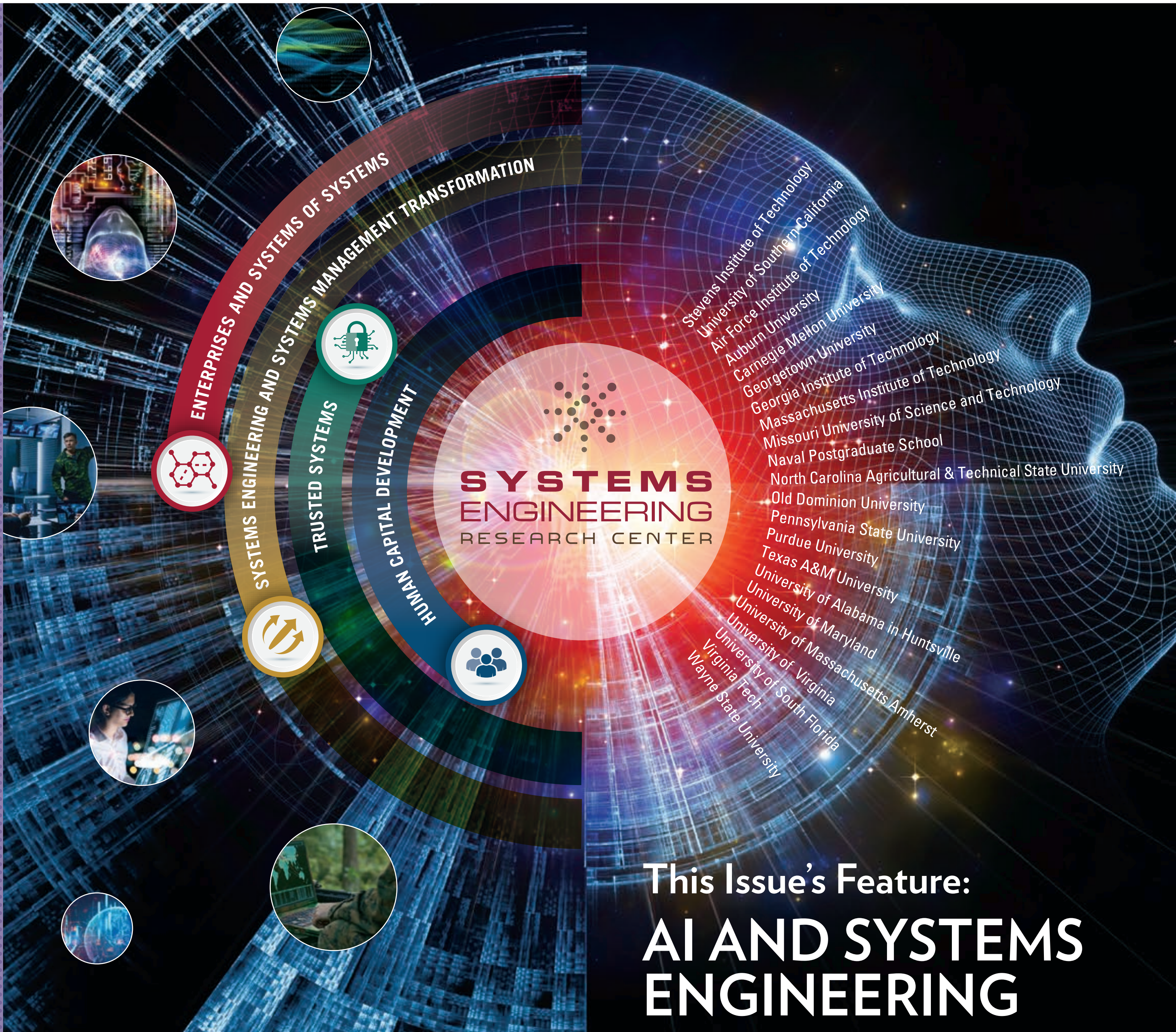
# INSIGHT




This Issue's Feature:

## AI AND SYSTEMS ENGINEERING

MARCH 2020
VOLUME 23 / ISSUE 1

LOCKHEED MARTIN

This issue is sponsored by the Lockheed Martin Corporation.

INCOSE

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

CSD&MAsia2020

## Submit Your Work
## to the 4th International Complex Systems Design & Management Asia Conference

## CALL FOR PAPERS

**New & Last Submission Deadline: March 21, 2020**

Mastering complex industrial systems is a strategic challenge. The CSD&M conference is a widely **open event dedicated to academic researchers and industrial & governmental actors**, interested in **Model-Based Systems Engineering and Digital Transformation in Engineering**. The conference language is English.

**September 21-22, 2020 | Renaissance Beijing Wangfujing Hotel, Beijing, China**

### CONFERENCE GENERAL CHAIRS

**KROB Daniel**
Institute Professor, Ecole Polytechnique, France
INCOSE Fellow
President, CESAMES

**Dr. ZHANG Xinguo**
Distinguished Professor,
Director of Complex Systems Engineering Research Center, Tsinghua University, China
INCOSE ESEP, President of INCOSE Beijing Chapter

### PROGRAM COMMITTEE

**Academic members**

**LI Lefei, Tsinghua University, China (Co-Chair)**
CARDIN Michel, Imperial College, United Kingdom
DE WECK Olivier, MIT, USA
FENG Yanghe, National University of Defense Technology, China
GAO Xinghai, Beihang University, China
GUAN Ruoxi, Tsinghua University, China
HAMMAMI Omar, ENSTA ParisTech, France
JACKSON Peter, SUTD, Singapore
KOO John, Hong Kong Applied Science and Technology Research Institute, China
LV Yisheng, Chinese Academy of Sciences, China
RAUZY Antoine, NTNU, Norway
WANG Chen, Tsinghua University, China
WANG Yingxun, Beihang University, China

**Industrial members**

**ZHANG Hongjun, China State Shipbuilding Corporation, China (Co-Chair)**
AMBROSO Annalisa, Framatome, France
DAURON Alain, Renault-Nissan Alliance, France & Japan
FERROGALINI Marco, Airbus, European Union
HOU Shudong, Huawei, China
ONG Robert, No Magic, Thailand
QIE Yongjun, Aviation Industry Corporation of China, China
SCHREINEMAKERS Paul, INCOSE, European Union
STARK Rainer, Fraunhoffer Institute, Germany
SUN Zhixiao, Aviation Industry Corporation of China, China
WEN Yuejie, Academy of Space Technology of China
XING Chenguang, Aviation Industry Corporation of China, China
YEE Victor, IBM, China
ZHANG John, Huawei, China

### CONFERENCE TOPICS

**Industrial domains**
Aerospace, automotive, communications & e-services, defense & security, electronics & robotics, energy & environment, healthcare & welfare services, high tech, logistics & supply chain, ship & transportation, software industry, urban & public infrastructures

**Scientific & technical topics**
Systems fundamentals, systems architecture & engineering, model-based systems engineering, big data for industrial applications, systems manufacturing, systems modeling & simulation, system optimization, system project management, systems safety, systems serviceability, systems technology & policy

### KEY INFORMATION

**Important dates**

- **Submission deadline:** ~~Feb. 21~~, *postponed to* **March 21, 2020**
- Results announcement: April 30, 2020
- Final version: May 29, 2020

**Publication & indexation policy**
Accepted papers will be included into the CSD&M Asia 2020 conference proceedings, published by **Springer Verlag** and referenced & indexed in **DBLP**, **EI Compendex**, **ISI Clarivate** and **Scopus**

**How to submit**
Access to submission procedure - https://www.2020.csdm-asia.net/call-for-papers-submission-procedure/

**Organizers** **Co-operators & Sponsors**

CESAMES AVIC CSSC INCOSE AIRBUS APSYS DASSAULT SYSTEMES

Contact: contact@csdm-asia.net / csaacsaa@126.com – Conference web site: https://www.2020.csdm-asia.net/

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

MARCH 2020 VOLUME 23 / ISSUE 1



# Inside this issue



*INSIGHT volume 23, no. 1 is sponsored by the Lockheed Martin Corporation.*

# About This Publication

## INFORMATION ABOUT INCOSE

INCOSE's membership extends to over 18,000 individual members and more than 100 corporations, government entities, and academic institutions. Its mission is to share, promote, and advance the best of systems engineering from across the globe for the benefit of humanity and the planet. INCOSE charters chapters worldwide, includes a corporate advisory board, and is led by elected officers and directors.

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

## OVERVIEW

***INSIGHT*** is the magazine of the International Council on Systems Engineering. It is published four times per year and features informative articles dedicated to advancing the state of practice in systems engineering and to close the gap with the state of the art. ***INSIGHT*** delivers practical information on current hot topics, implementations, and best practices, written in applications-driven style. There is an emphasis on practical applications, tutorials, guides, and case studies that result in successful outcomes. Explicitly identified opinion pieces, book reviews, and technology roadmapping complement articles to stimulate advancing the state of practice. ***INSIGHT*** is dedicated to advancing the INCOSE objectives of impactful products and accelerating the transformation of systems engineering to a model-based discipline.

Topics to be covered include resilient systems, model-based systems engineering, commercial-driven transformational systems engineering, natural systems, agile security, systems of systems, and cyber-physical systems across disciplines and domains of interest to the constituent groups in the systems engineering community: industry, government, and academia. Advances in practice often come from lateral connections of information dissemination across disciplines and domains. ***INSIGHT*** will track advances in the state of the art with follow-up, practically written articles to more rapidly disseminate knowledge to stimulate practice throughout the community.



## PERMISSIONS

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

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

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

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

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

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

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

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

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


**Permissions Department – 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.org

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

- March 2020 issue – 2 January
- June 2020 issue – 2 April
- September 2020 issue – 1 July
- December 2020 issue – 1 October

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 18,000 individual members and uncounted employees and students of more than 100 CAB organizations worldwide. Readership includes engineers, manufacturers/purchasers, scientists, research & development processionals, presidents and CEOs, students and other professionals in systems engineering.

| Issuance | Circulation |
|---|---|
| 2020, Vol 23, 4 Issues | 100% Paid |

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

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

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

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

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

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

**Or please contact:**
Dan Nicholas, Associate Director – Sciences, Corporate Sales Wiley
PHONE: +1 716-587-2181
E-MAIL: dnicholas@wiley.com

## CONTACT

**Questions or comments concerning:**

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

**Advertising—*please contact:***
Dan Nicholas, Associate Director
Sciences, Corporate Sales
PHONE: +1 716-587-2181
E-MAIL: dnicholas@wiley.com

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

## ADVERTISER INDEX March volume 23-1



***INSIGHT volume 23, no. 1 is sponsored by the Lockheed Martin Corporation.*** LOCKHEED MARTIN

## CORPORATE ADVISORY BOARD — MEMBER COMPANIES

321 Gang, Inc.
Aerospace Corporation, The
Airbus
Airbus Defense and Space
AM General LLC
Analog Devices, Inc.
Analytic Services
Aras Corp
Australian Department of Defence
Aviation Industry Corporation of China, LTD
BAE Systems
Bechtel
Boeing Company, The
Bombardier Transportation
Booz Allen Hamilton Inc.
C.S. Draper Laboratory, Inc.
CACI International, Inc.
Carnegie Mellon University Software Engineering Institute
Change Vision, Inc
Colorado State University
Cornell University
Cranfield University
Cubic Corporation
Cummins, Inc.
CYBERNET MBSE
Defense Acquisition University
Deloitte Consulting
DENSO Create, Inc.
Drexel University
Eindhoven University of Technology
Embraer S.A.
ENAC
Federal Aviation Administration (U.S.)
Ford Motor Company
Fundacao Ezute
General Dynamics
General Electric
General Motors
George Mason University
Georgia Institute of Technology
IBM
Idaho National Laboratory
ISAE SUPAERO
ISDEFE
ISID Engineering, LTD
iTiD Consulting, Ltd
Jacobs Engineering
Jama Software
Jet Propulsion Laboratory
John Deere & Company
Johns Hopkins University
KBR, Inc.
KEIO University
L3 Harris
Leidos
Lockheed Martin Corporation
Los Alamos National Laboratory
ManTech International Corporation
Maplesoft
Massachusetts Institute of Technology
MBDA (UK) Ltd.
Missouri University of Science & Technology
MITRE Corporation, The
Mitsubishi Aircraft Corporation (Mitsubishi Heavy Induftries Group)
National Aeronautics and Space Administration
National Security Agency – Enterprise
Naval Postgraduate School
Nissan Motor Co, Ltd
No Magic/Dassault Systems
Noblis
Northrop Grumman Corporation
Pacific Northwest National Laboratory
Penn State University
Perspecta (formerly Vencore)
Prime Solutions Group, Inc.
Project Performance International
Raytheon Corporation
Roche Diagnostics
Rolls-Royce
Saab AB
Safran Electronics and Defence
SAIC
Sandia National Laboratories
Shell
Siemens
Sierra Nevada Corporation
Singapore Institute of Technology
Skoltech
SPEC Innovations
Stellar Solutions
Stevens Institute of Technology
Strategic Technical Services
Swedish Defence Materiel Administration
Systems Engineering Directorate
Systems Planning and Analysis
Tetra Pak
Thales
TNO
Trane
Tsinghua University
TUS Solution LLC
UK MoD
United Technologies Corporation
University of Arkansas
University of Connecticut
University of Maryland
University of Maryland, Baltimore County
University of New South Wales, The, Canberra
University of Southern California
University of Texas at Dallas
University of Texas at El Paso, The
US Department of Defense, Deputy Assistant Secretary of Defense for Systems Engineering,
Veoneer, Inc
Virginia Tech
Vitech Corporation
Volvo Construction Equipment
Woodward Inc
Worcester Polytechnic Institute- WPI
Zuken, Inc

# From the Editor-in-Chief and Systems Engineering Research Center Executive Director

**William Miller,** insight@incose.org; and **Dinesh Verma,** dverma@stevens.edu

*INSIGHT's* mission is to provide informative articles on 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 research as captured in *Systems Engineering*, the Journal of INCOSE, also published by Wiley. INCOSE thanks corporate advisory board (CAB) member Lockheed Martin for sponsoring *INSIGHT* in 2020 and welcomes additional sponsors, who may contact the INCOSE director for marketing and communications at marcom@incose.org.

