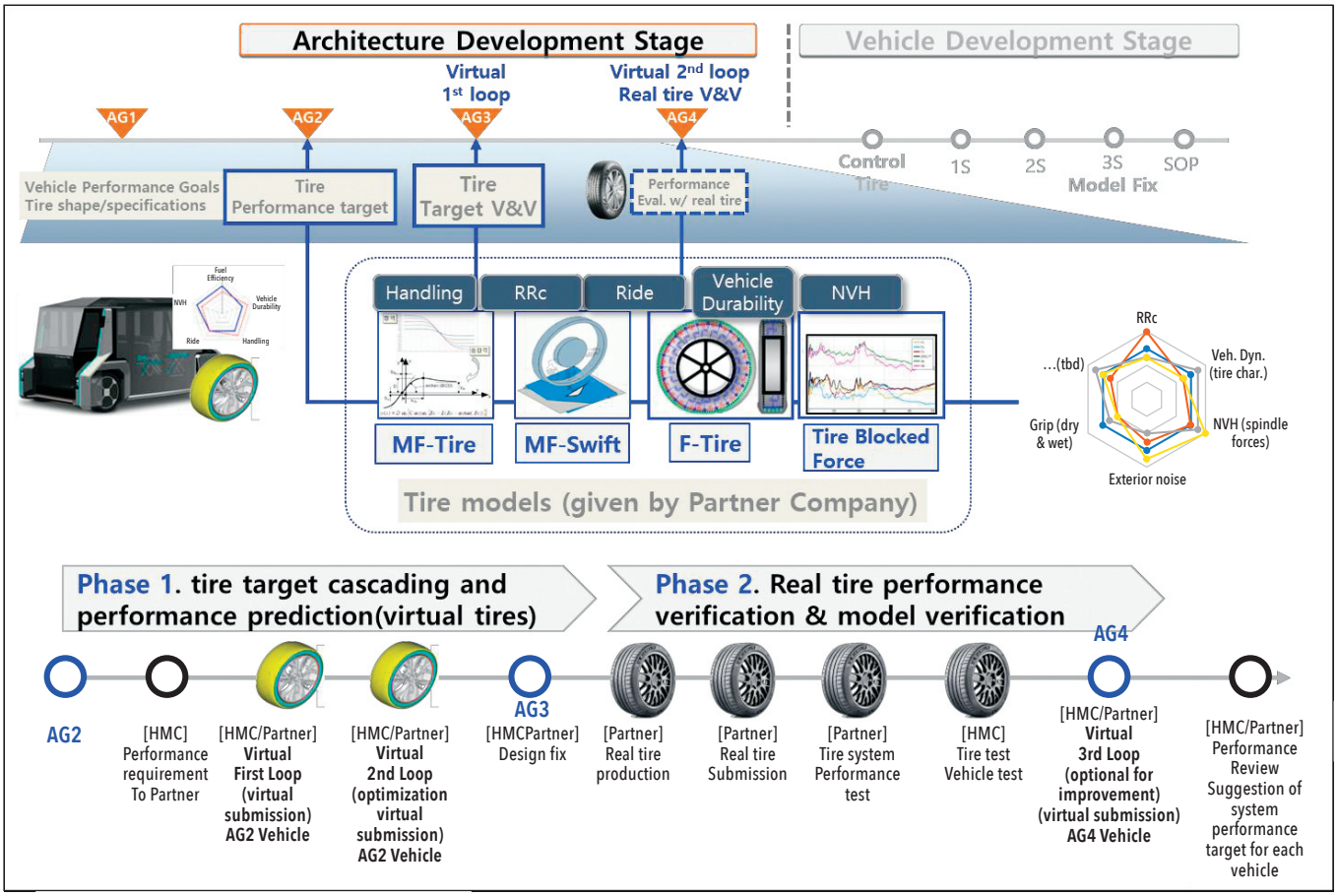


Figure 2. Virtual tire development process at HKMC (PBV 이미지 교체)

in the third section. The results of the MBSE approach applied to virtual tire development are presented in the fourth section followed by conclusion and future scope discussion in the final section.

LITERATURE REVIEW

MBSE in Automotive

The INCOSE Automotive Systems Engineering Vision 2025 describes seven engineering challenges the automotive industry is currently facing and clearly signals the need of MBSE modeling as one of the many actions to realize the 2025 vision (INCOSE 2020). MBSE has become widely accepted across various product manufacturing industries and the acceptance is only growing. This has certainly contributed to the increasing literature available on the topic. Previous studies have suggested usage and implications of MBSE in the automotive industry. An overview of general challenges of implementing systems engineering and particularly MBSE across the automotive industry is provided where the authors describe the growing complexity of modern automobile systems as one of the major challenges for OEMs that pose the need for a model-based systems engineering approach. A system architecture model can serve a key role in managing risk and complexity by capturing various stakeholder concerns in a descriptive source while connecting that with broader engineering teams across the enterprise. In addition to describing the benefits of implementing an MBSE approach to automotive development, the authors also suggest that the overall complexity of modern automotive systems will soon require applying MBSE to the system of systems problem (Ambrosio and Soremekun 2017). The consistent academic and industrial pursuit in studying the breadth and depth of MBSE application across different stages of the system development life cycle has given insights not only into its potential for enterprise-wide benefits but also the increasing possibilities of using models as true sources of knowledge and decision-making. MBSE has been shown to incorporate product line engineering (PLE) through modeling variability in SysML models (Young et al. 2017). The authors present cases of feature based MBSE across three industries, where the automotive company uses a combination of MBSE and features models to manage the complexity arising from numerous product variations in today's vehicles. One of the main areas where MBSE is also seen as beneficial is the ability to manage simulations for multi-level requirements verification during the early concept stage. A descriptive system architecture can serve as a useful reference point to initiate multiphysics simulations of various subsystems and components to verify system-level performance requirements (Sohier et al. 2021, Nowodzienski and Navas 2022).

Not only OEMs, but also the global automotive supply chain has embraced the use of MBSE in transforming their heavily document-centric processes. In one such case, an automotive supplier for intelligent driver assistance systems has shown the use of a consistent system architecture model developed using the Arcadia approach across two operational projects with the objectives of improving efficiency and value for their customers, reducing development costs and schedule, fostering, and securing collaborative work and mastering complexity (Continental 2017). Their pilot MBSE implementations provided key insights into the benefits of using an MBSE approach, mainly better and effective communication with various stakeholders throughout the project's life cycle and have resulted in a wider acceptance across the enterprise followed by increasing numbers of the projects planning or already having deployed MBSE capabilities. Interestingly, the same group of supplier companies is one of the major vehicle tire suppliers to HKMC, which is the system component-of interest for this study.

System of Systems

System of systems engineering has been recognized as an MBSE focus-point in aerospace and defense industries based on the numerous references in systems engineering literature (Jamshidi 2008, DoD 2008). Great emphasis is placed on SoS as an emerging field in systems engineering necessary to respond to changing global contexts (INCOSE 2015). The 'ISO/IEC/IEEE 21839:2019 Systems and software engineering' standard provides critical SoS considerations that apply to a system that is a constituent system-of-interest (SoI) within an SoS, and that must be addressed at the key points in the life cycle of the SoI. These considerations can apply to man-made SoS whose constituent elements must include one or more of the following: hardware, software, humans, procedures, and facilities (ISO 2019). The systems in SoS have operational and managerial independence (Dahmann and Henshaw 2016) which make them very relevant in automotive companies that are expected to react at a rapid rate to the changing market needs while adhering to the requirements posed by future mobility SoS. Considering the modern automobile systems as constituent systems of a larger SoS, authors have provided recommendations on addressing automotive challenges by applying the SoS approach from an automotive OEM's perspective with limited authority over the other constituent systems forming the automotive SoS (Hoehne and Rushton 2018). Particularly, a modular open systems approach to automotive SoS is presented based on an existing transportation SoS framework of interoperability standards. The approach includes three key steps in modularizing any automotive SoS: *defining the technical modules* (infrastructure, energy, rolling stock and command, control and signaling) that the constituent vehicle directly interfaces with, *identifying the key vehicle interfaces* with the modules, and *specifying the interface requirements* which are further broken down into mechanical, electrical, communication, electromagnetic compatibility (EMC), and other applicable interfaces. Once modularized, these constituent systems can be developed and evolved independently without a common managerial authority. The technologies and teaming evaluation (TATE) framework is another example of a combined top-down/bottom-up approach to SoS comprising of manned and unmanned vehicles (Peters et al. 2018). The authors propose a modular and flexible approach to synthesizing and quantitatively evaluating configuration options within a SoS mission which can be further extended with various analyses that can aid in informing SoS and system requirements. Although SoS engineering and architecting is a challenging task because of the sheer complexity and scale of SoS, some attempts have been made to explore model-based methodologies and approaches to SoS architecting. The I^5 framework (interoperability, interconnectivity, interfacing, integration, and interaction) is one such approach that describes a model-based framework to design complex interactions among disparate systems, using object-process methodology (OPM) (Mordecai and Dori 2013). The framework provides an integration-centric perspective to SoS integration programs. OPM provides textual and graphical formalism to support the unique aspects in modeling and integration, such as capturing emergent properties and behaviors and a top-to-bottom hierarchy of interaction aspects among the constituent systems.

METHODOLOGY

Making the Case for EV Mobility SoS

In this PoC, the electric vehicle (EV) mobility SoS is defined as the operational context that drives the PBV development. The main constituent systems that provide EV mobility capabilities are the 'purpose built electric vehicle (PBV)', its interfacing systems such as 'smart energy grid' that includes smart charging stations, 'OEM' and 'eMobility service provider' that provides the charging station network and software

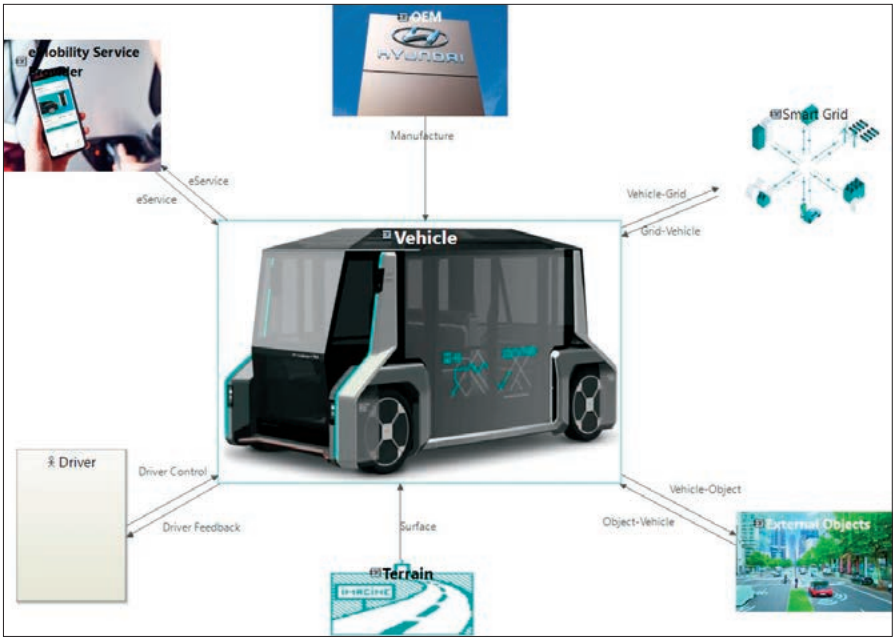


Figure 3. EV mobility SoS: operational context in Arcadia (PBV 이미지 교체)

services as shown in Figure 3.

Each of these systems are defined as modules that serve individual purpose(s) towards delivering the EV mobility mission. Because of the complexity and regulatory challenges noted in the introduction, HKMC believes that the most effective way to address the EV mobility challenge is by implementing a modular SoS approach that is supported with a robust underlying MBSE method. Such an approach will provide a strong MBSE foundation to the future mobility blueprint initiative laid out by HKMC (Hyundai 2022). It is therefore important to justify the case for EV mobility as an SoS problem. To do so, we look at the five main characteristics of SoS as defined by Maier (1998):

1. **Operational independence**. Each constituent system in an SoS must be able to operate independently of the SoS and the other systems. While the electric vehicles rely on external charging networks and services for high mileage, their operational independence does not require new development of every constituent system which may already exist and can rather be called upon to support a new capability.
2. **Managerial independence**. In addition, each constituent system in SoS is managed independently. With growing demands for EV technology and emerging interoperability standards, products must evolve rapidly to respond to these changing needs. The OEMs for EVs will continue to independently develop and evolve products while adhering to the SoS operational constraints whereas the energy management grids will evolve to respond to growing energy efficiency requirements.
3. **Geographical distribution**. SoS consist of geographically distributed systems. The EV, charging systems, smart grids, and mobility systems are geographically distributed.
4. **Evolutionary development processes**. SoS development is incremental as the constituent systems develop and evolve incrementally and/or asynchronously. The evolution of the EV mobility SoS is derived from the evolution of EV technology which may be more frequent in delivering constituent system capabilities.
5. **Emergent behavior**. SoS exhibit emergent behavior that results from the relationships and interactions between the constituent systems. In EV mobility SoS, each constituent system has a well-defined purpose. The EV must provide a transportation vehicle powered by electricity, the smart grid provides efficient energy generation and distribution to the electrical charging stations while the mobility services provide vehicle connectivity to communication networks for driver assistance, all leading towards the emergent behavior of reducing global carbon emissions.

MBSE Process Orchestration

This section describes the model-based systems engineering (MBSE) approach applied to the virtual tire development process at HKMC, which includes three major elements as shown in Figure 4:

1. **System Architecture Authoring**. Based on the requirements captured in a requirements repository, a concept architecture is used to describe the SoS operational context and the PBV architecture. The vehicle architecture modeling is significantly influenced by the vehicle's functional and non-functional constraints posed by the vehicle requirements which are then associated to the architecture elements. System Modeling Workbench (SMW) for Teamcenter® is used as the primary architecture authoring tool in this study. SMW is an integrated systems modeling environment that is used to apply MBSE concepts to the architecture development process using the Arcadia method.

Architecture analysis and design integrated approach (ARCADIA) is a system and software architecture engineering method, based on architecture-centric and model-driven

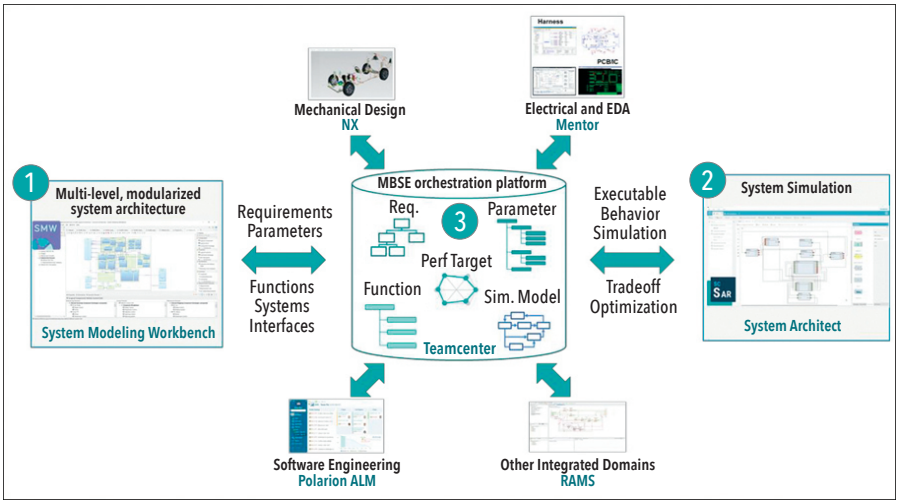
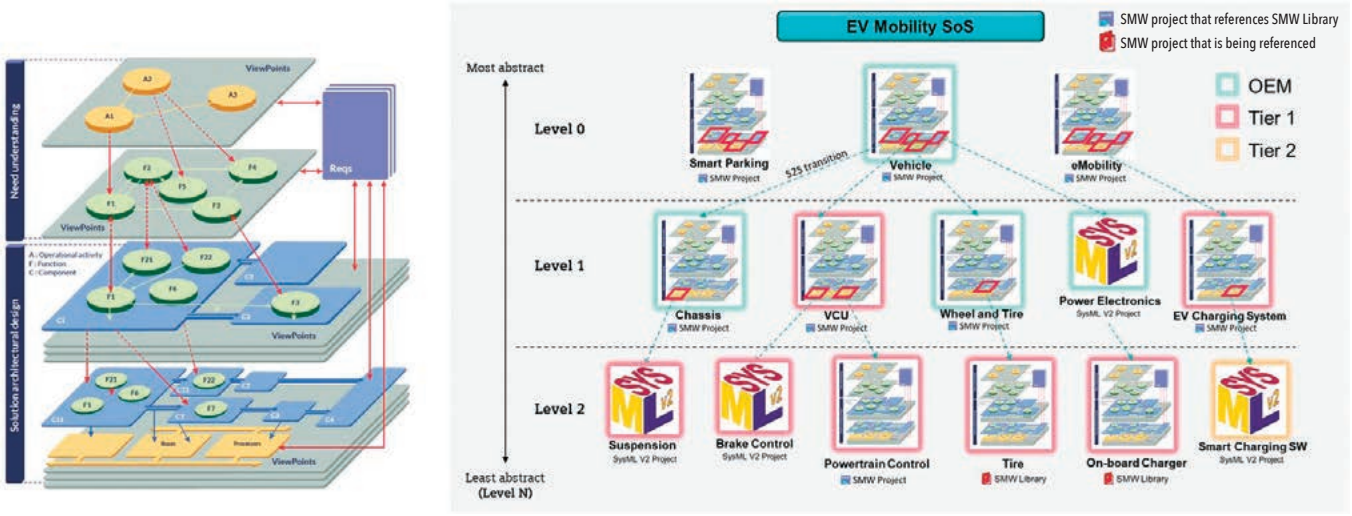


Figure 4. MBSE process elements



(a) The Arcadia method

(b) SoS modularization and classification

Figure 5. Applying the Arcadia method to EV mobility SoS

engineering activities (Voirin 2017). The Arcadia method shown in Figure 5 (a) provides a layered approach to modeling multiple levels of vehicle architecture at varying layers of abstraction, mainly *operational needs analysis (OA), system analysis (SA), logical architecture (LA) and physical architecture (PA)*, which is then orchestrated across engineering domains through an MBSE orchestration platform. Following the rationale for SoS, the Arcadia method is applied to model a multi-layered modular electric vehicle architecture that operates in the SoS context. Figure 5 (b) shows the EV Mobility SoS modularization concept that focuses on defining the SoS modules in Arcadia while independently managing their lifecycles through the orchestration platform. For this study, the focus is to define only the 'vehicle', 'wheel and tire' and the 'tire' modules, whereas the definition and management of other modules across all the levels is not part of the scope. The approach also takes into consideration the future of system modeling languages, including the upcoming SysML V2 specification (Bajaj et al. 2022), exploring possibilities of interoperability between Arcadia models and SysML V2 specification models, which is currently supported in SMW (Beta).