The March 2020 issue of *INSIGHT* addresses *augmented and artificial intelligence (AI) for systems engineering (AI4SE) and systems engineering for augmented and artificial intelligence (SE4AI)*. SE4AI addresses the transformation we need in methods, procedures, and tools (MPTs) to engineer systems with embedded AI to be fit for purpose and doing no (unintended) harm. AI4SE addresses challenges that have to be overcome to leverage AI in the practice of systems engineering much as a lever or pulley provides mechanical advantage to perform work in Newtonian mechanics.

Tenets of systems engineering from control systems engineering are that engineered systems be observable, controllable (to assure system stability with tolerable errors), and identifiable (Möller 2016). AI is in the forefront of technology advances with widely publicized innovations in image identification, diagnostics, and autonomy; AI applications gone awry are newsworthy in both the popular and scientific/engineering media.

AI methods can be broadly classed as rule-based and neural-network based. Rule-based methods are relatively mature with decades of experience in application and are well-understood. In contrast, much is unknown of the contextually driven behavioral characteristics of neural network-based AI, commonly referred to as machine learning and deep learning. Neural network performance is critically dependent on the datasets used to train the algorithms and whose actions currently cannot be guaranteed to be *fit for purpose* to meet the attributes of elegance that systems accomplish their intended purposes, be resilient to effects in real-world operation, while minimizing unintended actions, side effects, and consequences (Griffin 2010).

The articles represent ongoing research in the Systems Engineering Research Center (SERC), a university-affiliated research center (UARC) of the US Department of Defense, in addressing the AI4SE and SE4AI challenges as stated in their research roadmap for AI and autonomy (SERC 2019). The SERC (www.sercuarc.org), operated by Stevens Institute of Technology with the University of Southern California (USC) as principal collaborator, leverages the research and expertise of faculty and researchers from 22 collaborator universities throughout the US. The *INSIGHT* editorial staff and SERC theme editors Dinesh Verma, Tom McDermott, and Kara Pepe thank the authors for their contributions. The lead article by Tom McDermott, Dan DeLaurentis, Peter Beling, Mark Blackburn and Mary Bone overviews the AI and autonomy research roadmap and maps the subsequent articles to the parts of the roadmap.

The SERC articles are within the scope of the systems community initiative on the future of systems engineering (FuSE) that has AI4SE and SE4AI as an initial project. The May 2019 *INSIGHT* addressed other FuSE projects on systems engineering principles and foundations for systems engineering (F4SE). *INSIGHT* intends to report on FuSE related topics on an ongoing basis.

We hope you find *INSIGHT*, the practitioners' magazine for systems engineers, informative and relevant. Feedback from readers is critical to the quality of *INSIGHT*. We encourage letters to the editor at insight@incose.org. Please include "letter to the editor" in the subject line. *INSIGHT* also continues to solicit contributions for special features, standalone articles, book reviews, and op-eds. For information about *INSIGHT*, including upcoming issues, see https://www.incose.org/products-and-publications/periodicals#INSIGHT. ■

## REFERENCES


- Griffin, M. D. 2010. "How do we fix system engineering?" 61st International Astronautical Congress. Prague, CZ; 27 September – 1 October.
- Möller, D. P. F. 2016. *Guide to Computing Fundamentals in Cyber-Physical Systems: Concepts, Design Methods, and Applications*. Springer, CH.
- Systems Engineering Research Center. 2019. "Research Roadmaps 2019-2020." https://sercuarc.org/wp-content/uploads/2020/01/ROADMAPS_2.3.pdf.

## ABOUT THE AUTHORS

**Dinesh Verma** is professor in systems engineering at Stevens Institute of Technology and the former dean of its School of Systems and Enterprises. He is an INCOSE Fellow and 2019 chair of the Fellows Committee. Dr. Verma is executive director of the Systems Engineering Research Center (SERC), the first university affiliated research center (UARC) established by the US DoD for systems engineering research. Prior to these roles, he served as technical director at Lockheed Martin Undersea Systems, in Manassas, Virginia, US, in the area of adapted systems and supportability engineering processes, methods, and tools for complex system development and integration. He has a BS in mechanical engineering, MS in industrial and systems engineering, and PhD in industrial and systems engineering.

**William Miller** is *INSIGHT* editor-in-chief, principal executive analyst at Innovative Decisions, Inc., and adjunct professor in the School of Systems and Enterprises, Stevens Institute of Technology. He has forty-seven years of experience in the conceptualization and engineering application of communications and information technologies, products and services in both commercial and government sectors. Mr. Miller has managed projects including positions as chief systems engineer and program manager funded by Bell Laboratories, AT&T, and the US government. He leads the systems community initiative on the future of systems engineering (FuSE) and previously served as INCOSE secretary (1997-1998, 2003-2006) and technical director (2013-2014). He has BS and MS degrees in electrical engineering with emphasis in control systems and digital design.

*INSIGHT* vol 23-1 cover credit: Kristine Gallo

MARCH 2020
VOLUME 23/ ISSUE 1

# AI4SE and SE4AI: A Research Roadmap

**Tom McDermott,** tmcdermo@stevens.edu; **Dan DeLaurentis,** ddelaure@purdue.edu; **Peter Beling,** beling@virginia.edu; **Mark Blackburn,** mblackbu@stevens.edu; and **Mary Bone,** mbone@stevens.edu



**ABSTRACT**

In 2019, the Research Council of the Systems Engineering Research Center (SERC), a US Defense Department sponsored University Affiliated Research Center (UARC), developed a roadmap structuring and guiding artificial intelligence (AI) and autonomy research. This paper presents that roadmap and key underlying Digital Engineering transformation aspects both enabling traditional systems engineering practice automation (AI4SE), and encourage new systems engineering practices supporting a new wave of automated, adaptive, and learning systems (SE4AI).

**KEYWORDS:** systems engineering, artificial intelligence, machine learning, automation, research

## INTRODUCTION

Systems engineering is undergoing a digital transformation that will lead to transformational advances facilitating systems engineering use of artificial intelligence (AI) and machine learning (ML) technology to automate many routine engineering tasks. At the same time applying AI, ML, and autonomy to complex and critical systems encourages new systems engineering methods, processes, and tools. A 2019 future of systems engineering (FuSE) workshop, hosted by the International Council on Systems Engineering (INCOSE), first used the terms AI for systems engineering and systems engineering for AI to describe this dual transformation (Miller 2019). The "AI4SE" and "SE4AI" labels have become metaphors for an upcoming rapid evolutionary phase in the systems engineering Community. AI4SE applies augmented intelligence and machine learning techniques to support systems engineering practices. Goals in such applications include achieving scale in model construction and confidence in design space exploration. SE4AI applies systems engineering methods to learning-based systems' design and operation. Key research application areas include developing principles for learning-based systems design, life cycle evolution models, and model curation methods.

To better understand and focus on this evolution, the Systems Engineering Research Center's (SERC) research council, a US Defense Department sponsored university affiliated research center (UARC), developed a roadmap to structure and guide research in Artificial Intelligence (AI) and autonomy. This paper presents that roadmap. The SERC research strategy aligns three mission areas supported by four research areas, shown in Figure 1.

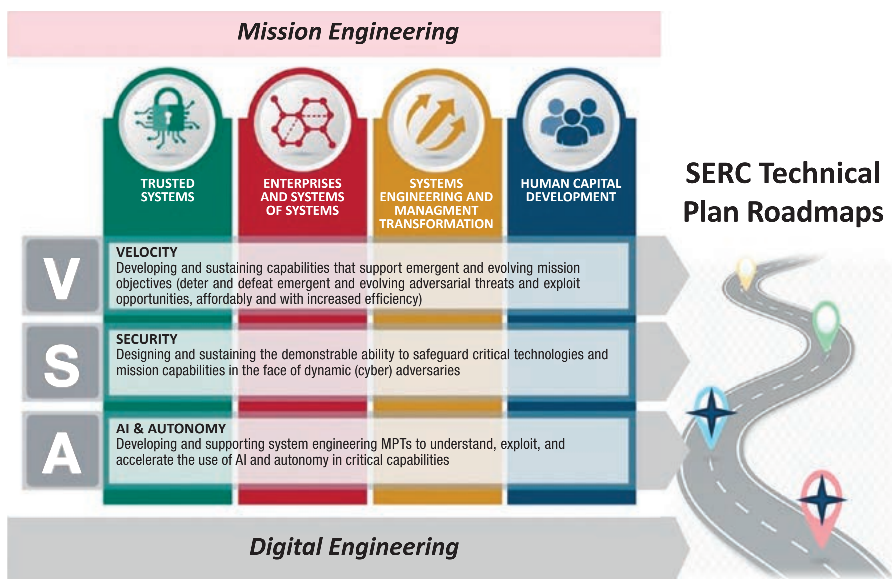



*Figure 1. SERC research areas and missions*

The research areas are enterprises and systems of systems (ESOS), trusted systems (TS), systems engineering and systems management transformation (SEMT), and human capital development (HCD). The mission areas the SERC is addressing are:

- Velocity: Developing and sustaining timely capabilities supporting emergent and evolving mission objectives (deter and defeat emergent and evolving adversarial threats and exploit opportunities affordably and with increased efficiency).
- Security: Designing and sustaining the demonstrable ability to safeguard critical technologies and mission capabilities in the face of dynamic (cyber) adversaries.

- Artificial Intelligence (AI) and Autonomy: Developing and supporting system engineering methods, processes, and tools to understand, exploit, and accelerate AI and autonomy use in critical capabilities.

Digital engineering, which transforms systems engineering from document-based methods and artifacts to linked digital data and models, enables these.

The SERC technical plan, which outlines a 5-year vision for each research area, guides the mission and research areas (Systems Engineering Research Center 2019). Four roadmaps, developed collaboratively by the SERC research council over a 5-month effort in 2019, provide crosscutting mission area details (Systems Engineering Research Center 2019). Each roadmap has a set of verticals leading to a visionary outcome or set of outcomes, and a set of capabilities needed to meet those long-term outcomes. The listed capabilities reflect SERC research and other research, both active and prioritized by our sponsors and the systems engineering community in general. By sharing this work, we hope to guide both SERC research and systems engineering transformations.

## DIGITAL ENGINEERING AS THE ENABLER

Digital engineering forms the basis for all three SERC crosscutting missions and resulting research roadmaps. Systems engineering is in a transformation process based on the data use (an authoritative source of truth) and collaboration using models (collaborative integrated modeling environments). The SERC developed a digital engineering research roadmap which aligns with the five goals of our DoD sponsor's strategy: 1) model use for decision making; 2) the authoritative source of truth (AST); 3) technological innovation; 4) collaborative environments; and 5) workforce and cultural evolution (Zimmerman 2017). Figure 2 shows this roadmap.