2. **System Simulation**. A multidisciplinary system simulation software is used to perform vehicle-level performance synthesis. This provides a means to verify the vehicle's key performance metrics early in the concept stage. The performance simulation is driven by metrics defined in the concept architecture to evaluate the vehicle dynamics and provide the best set of tire design parameter values which are then captured as configurable objects in a common database. Simcenter System Architect® is used as the multiphysics simulation software to analyze the vehicle performances.
3. **MBSE Orchestration**. Siemens Teamcenter® is used as the cross-domain platform in this study that enables the orchestration of the overall MBSE business process. In addition to providing model lifecycle management capabilities, the platform also enables managing granular architecture and simulation data and model files, parameters, requirements and creating cross-domain verification requests and workflows to facilitate multidisciplinary analysis and optimization. The MBSE orchestration platform provides the ability to share architecture models as whole and in parts with various stakeholders with granular traceability to numerous model elements, which is key to enabling the model-based workflows. The platform also provides the primary requirement and parameter authoring capabilities and traceability to the concept architecture elements which enables granular cross-probing across the system's RFLP definition, also called the *integrated system definition*.

MBSE Approach Applied to HKMC's Virtual Tire Development

This section describes the MBSE approach applied to HKMC's virtual tire development process as overlaid on the left-side of the V-diagram in Figure 6:

1. **Requirements and Metrics Definition**. As described before, the MBSE orchestration platform is used to author various levels of requirements starting from the SoS stakeholder needs all the way down to component design requirements. The SoS operational context captures the

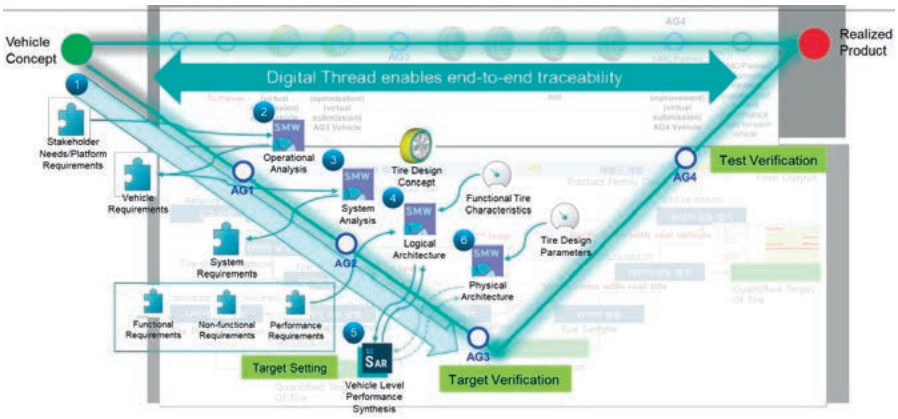


Figure 6. MBSE approach applied to HKMC's virtual tire development

numerous constituent systems that have their own sets of requirement specification that may be captured in the same environment depending on the 'managing authority' for those systems. Along with requirements, various levels of metrics and design constraints such as the FTC/TDP are captured in the same repository. The requirements and parameters are configurable objects that are linked to the various layers of the architecture model elements that are also shared across various domain assets managed in the cross-domain platform. This step usually starts early in the SoS concept exploration phases and is carried throughout the PBV concept development stage. In this study, the main drivers were the high-level stakeholder needs and platform requirements for the PBV.

2. **SoS Context and Operational Analysis (OA)**. The Arcadia OA layer is used to model the operational context of the SoS, capture the stakeholders, describe the SoS capabilities and high-level operational scenarios. The operational analysis results in identification of artifacts such as capability objectives, concept of operations (ConOps), and refined vehicle requirements specification that capture the stakeholder expectations. In other words, the vehicle requirements represent what the stakeholders expect from the 'to-be developed' system. This step is executed before AG1 in the virtual development process.
3. **System Analysis (SA)**. The Arcadia SA layer follows AG1 and is used to establish the PBV context as the SoI and define the functional dataflows and behavior along with the system boundaries. We modeled the functionalities of the performances using '*functional chains*' which describe the functional flows required to achieve desired capabilities. Functional analysis is the iterative and recursive process of identifying the functions that a system must perform to achieve the desired behavior, decomposing the system-level functions into their lowest level and defining relationships between the functions (Voirin 2017, Kossiakoff et al. 2020). The system functional analysis, executed before AG2, results in refined system functional and non-functional requirements that are captured in Teamcenter.
4. **Logical Architecture (LA)**. The Arcadia LA layer allowed us to capture the vehicle's logical systems and their interfaces that will deliver the required functional behavior, simultaneously capturing the tire design concept as parametric requirements in Teamcenter. The vehicle and tire-level FTCs are refined and scoped to the architecture project for the study and associated to the architecture elements. The *functional chains* enable creating functionality-specific views of the logical architecture that are useful to communicate the system architect's functional intent with the simulation engineers along with the associated vehicle requirements. This step is followed by AG2 and AG3 for multiple virtual verification loops.
5. **Architecture Verification.** At this point, early vehicle performance verification is initiated from the logical architecture where the FTCs for each performance area are linked. The performance of the vehicle is verified using multiphysics simulation by qualifying the FTC targets and a set of optimum TDPs is selected. The verification is orchestrated with the help of the Verification Request feature in Teamcenter that enables capturing all the relevant datasets pertaining to the simulation, including architecture model views and granular model objects, simulation parameters and simulation models conveniently packaged at the simulation engineer's disposal. The simulation engineer performs vehicle performance simulation to evaluate the FTCs and provides the TDPs in response to the verification request.
6. **Physical Architecture (PA)**. The Arcadia PA layer is used to identify and select technology choices for the logical subsystems and components, which can then be used to relate the design parameters such as the TDPs. The transition between the logical and physical layers is performed simultaneously with the orchestration of the verification request. For a smoother collaboration between the system architects, subsystem architects and designers, Arcadia 'system-to-subsystem transition' was used to demonstrate a case of carving out a part of the vehicle's logical architecture to be shared with the simulation engineer/designer such that the designer can access only the part of the architecture to which the design parameters must be associated. This facilitates robust and secure data sharing among participants in the systems engineering workflow. To the effect of establishing end-to-end traceability, the *operational processes* defined in the OA layer are traced to the *functional chains* in the SA, LA and finally the PA layer that capture the physical design constraints. Such traceability can enable analyzing the impact of changing SoS operational needs to the physical design details of its constituent system modules for a faster development response.

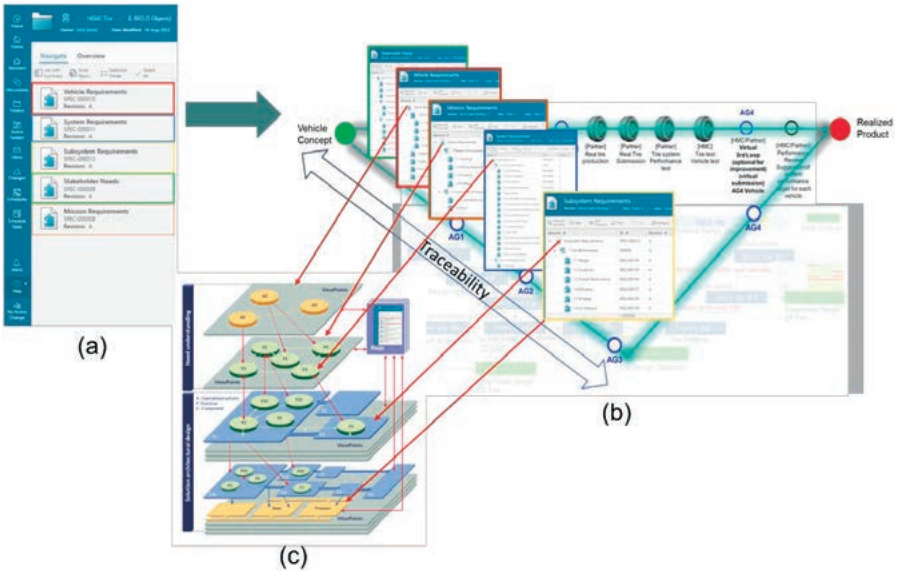


Figure 7. Requirements definition throughout the virtual tire development process

RESULTS

Requirements and Metrics Definition

Figure 7 (a) shows the various layers of requirement specification captured in Teamcenter®. Starting with the high-level needs, the SoS needs were captured in the form of need statements into configurable

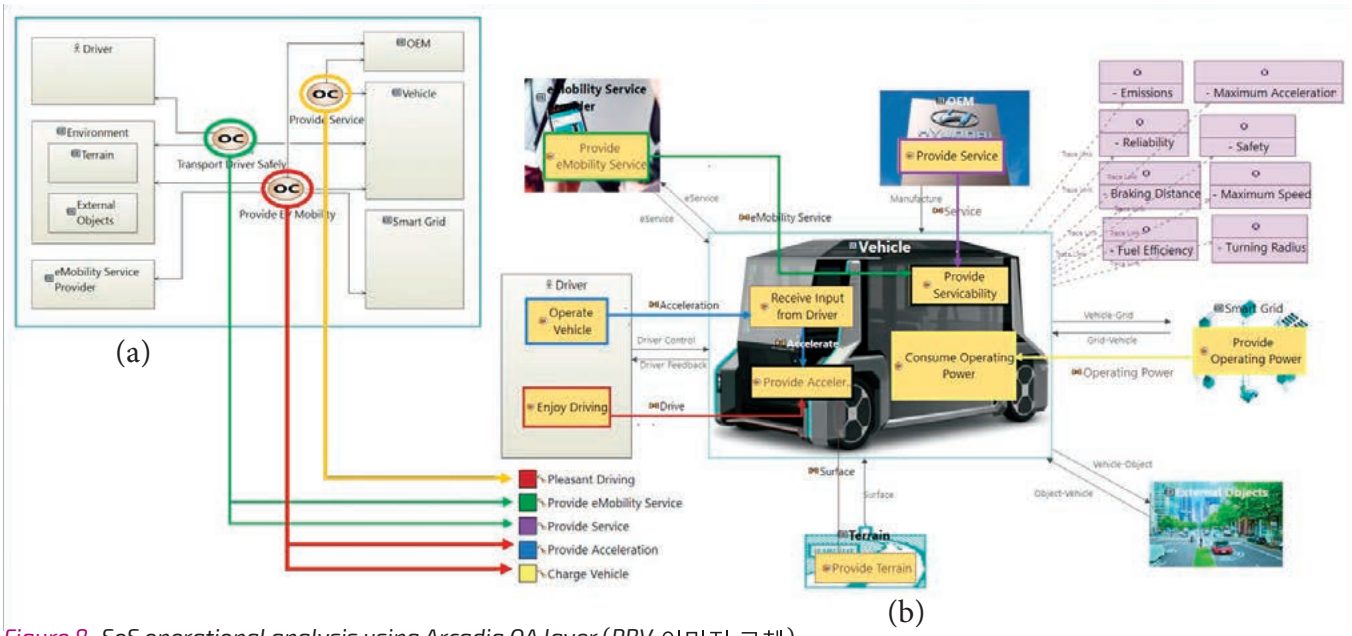


Figure 8. SoS operational analysis using Arcadia OA layer (PBV 이미지 교체)

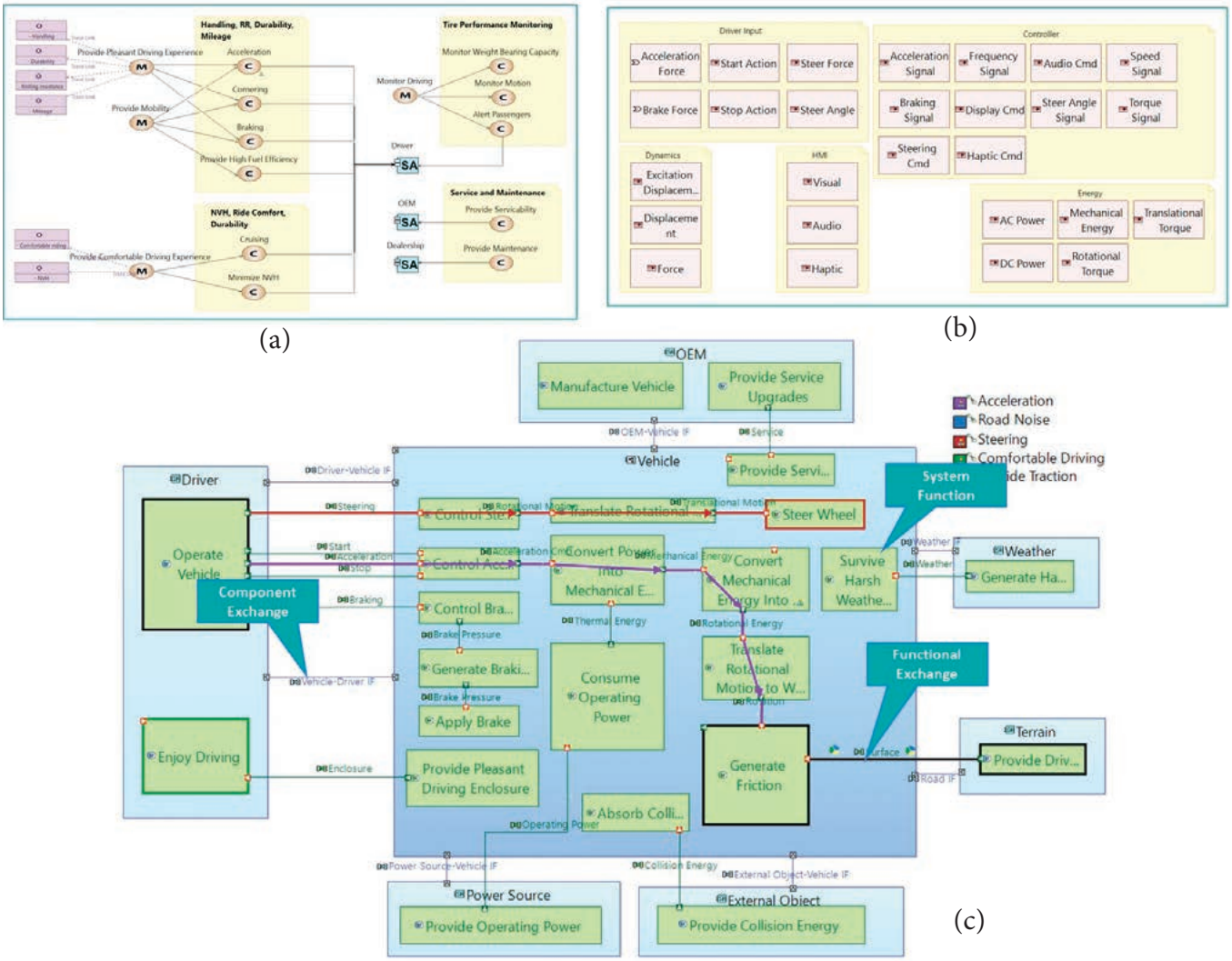


Figure 9. System mission and context analysis using Arcadia SA layer

requirement objects in the stakeholder requirements specification. The mission requirement specification captures the PBV mission in the context of the SoS thereby leading to operational analysis. The operational analysis resulted in clearly eliciting the vehicle requirements specification that captures the to-be developed PBV concept, including key measures of performance (MoP) such as *maximum acceleration, maximum speed, turning radius, braking distance*. This provided a basis to perform system analysis where vehicle was contextualized as a black-box entity, resulting in blackbox functional analysis to identify the key system-level functional (and non-functional) needs that are then captured in the system requirements specification. The system requirements specification includes requirements that also capture the key performance metrics (FTCs) that are eventually associated to the logical architecture view describing the vehicle performance functionalities namely, *ride comfort, handling, NVH, durability and rolling resistance*. The logical and physical architecture layers result in the subsystem requirements specification that capture the design requirements for the subsystems. For this PoC, the focus was on eliciting only the tire design requirements. Figure 7 (b) shows the requirements specification flow down along the HKMC virtual tire development process between the architecture gates and Figure 7 (c) shows the Arcadia method and its different modeling layers at which the requirements are linked to the architecture model elements. As described previously, the requirements definition process is applied throughout the modeling activity to achieve requirements flow down, and traceability is established with the model at desired levels.

SoS Context and Operational Analysi

Figure 8 (a) shows the *operational capabilities blank* diagram that captures the operational capabilities of the SoS, mainly 'provide EV mobility' and 'transport driver safely', which includes sub capabilities such as 'provide acceleration', 'provide pleasant driving experience', 'provide comfortable driving experience' that are hidden for simplicity. One of the main benefits realized during the architecture development in Arcadia was the ease of creating multiple views to represent several aspects of the same integrated model. Figure 8 (b) shows the *operational architecture blank* diagram that describes the operational context of the SoS. The operational architecture shows the allocation of a selected set of *operational activities* to the constituent *entities* and *actors* of the SoS and their *activity interactions* that capture the high-level dataflow between the activities. This view provided a high-level understanding of the operations that the SoS stakeholders expect from the constituent systems and actors to achieve the SoS capabilities. It also shows the *operational processes* 'charge vehicle', 'provide acceleration', 'pleasant driving', 'provide service' that describe the operational behavior required to achieve the SoS capabilities. As a result of the operational analysis, the PBV requirements were captured to specify the vehicle concept and traced to the operational architecture.

System Analysis

Figure 9 (a) shows the *mission capabilities blank* diagram that describes a subset of the vehicle's *mission* and the desired *system capabilities*. The PBV is expected to provide *system capabilities* that support the *missions* 'provide mobility,' 'provide pleasant driving experience' and 'provide comfortable driving experience' that realize the *operational capability* 'transport driver safely.' The *system architecture blank* diagram (also called system context/black box diagram) is shown in Figure 9 (c) which shows the primary functional allocation to the system and the system actors mainly, 'driver,' 'charging station,' 'weather,' 'external object,' and 'terrain.'