Digital engineering's progression begins with data integration in the AST, then semantic model integration. We expect to see advances in Augmented Intelligence – model use and "big data" bringing automation to engineering processes, system quality, and certification. Augmented Intelligence forms the fourth "wave" of the digital engineering transformation in our roadmap and is AI4SE's core. There are multiple ongoing developmental efforts in this roadmap (yellow circles) and a few solid transition areas (blue circles). Summarized below are those relevant to AI4SE:

- Tool and Domain Taxonomies and Ontologies: we look to underlying engineering and programmatic data interoperability through future ontologies. Graph databases for linked data are becoming more prominent in model-based systems engineering (MBSE). Taxonomies provide the starting point for building ontologies, ultimately enabling AI-based reasoning on the underlying data. This is the transformational infrastructure in AI4SE.
- Semantic Rules: based on knowledge representations such as ontologies, semantic rules provide the basis for reasoning (AI) about completeness and consistency.

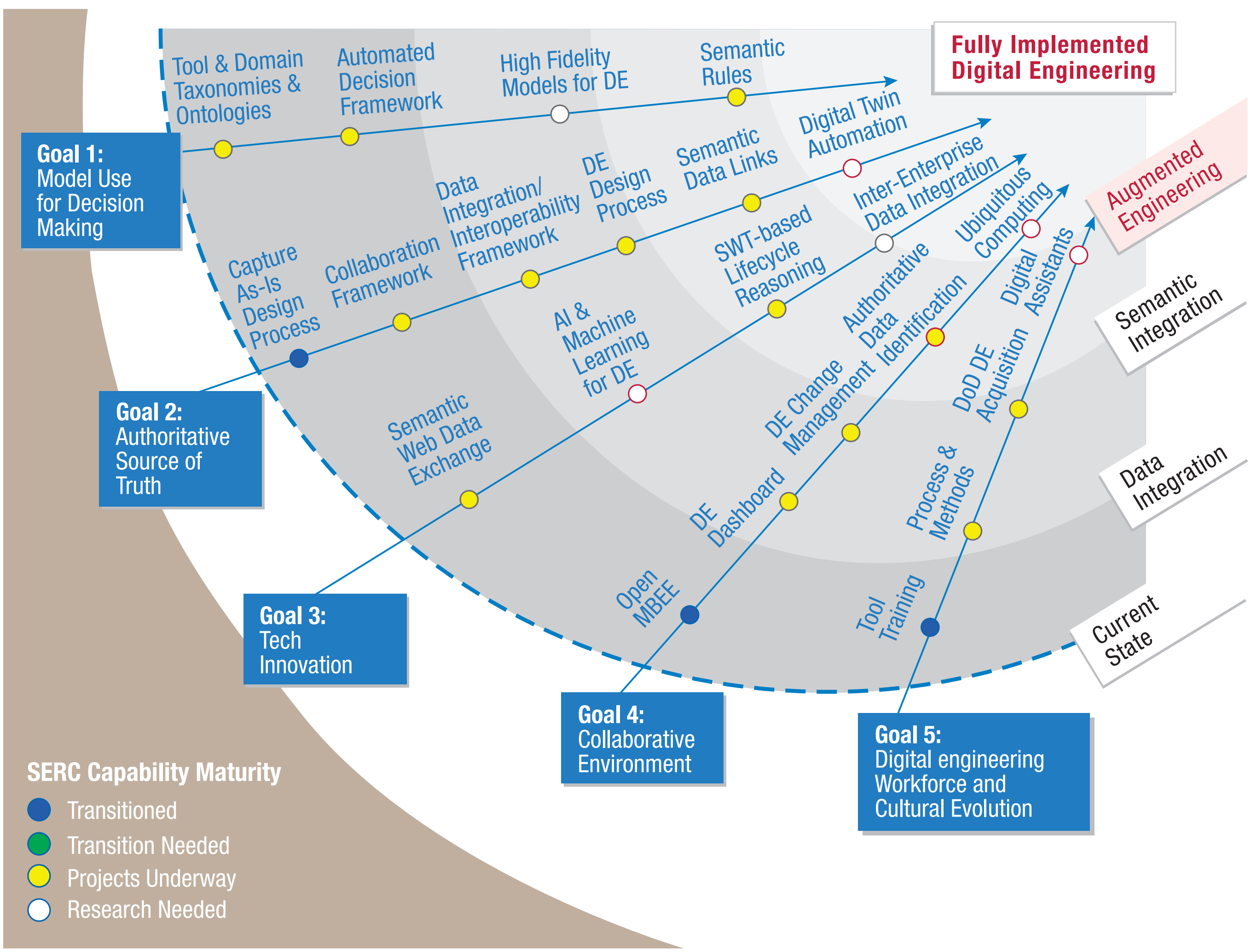



*Figure 2. SERC digital engineering roadmap*

In their article "Knowledge Representation with Ontologies and Semantic Web Technologies to Promote Augmented and Artificial Intelligence in Systems Engineering," Blackburn et al. provide a detailed discussion of the importance of a semantic basis for engineering as an enabler for AI.

- Inter-Enterprise Data Integration: data/information seamlessly update/exchange continuously in "real-time" cutting across the entire enterprise (technical, manufacturing, cost, risk).
- Automated Decision Frameworks: combining ontologies, descriptive models, and analytics provide a decision making framework related to alternative analysis across any decision type, characterized by an objective hierarchy (basis for decision) and the underlying data.
- Authoritative Data Identification: automating how to find the "authoritative data," assisted by AI/ML, and understanding what the user is looking for.
- Digital Assistants: trusted and automated AI guidance in engineering design and decision-making processes.
- Digital Twin Automation: this is the "end game" – fully dynamic virtual system copies built from the same models as the real systems running in parallel to physical systems and updating from the same data feeds as their real counterparts. This leads to opportunities for parallel learning in the real and simulated environments.

## A FRAMEWORK FOR AI AND AUTONOMY RESEARCH

The SERC goal in artificial intelligence and autonomy is to lead the systems engineering transformation to dynamic processes leveraging the speed and rigor of rapidly evolving modeling, simulation, and analysis. Computational technologies and computational intelligence enable this tranformation. The technical domain of AI, ML, and autonomy encompasses a broad range of emerging methods, processes, tools, and technologies. Although the technology will pass through "waves" led by digital data integration, semantic integration, and augmented engineering, we cannot yet define a concrete roadmap leading toward a central outcome. We can categorize research areas in an evolutionary framework expected to transform the engineering domain. Technological innovation's "double S" curve (Figure 3) provides an effective categorization of systems engineering research contributions to emerging technology and their application. These include abstraction and high-level design methods, design for "X", and design for test and certification, leading to ability to specify the technology into requirements, tools to accelerate and scale design, modeling and simulation at the mission level, and finally to operational test and incorporation.

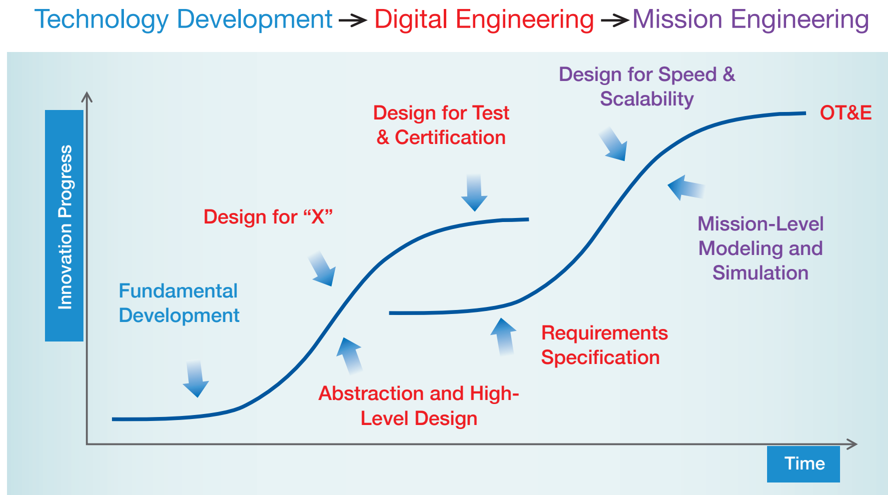



*Figure 3. Systems engineering support to technological innovation*

This domain's advance, measured by the levels in our framework, form emergence "waves" as research transitions to practice. In the AI and autonomy roadmap, shown in Figure 4, green circles identify mature research areas where transition into the systems engineering domain is possible today, the yellow circles are ongoing systems engineering research activities, and the white circles are areas where systems engineering research should begin. The roadmap centers around four major verticals:

- AI/ML Technology – technological advances supporting use in systems;
- Automation and Manned/Unmanned Teaming – methods and tools ensuring beneficial and safe use of resulting automation; and
- Systems Engineering Process Evolution through Digital Engineering – the evolution of systems engineering process to learning technologies and automation and the transition to a digital engineering data driven basis for engineering which allows automation and learning.

Described below, organized by the four verticals, are the research areas:

### *AI and Machine Learning Technology*

A continuing set of research needs in the basic technologies of AI/ML and autonomy both enabling AI4SE and evolving SE4AI processes are:

- Multi-Modal AI: holistic multi-modal data analysis (sensed, discussed, written, social, environmental) aiming to produce actionable intelligence for human decisions (systems engineering design decisions in this case). Most ML applications today are narrow, focusing on single modalities. However, AI applications fusing multiple modalities have been in service for years. Linking ML-based learning to AI-based decision support algorithms provides one foundation of AI4SE.
- Cognitive Bias: a need to counter intentional or unintentional misled decision-making in AI systems, caused by the training data, adaptative learning over time, or intentional misinformation. Current narrow ML applications are subject to such biases. Augmented intelligence algorithms for SE must be trustworthy.
- Hybrid Human/AI Systems: automated machine reasoning agents helping humans understand complex data. Current AI/ML applications work well with human reasoning in narrow well-defined contexts. Highly automated complex systems (self-driving vehicles) face human versus machine control challenges. Both the AI4SE process, and new systems engineering processes defining interactions in SE4AI evolution require seamlessly integrating human and machine tasking and learning. In future systems engineering practice, human systems integration will no longer be a specialty area but will be front and center to system definition.

In his article "As Smart as a Human? Leveraging models of human intelligence to assess the intelligence of systems," Brown provides a thoughtful approach to evaluate human and machine system "intelligence."

- Contextual Sensemaking: research into AI perceiving and learning context, such as DARPA's Third Wave initiatives (Launchbury 2015), will provide

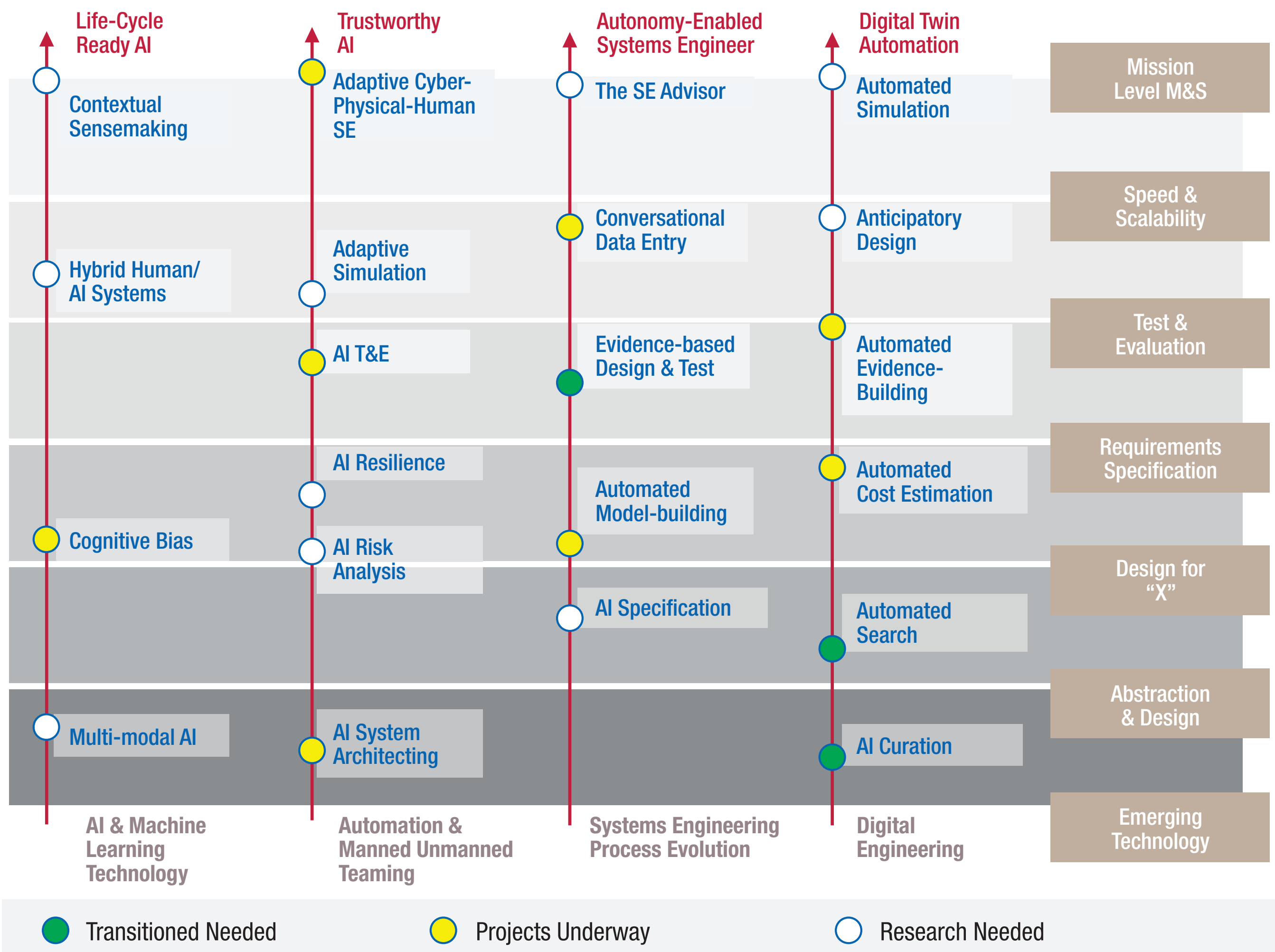



*Figure 4. SERC AI and autonomy roadmap*

foundational platforms for future hybrid human/AI systems.