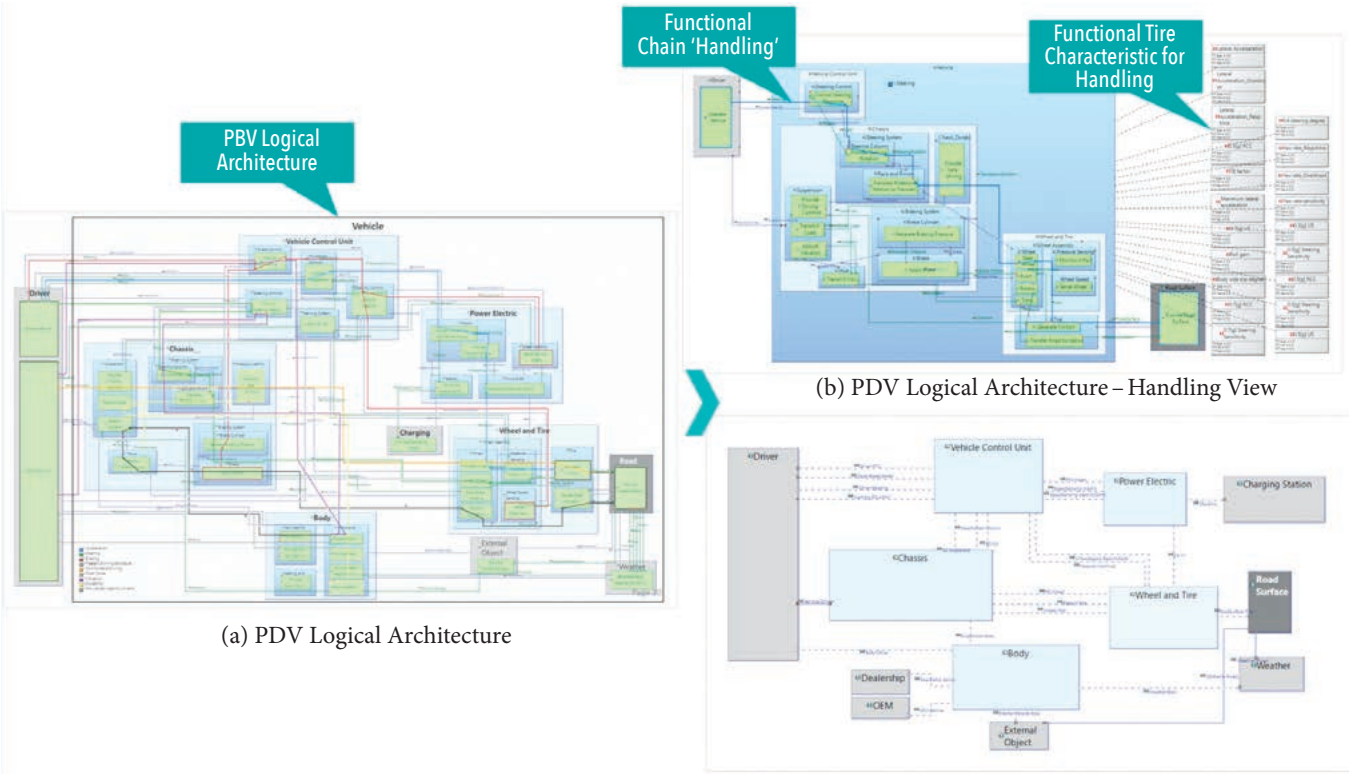
The system architecture shows an integrated view of the *functional interactions* and the system's external interfaces that carry the functional dataflow shown in Figure 9 (b), which consists of matter, energy and/or data. One of the main purposes of the modeling activity was to capture the vehicle behavior that affects the performances: *handling, rolling resistance, durability, NVH* and *ride comfort*, which are represented using the color ed *functional chains* as shown in Figure 9 (c). Other diagrams for system analysis that were developed for this study included the *system functional data flow* diagram, to capture the global functional dataflow expected from the system and its actors, *mode and state machine* diagram to capture the vehicle states, and automatically generated views such as *system function breakdown* diagram to represent the global functional breakdown and *exchange scenario* diagram to describe sequential flow of functions that describe the functional behavior. The SA layer in Arcadia efficiently supports the system functional analysis and the activities required by the system requirements definition process as described in the INCOSE *Systems Engineering Handbook* (INCOSE. 2015).

Logical Architecture

The *logical architecture blank* diagram is shown in Figure 10. The *system functions* defined in the SA layer were allocated to the *logical subsystems* and *components* in the LA layer as shown in Figure 10 (a). Most of the functional architecture defined in the SA layer was automatically transitioned into the logical elements thereby reducing significant rework. The key *logical subsystems* identified include the 'vehicle control unit', 'chassis', 'power electric,' 'wheel and tire' and 'body.' These subsystems and their *logical interfaces* represent the logical breakdown of the PBV and their allocated functional behavior. The 'wheel and tire' subsystem remains the key focus of project as the objective is to simulate the vehicle performance to capture the best set of design parameters for the vehicle tires (TDPs). Based on a consistent logical architecture definition, several simplified functionality-specific views expressing a particular stakeholder concern are generated using SMW diagram filters to communicate the architecture design between the various stakeholders involved in virtual tire development. Figure 10 (b) shows the functionality-specific view representing the *handling* performance and a tailored simulation-specific view (simulation concern) that enables communicating the logical architecture schematic to the system simulation engineer is shown in Figure 10 (c). The FTCs related to the *handling* performance are linked to the 'vehicle' block in the functionality view. The 2 views provide the basis for the simulation engineer to analyze the system performance with the goal to identify the optimum TDPs for the 'tire' component. Similar functionality specific views were generated for *ride, NVH, mileage*, and *durability* with their corresponding FTCs associated to the 'vehicle.'

Architecture Verification

Teamcenter® Verification Request encapsulates the verification process inputs mainly verification requirements, FTCs, system architecture models, test methods, test cases, simulation input/output parameters, etc. to effectively communicate verification request criteria and procedures to the domain experts. The logical architecture imported into the simulation environment provides a modifiable simulation architecture skeleton as the starting point as shown in Figure 11. A simplified logical architecture view of the vehicle was generated for the purpose of simulation with only the level 1 subsystems required to develop a physics-based simulation model. With the help of simulation libraries, physical systems are assigned to each subsystem/component such as the VCU, chassis, power electric, wheel and tire and body and a system simulation model is developed for each scenario with corresponding physical components, as different sets of boundary conditions (driver and road) and tire models are considered to

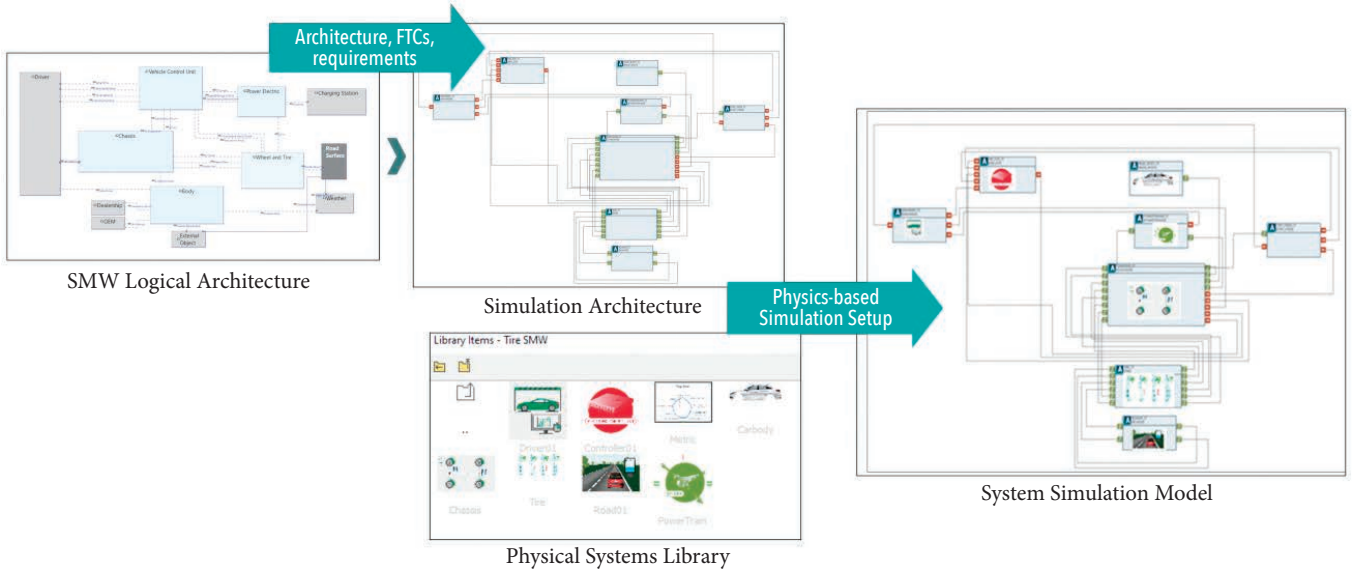


(a) PDV Logical Architecture

(b) PDV Logical Architecture – Handling View

(c) Logical Architecture – Component View

Figure 10. Logical architecture and FTCs



SMW Logical Architecture

Simulation Architecture

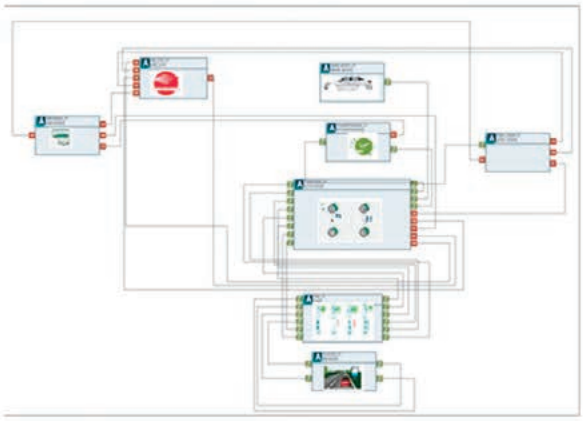
Physical Systems Library

System Simulation Model

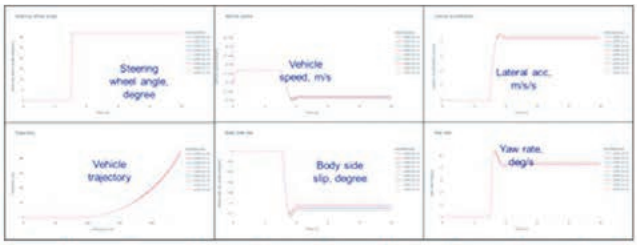
Figure 11. Transitioning From system architecture to simulation model

extract FTCs and identify optimum TDPs.

For this study, 'step steer' and 'constant radius' scenarios are simulated on flat road with 'Pacejka tire' model (Pacejka and Bakker 1992) to evaluate the handling performance whereas 'ride' scenarios are simulated with rough road and 'F-tire' model (Gipser 2003). For step steer, handling simulation architecture model is evaluated with constant speed of 100 kmph and step input of steer angle to achieve target 0.4g (4 m/s/s) acceleration to extract lateral acceleration and yaw rate. Similarly, vehicle performance evaluation for 'virtual turning' (constant radius cornering) and 'ride' scenarios was performed to extract the corresponding FTCs. Once the desired verification targets are achieved, the output parameter values are updated in Teamcenter and are synchronized with the system architecture model for updated traceability. Postprocessed simulation results as shown in Figure 12 were obtained for 9 different architecture choices and the optimum parameter set is baselined in Teamcenter to capture the TDP values. Simultaneously, the values of the extracted FTCs are saved to Teamcenter and synchronized with the parameters traced to the logical architecture in SMW for baselining. Such an integrated verification mechanism can enable efficient collaboration between the system architects and simulation engineers by providing a common dashboard for verification enabling real-time visibility



(a) Simulation Model



Architectur e	Yaw rate overshoot, %	Yaw rate response time, s	Lat acc overshoot , %	Lat acc response time, s	0.4g steering angle, deg	Yaw rate sensitivity deg/sec/1 00deg	Roll gain, deg/g	Body Sideslip, deg	TB factor, deg*sec
1	25.44	0.08	4.62	0.19	32	27	0.08	1.05	0.28
2	24.05	0.08	4.51	0.19	32	27	0.07	1.09	0.32
3	23.26	0.08	4.41	0.19	32	28	0.07	1.14	0.34
4	23.23	0.08	4.41	0.19	32	28	0.07	1.14	0.34
5	25.44	0.08	4.62	0.19	32	27	0.08	1.05	0.28
6	24.17	0.08	4.51	0.19	32	27	0.07	1.09	0.32
7	24.19	0.08	4.52	0.19	32	27	0.07	1.1	0.31
8	23.15	0.08	4.41	0.2	32	28	0.07	1.15	0.34
9	25.59	0.08	4.62	0.19	32	26	0.08	1.05	0.28

(b) Post-processing results

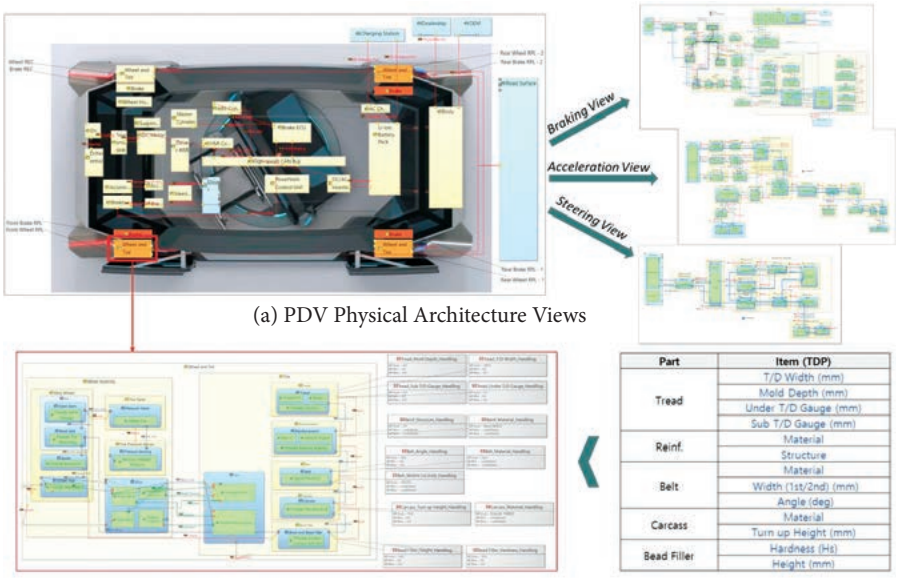
Figure 12. System simulation results and FTC extraction

to verification teams across the enterprise thereby improving the verification process efficiency.

Physical Architecture and Tire Design Parameters

Figure 13 shows some of the *physical architecture blank* diagrams developed in the PA layer. HKMC collaborates with a global supply chain to analyze vehicle's performance and share design data. The logical and physical architectures provide a means to collaborate with the design and manufacturing supply chain to exchange product information through a common source. The physical architecture diagram shows the selection of the *physical hosting components* that will deliver the desired vehicle behavior. Because of the sheer complexity of the vehicle architecture that results in the PA layer, it is difficult to capture the physical architecture in a single diagram, which is why a combination of different architecture views were developed resulting in one integrated model. Figure 13 (a) shows the simplified physical architecture diagram that includes the hosting physical components (wheel and tire, brake, DC motor, CAN bus, Li-Ion battery pack, etc.) that host the *behavior components* that realize the logical subsystems behavior along with several functionality-specific views describing parts of the overall physical architecture.

The system-to-subsystem transition concept demonstrated automatically transitioning from the 'vehicle' into a disparate model of the wheel and tire' module and the TDPs that are identified as a result of system simulation are captured in the physical architecture of the 'wheel and tire' module. This could potentially support model-based collaboration between the architects and the designers of a particular vehicle component, for instance the 'wheel and tire', without



(a) PDV Physical Architecture Views

(b) Physical Architecture—TDP for Handling

Part	Item (TDP)
Tread	T/D Width (mm)
	Mold Depth (mm)
	Under T/D Gauge (mm)
	Sub T/D Gauge (mm)
Reinf.	Material
	Structure
Belt	Material
	Width (1st/2nd) (mm)
	Angle (deg)
Carcass	Material
	Turn up Height (mm)
Bead Filler	Hardness (Hs)
	Height (mm)

(c) Tire Design Parameters

Figure 13. Physical architecture views and TDP

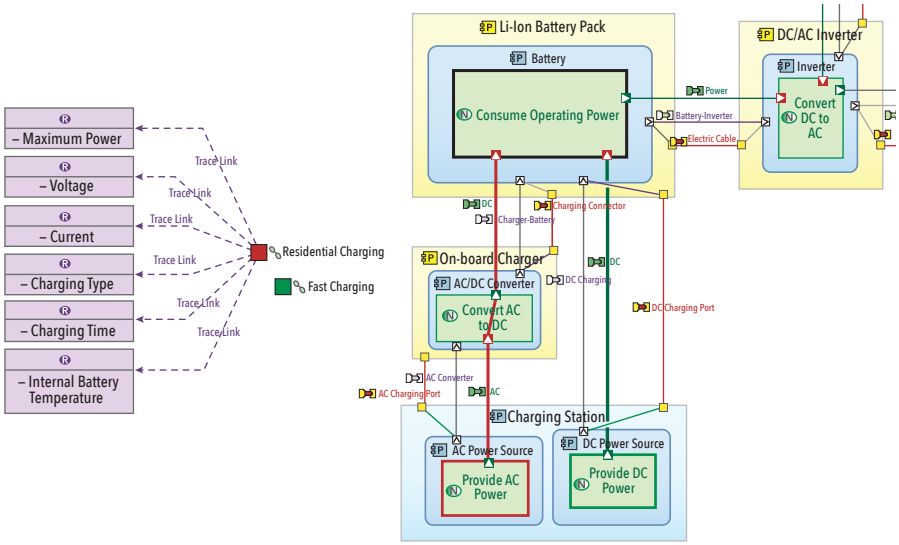


Figure 14. Physical architecture – 'Charging' view

having the need to share the entire 'vehicle' architecture. Figure 13 (b) and (c) show the TDPs identified for the optimum *handling* performance whose values are captured and updated in the parameters linked in the physical architecture view for 'wheel and tire'. These values are synchronized with the parameter objects in Teamcenter once the architecture is baselined. Such a process allowed HKMC to capture the optimum parameter values in a single source that could then be viewed, managed and consumed by several stakeholders involved with the overall vehicle development. Finally, the physical architecture provides operational traceability to the lowest level of the modular SoS approach. As an example, the 'residential charging' and 'fast charging' *functional chains* shown in the physical architecture view in Figure 14 realize the *operational process* 'charge vehicle' that is one of the many processes required to deliver the 'provide EV mobility' SoS operational capability that was captured in the OA layer as shown in Figure 8. These chains capture the physical design constraints posed by the vehicle-to-grid (V2G) interfaces at the SoS level with traceability all the way to the PA layer in the form of physical design/interface requirements between the vehicle and other constituent systems thereby maintaining end-to-end traceability.

CONCLUSION AND FUTURE WORK

The PoC described in this paper shows how an MBSE approach can provide significant benefits to HKMC's product development process. The paper demonstrated the use of system architecture models managed in the context of a PLM-based orchestration platform enabling a digital thread as the basis of traceability across the MBSE enterprise. Considering the magnitude of challenges posed by the future of electric vehicle mobility, the study showed the possibility of applying a top-down, modular system of systems perspective to the problem of virtual tire development using a structured, method-based approach to system architecture modeling. A consistent system architecture model(s) can serve as a powerful tool to enable effective cross-domain communication across several stakeholders involved in the automotive product lifecycle. The ability to share and manage architecture lifecycle data through the orchestration platform allows to create an integrated system definition that can provide a common source of product information to several stakeholders involved in the MBSE processes. In addition to the commonly known benefits realized by an MBSE approach such as end-to-end requirement traceability and effective communication, HKMC's tire engineering team emphasizes the following benefits that are realized by using such an approach:

Reduced Time to Model. The Arcadia method guidance provided a structured approach to modeling the vehicle architecture with automated transitions between the abstraction layers and model validation capabilities. This resulted in an easier learning curve thereby significantly improving the time to model.

SoS Knowledge Capture. Modeling the operational context of the SoS using Arcadia can provide a richer understanding of the needs of all the stakeholders and constituent systems involved in the SoS. This helps in potentially extending the scope of possibilities that can be analyzed to meet the stakeholder needs before contextualizing the system boundary. The operational behavior and interfaces can be used to capture necessary *operational processes* that can result in better product planning decisions at the very early stages of product development. These processes are traced from OA to PA through the *functional chain*s. This can potentially enable tracing each individual SoS capability realization to the lowest level of the constituent subsystems and components that deliver the functionality.

Reusability. The concept of reusable libraries of model elements was explored to demonstrate reusability within the architecture project. This can result in a dramatic reduction in the architecture development effort. With increasing numbers of parts and components that compose the modern products, reusability across projects and programs will likely be a game changer as more engineering teams choose to adopt MBSE approaches in their respective development processes.

Parameter Management and Traceability. In vehicle performance development, one of the more pressing challenges faced by the Tire engineering team is the ability to efficiently manage and trace different levels of parameters that exist across the development lifecycle. Starting from the vehicle concept phase to the detailed design, the paper shows the usage of parameters as configurable objects used to specify vehicle and tire-level performance metrics (FTCs) and their traceability to the design parameters (TDPs) both in the context of the architecture definition activity. Not only did specifying FTCs enable easier communication between the system architect and the performance simulation engineer, but the ability to associate the TDPs to the physical architecture components provided a robust and secure way of sharing design data between stakeholders.

Early Verification. The parameterized logical architecture was used to initiate early system simulation for performance verification. This enabled frontloading design decisions and communicating physical architecture design with the domain engineers using a model-based reference provided by a modularized vehicle architecture.

Cross-enterprise Collaboration. Another perceived benefit of using MBSE is the ability to share whole or part of the system architecture with cross-domain teams. The case study demonstrated a system-to-subsystem transition approach where the 'wheel and tire' component in the physical architecture was transitioned into a separate Arcadia project that could be shared across the design teams with parameters flow down and capture design parameter values after verification in the physical architecture. This use case can potentially be extended to the Tier1/2 tire design and manufacturing suppliers with a robust and secure lifecycle management framework thus promoting a model-based enterprise.

This study presents tremendous opportunities to apply similar MBSE practices across HKMC. Future work should focus on establishing a complete modular vehicle architecture that can address numerous stakeholder concerns across the product development lifecycle. The system-to-subsystem transition addon in the modeling tool can facilitate generation of architectures of vehicle modules between levels and should be explored as a communication mechanism to share architecture modules with subsystem architects and suppliers through a robust collaboration framework, thereby enabling an extended MBSE enterprise throughout the design supply chain. The modularity should span across many architecture levels as shown in Figure 5 and each level shall include architectures that are reusable, maintainable and configurable. Future work should also focus on exploring the possibilities of connecting the architecture modules with the product variants library for better product planning decisions during future product evolutions. To realize a shared vision, the MSBE approach should be applied at scale across engineering teams and across other automotive companies such that a concept architecture defined at early stages provides a reference to downstream design teams with controlled access to architecture data for cross-disciplinary collaboration. ■

REFERENCES

- Mihailovici, M. 2021. "When software writes software." *Porsche Engineering Magazine*, January, issue 1/2021, pp. 24-27, viewed October 2022. https://www.porscheengineering.com/filestore/download/peg/en/magazine-202101/default/ab93c8ce-7b56-11eb-80d3-005056bbdc38/Download-Magazine.pdf.
- Varanasi, A. 2022. "How Electric Vehicles Could Fix the Electrical Grid." *Columbia News*, 6 May, viewed October 2022. https://news.columbia.edu/news/how-electric-vehicles-could-fixelectrical-grid.
- Litman, T. 2022. "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning." Victoria Transport Policy Institute, viewed October 2022. https://trid.trb.org/view/1678741.
- Bhatti, G., H. Mohan, and R. R. Singh. 2021. "Towards the Future of Smart Electric Vehicles: Digital Twin Technology." *Renewable and Sustainable Energy Reviews*, vol. 141, article 110801.
- Freemark, Y., C. Stacy, O. Fiol, and J. Morales-Burnett. 2022. "Regulations to Respond to the Potential Benefits and Perils of Self-Driving Cars." Urban Institute, September, viewed October 2022. https://www.urban.org/sites/default/files/2022-09/Regulating%20Autonomous%20Vehicles%20Report.pdf.
- Eugensson, A., M. Brännström, D. Frasher, M. Rothoff, S. Solyom, and A. Robertsson. 2013. "Environmental, Safety Legal and Societal Implications of Autonomous Driving Systems." 23rd International Technical Conference on the Enhanced Safety of Vehicles (ESV), vol. 334, Seoul, KR, 27 – 30 May. https://trid.trb.org/view/1362102.
- Continental. 2022. "Smart Tires, Digital Solutions for Efficient and Sustainable Logistics." Continental Tires Press Release, 24 October, viewed 10 November 2022. https://www.continental.com/en/press/press-releases/20221024-whitepaper-digital-solutions/.
- Siemens. 2021. "Siemens Partners with Hyundai Motor Company and Kia Corporation for Digital Mobility Transformation." Siemens Press Release, 29 November, viewed November 2022. https://www.plm.automation.siemens.com/global/en/our-story/newsroom/siemenshyundai-kia-xcelerator-digital-twin/103280.
- INCOSE. 2018. "System of Systems Primer." International Council of Systems Engineering.
- Hyundai Motor Group. 2021. "The essence of future mobility lifestyle', Hyundai Press Release, 28 December, viewed August 2022. https://www.hyundaimotorgroup.com/story/CONT0000000000005028
- UNECE. 2014. "Proposal for a New Global Technical Regulation on the Worldwide Harmonized Light Vehicles Test Procedure (WLTP)." World Forum for Harmonization of Vehicle Regulations, 162nd Session, Geneva, CH, 11-14 March, viewed September 2022. https://unece.org/DAM/trans/doc/2014/wp29/ECE-TRANS-WP29-2014-027e.pdf.
- Riemersma, I., and P. Mock. 2012. "Influence of Rolling Resistance on CO_2." *The International Council on Clean Transportation*, 27 November, viewed October 2022. https://theicct.org/wpcontent/uploads/2021/06/ICCT_work_rollingresistance_nov2012.pdf.
- Barrand, J., and J. Bokar. 2008. 'Reducing Tire Rolling Resistance to Save Fuel and Lower Emissions." *SAE International Journal of Passenger Cars-Mechanical Systems*, 14 April, pp. 9-17.
- Koengkan, M., J. A. Fuinhas, M. Belucio, N. K. Alavijeh, N. Salehnia, D. Machado, and F. Dehdar. 2022. "The Impact of Battery-Electric Vehicles on Energy Consumption: A Macroeconomic Evidence from 29 European Countries." *World Electric Vehicle Journal*, 9 February, 13 (2): 36.
- Weiss, M., K. C. Cloos, and E. Helmers. 2020. "Energy Efficiency Trade-Offs in Small to Large Electric Vehicles." *Environmental Sciences Europe*, 18 March, 32 (1): 1-17.
- Jamshidi, M. O. 2008. "System of Systems Engineering — New Challenges for the 21st Century,' *IEEE Aerospace and Electronic Systems Magazine*, 16 May, 23 (5): 4-19.
- DoD. 2008. "Systems Engineering Guide for Systems of Systems." August, version 1.0. US Department of Defense (DoD), Washington, US-DC.
- INCOSE. 2015. "Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities." Hoboken, US-NJ: Wiley, ISBN: 978-1-118-99940-0.
- D'Ambrosio., J., and G. Soremekun. 2017. "Systems Engineering Challenges and MBSE Opportunities for Automotive System Design," IEEE International Conference on Systems, Man, and Cybernetics (SMC), *IEEE Explore*, November, pp. 2075-2080.
- Continental. 2017. "Driving Intelligent Transportation Systems with Capella." Continental Return of Experience, Capella Day 2017, Toulouse, FR, June, viewed October 2022. http://www.clarity-se.org/wp-content/uploads/2015/05/Driving-Intelligent-TransportationSystems-with-Capella.pdf.
- Young, B., J. Cheatwood, T. Peterson, R. Flores, and P. Clements. 2017. "Product Line Engineering Meets Model Based Engineering in the Defense and Automotive Industries." *Proceedings of the 21st International Systems and Software Product Line Conference*, vol. A, pp. 175-179.
- Sohier, H., P. Lamothe, S. Guermazi, M. Yagoubi, P. Menegazzi, and A. Maddaloni. 2021. "Improving Simulation Specification with MBSE for Better Simulation Validation and Reuse," *Systems Engineering: The Journal of the International Council on Systems Engineering*, 24 (6): 425-438.
- Nowodzienski, P., and J. Navas. 2022. "From Model-Based to Model and Simulation-Based Systems Architectures–Achieving Quality Engineering Through Descriptive and Analytical Models." 32nd Annual INCOSE International Symposium,Detroit, US-MI, 25 – 30 June , 32 (1): 1247-1266.
- International Organization for Standardization. 2019. ISO/IEC/IEEE 21839: 2019. Systems and Software Engineering–System of Systems (SoS) Considerations in Life Cycle Stages of a System." Geneva, CH: ISO.
- Dahmann, J., and M. Henshaw. 2016. "Introduction to Systems of Systems Engineering." *INSIGHT*, October, 19 (3): 12-16.
- Hoehne, O. M., and G. Rushton. 2018. "A System of Systems Approach to Automotive Challenges." *SAE Technical Paper*, April, no. 2018-01-0752.
- Peters, J. R., E. Jahangir, A. Surana, and Z. Mian. 2018. "Multi-Layer Framework for Synthesis and Evaluation of Heterogeneous System-of-Systems Composed of Manned and Unmanned Vehicles." *SAE Technical Paper*, October, no. 2018-01-1964.
- Mordecai, Y., and D. Dori. 2013. "6.5. 1 I5: A Model-Based Framework for Architecting System-of-Systems Interoperability, Interconnectivity, Interfacing, Integration, and Interaction." INCOSE International Symposium, Philadelphia, US-PA, 24-27 June, 23 (1): 1234-1255.
- INCOSE. 2020. "An Automotive World in Motion, Systems Engineering Vision 2025." INCOSE, viewed July 2022. https://conference.conflr.com/events/Demo/showcases/incose/INCOSE-automotive-vision-2025.pdf.

- Maier, M. W. 1998. "Architecting Principles for Systems-of-Systems." *Systems Engineering: The Journal of the International Council on Systems Engineering*, February 1999, 1 (4): 267-284.
- Voirin, J. L. 2017. "Model-Based System and Architecture Engineering with the Arcadia Method." *Elsevier*, ISBN: 9780081017944.
- Bajaj, M., S. Friedenthal, and E. Seidewitz. 2022. "Systems Modeling Language (SysML v2) Support for Digital Engineering', *INSIGHT*, April, 25 (1):19-24.
- Kossiakoff, A., S. M. Biemer, S. J. Seymour, and D. A. Flanigan. 2020. *Systems Engineering Principles and Practice* 3rd edition. Hoboken, US-NJ: Wiley, ISBN: 978-1-119-51666-8.
- Pacejka, H. B. and E. Bakker. 1992. "The Magic Formula Tyre Model." *Vehicle System Dynamics, International Journal of Vehicle Mechanics and Mobility*, 21 (1): 1-18.
- Gipser, M. 2004. "FTire, a New Fast Tire Model for Ride Comfort Simulations." International ADAMS User's Conference, Berlin, DE, November, pp. 1-11.

ABOUT THE AUTHORS

[Editor: Author biographies were current when the paper was initially published in 2023.]

Sunkil Yun works as a senior research engineer for model-based systems engineering at Hyundai Motor Company. Completed the doctor's course at the Graduate School of Systems Design Management, Keio University in March 2020. Doctor of Systems Engineering. Interested in model-based systems engineering and system modeling.

Shashank Alai is a solution consultant for model-based systems engineering solutions at Siemens DISW. In the past 4 years, Shashank has worked with customers from various product manufacturing industries, including automotive and aerospace on topics related to MBSE, particularly system architecture modeling. Shashank has a master's in mechanical engineering with a focus in systems engineering and likes to blend his theoretical understanding of systems engineering with practical SE challenges at industry customers.

Yongdae Kim received his PhD in mechanical engineering from Hanyang University in South Korea in 2016 and has been a professional engineer for noise and vibration from 2006. He is currently a senior researcher in Hyundai Motor Company. His research area includes dynamics and system engineering.

Jaehun Jo works as an engineer for model-based systems engineering at Hyundai Motor Company. Jaehun jo has a doctor's degree in automotive engineering with a focus in vehicle architecture development process. Jaehun jo is interested in multi-physics dynamics and model-based system engineering.

Tae Kook Kim works as a business development consultant at Siemens DISW for digital transformation of manufacturing industry. His experience in innovation includes various manufacturing industries including shipbuilding, automotive, and high-tech. He is now managing MBSE projects for a leading automotive OEM company and electronics company as a project manager. He holds MBA and BS for naval architecture and ocean engineering.

Dahyeon Lee is a presales consultant of product lifecycle management at Siemens DISW. Her work focuses specifically on providing an integrated MBSE environment that supports closed loop from requirements to verification management using the PLM system. She likes to configure the system that fit their work environment based on customer's needs and provide a system that can increase work and data management efficiency.

Lokesh Gorantla works as a senior simulation engineer for model-based systems at Siemens DISW. Lokesh has a master's in mechanical engineering from Indian Institute of Technology, Bombay. In the past 11 years, Lokesh has had the opportunity to work with customers from various organizations in automotive and aerospace domain on topics related to system modelling and analysis of chassis systems, vehicle dynamics, and hydraulic systems.

Michael Baloh is a solution consultant for model-based systems engineering solutions at Siemens DISW. He received his PhD in controls and systems engineering from University of Virginia and a post-doc fellowship at INRIA France with a focus on autonomous vehicles. Michael has since worked as lead developer for controls and for EV, HEV, PHEV applications. He has a broad experience in planning and executing projects in model-based control system development process with special expertise in controller design and verification and validation.

Flanigan and Robinson continued from page 36

- Wang, Michael. 2022. "Biofuel Life-cycle Analysis with the GREET Model." Argonne National Laboratory, 1 March. https://www.epa.gov/system/files/documents/2022-03/biofuelghg-model-workshop-biofuel-lifecycle-analysis-greet-model-2022-03-01.pdf, last accessed 21 September 2022.
- Williams, Eric D., Robert U. Ayres, and Miriam Heller. 2002. "The 1.7 Kilogram Microchip: Energy and Material Use in the Production of Semiconductor Devices." *Environmental Science & Technology* 36.24: 5504-5510.
- Yung, Winco K. C., Subramanian Senthilkannan Muthu, and Karpagam Subramanian. 2018. "Carbon Footprint Analysis of Printed Circuit Board." *Environmental Carbon Footprints*. ButterworthHeinemann, 365-431.

ABOUT THE AUTHORS

[Editor: Author biographies were current when the paper was initially published in 2023.]

Kevin Robinson has worked in numerous roles across both the UK's Ministry of Defence and Australia's Department of Defence and industry. Throughout his career, Kevin has taken a leadership role in advancing the field of model-based systems engineering (MBSE) via his publications and contributions to the systems engineering community. He initiated and chaired Australia's first annual MBSE symposium, formed and chaired INCOSE's Model-Based Conceptual Design Working Group, delivered a keynote address to INCOSE's international symposium in 2016, and has made contributions to INCOSE's systems engineering Handbook and related standards

David Flanigan is a member of the principal professional staff for The Johns Hopkins University Applied Physics Laboratory, providing systems engineering services to various Department of Defense and Department of Homeland Security clients, and has 20 years of active duty and reserve service with the US Navy. A graduate of the University of Arizona, he holds a MS in information systems and technology, a MS in systems engineering from the Johns Hopkins University, and a PhD in systems engineering and operations research from George Mason University. Dr. Flanigan is a member of INCOSE and MORS.

APRIL 2026
VOLUME 29/ ISSUE 2

Systems Engineers – Value Added Product Owners

Aswin Sukumaran Nair, Aswin.Nair@Carrier.com

■ **ABSTRACT**

Agile Methodology requires specialized roles like Product Owners to act as the bridge between the business and the product delivery teams. This calls for specific skills such as a customer-centric, design-thinking mindset and a good vision of the overall system, its capabilities, and the ability to provide the right information at the right time.

The primary duty of a product owner is to maintain a well-refined, prioritized backlog of work items. Systems Engineers are enabled with the right methods and tools to perform solution architecture, design synthesis, system verification, and validation. Systems Engineers have visibility into the overall system, interfaces between sub-systems, and external interface requirements. Enabled with the knowledge of Technical Processes, Systems Engineers are able to describe System Elements, their behavior, and interactions in the best possible detail. This allows them to ensure that functional and non-functional backlog items are defined in unambiguous adequate detail.

By applying their knowledge and expertise as mentioned above, Systems Engineers can effectively perform Product Owner duties and enables Agile teams to work efficiently to deliver the correct system increment at the end of each pre-defined time box.