- Life-Cycle Ready AI: the technology development vertical outcome is a body of new systems engineering methods and tools addressing increasingly complex learning and adapting systems with the rigor traditional to the systems engineering domain. We are beginning to see system level issues addressed in basic AI/ML applications research. Figure 5 details multiple research topics addressing dependable/safe AI produced by DeepMind, a leading company in basic AI research, (Ortega Maini and DeepMind 2018). AI safety researchers can categorize key topics in terms of traditional systems engineering and control areas, such as system modeling, stability, and resilience. However, SE4AI processes specific to the management and control AI/ML system life cycles remain lacking.

### *Automation and Manned-Unmanned Teaming*

Most AI/ML application's goals are automating previously manual or partially automated human processes. Thus, automation and human interaction research (manned-unmanned

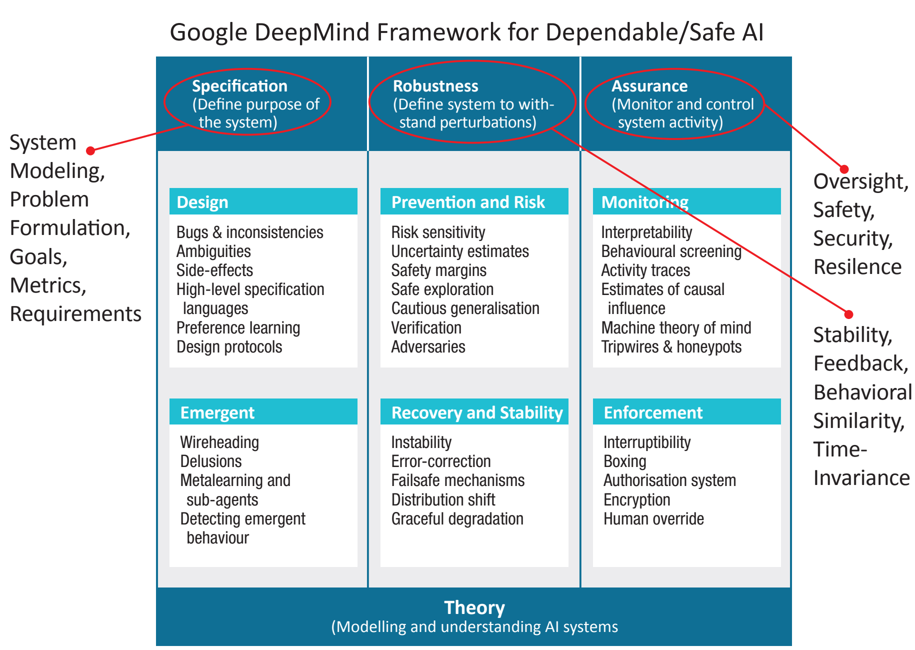



*Figure 5. SE practices influencing current AI/ML research (adapted from (Ortega Maini and DeepMind 2018))*

teaming) is an essential part of the systems engineering of these systems. Systems engineering included automation for many years, but systems adapting and learning in their deployment, particularly at scale, are a challenge for current systems engineering processes. Research will help develop the methods, processes, and tools that ensure these systems are trustworthy.

- AI Risk Analysis: methods, processes, and tools need to connect system risk analysis results with AI software modules related to those risks. A fundamental risk ontology for AI/ML systems needs development.
- AI Resilience: future systems should include resilience capabilities addressing AI related failure modes. This will involve system level behavior monitoring preventing the AI algorithms from performing operations outside predefined boundaries. research includes methods to evaluate AI/ML predictive algorithm generation (generally part of "explainable AI" research).
- AI Test and Evaluation: methods for addressing AI-related system test and evaluation addressing these systems' ability to adapt and learn from changing deployment contexts.

In her article "Test and Evaluation for Artificial Intelligence," Freeman reviews the challenges facing the test and evaluation community to evaluate learning systems.

- Adaptive Simulation: computer-based simulation and training supporting non-static objectives and/or goals (pickup games, course of action analysis) necessary to provide contextual learning environments for these systems. Most simulations today derive from static decomposition and will evolve to include learning agents which evolve the simulation over time. Real and simulated co-learning (digital twins) will be a standard systems development form.
- Adaptive Cyber-Physical-Human Systems: future systems will employ different types of human and machine learning to flexibly respond to unexpected or novel situations, using plan and goal adjustment and adaptation, learning from experience, and continuous task adaptation. Research challenges in the area include simulations reflecting human behavior and learning, and methods to address different time domains in the real and machine worlds.

In his article "Exploiting Augmented Intelligence in Systems Engineering and Engineered Systems," Madni discusses a framework for incorporating Augmented Intelligence in both engineered systems and systems engineering processes.

- Trustworthy AI: resulting AI systems that self-adapt while maintaining rigorous safety and security and policy constraints are this research vertical's primary outcome. Trust includes the machine's ability to meet human behavior standards and the impact to human situational awareness as trust is turned over to the machine.
- System Architecting of AI: the system architecting process may change as automation grows into more complex systems. System architectures must support learning and adaptation, and more agile change processes. A primary component of future architecting will be parallel development and comparison of in vivo (real) and in silico (virtual) deployments.

In their article "Motivating a Systems Theory of AI," Cody, Adams, and Beling discuss a system theoretic approach for architecting AI/ML systems.

*Systems Engineering Process Evolution*

Both AI/ML technology development and the needs to manage and control the resultant automation will drive Systems Engineering evolution. The following are some critical research areas driving the systems engineering disciplines.

- AI Abstraction: achieving AI4SE at scale will require library tools and methods abstracting AI algorithms and programming modules to engineering's general use. Developing such abstractions transformed the microelectronics industry. Similar developments will bring AI/ML to a broader engineering domain.

In their article "A Systems Engineering Approach for Artificial Intelligence: Inspired by the VLSI Revolution of Mead & Conway," Wade and Buenfil discuss a model using abstraction to bring AI/ML to the systems engineering community based on similar experience in the VLSI revolution.

- AI specification: a requirements specification and management process for adaptation and learning in systems does not exist today and needs developing. Coding environments will integrate much of this, building from today's Devops environments to systems models. Static requirements specification will only partially support future systems.
- Evidence-based Design and Test: SE4AI requires formal methods and processes moving from explicit composition verification to evidence building strategies focusing on adaptation acceptance or prevention. The SE4AI domain needs assurance methods linked to validation methods. In the AI4SE practice, ML based tools to find and reason on evidence will be a key enabler.

In their article "Validation of AI-Enabled and Autonomous Learning Systems," Collopy and Sitterle discuss the challenges of AI/ML system validation and suggest a framework for formal validation.

- Conversational Data Entry: human-computer interaction processes to convert natural language and other media to formal models will greatly improve knowledge transfer and consistency in future systems engineering tools.
- The systems engineering Advisor: we envision, as an outcome, a conversational system automating many mundane data entry, exploration, engineering calculation tasks, and many workflows.

In his article "AI as SE: Augmented Intelligence for Systems Engineers" Rouse reviews what this advisor might do and an architectural framework for its development.

- Autonomy Enabled Systems Engineers: in the long-term, system engineers become masters at deploying autonomy as a design variable and AI/ML as a design tool. Although not listed as a vertical in this roadmap, autonomy and AI skillset development in the engineering domain is quickly becoming a need.

*Digital Engineering*

Systems Engineering's transformation to first a model-based discipline then to a data-driven discipline, as discussed in the digital engineering roadmap, will enable rapid evolution of AI/ML technologies into our tools. Some of the critical areas are:

- AI Curation: data management and digital engineering data and model curation is a core transformational need for systems engineering. Data curation will be necessary to support evolving AI capability application, and AI/ML tools will see extensive use in managing the data sets. Training data curation is a new role in systems engineering and will be critical to future systems.
- Automated search, model-building, and cost estimation: applying ML to historical data and relationships will greatly

improve systems engineering process speed and consistency. Applying statistical methods and AI algorithms in these activities has been present for a number of years, but ML technologies will greatly increase these tools' accessibility and use.

- Automated evidence building: certification process automation via models and quality assurance will be a helpful AI/ML evolution outcome, particularly for systems-of-systems and distributed development and test activities.
- Anticipatory design: an expected and desired outcome of increasing engineering practice automation is a tool anticipating system emergence (failures) from design & operational data. This should lead to more trustworthy systems.
- Automated Simulation: simulation use to train and evaluate ML in an emerging research area. This is a generative adversary network's evolution. System theoretical methods (system mathematical frameworks) for training AI/ML algorithms in complex systems is an emerging research area.
- Digital Twin Automation–Industry 4.0 initiatives envision real-time continuous learning from real system and shadow simulations. The digital twin evolution from generally static models to fully dynamic simulations, even at mission level, will transform not only the systems engineering process, but the products we develop.

## VELOCITY

Velocity and agility are critical future system characteristics, both for the deployed system and the system developing and maintaining the deployed system. With the fusion of development in operations, DevOps, the delineation between these is disappearing. A research roadmap for velocity (Systems Engineering Research Center 2019) is the most difficult to articulate as current organizational practice and methodology implementations root it. One might ask where is the needed research? Velocity, with SERC's defense and other government sponsors, centers on three goals: architecting systems for continuous development and deployment, leading an agile transition across large government and contractor systems, and collaborative integrated modeling environment role as an enabler. These interweave with the AI/autonomy research thrusts (and their motivating needs) highlighted earlier.

Overall our vision is to enable systems engineering's transformation from sequential, document-driven, highly constrained practices toward much faster, flexible mission and enterprise-oriented approaches enabled by advances in modeling, simulation, data-driven analysis, and artificial intelligence. This paper does not discuss the velocity roadmap in detail, but the primary research verticals in this area strive for application in two areas: improved mission engineering processes and adaptive system creation. Research areas include rapid development of systems as platforms, architecting these platforms for DevOps enabled systems and environments, and DevOps practice execution in the systems engineering process.

## SECURITY

The SERC security roadmap (Systems Engineering Research Center 2019), also not presented in detail here, focuses on critical engineered systems such as cyber-physical systems, embedded systems, and weapon systems. These are critical, highly assured systems. The roadmap recognizes attributes such as security and resilience as critical system properties, and assurance as a process that yields an evidentiary case that a system is trustworthy with respect to the properties its stakeholders legitimately rely upon. Ongoing SERC security research focuses on three areas: (1) prevent, detect, and mitigate security vulnerabilities; (2) design, model, and conduct analysis of trustworthiness (safe and secure aspects) of complex cyber-physical system capabilities and behaviors; and (3) develop models, processes, and tools to assure the trustworthiness of system behaviors/ performance evolves increasingly driven by machine learning, autonomous capabilities, and manned-unmanned teaming.

Systems based on AI/ML are vulnerable to adversarial attacks on the learning algorithms. This might be through conventional cyber-attack vectors or through learning algorithm input data disruption (sensor data) via physical means in the environment. Additionally, the learning process' complexity and non-determinism mean, even in the absence of adversarial action, field condition performance may diverge from testing performance.

Engineering for AI/ML resilience is beginning. Today, few understand how to design learning-based systems resilient to an intelligent adversary's attacks or to use, environment, and condition changes occurring over a critical system's long lifecycle. However, these goals are similar to engineering resilience against cyber threats where the aim is inhibiting adversarial action, enduring an attack by maintaining minimum mission capability, and finally restoring full functionality in a timely manner. As part of the security roadmap research, SERC has developed numerous design patterns and technical methods for detecting and responding to cyber-attacks, alongside methodologies and tools supporting cyber-resilient design and vulnerability assessment (Beling Horowitz and Fleming 2019)( Carter Adams Barkirtzis Sherburne Beling Horowitz and Fleming 2019). This engineering for cyber resilience will provide a foundation for the study of engineering for AI resilience.

## SUMMARY

Roadmapping various research opportunities is important to communicate broad research goals and to connect researchers working on individual projects to larger impact opportunities. It also serves to help professional societies, government, industry, and academia focus their strategic goals. Systems engineering is transforming, enabled by digital engineering and shifted by the SE4AI and AI4SE concepts. These interlink with the need for methods, processes, and tools leading to adaptive, continually developed and deployed systems (velocity) while ensuring security and resilience are inherent in the outcomes. Perhaps the systems engineering community's biggest challenge is to adapt itself to these transformative concepts' challenges. ■

## REFERENCES


- Beling, P., B. Horowitz, and C. Fleming. 2019. "Model-Based Engineering for Functional Risk Assessment and Design of Cyber Resilient Systems." SERC Technical Report SERC-2019-TR-002.
- Carter, B., S. Adams, G. Bakirtzis, T. Sherburne, P. Beling, B. Horowitz, and C. Fleming. 2019. "A Preliminary Design-Phase Security Methodology for Cyber-Physical Systems." *Systems*. 7(2):21. doi:systems 7020021
- Launchbury, J. 2015. "A DARPA Perspective on Artificial Intelligence." https://www.darpa.mil/about-us/darpa-perspective-on-ai.
- Miller, W. 2019. "Future of Systems Engineering." Paper presented at INCSOE 2019 International Symposium. Torrance, US-CA, January 28.
- Ortega, P.A., V. Maini, and DeepMind Safety Team. 2018. "Building Safe Artificial Intelligence: Specification, Robustness, and Assurance." https://medium.com/@deepmindsafetyresearch/building-safe-artificial-intelligence-52f5f75058f1.
- Systems Engineering Research Center. 2019. "11th Annual SERC Sponsor Research Review (SSRR)." https://sercuarc.org/event/ssrr-2019/.
- Systems Engineering Research Center. 2019. "SERC 2019-2023 Technical Plan." https://sercuarc.org/wp-content/uploads/2019/03/SERC-Technical-Plan-2019-v1.0.pdf.

- Zimmerman, P. 2017. "DoD Digital Engineering Strategy." Paper presented at the 20th Annual National Defense Industrial Association (NDIA) Systems Engineering Conference, Springfield US-VA, October 23-26.

## ABOUT THE AUTHORS

**Tom McDermott** is a systems engineering consultant, practitioner, and research leader involved in numerous forward-leaning systems engineering initiatives. He serves as SERC's deputy director at Stevens Institute of Technology in Hoboken, NJ. With the SERC he develops new research strategies and leads research on digital engineering transformation, education, security, and artificial intelligence applications. Mr. McDermott also teaches system architecture concepts, systems thinking and decision making, and engineering leadership. He is a lecturer in Georgia Tech's Professional Education college, where he leads a masters level course on systems engineering leadership and offers many continuing education short courses. He consults with multiple organizations on enterprise modeling for transformational change, and often serves as a systems engineering expert on government major program reviews. He serves on the board of directors for INCOSE as director of strategic integration.

**Peter A. Beling** is a professor of systems engineering at the University of Virginia. Dr. Beling's research interests include decision-making in complex systems, emphasizing artificial intelligence, adaptive decision support systems, and model-based approaches to system-of-systems design and assessment. His research has found application in a variety of domains, including mission-focused cybersecurity, reconnaissance and surveillance, and prognostic and diagnostic systems. He directs the University of Virginia site of the Center for Visual and Decision Informatics, a National Science Foundation industry/university cooperative research center, and the Adaptive Decision Systems Laboratory, which focuses on data analytics and decision support in cyber-physical systems.

**Daniel DeLaurentis, PhD,** is a professor in Purdue's School of Aeronautics & Astronautics. Since 2017, Dr. DeLaurentis also serves as the director of Purdue's Institute for Global Security and Defense Innovation (i-GSDI) in the University's Discovery Park. Dr. DeLaurentis is chief scientist and member of the research council for the SERC, where he has led multiple projects including one that developed an analytical workbench for system of systems (SoS) architecture analysis, design and evolution. He is a longtime member of INCOSE.

**Mark R. Blackburn, PhD**, is a senior research scientist with Stevens Institute of Technology and principal at KnowledgeBytes. Dr. Blackburn's is the principal investigator (PI) on several SERC research tasks for both US Navy NAVAIR and US Army CCDC Armaments Center on systems engineering transformation through model-centric engineering. He has also been PI on a FAA NextGen and National Institute of Standards and Technology projects and has received research funding from the National Science Foundation. He develops and teaches a course on systems engineering for cyber physical systems. He is a member of the SERC research council, and core member of the semantic technologies for systems engineering initiative. Prior to joining Stevens, Dr. Blackburn worked in industry for more than 25 years. Dr. Blackburn holds a PhD from George Mason University, a MS in mathematics (emphasis in computer science) from Florida Atlantic University, and a BS in mathematics (computer science option) from Arizona State University.

**Mary Alice Bone, PhD**, is a staff researcher for the Systems Engineering Research Center (SERC) at Stevens Institute of Technology and an associate professor II for the University of South-Eastern Norway. She received a BS in aerospace engineering from Missouri S&T in 2001, a MEng in systems engineering from Iowa State University in 2007, and a PhD in systems engineering from Stevens Institute of Technology in 2015. Currently she is helping lead the SERC digital engineering transformation research tasks for both US Army CCDC Armaments Center and Naval Air Systems Command. Her research includes semantic web technology to enable digital engineering, system reference architectures, system complexity, system modularity, and model-based systems engineering to enable synchronized-continuous engineering throughout the whole system life cycle.

# Knowledge Representation with Ontologies and Semantic Web Technologies to Promote Augmented and Artificial Intelligence in Systems Engineering

**Thomas Hagedorn**, thagedorn@umass.edu; **Mary Bone**, mbone@stevens.edu; **Benjamin Kruse**, bkruse@stevens.edu; **Ian Grosse**, grosse@umass.edu; **Mark Blackburn**, mblackbu@stevens.edu


**ABSTRACT**

This article discusses knowledge representation using ontologies and semantic web technologies to enable artificial intelligence (AI) for Systems Engineering. Technology trends indicate new methods and tools for digital engineering will incorporate AI and machine learning (ML) technologies . ML techniques support classification, clustering, and association identification, but struggle to explain the rationale for decision making, where multi-domain semantic modeling and rule-based reasoning can excel. Knowledge representation plays a key role in applying this type of AI. Ontologies are a means to domain modeling and reasoning required across Digital Thread domains instantiated in digital system models (DSM). These evolve over time as digital twins, which co-evolve with physical instantiations of a DSM. Semantic technologies and ontologies formalize knowledge as an enabler for reasoning, with interoperable ontologies enabling reason about systems engineering across domains.

## INTRODUCTION

As engineered systems have evolved to higher complexity, formal knowledge representation has become a solution to help efficiently manage complex system design and manufacturing. Knowledge representation refers to many techniques attempting to encode information about the world, often into machine readable syntaxes (Chandrasegaran, Ramani, Sriram, Horváth, Bernard, Harik, and Gao 2013). These representations facilitate automated reasoning using logical deductions and rules to infer new information from datasets. This is powerful in systems engineering contexts where subject matter experts often make decisions based on understanding cross domain knowledge.

The challenge researchers from the Systems Engineering Research Center (SERC) have faced since 2013 is finding ways to develop complex systems in less time (Bone, Blackburn, Rhodes, Cohen, and Guerrero 2018), leading to the belief that a digital thread is the path forward to meet this challenge. This paper discusses a case study demonstrating how artificial/augmented intelligence (AI) may facilitate augmented design tools and enable a digital thread distinct from, although potentially synergistic with, techniques such as machine learning. AI in a context based on theoretical knowledge rather than big data, focuses applications like digital assistants. Early in this effort SERC researchers concluded a drastic change in how engineers design systems was not an option, and the solution must allow engineers to work naturally allowing creativity in system design. The key is to provide engineers with information and knowledge they need when they need it using computer reasoning and semantic web technologies (SWT).

## BACKGROUND

Knowledge representation for systems engineering offers broad potential to introduce AI capabilities into existing model-based systems engineering (MBSE) and digital engineering (DE) practices. Engineering fields are technical, and systems engineering involves multiple disciplines. Moreover, projects require multiple non-compatible, discipline-specific tools. This poses challeng-

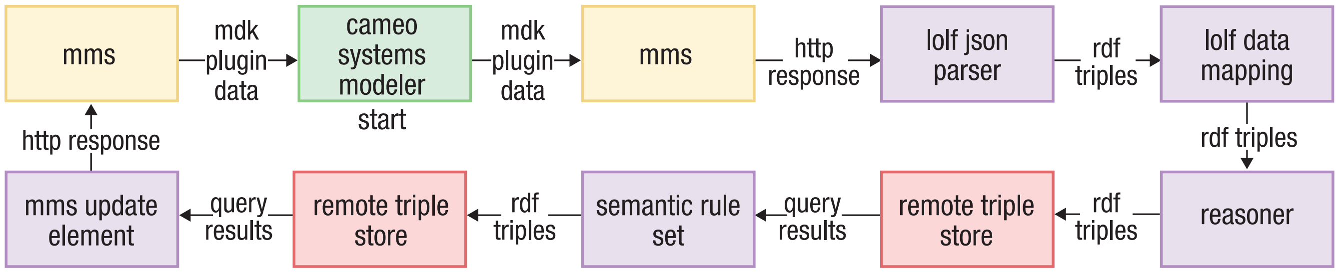



*Figure 1. Round trip of demonstration data*

es for a knowledge-based AI tool for systems engineering. Representation of knowledge must be consistent across domains, or the AI tools will replicate existing data siloes. The AI tool must also accommodate tool-specific data models. SWT and ontologies offer tools suited to these challenges.

SWT refers to standards, software tools, and methodologies aiming to enrich data with semantically meaningful tags. This semantically enhanced data can inform intelligent software tools (Berners-Lee, Hendler, and Lassila 2001). Ontologies are label taxonomies corresponding to classes of things and a permissible relation between these types (Gruber 1995) (Gruber 1993). Unlike conventional database schema, ontologies codify knowledge domains rather than specific data, and so can be consistent across applications. In systems engineering, knowledge representation with ontologies and SWT permit data about a system from different tools to be made interoperable (Bone, Blackburn, Kruse, Dzielski, Hagedorn, and Grosse 2018), and facilitate automated reasoning upon combined data.

## CASE STUDY

### *Overview*

The case study system has multiple subsystems which have their own components and properties such as weight. A system model specifies these traits, and various analysis tools provide their value estimates. In a simple case, deducing the entire system's weight is simple. System models can use tools like parametric diagrams to obtain the final weight, but these need repeated defining across models. If one wanted to add another trait like cost, the process would be similar but would need a separate definition. As the model becomes more complicated and more tools start informing the system analysis, this approach becomes arduous.

The same systems engineering problem can use ontologies and SWT to provide software platforms aggregating disparate information into a single, tool-agnostic source. The software uses its semantics to define general solutions enabled by automated, AI type reasoning and simple rule-based inferences. This platform, called the Interoperability and Integration Framework (IoIF), provides a platform for using one or more ontologies and associated reasoning.

This paper discusses a system model implemented with blocks describing the system's physical make-up, requirements specifying permissible bounds on various system traits, and secondary elements representing analysis data about the system. A duplicate system with intentional errors facilitated error finding demonstration purposes. The model also included a signoff element set in the Open Model Based Engineering Environment's (OpenMBEE) View Editor, allowing an individual to indicate part(s) of the model as reviewed and approved http://www.openmbee.org.

OpenMBEE Model Management System (MMS) implements a data server for models, web script layer, and representational state transfer application programming interface (REST API), while the view editor allows document creation of documents standard "views" to explore models stored in MMS. The REST API allows remote interactions with data stored in MMS's graph data server via HTTP requests, allowing model modification or content retrieval as elements written in JavaScript Object Notation (JSON), a lightweight data interchange format.