OVERVIEW

Many organizations have already adopted or are in the process of adopting Agile as a standard method for product/service development and delivery. This requires specialized roles like Product Managers and Product Owners to define and support the building of products that meet customer needs. Product Managers and Product owners act as the bridge between the business and stakeholders and the product delivery teams. This requires specific skills such as having a customer-centric, design-thinking mindset and a good vision of the overall system, its capabilities, and the ability to enable cross-functional teams with the right information at the right time. Systems Engineers possess the traits and skills necessary to perform these responsibilities (INCOSE 2015).

Agile teams work on "Backlogs" or holding areas that describe features that address user needs and deliver business benefits. It is always crucial to maintain a well-refined backlog. Systems Engineers can consider both the business and technical needs of customers and can translate stakeholder needs into requirements that can be easily understood by engineering and other functional teams since they are enabled with the right methods and tools to perform solution architecture, design synthesis, and system verification and validation. This allows them to effectively manage stakeholder needs and decompose them into various levels of "Backlog" with the right amount of detail.

Agile teams are generally focused on delivering a feature increment. This means that they need to be enabled with the underlying system, communication platforms, interfaces, and such for seamless feature delivery. Systems Engineers have visibility into the overall system and interfaces between various sub-systems within the system and external interface requirements. This allows them to ensure that enabler and non-functional backlog items are properly defined – which sometimes general product owners and business analysts find difficult to do.

When multiple self-organized agile teams are trying to deliver a product in increments, coordination among these teams becomes vital. Product Managers and Product owners need to ensure that each agile team is delivering the right "ingredient" that rolls up into the overall system or product. Enabled with the knowledge of Technical Processes, especially with the Implementation process, Systems Engineers can describe System Elements, their behavior, and interactions in the best possible detail. This allows Agile teams to work independently and deliver a system increment at the end of each pre-defined time box.

The points listed above are only a few among many reasons why Systems Engineers can effectively handle Product Owner duties in an Agile ecosystem and deliver added value by applying their knowledge and expertise. The article discusses this in detail in the next few sections.

BEING "AGILE" DURING SYSTEM DEVELOPMENT

In the changing times, it is imperative that organizations need to be able to deliver solutions that are more efficient, cost-ef-

fective, and available at a faster pace than their competitors. Many organizations over time have adopted or are in the process of adopting lean methods built on Agile practices. These practices enable organizations to work with greater agility, deliver in a continuous fashion, and can incorporate changes better.

Changes are typically introduced mostly by the following causes, which arise in the due course of system development.

SHIFT IN MARKET TRENDS:

The traditional waterfall method of solution/system/product development is linear in nature and requires a clear establishment of expectations and deliverables from a project at the outset itself. This poses a challenge – the market evolves very rapidly, and the initial set of needs and requirements can quickly change over time – in many cases during the execution of the project itself. Since the waterfall method relies heavily on the completion of an earlier phase in its entirety, for the next phase to begin, the cost of introducing changes at later phases is quite high.

EVOLVING STAKEHOLDER NEEDS:

Waterfall methods also require initial agreements with stakeholders that are frozen during early stages of the project. During later phases of product development, when the initial samples evolve, and teams present to stakeholders during scheduled phase gate reviews, it is possible that there could be differences in opinion from stakeholders regarding how they envisioned the product to be in the first place.

While good systems engineering practices in the early stages of needs analysis and effective requirement elicitation processes can address this to a certain extent, technical limitations, performance considerations, and more can have a detrimental effect on the overall experience that the stakeholders have once the initial samples of the product are ready. At this stage, since most of the design considerations are identified and locked down, the amount of flexibility in making changes tends to be on the lower side.

INTRODUCTION OF DEFECTS IN LATER STAGES:

During the phased product development approach — in many cases, verification and validation of the product and systems also happen in a phased manner. This means that the entire solution or product is picked up for verification after the design and implementation of the design are final. This can lead to late identification of unintended behaviors, issues occurring during boundary conditions, integration issues, and such. Since the design at that point is finalized, the cost of fixing these issues is significantly higher.

A good application of technical Processes such as system analysis process, integration process, and verification process reduces this risk significantly. However, with distributed teams, teams operating with higher autonomy, dependency on simulations, inability to put together real-world test conditions, and such can still cause defects to be identified late in the process.

IMPROVEMENTS WITH AGILE

In the Agile world, there are different levels at which "backlog" items are constantly reviewed and refined — At the Business Portfolio Level, At the Product or System level, At project levels, and at Agile team levels. Product Managers, Product Owners, Systems Engineers, and Implementation teams come together during pre-defined periodic intervals and evaluate if there are any changes to the initial set of requirements identified. These changes could have originated based on shifts in market trends, and evolving needs based on stakeholder experience.

In addition to this, Agile teams conduct a demonstration of the system capability and state achieved to date — providing the stakeholders and other decision-makers valuable inputs regarding system behavior progression in real-time. This enables the team to assess, prioritize, triage, and assign team members to solve defects in the system, and to prevent potential issues that could arise in the future in a methodical manner.

These periodic reviews or "the refinement process" and demonstrations allow for all stakeholders to convene and assess the state of the system or product development. Based on the inputs coming in from various sources as mentioned above, changes required are added to the appropriate backlog levels. The changes funnel down to the subsequent levels where appropriate course correction occurs. This provides great flexibility in terms of incorporating change and ensuring continuous delivery of system increments with built-in quality checks.

ROLE OF PRODUCT MANAGERS AND PRODUCT OWNERS

It is evident that the key to ensuring agility in managing change is continuous review and refinement of requirements and backlog items at various levels. Product Managers and Product Owners play a vital role in enabling this process.

PRODUCT MANAGERS

Product Managers are responsible for defining the system or product landscape by detailing the features and requirements (Scaled Agile Framework 2021a). They consult and collaborate with multiple functions — including and not limited to end customers, sales, marketing, regulatory and compliance entities, engineering, manufacturing, and others. They analyze and synthesize customer needs, provide insights into the overall Solution Context, and establish a vision and roadmap for the product and the program. They are vital for enabling Agile Release Trains (a team of multiple Agile teams that are responsible for developing system increments) to deliver value continuously.

PRODUCT OWNERS

Product Owners are part of Agile Teams themselves. They are "customer proxies" for the agile teams, and they work with Product Managers, Systems Engineers, other stakeholders, and Product Owners from other agile teams (Scaled Agile Framework 2021b). They enable distilling features into smaller "stories" and help the agile team prioritize them. They are primarily responsible for maintaining the backlog for Agile Teams and ensuring that program priorities are considered while streamlining program execution.

APPLYING SYSTEMS ENGINEERING PRACTICES AND SKILLS TO PRODUCT OWNER DUTIES

Systems Engineers are enabled with knowledge of tools and processes that help in analyzing problems and opportunities that a system presents in its entirety. They can understand the big picture, and they observe and study how different elements in the system interact and change over time. They can assess an issue fully and can contribute towards developing solutions that are optimal by considering multiple aspects — including and not limited to - short- and long-term consequences of actions, impact on the system structure, impact of time delays, and emergent behaviors.

A Systems Engineer's unique skillset developed by consciously being a Systems Thinker allow them to be able to convert ambiguous problem statements into clear and precise work items for the team. They can establish an environment where different teams collaborate and are conducive to collective decision-making and conflict resolution. They can communicate effectively to Organizational leadership and facilitate decision-m aking by presenting solutions, alternatives, implications of decisions, and putting forward recommendations and actions. Let us look at how these skillsets apply to the duties and responsibilities of a Product Owner below.

PLANNING:

Product Owners have multiple duties in terms of Planning and Execution. They are part of the refinement process and ensure

continuous refinement of the program backlog by providing information on technical implementation details, issues faced by the teams, risks, and changes in priorities. They ensure that there is adequate capacity allocated for completion of functional features, making quality improvements, closing technical debts, addressing non-functional requirements, and making appropriate prioritization and adjustments to the execution plan as needed.

Systems Engineers see the big picture and understand how different system elements interact with one another and with external systems. They have an extensive understanding of system interfaces, capabilities, life cycle, feedback loops, and cause-and-effect relationships that individual Agile teams may not be able to comprehend. They are well versed in the Business or Mission Analysis Process and Stakeholder Needs and Requirements Definition Process — allowing them to characterize the solutions space and define the capabilities of the system. This puts them in a unique position to facilitate and contribute to refinement sessions at each stage. As a result, they can add immense value as a Product Owner in refining Product Landscape, Features, and other program backlog items and prioritizing them in accordance with how they envision the system to evolve.

ELABORATION AND ACCEPTANCE OF WORK ITEMS:

Backlog items are drilled down into and elaborated on further for implementation into a work item called "User Story." *Product Owners* ensure that the elaboration of user stories is complete, unambiguous, well defined, and the criteria for the user story to be "accepted" are well defined. They also have constant reviews of the stories with the agile teams that implement the user stories in "backlog refinement sessions." Product Owners also "accept" user stories by validating that they meet the acceptance criteria defined once they are complete. They work with test engineers in the Agile Teams to identify, create, and execute the right set of tests. Any defects arising from these tests are triaged, residual risk assessed, open items prioritized, and added to appropriate backlog queues. (Scaled Agile Framework 2021c)

System Engineers use the knowledge of the System Requirements Definition process, which transforms work items at a capability or feature level into smaller pieces that specify the operational needs of an end user. They apply the practices from this process and the Implementation Process iteratively to system elements impacted by feature implementation to generate a techno-functional view that can easily turn into user stories. Since systems engineers are knowledgeable about system behavior and associated complexity in implementation, they can assist the team in arriving at complexity sizing for features and stories — enabling optimal scheduling of iterations. Systems Engineers are also able to objectively analyze test results based on their understanding of the system and hence be judicious in deciding to "accept" a user story. They can make decisions in assigning priority to residual defects and can help the Agile teams plan to address defects along with the implementation of functionality.

ENABLING TECHNICAL INFRASTRUCTURE:

Product Owners are responsible for understanding the amount of effort and the scope of work to build basic technological infrastructure that enables the implementation of functionality. Product Owners need not necessarily provide technical guidance in developing these capabilities but can prioritize and schedule their delivery so that they are available for Agile Teams in time. This requires them to work with system and solution architects to understand the complexity involved and the effort required in delivering technical infrastructure to ensure that adequate capacity is allocated for Agile Team members to accomplish technical infrastructure delivery.

In addition to functional and enabler work items, Product Owners are responsible for working with Product Management and other key stakeholders to help identify and understand Non-Functional Requirements and decompose them into smaller work items that Agile Teams can work upon. Non-Functional Requirements are critical for meeting stakeholder needs and ensuring that systems work according to their intended behavior.

Systems Engineers, as part of applying the Architecture Definition Process, can define multiple system architecture alternatives and enable the selection of one or more solution architectures that meet system requirements and stakeholder needs. They are aware of technical expectations from the solution architecture and can clearly capture the requirements for implementing the technical infrastructure needed to support the selected solution architecture. This makes it effective for Systems Engineers to work with Agile teams to help them prioritize and implement enabler work items.

The Systems Requirements Definition process empowers Systems Engineers with a set of tools and process steps to identify non-functional and performance requirements, and the constituent parts of the System Specification that emerges as the output of this process can iteratively feed into the program backlog at appropriate intervals — especially during Agile ceremonies like Program Increment Planning.

GATHERING FEEDBACK AND CONTINUOUS IMPROVEMENT:

Product Owners are a key part of periodic capability demonstrations carried out by development teams to key stakeholders. They receive firsthand feedback from stakeholders that expect them to process this and ensure that all items that need to be acted upon and would result in a change of system behavior gets into the backlog funnel and are elaborated and prioritized.

Since they are also an integral part of Agile Teams themselves, they take part in an Agile ceremony called the Iteration Retrospective. The objective of this ceremony is to review and retrospect the completed iteration and find improvement opportunities. The improvement opportunities could be based on actual metrics like work items delivered, defects introduced and addressed, and others, or could be based on practices, processes, challenges, and such. Product Owners can contribute to identifying issues and shaping solutions for improvement.

As part of the Project Assessment and Control Process that *Systems Engineers* are well versed in, they have the right set of tools to assess feedback coming in periodically from stakeholders and Agile Teams. These assessments also monitor progress made on the technical front. When this process applies to feedback emerging from retrospective ceremonies, potential corrective and preventive actions can be identified to ensure that the program can meet the schedule and budget constraints and improve the overall efficiency of processes.

Systems Engineers also review designs, specifications, at times, source code, and other artefacts periodically and can provide technical feedback to Agile Teams and steer them towards course correction if required during implementation. This is an added benefit as regular Product Owners are generally non-technical and they will need to rely on other technical staff to accomplish this.

CONCLUSION

In conclusion, Systems Engineers enabled with Systems Thinking behavior, tools, practices, and skills developed by applying Systems Engineering processes make them efficient and well-suited to play the additional role of Product Owners in the "Agile" world. Organizations and programs would benefit from this approach — both in terms of overall quality and execution efficiency. ■

> continued on page 61

Informing the Delineation of Input Uncertainty Space in Exploratory Modelling Using a Heuristic Approach

Enayat A. Moallemi, Sondoss Elsawah, and **Michael J. Ryan**

ABSTRACT
Exploratory modelling is an emerging approach which can address the challenge of model-based decision making in dealing with input model uncertainties. Exploratory modelling samples from an input uncertainty space and generates extensive computational experiments to analyse possible model behaviours in an output solution space. The way that the input uncertainty space is delineated influences the results of exploratory modelling and its computational cost. In this article, we show the statistical significance of the implication of the size of an input uncertainty space on the resulted output solution space. We also propose a heuristic approach which informs the delineation of input uncertainties by screening the relevant model behaviour in the solution space. An illustrative example of an aircraft fleet management system is used to demonstrate the implementation of our approach in practice. We conclude that the delineation of input uncertainty space can be a way to control simulations in exploratory modelling and to enhance the efficiency of the exploration process and the confidence of the final results.

INTRODUCTION

Models are used widely in the decision science to represent a system of interest and to assist in the decision-making process. The analysis of model behaviour for decision making is challenged by the presence of various forms of uncertainty. Exploratory modelling is an emerging field which addresses this challenge through exploring the implications of various possible values of input uncertainties on the model behaviour (Bankes 1993, Kwakke, 2017, and Moallemi *et al.* 2017). Exploratory modelling uses one or more simulation models to generate the possible impacts of the input uncertainty space on the output solution space in the format of an ensemble of computational experiments (Bankes *et al.* 2001). The design of these experiments—in terms of which uncertainty factor to choose, how to delineate the uncertainty space, how to sample from the uncertainty space, and how many samples to collect—is critical in the exploratory modelling process (Pianosi *et al.* 2016, Schulze *et al.* 1999, and Bergmann *et al.* 2011). Experimental design is critical as different designs can result in different ensembles of generated experiments and varied output solution spaces, and therefore, can sometimes result in divergent decision insights (Pianosi *et al.* 2016, and Kwakkel and Pruyt 2015).

Previous studies have discussed the aspects of the design of experiments in exploratory modelling to different extents (for example Lempert *et al.* 2003, Kwakkel *et al.* 2010, and Haasnoot *et al.* 2013). In this article, we focus on one of these aspects: the delineation of the input uncertainty space. We assume that the behaviour of the output solution space is dependent on the areas of the input uncertainty space from which samples are taken, and therefore, not all areas of the input uncertainty space should be investigated if only a specific behaviour in the output solution space is of interest to decision makers. Accordingly, this article answers the two following questions:

- Question 1: How sensitive is the output solution space to changes in the input uncertainty space?
- Question 2: How can we inform the delineation of the input uncertainty space by screening a behaviour of interest in the output solution space?

We first show the significance of the input uncertainty space on the distribution of the output solution space, with an illustrative exploratory modelling example in the aircraft fleet management system. We then propose and demonstrate a heuristic approach to inform the delineation of the input uncertainty space based on screening output solutions. An informed delineation of input uncertainties can reduce the computational burden of the exploratory modelling process by avoiding extra

simulations based on the samples from the irrelevant areas of input uncertainty space. This screening also presents a more detailed picture of the particular area of interest within the solution space.

The remainder of the article is structured as follows. The second and third sections present an overview of the research background and methods used in this work. The fourth section answers the two research questions posed. The final section concludes the paper and draws future research directions.

BACKGROUND

This section gives a broad overview of exploratory modelling and why experimental design, of which the delineation of uncertainty space (the focus of this article) is one part of design, that is critically important.

EXPLORATORY MODELLING

Exploratory modelling (Lempert *et al.* 2003, Bankes *et al.* 2001, and Bankes 1993) challenges the reliance on deterministic (or even probabilistic with known distributions) sets of model structures and input parameters in modelling in the face of future uncertainties. Exploratory modelling supports the exploration of the impacts of a diversity of parametric and non-parametric assumptions on the output solution space, when the model operates in 'deep or severe uncertainty' (Lempert et al. 2003, and BenHaim 2006). Exploratory modelling generates various model responses in thousands of computational experiments, using a simulation model and sampling from an input uncertainty space. It then analyses the generated experiments with a range of analytical techniques to draw various decision insights and modelling conclusions (Moallemi *et al.* 2017, Kwakkel 2017, and Lempert 2013). See Moallemi *et al.* (Submitted-a), and Walker *et al.* (2013) for a further explanation of exploratory modelling. Exploratory modelling shares similarities with sensitivity analysis as they both use computational experimentation for the treatment of uncertainty. However, contrary to sensitivity analysis, exploratory modelling does not have any pre-assumptions regarding the probability distribution of the uncertainty space. In other words, exploratory modelling can deal with Knightian form of uncertainty where no information (for example, ranking, probability distributions, estimates) exist about the uncertain parameters.

Experimental design

Exploratory modelling is based on the generation and analysis of computational experiments. The way that computational experiments are established and performed—that is, their experimental design— influences the nature and number of generated results, the insights gained from them, as well as the computational cost and time to complete the experiments (Kwakkel and Pruyt 2015). Experimental design includes a decision regarding the list of critical uncertainties, the space of the uncertainty, a sampling strategy for choosing from this space, an appropriate number of samples, and the response variables (outcomes of interest) in the experiments. Among them, delineating the uncertainty space is a delicate aspect of experimental design. An extreme uncertainty space can lead to an extreme computational burden and too plural (many) conclusions, and a very narrow uncertainty space can lead to the risk of missing potential future possibilities from the analysis. Several approaches have been introduced to delineate the uncertainty space, including: setting independent uniform distributions with lower and upper bounds (Pianosi *et al.* 2016), limiting the feasible uncertainty space using *a priori* knowledge for filtering subsets associated with a certain outcome (Kasprzyk *et al.* 2013), and assigning likelihood weights to different values from the uncertainty space based on a comparison between model-generated and observed values (Beven and Binley 1992).