The case study comprises several steps (Figure 1). First, system model creation or update in Cameo System Modeler, a graphical tool for creating SysML models. The model posts to a cloud-based model management server implementing MMS and the view editor via the model development kit plugin. Once signed off in the view editor (Kruse and Blackburn 2019), IoIF retrieves the model through the MMS REST API. MMS returns JSON elements parsed into IoIF's SWT architecture. Ingested data then transforms into a tool-agnostic data representation pattern derived from an ontology ecosystem. Logical reasoning and rules inspect the transformed data to tabulate the system weight, identify inconsistencies, and generate documentation and updated elements.

Updates and signoff changes return to the model in MMS. The view editor or in Cameo Systems Modeler can then observe these changes by synchronizing the tool with MMS.

The next sections discuss SWT theory and implementation in this workflow's context.

### *Ontologies and Ontology Systems*

Ontologies are human and machine-readable domain knowledge formalizations (Gruber 1993; Gruber 1995). Human readable portions provide a taxonomy of domain terms, definitions, guidance for use, and information facilitating deployment within a larger SWT ecosystem. Machine-readable portions contain logical axioms enabling automated reasoning on the ontology itself and data aligned to it (Smith 1998). Together these provide a controlled lexicon used to tag data and representations to contextualize it. The ontology and aligned data form a directed data graph with vertices indicating relations and nodes representing explicitly stated entities (W3C Owl Working Group 2009)(Motik Grau Horrocks Wu Fokoue and Lutz 2009), resulting in a highly interconnected and semantically meaningful dataset. Since engineering terminology evolves slowly, ontologies remain stable across time. If one formalized the term "force" a century ago, it would remain accurate today.

Ontologies can be reusable, shareable, and extensible (Gruber 1991), but in practice these properties are contingent on good development principle adoption. Such principles include using shared development strategies and standards; using a top-level ontology; coordinated ontology co-development; using and re-using prior work; and curating and openly disseminating existing, vetted ontologies (Arp, Smith, and Spear 2015). Though challenging, ontology development yields an expanding, specialized as needed, ecosystem. In the "force" example one might use subclasses, like "drag force," to refine the term without affecting the existing work's validity or usefulness.

The case study's ontology used freely available ontologies to represent the system. A top-level ontology called the *Basic Formal Ontology* (BFO) (Bittner and Smith 2004) provided the semantic layer's philosophical foundations. BFO provides a framework and guidance under which to define more specific domain terms. The adopted BFO-aligned common core ontologies (CCO) then represent things like manmade systems (artifacts) and the specific weight case. CCO forms an ontology "mid-layer" which bridge the divide between a high-level philosophical basis (BFO) and a domain expert's terms of interest. Prior work used in IoIF (Bone, Blackburn, Kruse, Dzielski, Hagedorn, and Grosse 2018) extended this base to address model alternative analyses for decision making, based on a decision ontology.

This case study's specific requirements also mandated ontology development for the SysML language and its implementation in Cameo Systems Modeler. These highly specialized terms, called "application ontologies," align tool or institution specific data to tool-agnostic domain ontologies through taxonomical relations.

***Semantic Web Technology***

While ontologies enable knowledge representation, SWT comprises tools realizing the knowledge representation benefits in a software system. A resource description framework (RDF) captures semantic data (Horrocks, Patel-Schneider, and Van Harmelen 2003)(Antoniou and Van Harmelen 2004)(W3C Owl Working Group 2009) in the "triples" form. Triples are essentially statements comprising a subject, predicate, and object. Visualized as a graph, each triple has two nodes (the subject and object) connected by a vertex (the predicate). Each triple defines new nodes or connections between nodes, creating an ever-larger data graph aligned to a controlled lexicon. The web ontology language (W3C Owl Working Group 2009)(Antoniou and Van Harmelen 2004) (OWL) extends RDF with more expressive semantics. Triple stores, data repositories implementing language semantics, scale well and can store and query RDF data (Morsey et al. 2011).

SWT also helps implement AI through automated reasoning. Automated reasoning uses logical deduction from ontology axioms to make inferences about the entities described in the graph. In the ontology itself (referred to as the "Terminology Box" (TBox)) a reasoner reclassifies terms or assets new class-level axioms. Data (the "assertion box" or ABox), reasoned upon according to the same axioms, can evolve through semantic rules (Horrocks et al. 2004), which implement full first order logic in if-then type statements called Horn Rules (Horn 1951).

Standards such as the SPARQL Protocol and RDF Query Language (SPARQL) (Prud and Seaborne 2006) can query semantic data. SPARQL queries describe graph patterns and return nodes, data values, or new triples based on conforming graph data. SPARQL is highly expressive (Angles and Gutierrez 2008) and implements complex logical rules outside of a reasoner.

The software developed for IoIF implements key methods used to manage, store, and manipulate semantic data, and is not case-study specific. The Owlready2 (Lamy 2017) Python package populated the ontologies and call Pellet (Parsia and Sirin 2004), an automated reasoning software. RDF4J server was a triple store and SPARQL query endpoint. The knowledge representation combination through OWL, querying through SPARQL, and automatic reasoning through Pellet cumulatively provide a foundation for knowledge representation and AI project data enhancement. We view this semantically enhanced data as the authoritative information source for the modeled system.

***Data Ingestion***

Most tools do not natively output RDF data. The exact output (format, data type, other) varies by tool. As a result, it is necessary to pre-process data to express it as a graph of triples. Commercial linked dat products refer to this as "data ingestion". In practice, the ontology ecosystem must inform the data ingestion process. Ontology-aligned data includes graph patterns repeated at differing generality levels, helping with reuse and effective querying (Blomqvist and Sandkuhl 2005). Ingestion uses tool data and metadata to populate a triples graph conforming to these patterns. This often entails translation from a "tool" lexicon to an ontology lexicon. IoIF accomplishes this by using taxonomic relations expressed in an application ontology. SysML uses terms like "class" and "property," and these all function as information describing something of interest queried using general graph patterns for all information content.

The case study retrieved a system model from MMS as a JSON element list. A parser used the JSON elements to write an RDF representation aligned to a SysML tool ontology.

***Interpreting the System Model:***

Since system modelers do not use tightly controlled naming conventions and operate outside ontological constraints (such as ontological disjoints, the open-world assumption, and more), integration between IoIF and a system model requires extra work. Prior efforts used element naming conventions to tie a model to the ontology lexicon (Bone et al. 2018), but the process proved time consuming, knowledge intensive, and had limited reuse potential. In IoIF, providing the system modeler with ontologically meaningful profiles with consistent and reusable tool metadata solves these issues. These profiles recreate the controlled ontology terminology in the modeling environment through introducing stereotypes corresponding to ontology terms. The presence of ontologically aligned stereotypes helps translate the model into repeatable graph patterns by identifying precisely what the modeler described and flagging relevant elements to the broader system analyses and verification facilitating parsing elements into repeatable, ontology-aligned graphs. This approach allows IoIF to incorporate a system model into its knowledge base and automatically use it as a basis to understand the system's nature.

Thus, in the IoIF context, system models can create a system and mission representation in the triple store. Subsequent data from any given tool links to this representation. A system model becomes central to defining the data structure ultimately used to semantically relate data throughout the digital thread. The next section describes the data transformation process from the native modeling language to a tool invariant ontological representation.

***Data Transformation***

Ontologically aligned graph pattern creation requires data transformation from a tool information model and lexicon into one represented ontologically. For example, modeling languages often lack a one-to-one mapping between language constructs and a domain view. In the transformation process a graph identifies tool specific patterns of interest which translate to equivalent ontological representation.

The profiled SysML model required such a transformation. The ontologically profiled stereotypes in the SysML model guides the transformation process. Though a modeler might not follow some preferred representation pattern sets, the stereotypes provide starting points. A transformation might select stereotyped blocks and properties and use them to infer the model addresses certain items, for example a vehicle block having weight properties. It might also add nodes to represent data about the system and assert relations between the entire pattern (Figure 2). This process repeats across tool-specific patterns and translations; the model helps describe related system parts, and the profile, stereotypes, and ontology terms help determine their exact relation.

In IoIF, SPARQL targets patterns of interest in the tool-aligned graph. The results construct a parallel, ontology conformant graph according to transformation instructions (Figure 3). The new graph contains distinct individuals in ontology prescribed graph patterns, and these link back to the tool-specific graph. This

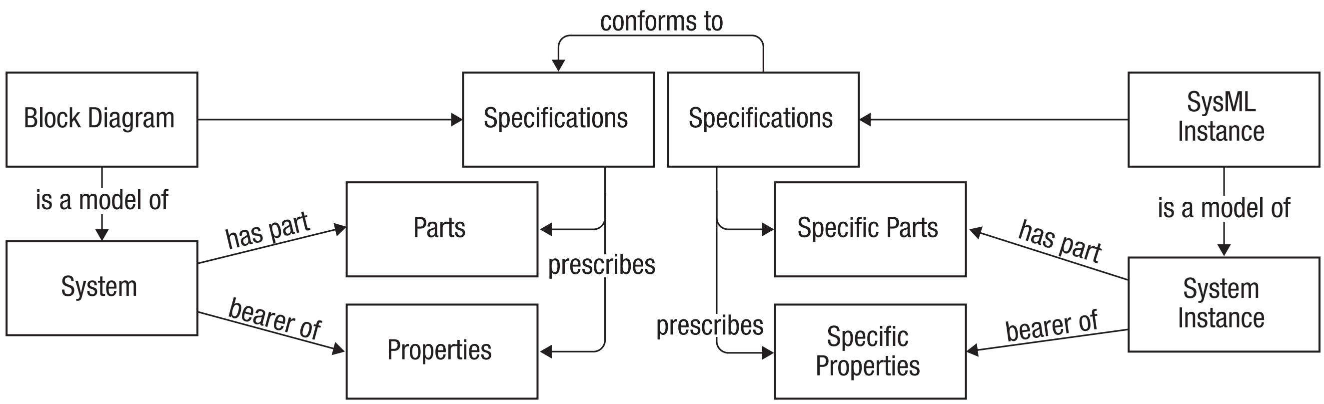



*Figure 2. Data transformation and handling of SysML class and instance relations*

ontology-aligned graph provides the tool-agnostic data against which AI applications then make inferences.

Absent the clear metadata provided by the ontologically aligned profile, this transformation process may become arduous and unrepeatable. The profiled case study model could fully automate the transformation. Ongoing work in other applications has applied the transformation instructions used in the case study across substantially different models, with the ontology scope and the implemented transformation instruction extent being the main limiting factors.

### *Reasoning Upon Data*

Reasoning is possible once triples represent the model, but is easier to implement and more reusable if used upon domain ontology aligned data. Many reasoning kinds, including automated consistency checking, description and first order logical deduction, data completeness verification, and rule-based reasoning. There are several ways to accomplish these, though all stem from information or context embodied in the ontology layer. A reasoner uses explicit and inferred relations in an ontology and data to make new inferences. For example, an ontology might assert every material thing bears a single mass, thus inferring every node tagged as a material thing would have mass, even if mass does not explicitly exist in the graph. Automated reasoning results in inserting new triples into the graph, with each new inference potentially causing additional ones. This makes reasoners reliable but computationally intensive for larger graphs (Abburu 2012). SPARQL queries are useful in cases where one requires inferences at scale, new graph node creation, or extended semantics. This provides considerable power, but does not forward chain inferences leaving them incomplete. Used opportunistically in tandem with another, automated and query-rule based reasoning provide very powerful AI potential to ontology-aligned data.

The IoIF case study used an automated reasoning software called Pellet (Parsia and Sirin 2004) and SPARQL focusing on summing system weight from its component parts and using the results to assess the model's internal validity. When accomplished in SWT systems and projects undergoing ingestion and transformation may reuse this. Determining system weight is a relatively straightforward addition problem. The main challenge is knowing what to add. The ontology facilitates identifying trait types aggregating (sum) between a whole and its parts. Weight is one such trait; anything having multiple parts or including a repetition aggregate can weigh the sum of all those parts provided the graph includes all parts. SPARQL locates these values and returns the weight sum. The resulting values can return to the tool-agnostic graph, and eventually associate with the tool-specific data, model elements, and views (Figure 4 on next page).

IoIF implements this logic by first calling a reasoner to make classification inferences. It then provides a wrapper so the query can run until it knows the entire system's weight. The query itself modifies the tool-agnostic graph to include the inferred weight. Automated reasoning, rules, or queries then determine the consistency, validity, and conformance and insert new triples into the graph.

### *Pushing Data Back to Models*

Thus far, SWT has addressed a unidirectional link from tools to a triple store acting as a tool-agnostic information source. For the knowledge representation and AI to deliver benefits, this link likely needs to be bidirectional and the resulting information derived from reasoning should push back into the system model. The first issue is identifying what data is relevant to the tool. Terms used to denote inconsistencies and changed values identify a node set in the tool-agnostic graph linked tool-specific graph. Documentation, new data values, and

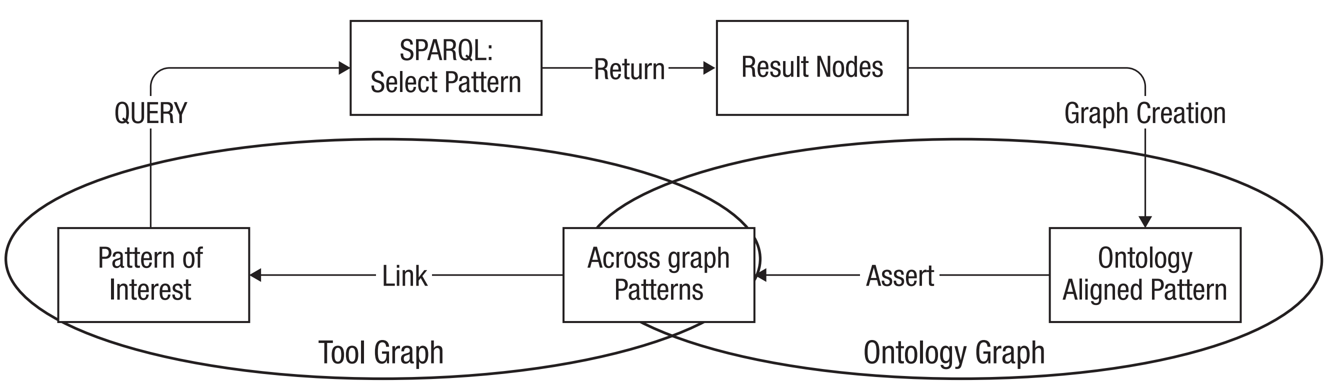



*Figure 3. Transformation process from tool-specific to tool-agnostic data*

| Owner | Value | Unit or Type |
|---|---|---|
| System | 16500.0 | lbs |
| System.Part1 | 9200.0 | lbs |
| System.Part2 | 6500.0 | lbs |
| System.Part3 | 2550.0 | lbs |
| System.Part4 | 5000.0 | lbs |
| System.PartAggregate | 150.0 | lbs |

| Owner | Value | Unit or Type |
|---|---|---|
| System | 15500.0 | lbs |
| System.Part1 | 9050.0 | lbs |
| System.Part2 | 6500.0 | lbs |
| System.Part3 | 2550.0 | lbs |
| System.Part4 | 5000.0 | lbs |
| System.PartAggregate | 1500.0 | lbs |

*Figure 4. Updated System Weight Table in the View Editor*

| Approved Elements | Risk | Approval Status |
|---|---|---|
| Weight Document | high | approved |
| System | medium | approved |
| System Model | medium | approved |
| Requirements | medium | approved |

| Approved Elements | Risk | Approval Status |
|---|---|---|
| Weight Document | high | approved |
| System | medium | approved |
| System Model | medium | approved |
| Requirements | medium | approved |

*Figure 5. Signoff table before and after Reasoning Via IoIF*

Element Differences

| Attribute | Local Value | MMS Value |
|---|---|---|
| "visibility" | null | null |
| "templateParameterId"" | null | null |
| "typeId" | null | null |
| "value" | 16500.0 | 15550.0 |
| "definingFeatureId" | V weight "_18_5_3_8db028d_156207416493... | V weight "_18_5_3_8db028d_15620741649... |

*Figure 6. Round trip of weight update model viewed in Cameo Systems Modeler*

the like pass to the tool based upon these links. The demonstration case also included an update to the signoff mechanism built into the system model (Figure 5). Queries computed a "affected elements" set based on known changes and relations in the ontology denoting dependencies in information. Tracing these back to their associated signoff allowed an updated status to return to the tool-specific graph's corresponding nodes.

To synchronize to MMS, MMS requested the changed elements as JSON and overwrote the changed fields. The resulting JSON elements pass back to the MMS via the MMS REST API and become visible in the View Editor and synchronize to a system modeling tool (Figure 6). A modeler can then inspect the model and changes, or documentation not explicitly captured in model views.

## CONCLUSIONS:

Knowledge representation through ontologies is a foundation for enabling automating reasoning about systems engineering methods. The case study identified areas requiring effort. Well-defined ontologies facilitate reuse across systems, disciplines, and domains. Both in and beyond the ontologies the case study used, it is necessary to represent vast domain knowledge. Doing so requires multiple SWT tools. It is also necessary to develop software systems and tools to support integration with SWT. IoIF made the case study possible using multiple parsers, clients, and links to external tools, many requiring independent development.

This data-driven systems engineering view is a necessary step for AI in systems engineering, and this exemplifies how system engineers define future systems to support AI applications. This case study demonstrated current technology enables a digital thread providing information such as system weight and helps identify inconsistent design data. These checks and system data enhancement with semantically meaningful information demonstrates AI for SE's art-of-the-possible.

Other contexts, such as cyber security, have reused the same approach and software. In this case IoIF performed a round trip from a SysML model, into a triple store, and then associate vulnerabilities with model elements. These propagated back into MMS and finally the SysML model. The process used nearly identical SWT and software despite using different models and moving from a physical to a cyber system. This ease of re-application suggests the approach may be useful in many domains and AI applications. Future work plans to investigate other ways ontologies, SWT, and IoIF apply to AI for systems engineering. ■

## REFERENCES


- Abburu, S. 2012. "A Survey on Ontology Reasoners and Comparison." *International Journal of Computer Applications* 57 (17).
- Angles, R. and C. Gutierrez. 2008. "The Expressive Power of SPARQL."Springer.
- Antoniou, G. and F. Van Harmelen. 2004. "Web Ontology Language: Owl." In *Handbook on Ontologies*, 67-92: Springer.
- Arp, R., B. Smith, and A. D. Spear. 2015. *Building Ontologies with Basic Formal Ontology* MIT Press.

- Berners-Lee, T., J. Hendler, and O. Lassila. 2001. "The Semantic Web." *Scientific American* 284 (5): 28-37.
- Bittner, T. and B. Smith. 2004. "Normalizing Medical Ontologies using Basic Formal Ontology."
- Blomqvist, E. and K. Sandkuhl. 2005. "Patterns in Ontology Engineering: Classification of Ontology Patterns."Citeseer.
- Bone, M. A., M. R. Blackburn, D.H. Rhodes, D.N. Cohen, and J.A. Guerrero. 2018. "Transforming Systems Engineering through Digital Engineering." *The Journal of Defense Modeling and Simulation*: 1548512917751873.
- Bone, M., M. Blackburn, B. Kruse, J. Dzielski, T. Hagedorn, and I. Grosse. 2018. "Toward an Interoperability and Integration Framework to Enable Digital Thread." *Systems* 6 (4): 46.
- Brickley, D. and R.V. Guha. 2004. "RDF Vocabulary Description Language 1.0: RDF Schema."
- Chandrasegaran, S.K., K. Ramani, R.D. Sriram, I. Horváth, A. Bernard, R.F. Harik, and W. Gao. 2013. "The Evolution, Challenges, and Future of Knowledge Representation in Product Design Systems." *Computer-Aided Design* 45 (2): 204-228.
- Gruber, T.R. 1991. "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases." Kr 91: 601-602.
- ——. 1995. "Toward Principles for the Design of Ontologies used for Knowledge Sharing?" *International Journal of Human-Computer Studies* 43 (5): 907-928.
- Gruber, T. 1993. "What is an Ontology." *http://www-Ksl.Stanford.Edu/Kst/Whatis-an-Ontology.html (Accessed on 07-09-2004).*
- Horn, A. 1951. "On Sentences which are True of Direct Unions of Algebras." *The Journal of Symbolic Logic* 16 (1): 14-21.
- Horrocks, I., P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. 2004. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." *W3C Member Submission* 21: 79.
- Horrocks, I., P.F. Patel-Schneider, and F. Van Harmelen. 2003. "From SHIQ and RDF to OWL: The Making of a Web Ontology Language." *Web Semantics: Science, Services and Agents on the World Wide We*b 1 (1): 7-26.
- Knublauch, H. and D. Kontokostas. 2017. „Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017." https://www.W3.0rg/TR/Shacl.
- Kruse, B. and M. Blackburn. 2019. "Collaborating with OpenMBEE as an Authoritative Source of Truth Environment." *Procedia Computer Science* 153: 277-284.
- Lamy, J-B. 2017. "Owlready: Ontology-Oriented Programming in Python with Automatic Classification and High Level Constructs for Biomedical Ontologies." *Artificial Intelligence in Medicine* 80: 11-28.
- Lassila, O. and R.R. Swick. 1999. "Resource Description Framework (RDF) Model and Syntax Specification."
- Morsey, M., J. Lehmann, S. Auer, and A-C.N. Ngomo. 2011. "DBpedia SPARQL Benchmark–performance Assessment with Real Queries on Real Data."Springer.
- Motik, B., B.C. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. 2009. "OWL 2 Web Ontology Language Profiles." *W3C Recommendation* 27: 61.
- Parsia, B. and E. Sirin. 2004. "Pellet: An Owl Dl Reasoner."Citeseer.
- Prud, E. and A. Seaborne. 2006. "SPARQL Query Language for RDF."
- Shearer, R., B. Motik, and I Horrocks. 2008. "HermiT: A Highly-Efficient OWL Reasoner."
- Smith, B. 1998. "Basic Concepts of Formal Ontology." In *Formal Ontology in INformation Systems*, edited by Nicola Guarino, 19: IOS Press.
- Tsarkov, D. and I. Horrocks. 2006. "FaCT Description Logic Reasoner: System Description." *Automated Reasoning*: 292-297.
- W3C Owl Working Group. 2009. "{OWL} 2 Web Ontology Language Document Overview."


## ABOUT THE AUTHORS

**Mark R. Blackburn, PhD**, is a senior research scientist with Stevens Institute of Technology and principal at KnowledgeBytes. Dr. Blackburn is the principal investigator (PI) on several System Engineering Research Center (SERC) research tasks for both US Navy NAVAIR and United States Army CCDC Armaments Center on systems engineering transformation through model-centric engineering. He was PI on a FAA NextGen and National Institute of Standards and Technology projects and has received research funding from the National Science Foundation. He develops and teaches a course on systems engineering for cyber physical systems. He is a member of the SERC research council, and core member of the semantic technologies for systems engineering initiative. Prior to joining Stevens, Dr. Blackburn worked in industry for more than 25 years.

**Mary Alice Bone, PhD**, is a staff researcher for the Systems Engineering Research Center (SERC) at Stevens Institute of Technology and an associate professor II for the University of South-Eastern Norway. She received a BS in aerospace engineering from Missouri S&T in 2001, a MEng in systems engineering from Iowa State University in 2007, and a PhD in systems engineering from Stevens Institute of Technology in 2015. Currently she is helping lead the SERC digital engineering transformation research tasks for both United States Army CCDC Armaments Center and Naval Air Systems Command. Her research includes semantic web technology to enable digital engineering, system reference architectures, system complexity, system modularity, and model-based systems engineering to enable synchronized-continuous engineering throughout the whole system life cycle.

**Benjamin Kruse, ScD**, is a research assistant professor at Stevens Institute of Technology, working as a SERC researcher on systems engineering transformation through model-centric engineering research tasks, focusing on the use of SysML and OpenMBEE with their view and viewpoint mechanism, and approaches for mapping SysML models to ontologies and semantic web technologies.

**Tom Hagedorn, PhD**, is a senior research fellow in the Department of Mechanical and Industrial Engineering at the University of Massachusetts Amherst. His research interests include applications of semantic technologies to systems engineering, decision making, advanced manufacturing, and biomedical design.

**Ian R. Grosse** is a full professor in the Department of Mechanical and Industrial Engineering at the University of Massachusetts Amherst. Professor Grosse conducts research in the areas of 1) semantic information modeling and technologies to support engineering design, finite element analysis, and decision making, 2) advanced additive manufacturing, and 3) biomechanics. Professor Grosse has served as a co-director for the NSF Center for e-Design and as an associate editor for the ASME/ACM Journal of Computing and Information Science in Engineering. In 1990 and 2003 he received the best paper award in ASME Computers and Information in Engineering Conference. In 2012 Professor Grosse was made a fellow of ASME, and in 2018 he received the distinguished service award from ASME's Computers and Information in Engineering Division.