METHODS

We use four methods for answering the two questions of this article. This section introduces these methods briefly. The way that we use each method is explained later.

ANOVA

ANalysis Of VAriance (ANOVA) is a statistical technique to test the statistical significance of difference between means of multiple groups (Montgomery 2001). ANOVA is based on a null hypothesis of no significant difference among groups and an alternative hypothesis of at least one significant difference among groups. Assuming initially that the null hypothesis is true, the observed difference of group means is called statistically significant if it is concluded unlikely to happen by chance. ANOVA uses the F-statistic (that is, a ratio of two variances) to test the statistical difference. ANOVA generate the F-statistic and compared its associated probability of occurrence (p-value) with a threshold (significance level). A p-value less than a threshold justifies the rejection of the null hypothesis and the support of the alternative hypothesis. See Iverson and Norpoth (1987) for further explanation of this technique.

Multi-dimensional clustering

Multi-dimensional clustering (Gerst *et al.* 2013) groups many potential model behaviours, generated based on different samples from the input uncertainty space, into clusters of similar behaviours. The appropriate number of clusters is decided based on the value of Bayesian information criterion (BIC) and Aikake's information criterion (AIC) (McLachlan and Peel 2004). Multi-dimensional clustering uses this appropriate number of clusters and generates a mixture of Gaussian distributions to estimate the distribution of system behaviour in the chosen number of clusters.

Scenario discovery

Scenario discovery is a statistical, data-mining process used to identify subsets of the input uncertainty space that result in similar classes of behaviour in the output solution space (Bryant and Lempert 2010, and Groves and Lempert 2007). Scenario discovery starts by generating computational experiments using a model based on input samples from the input uncertainty space. It distinguishes similar classes of behaviour among experiments and selects alternative subsets from the input uncertainty space (in the format of hyper-dimensional boxes) to describe the classes of behaviour. Scenario discovery uses two measures of quality (coverage and density) and a p-value for comparing the generated subsets and for choosing the best subset of the uncertainty space responsible for the creation of the behaviour of interest in the model. Coverage describes how universally a subset can cover all experiments from a same class of behaviour, and density describes to which extent samples from a subset can only result in experiments with a certain class of behaviour and no other behaviours. Scenario discovery has been implemented using a number of algorithms (Lempert *et al.* 2008) such as classification and regression tree (CART) (Breiman et al. 1984) and patient rule induction method (PRIM) (Friedman and Fisher 1999). See Moallemi *et al.* (2017), Lempert *et al.* (2013), Lempert and Groves (2010), and Moallemi and Malekpour (2018) for implementations of scenario discovery.

Multi-objective robust optimisation

Multi-objective robust optimisation is a group of methods which generate alternative solutions for maximising or minimising multiple objective functions under constraints. The solutions much remain valid under any future conditions (Deb 2001, Marler and Arora 2004, and Ben-Tal and Nemirovski 2000). The result of multi-objective robust optimisation is not a single optimal solution. The multiplicity

of (conflicting) objectives necessitates the generation of *Pareto optimal* solutions, which can compromise among multiple objectives. The presence of future uncertainties also necessitates *robust* solutions where the performance of solutions remains insensitive to drastic changes in the input parameters in the face of future uncertainties. Multi-objective evolutionary algorithms such as NSGAII (Deb and Pratap 2002) and Borg (Hadka and Reed 2013), also see Maier *et al.* (2014) and the non-dominated sorting search algorithm such as Woodruff and Herman (2013) are among popular methods for generating and post-processing (respectively) of Pareto and robust solutions in multiobjective optimisation problems. In both cases, a simulation model is used to assess the impact of the solution space on objective functions and to identify the Pareto robust solutions. See Moallemi *et al.* (Submitted-b), Kasprzyk *et al.* (2013), and Hamarat *et al.* (2014) for the implementations of multiobjective robust optimisation.

THE DELINEATION OF THE INPUT UNCERTAINTY SPACE IN EXPLORATORY MODELLING

This section addresses the two questions raised in Introduction using an illustrative example of asset acquisition and management of aircraft fleets.

THE SIGNIFICANCE OF THE INPUT UNCERTAINTY SPACE

This section shows the significance of the way we delineate the input uncertainty space. While it is clear that changing the delineation of uncertainties impacts the results, this section aims to show the statistical significance of this impact and also to demonstrate the variation in results visually. We explain the steps taken, the generated results and discussion as follows.

Process

We assess the significance of the input uncertainty space in three steps based on the methods explained in Methods:

- Step 1: Different ensembles of computational experiments are generated based on sampling from full and truncated input uncertainty spaces.
- Step 2: A joint kernel density estimate (KDE) diagram is used to show the variation the output solution space—in terms of the state of selected model outputs—in response to the subsetting of the input uncertainty space. ANOVA is also used to test the statistical significance of variation among the multiple ensembles of generated experiments with full and truncated input uncertainty space.
- Step 3: The distributions of the output solution space—in terms of states of selected model outputs in the last time step of the simulation duration—are represented in series of scatter plots, each plot based on sampling from one specific area of the input uncertainty space. The states of the model outputs in the solution space are also clustered using the multi-dimensional clustering technique. We compare the clusters and scatter plots to show how the choice of delineation in the input uncertainty spaces impacts the distribution of the output solution space.

Table 1. List of uncertain parameters with their ranges of variation

Uncertain parameter	Range
The risk that an aircraft is lost during operation	0.00026 – 0.00234 (–)
Lifetime of aircraft	37440 – 336690 (hour)
Total required flying hours	12 – 200 (hour/week)
Expected time spent by an aircraft in CAP	8 – 45 (week)
Time between CAP events	16 – 40 (week)
Expected time spent by an aircraft in DM (Time in DM)	5 – 25 (week)
Time (flying hours) between DM events	200 – 1800 (hour)
Expected time spent by an aircraft in OM (Time in OM)	3 – 15 (week)
Time between OM events	50 – 450 (hour)
Available capacity for CAP	1 – 7 (–)
Number of purchased aircrafts	1 – 7 (–)
OM available capacity	1 – 7 (–)
DM available capacity	1 – 7 (–)

Table 2. The decision space

Decision variable	Range
Available capacity for CAP	1 – 7 (–)
Number of purchased aircrafts	1 – 7 (–)
OM available capacity	1 – 7 (–)
DM available capacity	1 – 7 (–)

Results and discussion

To analyse the impact of sampling from different areas of the input uncertainty space, we generated three ensembles of experiments, each including 200 model runs. Ensembles are generated based on sampling from the full, the first quartile, and the fourth quartile of the uncertainty space and the decision space—where different combinations of decisions regarding the size of acquisition and maintenance can be chosen from. See Table 1 for the full range of uncertainties (also see Appendix A). See Table 2 for the range of decision variables. Experiments in each ensemble represent the model response in terms of (average) in-service aircraft and total (acquisition and maintenance) costs, as two outcomes of interest.

The distribution of the output solution space—in terms of the state of in-service aircraft and total costs—in each ensemble was drawn in a joint KDE diagram (see Figure 1). It is observed that the choice of the input uncertainty space in each ensemble creates a different distribution of the solution space. This difference is more visible in the distribution of in-service aircraft (Figure 1 (b)). This observation can prove our initial assumption regarding the significance of the delineation of the input uncertainty space. We also performed ANOVA to statistically verify the visual observation.

The result of ANOVA in Table 3 shows that in both model outcomes, the null hypothesis (similarity of the means of distributions) is rejected and the alternative hypothesis is supported.

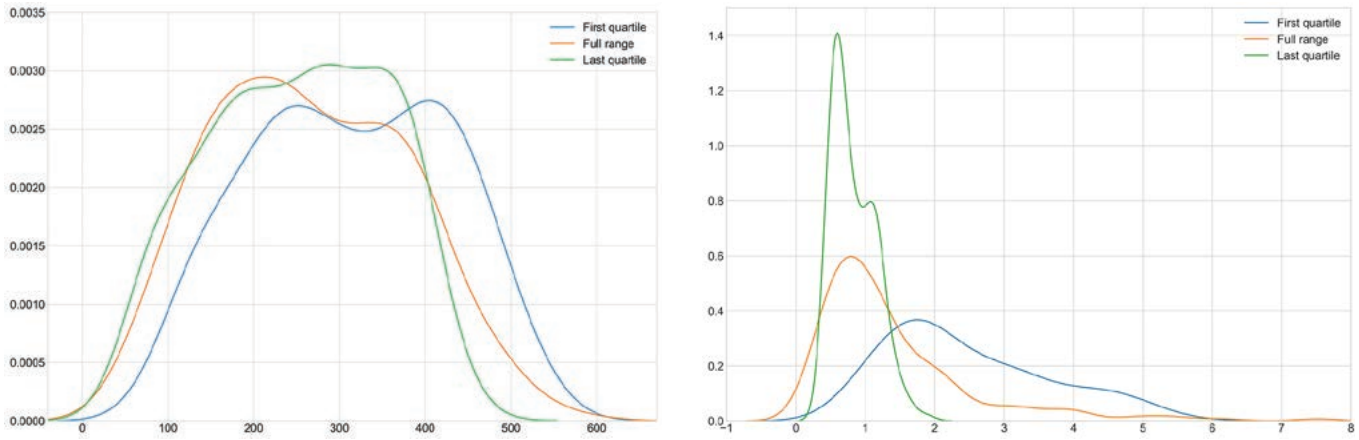


Figure 1. KDEs for (a) total costs and (b) in-service aircraft in three ensembles of experiments for the full range, first quartile, and fourth quartile of uncertain parameters

Table 3. The results of ANOVA (5% significance level) for the thhree ensembles of experiments for (a) total costs and (b) in-service aircraft

SUMMARY				
Ensemble	*Count*	*Sum*	*Average*	*Variance*
Full range	200	53070	265.35	12574.16
First quartile	200	62661	313.305	12917.03
Last quartile	200	50184	250.92	10148.98

ANOVA		
F	*P-value*	*F critical*
17.95669	2.67E-08	3.010815

(a)

SUMMARY				
Ensemble	*Count*	*Sum*	*Average*	*Variance*
Full range	200	278.6238	1.393119	1.323377
First quartile	200	496.2475	2.481238	1.387502
Last quartile	200	166.5347	0.832673	0.096493

ANOVA		
F	*P-value*	*F critical*
150.1713	1.48E53	3.010815

(b)

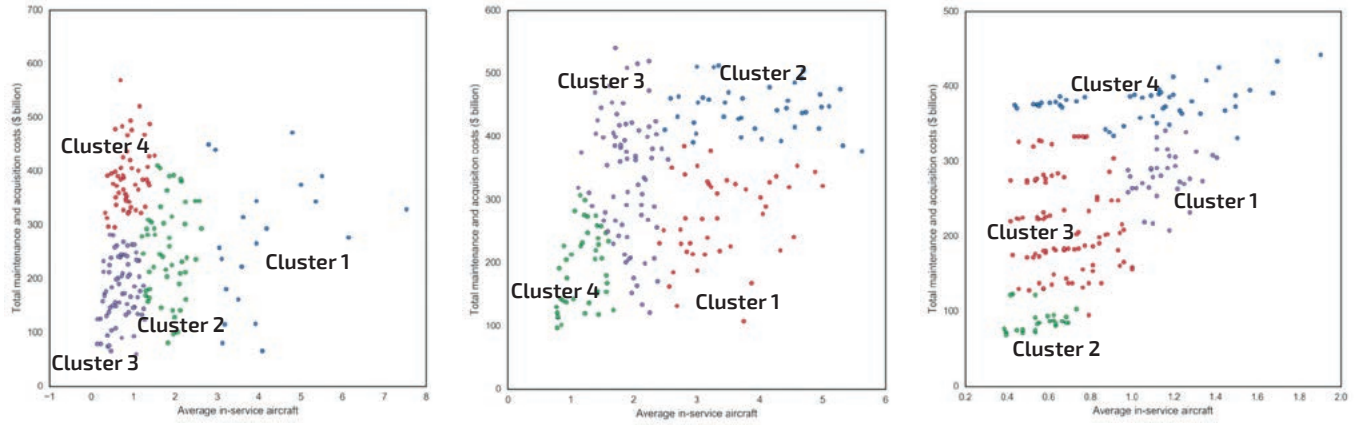
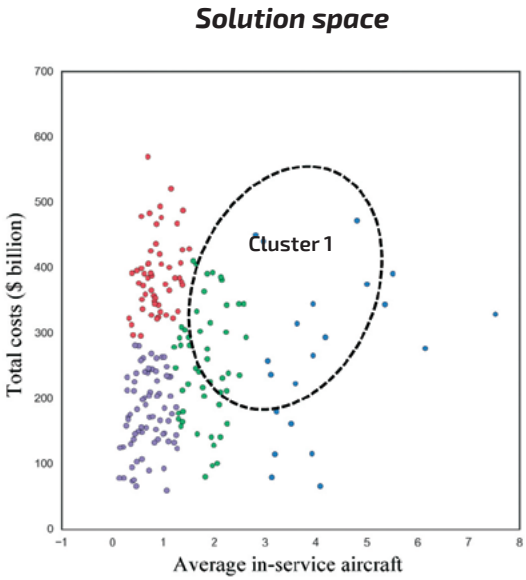


Figure 2. Clusters of experiments with similar behaviour regarding in-service aircraft and total costs based on (a) the full range, (b) first quartile, and (c) fourth quartile of uncertain parameters. Note that clusters are named randomly and the features of clusters from one plot to another do not remain similar.



The truncated uncertainty ranges. For the rest of uncertainties in Table 1, the full ranges are considered.

Measure of merits	Uncertainty	Range	P-value
Coverage: 0.28	Required flying hours	13–61 (hours/week)	7.0e–05
Density: 1.00	DM available capacity	4–6 (–)	4.6e–02
Last quartile	200	166.5347	0.832673

Figure 3. The relationship between Cluster 1 in the solution space and the input uncertainty space

We clustered and plotted generated experiments in each ensemble with respect to the state of in-service aircraft and total costs in a scatter plot (see Figure 2). The results show that truncated uncertainty spaces result in a high resolution of the solution space, but they also disregard some possible variations in the output solution space. For example, Figure 2 (a), based on the full range of uncertainties, shows the possible variations in the solution space (i.e., from about 0 to 8 in-service aircraft and from about B$50 to B$600 total costs) in a high-level picture. However, Figure 2 (b) and Figure 2 (c), which are based on truncated ranges of uncertainties, dismiss this wide possible variation of in-service aircraft and instead present a higher resolution and more accurate clustering of the solution space in regions with high (that is, 1 to 6) in-service aircraft and low (that is, 0.4 to 2) in-service aircraft respectively. A high resolution and accurate clustering of the solution space and the inclusion of a wide diversity of variations can be achieved together if we consider the full uncertainty space and also increase the number of samples (experiments) at the same time. However, this can come at the costs of adding to the computation burden of the process. This leads us to conclude that there is a need for a smart approach which can truncate the input uncertainty space effectively; in a way that improves the resolution and accuracy of the output solution space while minimising the exclusion of relevant solution possibilities from this space. We introduce a heuristic to address this need in the next section.

To answer the first question posed in the Introduction, the output solution space is sensitive to changes in the input uncertainty space as full and truncated areas of the input uncertainty create different (statistically significant) means and dispersion for the distribution of the output solution space.

A HEURISTIC TO INFORM THE DELINEATION OF THE INPUT UNCERTAINTY SPACE

The previous section demonstrated the implications of sampling from the different areas of the input uncertainty space for the output solution space. This section presents a heuristic to inform the delineation of the uncertainty space. The idea behind this heuristic is that although exploratory modelling can generate a wide solution space, not all solutions in this space are relevant for the context of study. These solutions are technically possible to be generated based on random samples from the input uncertainty space. However, the solutions are not possible in reality or, even if they happen, they are not intended by decision-makers or are not relevant to the context. Stakeholders, with their practical knowledge of the context, can help to identify these irrelevant parts of the solution space. We then sample only from those areas of the input uncertainty space which can generate the relevant parts of the solution space. We explain the steps taken in this heuristic, the generated results, and discussion as follows.

Process

To answer the second question posed in Introduction, we suggest a heuristically informed data-mining process which modifies the input uncertainty space based on screening the output solution space in the following steps:

- Step 1: The process starts by projecting the clusters of similar parts of the output solution space (in terms of selected model outcomes) in potential futures with a data-mining technique, called multi-dimensional clustering.
- Step 2: Stakeholders are asked about the desired/relevant parts of this solution space. This stakeholder opinion is used as a heuristic to limit the solution space.
- Step 3: A data-mining method, called scenario discovery, is used to identify which regions of the input uncertainty space are more likely to be responsible for the generation of the desired/relevant parts of the solution space.

Results and discussion

To demonstrate the implementation of this process, we plotted the potential solution space (in terms of in-service aircraft and total costs) based on the full ranges of uncertainties in a scatter plot (see Figure 3). We assumed that the behaviour of experiments in Cluster 1, which are featured with higher in-service aircraft, is of more interest to decision-makers and relevance to the context compared to the other clusters. We, therefore, discard other clusters and only focus on sampling from the area of the input uncertainty space responsible for the generation of experiments in Cluster 1. To delineate this specific area of the uncertainty space, we applied scenario discovery.

The results of the scenario discovery show that the input uncertainty space would be more likely to result in Cluster 1 if the required flying hours and DM capacity available (from Table 1) are truncated to the specified ranges in Figure 3 while the

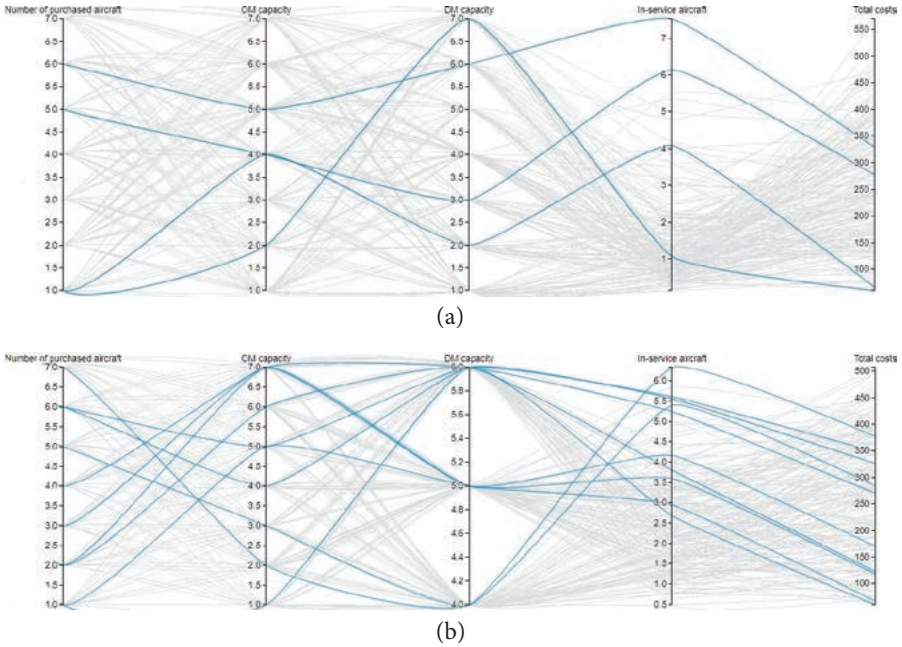


Figure 4. Pareto optimal solutions based on sampling from (a) the full input uncertainty space and (b) the truncated uncertainty space

ranges of the rest of uncertainties in Table 1 remain the same. This truncated input uncertainty space improves the efficiency of the exploration process by avoiding having to sample from the areas of the input space insensitive to the desired solutions.

We show the implication of our heuristic approach for the exploratory modelling process with a multi-objective decision-making problem aiming to find the number of purchased aircraft, OM capacity and DM capacity, which could maximise in-service aircraft and minimise total costs. We generated Pareto optimal solutions with multi-objective robust optimisation under two conditions: the full input uncertainty space and the uncertainty space truncated according to the results of scenario discovery (in Figure 3). This resulted in two different sets of solutions in Figure 4. One reason for their difference is that solutions under each condition are generated based on two series of random sampling and simulation runs. However, the more insightful reason of the difference lies in the different areas of the input uncertainty on which the generation solutions are based. The comparison of the Paretooptimal solutions in two conditions (see Figure 4) shows that running multi-objective robust optimisation under an informed-delineated input uncertainty space can result in more solutions with desired performance for decision-makers. The solutions in Figure 4 (b) can deliver a better trade-off among the multiple decision objectives by delivering a higher in-service aircraft and not necessarily resulting in a higher total cost compared to solutions in Figure 4 (a).

CONCLUSIONS

The delineation of an appropriate uncertainty space in the exploratory modelling process is significant as it needs to capture a picture of model behaviours that is as wide as possible bit does not incur a high computational cost. This led us to think about the ways that we can effectively adjust the delineation of the input uncertainty space based on a feedback control from output solutions. We argued that stakeholders can screen the solution space and identify the desired behaviour of the model. A feedback control, then, can identify which area of the input uncertainty space is responsible for the generation of the desired model behaviour and can inform the adjustment of input uncertainty space accordingly.

The feedback control that we suggested in this article is based on data mining and statistical analyses of both input uncertainties and output solutions. However, this is not the only way to design such feedback control. An alternative approach would be to develop a simple form of control model which can relate outputs to inputs. This control model is much simpler than the original simulation model which generated the outputs from inputs in the first place. Therefore, the control model can be run very quickly to inform the delineation of the input uncertainty space as the desired model behaviour is identified in the solution space. Although this control model would not be accurate in explaining the precise input-output relationship because of its simple structure, the accuracy of the control model can be improved by training the model with many ensembles of inputs and outputs. We suggest the development of this control model in the exploratory modelling process and its comparison with our suggested control data mining and statistical analyses as a future research direction. ■

REFERENCES

- Bankes, S. 1993. "Exploratory modeling for policy analysis." *Operations Research* 41: 435-449.
- Bankes, S. C., R. J. Lempert, and S. W. Popper. 2001. "Computer-assisted reasoning." *Computing in Science & Engineering* 3: 71-77.
- Ben-Haim, Y. 2006. *Info-Gap Decision Theory: Decisions Under Severe Uncertainty*, London, GB: Academic Press.
- Ben-Tal, A. and A. Nemirovski. 2000. "Robust solutions of linear programming problems contaminated with uncertain data." *Math. Program.* 88: 411.
- Bergmann, S., S. Stelzer, and S. Straßburger. 2011. *Initialization of simulation models using CMSD. In Proceedings of the 2011 Winter Simulation Conference (WSC)*, pp. 2223-2234. IEEE.
- Beven, K., and A. Binley. 1992. "The future of distributed models: model calibration and uncertainty prediction." *Hydrological processes* 6: 279-298.
- Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen. 1984. *Classification and regression trees*. CRC Press.
- Bryant, B. P. and R. J. Lempert. 2010. "Thinking inside the box: A participatory, computer-assisted approach to scenario discovery." *Technological Forecasting and Social Change* 77: 34-49.
- Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.
- Deb, K., and A. Pratap. 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE Trans. Evol. Comput.* 6: 182.

- Friedman, J. H., and N. I. Fisher. 1999. "Bump hunting in high-dimensional data." *Statistics and Computing* 9: 123-143.
- Gerst, M. D., P. Wang, and M. E. Borsuk. 2013. "Discovering plausible energy and economic futures under global change using multidimensional scenario discovery." *Environmental Modelling & Software* 44: 76-86.
- Groves, D. G., and R. J. Lempert. 2007. "A new analytic method for finding policy-relevant scenarios." *Global Environmental Change* 17: 73-85.
- Haasnoot, M., J. H. Kwakkel, W. E. Walker, and J. ter Maat. 2013. "Dynamic adaptive policy pathways: A method for crafting robust decisions for a deeply uncertain world." *Global Environmental Change* 23: 485-498.
- Hadka, D. and P. Reed. 2013. "Borg: An auto-adaptive many-objective evolutionary computing framework." *Evol. Comput.* 21: 231.
- Hamarat, C., J. H. Kwakkel, E. Pruyt, and E. T. Loonen. 2014. "An exploratory approach for adaptive policymaking by using multi-objective robust optimization." *Simulation Modelling Practice and Theory* 46: 25-39.
- Iverson, G. R. and H. Norpoth. 1987. *Analysis of variance*, Thousand Oaks, US-CA: Sage Publications.
- Kasprzyk, J. R., S. Nataraj, P. M. Reed, and R. J. Lempert. 2013. "Many objective robust decision making for complex environmental systems undergoing change." *Environmental Modelling & Software* 42: 55-71.
- Kwakkel, J. H. 2017. "The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making." *Environmental Modelling & Software* 96: 239-250.
- Kwakkel, J. H. and E. Pruyt. 2015. "Using System Dynamics for Grand Challenges: The ESDMA Approach." *Systems Research and Behavioral Science* 32: 358-375.
- Kwakkel, J. H., W. E. Walker, and V. A. W. J. Marchau. 2010. "Adaptive airport strategic planning." *European Journal of Transport and Infrastructure Research* 10: 249–273.
- Lempert, R. 2013. "Scenarios that illuminate vulnerabilities and robust responses." *Climatic Change* 117: 627-646.
- Lempert, R. J., B. P. Bryant, and S. C. Bankes. 2008. "Comparing algorithms for scenario discovery." RAND, Santa Monica, US-CA.
- Lempert, R. J. and D. G. Groves. 2010. "Identifying and evaluating robust adaptive policy responses to climate change for water management agencies in the American west." *Technol. Forecast. Soc.* 77: 960.
- Lempert, R. J., N. Kalra, S. Peyraud, Z. Mao, S. B. Tan, D. Cira, and A. Lotsch. 2013. Ensuring Robust Flood Risk Management in Ho Chi Minh City. *Policy Research Working Paper 6465*. World Bank.
- Lempert, R. J., S. W. Popper, and S. C. Bankes. 2003. *Shaping the next one hundred years: new methods for quantitative, long-term policy analysis*. RAND Corporation.
- Maier, H. R., Z. Kapelan, J. Kasprzyk, J. Kollat, L. Matott, M. Cunha, G. C. Dandy, M. S. Gibbs, E. Keedwell, and A. Marchi. 2014. "Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions." *Environmental Modelling & Software* 62: 271-299.
- Marler, R. T., and J. S. Arora. 2004. "Survey of multi-objective optimization methods for engineering." *Structural and multidisciplinary optimization* 26: 369-395.
- McLachlan, G., and D. Peel. 2004. *Finite mixture models*. John Wiley & Sons.
- Moallemi, E. A., F. de Haan, J. Kwakkel, and L. Aye. 2017. "Narrative-informed exploratory analysis of energy transition pathways: A case study of India's electricity sector." *Energy Policy* 110: 271–287.
- Moallemi, E. A., S. Elsawah, and M. J. Ryan, M. J. Submitted-a. "Robust Decision Making and Epoch-Era Analysis: A comparison of two model-based approaches for decision making under uncertainty." *Technological Forecasting & Social Change*.
- Moallemi, E. A., S. Elsawah, and M. J. Ryan. Submitted-b. "Model-based multi-objective decision making under deep uncertainty from a multi-method design lens." *Simulation Modelling Practice & Theory*.
- Moallemi, E. A., and S. Malekpour. 2018. "A participatory exploratory modelling approach for long-term planning in energy transitions." *Energy Research & Social Science* 35: 205-216.
- Montgomery, D. C. 2001. *Design and Analysis of Experiments*. New York: Wiley.
- Pianosi, F., K. Beven, J. Freer, J. W. Hall, J. Rougier, D. B. Stephenson, and T. Wagener. 2016. "Sensitivity analysis of environmental models: A systematic review with practical workflow." *Environmental Modelling & Software* 79: 214-232.
- Schulze, T., S. Strassburger, and U. Klein. 1999. "On-line data processing in simulation models: New approaches and possibilities through HLA." In *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 2*, pp. 1602-1609.
- Walker, W. E., M. Haasnoot, and J. H. Kwakkel. 2013. "Adapt or perish: a review of planning approaches for adaptation under deep uncertainty." *Sustainability* 5: 955-979.
- Woodruff, M., and J. Herman. 2013. *Nondominated sorting for multi-objective problems* [Online]. Available: https://github.com/matthewjwoodruff/pareto.py [Accessed 2017].

ABOUT THE AUTHORS

[Editor: Author biographies were current when the paper was initially published in 2018.]

Enayat A. Moallemi is research associate at the Capability Systems Centre, the University of New South Wales, Canberra. His research lies at the intersection of robust decision making, exploratory modelling, and sustainability transitions. He specifically explores the long-term behaviour of complex systems under deep uncertainty for informing robust decision making.

Sondoss Elsawah is a senior lecturer in the School of Engineering and IT (SEIT) and leads the modelling and simulation effort in the Capability Systems Centre. Dr Elsawah's research program focuses on understanding the behaviour of large complex problems and systemic risks that arise from the interactions between social, ecological, and technological systems. This focus includes how to use that understanding to create computational tools that enable decision makers design and evaluate options to deal with such complexity.

Michael J. Ryan is associate professor and the director of the Capability Systems Centre. In addition, he is the program coordinator for the Master of systems engineering program in the School of Engineering and Information Technology (SEIT). He is the author or co-author of twelve books, three book chapters, and over 200 technical papers and reports.

Appendix A

Table A.1. The first quartile of uncertainty ranges

Uncertain parameter	Range
The risk that an aircraft is lost during operation	0.00026 – 0.00078 (–)
Lifetime of aircraft	37440 – 112252.5 (hour)
Total required flying hours	12 – 59 (hour/week)
Expected time spent by an aircraft in CAP	8 – 17.25 (week)
Time between CAP events	16 – 22 (week)
Expected time spent by an aircraft in DM (Time in DM)	5 – 10 (week)
Time (flying hours) between DM events	200 – 600 (hour)
Expected time spent by an aircraft in OM (Time in OM)	3 – 6 (week)
Time between OM events	50 – 150 (hour)

Table A.2. The fourth quartile of uncertainty ranges

Uncertain parameter	Range
The risk that an aircraft is lost during operation	0.00182 – 0.00234 (–)
Lifetime of aircraft	261877.5 – 336690 (hr)
Total required flying hours	153 – 200 (hour/week)
Expected time spent by an aircraft in CAP	35.75 – 45 (week)
Time between CAP events	34 – 40 (week)
Expected time spent by an aircraft in DM (Time in DM)	20 – 25 (week)
Time (flying hours) between DM events	1400 – 1800 (hour)
Expected time spent by an aircraft in OM (Time in OM)	12 – 15 (week)
Time between OM events	350 – 450 (hour)

Aswin Sukumaran Nair continued from page 53

REFERENCES:

- INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 4. Revised by D. Walden, G. Roedler, K. Forsberg, R. D. Hamelin, and T. M. Shortell. San Diego, US-CA: INCOSE.
- Scaled Agile Framework. 2021a. "Product Management." https://www.scaledagileframework.com/product-management/.
- ———. 2021b. "Product Owner." https://www.scaledagileframework.com/product-owner/.
- ———. 2021c. "Program and Solution Backlogs." https://www.scaledagileframework.com/program-and-solution-backlogs/.

ABOUT THE AUTHOR

[Editor: Author biography was current when the paper was initially published in 2022.]

Aswin Sukumaran Nair is a Principal Engineer with Carrier Corporation, based out of Hyderabad, India. He is a Systems Engineering Practitioner with over 17 years of experience in software and systems development. He is a certified ASEP and has completed PO/PM certification from Scaled Agile Framework.

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

Call for Papers

The *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 activities involve the technologies and system management approaches needed for

- definition of systems, including identification of user requirements and technological specifications;
- development of systems, including conceptual architectures, tradeoff of design concepts, configuration management during system development, integration of new systems with legacy systems, integrated product and process development; and
- deployment of systems, including operational test and evaluation, maintenance over an extended life cycle, and re-engineering.

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 *Systems Engineering* journal is dedicated to all aspects of the engineering of systems: technical, management, economic, and social. It focuses on the life cycle processes needed to create trustworthy and high-quality systems. It will also emphasize the systems management efforts needed to define, develop, and deploy trustworthy and high quality processes for the production of systems. Within this, *Systems Engineering* is especially concerned with evaluation of the efficiency and effectiveness of systems management, technical direction, and integration of systems. *Systems Engineering* is also very concerned with the engineering of systems that support sustainable development. Modern systems, including both products and services, are often very knowledge-intensive, and are found in both the public and private sectors. The journal emphasizes strategic and program management of these, and the information and knowledge base for knowledge principles, knowledge practices, and knowledge perspectives for the engineering of systems. Definitive case studies involving systems engineering practice are especially welcome.

The journal is a primary source of information for the systems engineering of products and services that are generally large in scale, scope, and complexity. *Systems Engineering* will be especially concerned with process- or product-line–related efforts needed to produce products that are trustworthy and of high quality, and that are cost effective in meeting user needs. A major component of this is system cost and operational effectiveness determination, and the development of processes that ensure that products are cost effective. This requires the integration of a number of engineering disciplines necessary for the definition, development, and deployment of complex systems. It also requires attention to the lifecycle process used to produce systems, and the integration of systems, including legacy systems, at various architectural levels. In addition, appropriate systems management of information and knowledge across technologies, organizations, and environments is also needed to insure a sustainable world.

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, with the intent to achieve published papers that

- relate to the field of systems engineering;
- represent new, previously unpublished work;
- advance the state of knowledge of the field; and
- conform to a high standard of scholarly presentation.

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.

Proving the Path: Validating Functional-Outcomes-Driven Tailoring (FODT) Through Practical Applications

Barend Botha, Chief Engineer, Halcon, bwbotha@tadhole.com

ABSTRACT

In a previous article, we proposed functional-outcome-driven tailoring (FODT) as a fresh and structured approach to managing the lifecycle of complex engineered systems. Recognizing that FODT introduces a shift from traditional methodology-centric thinking to functional and outcome-driven tailoring, this follow-up addresses the natural next step: exploring whether the approach holds up in practice and how it can be implemented with confidence.

This article investigates the validity of FODT by referencing real-world analogs, policy alignment, and engineering trends across both traditional sectors (such as defense and aerospace) and broader industries (such as transportation, energy, civil infrastructure, and healthcare). While FODT formalizes a novel name, it is not untested—it draws on principles and practices already emerging in the defense and aerospace sectors. The goal is to support program and engineering leaders in confidently adopting FODT by grounding the approach in current evidence, operational precedents, and structured rollout guidance.

INTRODUCTION: REAFFIRMING THE NEED FOR TAILORED ENGINEERING

Modern engineered systems operate in increasingly dynamic environments, requiring integration across hardware, software, and cyber-physical domains. In this context, rigid adherence to a single systems engineering methodology often falls short [1][2][8]. As outlined in the initial *INSIGHT* FODT article (February 2026), there is a growing need to tailor development methods to the functional goals and operational outcomes of each lifecycle phase [2][8].

This article seeks to validate FODT as both conceptually sound and practically achievable. It draws from established standards, cross-industry practices, and recognized frameworks—including the DoD's Adaptive Acquisition Framework [3], the Digital Engineering Strategy [4], INCOSE's hybridization guidance [2], and NASA's phase-aligned model-based systems engineering (MBSE) approach [5]—to support the transition from strategic intent to actionable engineering guidance.

WHAT IS FUNCTIONAL-OUTCOME-DRIVEN TAILORING (FODT)?

FODT is a lifecycle tailoring framework that aligns methodology selection with the functional purpose and desired outcome of each development phase. Unlike one-size-fits-all frameworks, FODT supports structured flexibility—methodologies are selected, adapted, or combined based on:

- The core engineering function of each phase
- The maturity of requirements and architecture
- The level of technical and operational risk
- The need for traceability, speed, or innovation.

In essence, FODT supports risk-aware engineering without compromising lifecycle rigor. It provides a middle ground between method orthodoxy and uncontrolled ad hoc tailoring, as often advocated in modern engineering governance literature [2][8].

STANDARD ALIGNMENT AND BEST PRACTICE TRENDS

Functional-Outcome-Driven Tailoring (FODT) is not a departure from established systems engineering principles—it represents a structured realization of values already embedded in leading international frameworks. Several standards and industry best practices explicitly advocate tailoring as a key aspect of successful systems engineering, particularly in high-complexity and regulated domains.

At the foundation of lifecycle tailoring lies ISO/IEC/IEEE 15288 [1], which establishes essential processes for system development and encourages adaptation based on the system-of-interest and its context. This standard promotes a pragmatic approach to engineering process application, where method use is conditioned by mission profile, stakeholder environment, and technological maturity. FODT builds on this by translating abstract tailoring advice into a practical method-to-function roadmap.

Complementing this, the INCOSE *Systems Engineering Handbook* (v5.0) [2] advances the case for hybridized methodology deployment. It underscores the importance of tailoring lifecycle processes to suit specific project conditions rather than prescribing rigid procedural templates. FODT operationalizes this hybridization logic by offering a structured reasoning framework for selecting and sequencing methodologies—including V-model, agile, model-based systems engineering (MBSE), and prototyping—based on their alignment to phase-specific functional objectives.

Tailoring is also reinforced in the US DoD Digital Engineering Strategy [4] and its accompanying Systems Engineering Guidebook [3], which call for model-based, outcome-aligned processes that evolve with system maturity. These policies advocate the integration of authoritative data sources, digital thread strategies, and agile responsiveness. All of these are mirrored in FODT's foundational tenets—namely, aligning tools and methods to phase needs and emphasizing engineering intent over documentation formality.

The principle finds further validation in NATO STANAG 4728 [6], which promotes tailoring lifecycle controls according to mission scope and national interpretation. It explicitly recognizes that uniformity at the method level may hinder effectiveness, especially in joint or multinational programs. FODT enables this diversity within alignment by offering structure without imposing uniformity—allowing various teams to contribute to shared outcomes through context-appropriate practices.

Taken together, these international benchmarks confirm that tailoring is not only accepted—it is expected. What FODT contributes is a coherent operational model that turns this expectation into executable guidance. It translates conceptual flexibility into actionable structure, improving how organizations plan, train, and assure their engineering activities.

REAL-WORLD APPLICATIONS ACROSS INDUSTRIES

The principles of functional-outcome-driven tailoring are increasingly reflected in real-world engineering practice, even if not always explicitly labeled as such. Across diverse domains—from government space programs to urban infrastructure, manufacturing, and medical devices—tailoring methodologies based on lifecycle phase function and system maturity has become a key enabler of efficiency, risk control, and stakeholder relevance. The following examples illustrate how different sectors are already applying FODT-aligned logic in their development and delivery strategies:

Civil Engineering: Large-scale civil infrastructure projects—such as urban transport systems, bridges, and water resource management programs—often follow FODT-aligned practices without formally naming them as such. These projects utilize staged prototyping, risk-based validation, and stakeholder-aligned milestones to phase development activities. Functional tailoring enables adaptive engagement with public stakeholders and regulatory bodies throughout project evolution. References such as the UK Infrastructure and Projects Authority's Project Routemap and ASCE lifecycle guidelines illustrate how outcome-driven tailoring supports large-scale delivery while managing risk and complexity [15][16]. Functional tailoring enables adaptive engagement with public stakeholders and regulatory bodies throughout project evolution.

Industrial & Automotive: In industrial sectors, firms such as Airbus and Lockheed Martin, as well as automotive original equipment manufacturers (OEMs), integrate simulation, MBSE, and agile practices to manage complexity across system layers. These companies apply different development methods at different stages—such as agile sprints for embedded software, traditional V-model for physical systems, and concurrent engineering for integration phases—demonstrating phase-function alignment as embodied by FODT principles [9][10]. BAE Systems reinforces this approach by formalizing governance policies for tailoring through their internal engineering framework [11].

Healthcare and Energy Systems: Tailored lifecycle management is particularly critical in highly regulated sectors such as healthcare devices and nuclear energy. Organizations in these sectors customize validation, verification, and documentation practices based on compliance obligations, technology maturity, and patient or environmental risk. FODT provides a valuable logic for these domains, helping teams balance innovation with regulatory precision while adapting methodologies to shifting risk profiles. Standards such as IEC 62304 for medical software, US Federal Drug Administration (FDA) software validation guidance, and International Atomic Energy Agency (IAEA) SSR-2/1 Rev.1 for nuclear systems reflect the demand for risk-based, phase-specific tailoring in lifecycle practices [17][18][19]. Tailored validation and verification strategies are used to manage compliance, risk, and innovation timelines in complex regulatory environments.

Defense & Aerospace: Defense and aerospace programs are at the forefront of structured tailoring, with initiatives such as the US Department of Defense's Adaptive Acquisition Framework (AAF) and NASA's phased MBSE implementation [3][5]. These frameworks demonstrate how method selection can be aligned with mission profiles, technical risk, and maturity milestones, facilitating flexibility without compromising assurance or traceability. The Project Scorpion initiative—led by France in partnership with NATO members—demonstrates how tailoring enables diverse national teams to contribute subsystems using domestically preferred engineering methods, while still aligning to a shared mission outcome [12]. This is a textbook example of FODT's potential in multinational and system-of-systems environments.

CASE ANALYSES: LIFECYCLE-PHASE ALIGNMENT IN ACTION

In support of functional-outcome-driven tailoring, several organizations provide rich illustrations of how lifecycle methodologies are successfully tailored to serve the function and maturity of each phase.

The Adaptive Acquisition Framework (AAF) [3] exemplifies how tailoring at the program level can be guided by a triad of key drivers: urgency, complexity, and risk. Through its flexible structure, AAF offers multiple acquisition paths—each with distinct expectations for documentation, oversight, and speed. The result is an acquisition strategy that reflects the nature of the mission rather than the constraints

of a default process. This embodies FODT's emphasis on dynamic method selection anchored to the strategic purpose of each lifecycle stage.

Similarly, NASA's use of MBSE [5] reveals a differentiated application of modeling that evolves across the development lifecycle. During early-phase trade space exploration, high-level behavioral and architectural modeling is used to compare alternatives and assess feasibility. As the project advances into integration and verification, the modeling becomes progressively detailed and formalized, supporting configuration control and system validation. This phase-aware adaptation of MBSE mirrors the tailored maturity approach that FODT formalizes.

A comparable pattern emerges in industrial practice at Airbus [9] and BAE Systems [11], where methodology is flexibly adapted according to both system maturity and the phase's dominant engineering objective. Teams move fluidly between MBSE, simulation, prototyping, and lean design practices—selecting each based on where it best addresses the challenges of a given phase. For instance, MBSE may guide interface definition in system architecture, while simulation supports failure mode exploration during subsystem validation. These decisions are not driven by doctrine, but by function—just as FODT proposes.

What all these examples reveal is a unifying insight: effective engineering governance does not emerge from applying a method consistently, but from applying it appropriately. FODT captures this insight and turns it into an implementation-ready framework—offering teams not only the freedom to adapt, but the structure to justify and synchronize that adaptation across an entire program lifecycle.

RISK MITIGATION AND MATURITY ASSURANCE THROUGH FODT

Managing complexity in engineered systems—particularly in regulated, safety-critical, or high-stakes environments—requires both adaptability and the ability to mitigate risk while ensuring development maturity under increasingly compressed timelines. One of FODT's core strengths is its capacity to reduce lifecycle risk by aligning engineering methods with phase-specific functional needs [8].

FODT reduces systemic risk by:

- Preventing late rework through early-phase simulation, stakeholder co-design, and high-fidelity conceptual modeling [5][8].
- Enabling modular reviews based on subsystem readiness, supporting staggered verification strategies and incremental maturity assessment [7][8].
- Avoiding overengineering by tying technique fidelity to functional value, especially in low-risk or well-understood domains [2][4].

These risk mitigation practices are echoed in globally accepted guidance: the Guide to the Systems Engineering Body of Knowledge (SEBoK) emphasizes early validation, functional milestone reviews, and flexible verification strategies as critical to lowering lifecycle cost and failure rates [8], while the US Defense Acquisition University (DAU) promotes tailoring based on system-specific risks and complexity [7]. By connecting method choices directly to functional intent and risk posture, FODT equips organizations to manage uncertainty while preserving engineering control.

TOWARD STRUCTURED ADOPTION: MAKING FODT WORK

Successfully implementing functional-outcome-driven tailoring does not require organizations to abandon their existing lifecycle frameworks. Rather, FODT complements and strengthens them by introducing structured logic for selecting the most appropriate method at the right time. This ensures that tailoring is not ad hoc but governed, auditable, and aligned with engineering objectives and stakeholder outcomes.

To transition FODT from theory to routine practice, organizations can take the following enabling actions:

- **Establish a Tailoring Governance Board**: Create or assign authority within program or engineering governance to review and approve lifecycle tailoring decisions. This board ensures that tailoring is justified based on project complexity, risk, and functional intent, and promotes consistency while enabling flexibility [3][7].
- **Develop Lifecycle Fluency Across Teams**: Equip technical and managerial staff with the ability to think critically about lifecycle phases. Training, mentoring, and community-of-practice initiatives can help build shared understanding of function-phase alignment and tailoring rationale. This supports a shift from rigid process following to purpose-driven decision-making [2][8].
- **Embed Tailoring Tools into Program Infrastructure**: Tools such as heatmaps, method-function matrices, and tailoring logs should be incorporated into project artifacts, milestone templates, and digital engineering environments. These artifacts provide traceability and strengthen governance of tailoring practices [4].
- **Promote Functional Decision Logic at Key Milestones**: Encourage teams to routinely ask: "*Does this method best support the function and outcome of this phase?*" Embedding this thinking in planning, design, and review stages fosters responsiveness, outcome alignment, and engineering transparency.

This structured approach allows organizations to institutionalize tailoring decisions that align with strategic intent and engineering rigor. The core adoption activities—governance setup, team enablement, digital integration, and decision logic—should be embedded within the organization's standard engineering practices to ensure consistency, traceability, and long-term value realization.

IMPLICATIONS FOR POLICY AND PROCUREMENT

One of the defining strengths of FODT lies in its compatibility with established engineering policies, regulatory frameworks, and procurement practices. Rather than contradicting formal standards, FODT operationalizes them—providing a logic-driven, traceable pathway for lifecycle tailoring that enhances compliance and confidence among both practitioners and auditors.

From a governance and assurance standpoint, FODT elevates tailoring from an informal exception to a rigorously justified process. By aligning method selection with phase-specific functions, risks, and system maturity, it creates a transparent rationale that satisfies both engineering oversight teams and external regulators [1][2][4].

This compatibility is clearly reflected in ISO/IEC/IEEE 15288 [1], which advocates tailoring lifecycle processes to suit the system-of-interest. FODT transforms that abstract guidance into tangible engineering logic. It also aligns with NATO STANAG 4728 [6], which permits tailoring for national interpretation while maintaining shared capability outcomes—especially valuable in multinational programs.

Beyond compliance, FODT offers strategic benefits to acquisition and supplier evaluation processes. Modern procurement frameworks increasingly require bidders to demonstrate a contextual understanding of system complexity, integration risk, and lifecycle strategy [3][7]. Embedding FODT principles into tender requirements—such as requesting a justification for tailored lifecycle methodology—enables selection committees to distinguish between checklist-driven proposals and those grounded in functional engineering logic.

Tools such as method-function alignment matrices and tailoring heatmaps provide structure for evaluating whether

proposed development approaches fit the system maturity and risk profile of each phase [4]. In this way, FODT not only strengthens internal development discipline but also elevates the quality and transparency of supplier engagement.

Ultimately, FODT bridges the often-competing needs for innovation and regulatory assurance. It replaces arbitrary method selection with defensible logic, positioning tailoring not as a governance loophole but as a disciplined, auditable process in line with the highest international systems engineering standards.

Positioning FODT in the Systems Engineering Landscape

While FODT introduces a distinctive vocabulary and structured reasoning, it does not diverge from the systems engineering canon. Rather, it consolidates and formalizes tailoring principles that are increasingly essential in modern engineering environments.

FODT most directly aligns with ISO/IEC/IEEE 15288's guidance on tailored lifecycle processes [1], extending it through structured heuristics, function-phase mapping, and documentation traceability. It gives form to the high-level tailoring philosophy articulated by that standard and others.

Its phase-by-phase decision logic also parallels risk-based and context-sensitive approaches advocated by the US Defense Acquisition University (DAU) [7], and resonates with the tailoring flexibility embedded in MBSE and agile practices.

FODT complements adjacent frameworks such as value-focused systems engineering (VFSE) and capability-based planning by extending value orientation into the methodological layer. Rather than optimizing only requirements and architectures, FODT ensures the development method itself supports value delivery in each phase.

It further reinforces concepts critical in mission engineering and system-of-systems integration—domains where diverse teams and heterogeneous systems must harmonize their practices toward shared operational goals [3][6].

In this sense, FODT is not a new methodology, but a meta-framework that enables structured, justifiable use of existing methods. It provides teams with governance scaffolding and shared language to apply engineering discipline with purpose, consistency, and flexibility.

CONCLUSION: A RESEARCH-GROUNDED EVOLUTION, NOT A LEAP

As engineering organizations face escalating complexity, tighter development cycles, and evolving operational demands, traditional lifecycle models—while foundational—can no longer be applied as one-size-fits-all solutions. The functional-outcome-driven tailoring (FODT) approach presents itself not as a radical alternative, but as a structured evolution. It distills decades of systems engineering wisdom, refines it through contemporary needs, and integrates it with real-world practices already visible across defense, aerospace, infrastructure, healthcare, and industry [1][2][3].

FODT is far from being speculative. It reflects the tailoring provisions of ISO 15288 [1], the value-driven guidance of INCOSE [2], and the digital lifecycle strategies embedded in frameworks such as the US DoD Adaptive Acquisition Framework (AAF) [3]. Rather than discarding traditional methods, FODT applies them with greater discernment—selecting the right method for the right phase, for the right reason.

The core value of FODT lies in its ability to:

- **Provide a structured, explainable basis for tailoring**, enabling decisions to be justified, audited, and improved over time.
- **Foster coherence between engineering function and operational outcome**, ensuring that system development remains mission-relevant across lifecycle stages.
- **Increase transparency into maturity and readiness**, helping teams assess where they are, what remains uncertain, and how best to proceed.

In short, FODT offers a way forward that balances control with adaptability. It empowers engineering teams, acquisition leaders, and assurance authorities with a shared, logic-based framework that aligns process with purpose. The future of complex systems engineering will not be shaped by abandoning legacy practices—but by transforming them into a more intelligent, agile, and outcome-aware discipline. FODT offers exactly that bridge. ■

REFERENCES

- [1] ISO/IEC/IEEE 15288:2015. "Systems and Software Engineering — System Life Cycle Processes." International Organization for Standardization, Geneva, CH.
- [2] INCOSE. 2023. *Systems Engineering Handbook*, Version 5.0. International Council on Systems Engineering.
- [3] US Department of Defense. 2022. "Adaptive Acquisition Framework (AAF) and Systems Engineering Guidebook." Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)).
- [4] US Department of Defense. 2018. "Digital Engineering Strategy." Office of the Deputy Assistant Secretary of Defense for Systems Engineering.
- [5] US NASA Systems Engineering Handbook. 2016. SP-2016-6105 Rev2, National Aeronautics and Space Administration.
- [6] NATO STANAG 4728. nd. "Allied Systems Engineering Management." North Atlantic Treaty Organization, Brussels, BE.
- [7] US Defense Acquisition University (DAU). 2020. "Engineering of Defense Systems, Tailoring Guidelines." DAU Press.
- [8] SEBoK v. 2.8. 2023. "Guide to the Systems Engineering Body of Knowledge." The Trustees of the Stevens Institute of Technology. Available: https://www.sebokwiki.org.
- [9] Airbus Defence and Space. 2022. "Digital Design and Engineering Initiatives." Corporate Technical Briefing.
- [10] Lockheed Martin. 2021. "Model-Based Engineering and Digital Thread Strategy." Internal White Paper.
- [11] BAE Systems. 2022. "Engineering Governance and Tailoring Policy." Internal Engineering Policy Manual.
- [12] Project Scorpion (France). 2021. "Programme SCORPION: Capacité Collaborative Interarmes." French Ministry of the Armed Forces, DGA.
- [13] Systematic A/S. 2022. "SitaWare Suite for Command and Control." NATO Interoperability Case Study.
- [14] US Department of Defense. 2023. "Contract Requirements Checklist and Proposal Evaluation Templates." Defense Contract Management Agency (DCMA).
- [15] Infrastructure and Projects Authority. 2023. Project Routemap: A Guide to Governance and Delivery, UK Government.
- [16] American Society of Civil Engineers (ASCE). 2022. Infrastructure Lifecycle Guidelines. ASCE Publications.
- [17] IEC 62304:2006. Medical Device Software – Software Life Cycle Processes. International Electrotechnical Commission.
- [18] US Food and Drug Administration (FDA). General Principles of Software Validation; Final Guidance for Industry and FDA Staff. FDA.
- [19] International Atomic Energy Agency (IAEA). 2016. Safety of Nuclear Power Plants: Design, SSR-2/1 Rev.1. IAEA, Vienna, AT.

CUSA 2026

Where Systems Engineers Shape the Future of Integrated Engineering

May 18–21, 2026 | Reno, Nevada

For decades, the INCOSE community has led the evolution of systems thinking—advancing the methods, rigor, and discipline that modern engineering depends on. Today, that influence is expanding into a new era of connected, model0driven, simulation-powered innovation.

Welcome to CUSA 2026—the next evolution of the MCSS community. A unified engineering symposium built for Systems Engineers, MBSE practitioners, Design Engineers, MODSIM experts, and innovators across the Americas. A place where the virtual twin of the product becomes the backbone of end-to-end engineering.

Why INCOSE Members Belong at CUSA

A Symposium Rooted in Systems Engineering CUSA is built on the legacy of the MCSS community—preserving the depth, discipline, and technical excellence that INCOSE members value most.

A Broader, More Connected Engineering Ecosystem As engineering evolves, so does the community supporting it. CUSA brings together Systems Engineering, MBSE, design, simulation, and MDSIM experts into one integrated environment—reflecting how real teams work today

A Platform to Influence the Future From requirements to architecture, from simulation to manufacturing, from concept to sustainment—CUSA is where practitioners shape the next generation of engineering practices and the CATIA roadmap itself.

What You'll Experience

Deep-dive MBSE and Systems Engineering sessions
Cross-discipline collaboration with design, simulation, and MODSIM experts
Customer case studies from leading innovators
Hands-on learning with the 3DEXPERIENCE platform
A community committed to elevating engineering excellence across the Americas

Exclusive INCOSE benefits: $300 off registration + onsite SEP Exam availability

Be Part of the Movement

CUSA is more than a conference—it's the launch of a new North American flagship for the CATIA and MBSE community. A place where INCOSE members can lead, influence, and accelerate the future of integrated engineering.

Join Us in Reno

CUSA 2026 | May 18–21, 2026 Be part of the story from the very beginning.

Register Now

DASSAULT SYSTEMES

INCOSE EMEA

EMEASEC 2026
Europe, Middle East, Africa
29 Aug - 2 Sep 2026
Linköping, Sweden

SAVE THE DATE FOR EMEASEC2026

Join us at EMEASEC2026, the Systems Engineering Conference for the Europe, Middle East, and Africa (EMEA) Sector of INCOSE.

29 AUG – 2 SEP 2026

LINKÖPING, SWEDEN