# As Smart as a Human? Leveraging Models of Human Intelligence to Assess the Intelligence of Systems

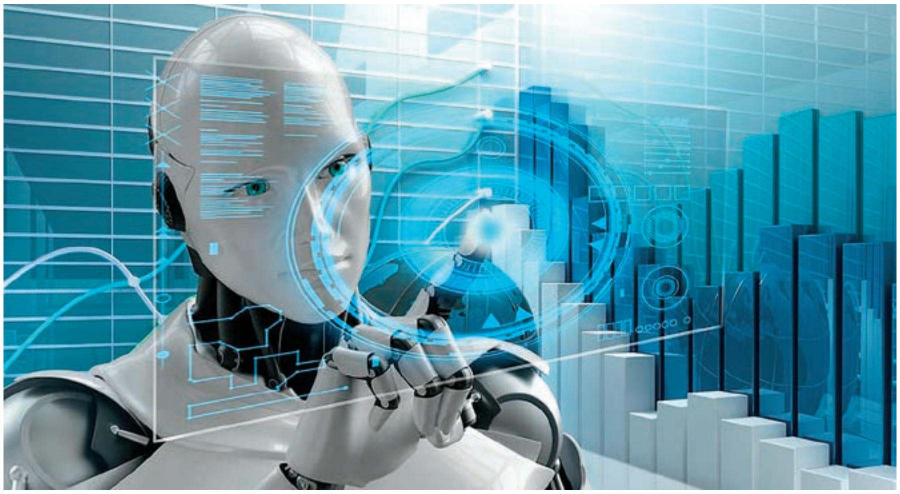
**Barclay R. Brown,** barclay@barclaybrown.com


**ABSTRACT**
As we evolve artificial and machine intelligence concepts, and consider their extension into intelligent systems, it becomes important to be able to assess the system's intelligence level. This assessment serves several important needs in systems engineering. First, it enables the tradeoff of system design alternatives based on a system's intelligence, along with other factors including cost and performance. Second, it influences system verification and validation methods. Lastly, it will help stakeholders specify the required system intelligence. Assessing or specifying the system's intelligence presents many challenges and difficulties similar to those faced by psychologists and neurologists in measuring the intelligence of human beings and animals. This article explores some human and animal intelligence assessment concepts and shows their application to assessing a system's intelligence.

## INTRODUCTION

The only systems currently on planet earth rivaling human intelligence levels, are human beings themselves. When we interact with these systems (human beings), we appraise the system's intelligence, and adjust our behavior accordingly. We interact with a 1-year old differently than we would a 3-year-old, 10-year-old or 35-year old. Independent of age, humans have varying abilities to understand, reason, and respond—the commonly recognized intelligence characteristics. What of engineered systems—can we sensibly appraise their intelligence?

Intelligence is a systems phenomenon. Human beings did not rise to preeminence on earth by thinking alone, but by developing and applying spoken and written language, building and using tools, and harnessing fire (Asimov 1979). Artificial intelligence technologies, including rapidly progressing deep neural network methods, are providing intelligent building blocks with which to create intelligent systems. The resulting systems' usefulness will be a function of a combination of elements, with systems intelligence as an emergent property. Whether enough known and creatable elements can eventually combine to produce an artificial general intelligence, is a debatable topic. Some claim this evolution is inevitable and others believe it will never happen—that we are jungle dwellers, building mud huts and assuming if we only had enough mud we could build skyscrapers! In either case, systems engineers are in a unique position to lead and guide intelligent system dynamic, uncertain, and exciting evolution.



(*Source:* https://www.pxfuel.com/en/free-photo-qaulj, CC0 license)

## WHAT IS AN INTELLIGENT SYSTEM?

We are sure human beings, and some animals, have various levels of what we call intelligence, so it is tempting to define an intelligent system as one constructed the way humans and animals are—with a

MARCH 2020 VOLUME 23/ ISSUE 1

brain. To create an intelligent system this way would require creating an engineered system accomplishing what a brain does. Attempts to create artificial brains exist, but are limited by the understanding current science has about how brains work, and more importantly, how they produce what we observe as intelligence, let alone consciousness (Li et al, 2012). Creating an artificial brain is intriguing and may produce a system demonstrating both intelligence and consciousness, yet there is significant doubt about its feasibility, at least until we fully understand animal and human brains. By analogy, once we understood the human heart's function, it was a straightforward matter to create an artificial system doing what the heart does. We lack this understanding of the brain, but do we need to create a brain to create an intelligent system?

The word intelligence, which comes from roots related to understanding, has come to mean "the ability to acquire and apply knowledge and skills" (Lexico/Oxford, 2019), casting a much wider net and including numerous skills, provided the system can acquire and apply them. Intelligence, understanding, and consciousness, considered closely related before computers, occurred together as an emergent brain function. Progress in artificial intelligence and machine learning, however, has shown constructed systems can act intelligently while having neither humanlike understanding nor anything resembling consciousness.

From an engineering perspective, it seems there must be more to intelligence than responding to an input and producing a result or decision. A thermostat can control the temperature in a home and make decisions regulating the heating, cooling, and ventilation systems, but it seems a stretch to call it intelligent. Replace the thermostat with a human caretaker, and we expect consideration of the home's occupants, their current activities, their needs, their schedules (both regular and anticipated), the weather, and the heating and cooling system efficiencies and economics. Advanced thermostats on the market now have these capabilities and we (or at least their marketers) call them intelligent. We can say the more advanced thermostat is more intelligent than last century's simpler ones, but even the advanced thermostat would score low on intelligence when compared to a human caretaker performing the same function. Considering our intuitions on what counts as intelligent behavior starts us on the path to a conception of intelligent systems.

IBM Deep Blue's victory over world chess champion Garry Kasparov in 1997 showed a computer could achieve a superhuman performance level in a task previously thought to require high intelligence, understanding, and consciousness. Deep Blue, however, had nothing resembling understanding or consciousness, playing openings and endgames from rote tables and exploiting fast computation to evaluate midgame board positions and choose moves. Deep Blue could play chess, but could not do things we would expect of a human who can play good chess—it could neither explain chess to a child, nor develop a strategic attack plan, nor be an interesting guest on the Tonight Show. On the other hand, it could not experience regret, overconfidence, or fear (as Kasparov himself observed in 2018). Deep Blue had a *specific intelligence*—chess—but not the general chess intelligence we would expect from a human chess player.

Deep Blue is an intelligent system—a useful system, with specific abilities to acquire and apply knowledge and skills at human (even superhuman) performance levels in specific circumstances. An intelligent system exhibits human-like intelligence in a specific area or domain but need not be—and is likely not to be for some time—intelligent in *all* human activity areas. Intelligent systems will be specialists, not generalists for one main reason: we do not know how specific intelligences combine to produce general intelligence. We could try combining a Tesla car's auto-pilot capabilities with Deep Blue's chess-playing abilities, but would that improve either the driving or the chess game? In a human, learning chess can reasonably improve mental capabilities which may transfer to other activities, like driving. We know this works—it's the reason we give our children broad, general educations, rather than simply training them from birth to perform a specific vocational task. Our dogs, on the other hand, are more in the *specific task* category.

## MEASURING INTELLIGENCE

Not all systems performing some function are *intelligent* systems. Risking a circular explanation, a system is intelligent if it performs a function, which were it performed by a human, would be considered as intelligent behavior. For example, a child can answer, "What is 2 + 2?" by rote. More learning and time expands this to any single number pair, then larger numbers, more than two numbers, and so on—a growing intelligence level in arithmetic computation. Incidentally, Google, Amazon Echo, and Bing handle "What is 2 + 2?" just fine, but Yahoo and DuckDuckGo do not. DuckDuckGo gives the correct answer when inputting "2+2=", much as a pocket calculator would. Google and Bing, with their superior ability to handle various question forms, are intuitively more intelligent with respect to plain language arithmetic computation than their rivals. However, simply programming a system to do something a human can do, such as add 2 + 2, does not make it intelligent. Intelligence requires something more than strict function performance.

The technology to implement a system does not matter with respect to intelligence in the sense we are describing. A system need not add numbers, or play chess, the way we do to appear intelligent and be useful to its users. A system might not use current artificial intelligence and machine learning techniques, but still seem intelligent. Deep Blue seems quite intelligent, though it plays the entire chess endgame by rote table lookup. Conversely, a system may use the latest machine learning deep neural network algorithms but not appear to humans as intelligent, depending on the example data on which it was trained and the neural network design. A system might include multiple component subsystems, some using AI or machine learning, and the whole system may behave in intelligent or unintelligent ways. System implementation does not matter, only successfully and beneficial intelligent behavior that meets stakeholder requirements.

We have concluded an intelligent system is one exhibiting intelligent behavior and acting in intelligent ways to benefit its users. Before breaking this down into more measurable intelligence components, first consider some intelligent system examples.

Imagine a McDonald's hamburger restaurant. Consider the entire restaurant, including employees, to be a system. Specifically, it is a human activity system since human actions are its primary motivator, distinct from an engineered system which does not contain humans as a system part, or a natural system which contains only biological components (Checkland 1999). Since the first McDonald's opened in 1955, a customer's interaction with the McDonald's system has been the same—approach an employee, dictate the order as he or she keys it into an order keyboard, and pay. Of course, this system is intelligent at some level, relying mostly on the McDonald's employee to contribute the intelligence. In 2018, McDonald's began deploying large kiosks in its restaurants, allowing customers to enter their own orders, select customizations, and make payment. A McDonald's host hovers nearby to help with kiosk order entry if needed. Is the original McDonald's a more intelligent system than the kiosk-plus-host McDonald's, or is McDonald's simply trying to replace human workers with automation?

“The popular narrative is that kiosks and mobile ordering are here to take jobs and hours away from underpaid cashiers… but the data suggests that isn’t true… with such a tight labor market, many chains are struggling to hire and retain customer-facing employees… automated ordering can help wait times and improve order accuracy, and it doesn’t negatively affect labor as much as some think” writes Hollis Johnson in Business Insider (2018), suggesting a kiosk might produce a better or more intelligent interaction with customers than under-trained, low-tenure, error-prone employees. If the former is a more intelligent system, how much more intelligent is it? To move toward an answer for systems, consider human intelligence in more depth.

## THE COMPONENTS OF INTELLIGENCE

Everyday notions of intelligence when applied to humans include characteristics like language and conversation mastery; abstract or symbolic reasoning; ability to recall related fact sets; wide-ranging world knowledge (both on trivial and significant topics); worldliness; logic and analysis, creativity in science, technology, or art; and various puzzle solving forms. Engineered systems cannot compete with intelligent humans unless the competition limits its scope to a certain task, puzzle, or game, such as chess, arithmetic, or simple fact retrieval. We would not consider a human being intelligent whose *only* ability is performing a narrow, even though complex function, just as we would not consider a human who can recite $e{=}mC^2$ to be as intelligent as one who can also explain it, analyze it, and apply it. In any function requiring intelligence, we expect a system (or human) to be able to perform a set of *subfunctions*.

If a human can play chess, we would expect the human to be able to explain the rules for how a knight can move to a six-year-old, and to be able to play a modified game where, say, kings can move two squares at a time instead of the normal one. Interestingly Deep Blue would fail at both, the latter because it plays openings and endgames from tables now invalidated by the rule change. For a system to be intelligent, as we gauge intelligence, it must perform a complex function, but also have additional characteristics including subfunctioning, flexibility, adaptation, and others. Developing a comprehensive characteristics set along with means for specifying and measuring them in systems is an important task for systems engineers.

In addition to performing subfunctions, an intelligent system would be able to rearrange these subfunctions to perform new tasks. When we teach our children to play basketball, we begin with holding the ball, then catching the ball, then throwing the ball with two hands, then with one hand, then at a target, and more. When it comes to playing soccer, many skills will transfer over, even though throwing the ball plays a different role in soccer than in basketball. An intelligent system can adapt to new but similar situations. A competent tennis player, confronted for the first time with racquetball, will play better than a beginning racquetball player. The tennis player will use some skills unmodified, such as footwork, and modify others such as wrist position and use. In software, including flexible and multi-level APIs (application programming interfaces) allows other programs to separately invoke a system’s capabilities in new sequences and arrangements to accomplish a new purpose. Function definitions and calls work similarly *in* a software program.

An intelligent system design should handle situations where it does not have confidence in its answer or decision. A human, say a two-year-old, when given a complex sentence, such as one requiring more working memory (Blything and Cain 2016), will likely find something in the sentence that sparks some idea and provides some response. “Honey, did you know the latest research on fat-content and calorie-consumption indicates that it might be better to eat more ice cream and fewer chips?” will bring a response of “Ice cream!” rather than an insightful response to the whole sentence’s meaning. Try this sentence with an “intelligent” voice assistant and you’ll likely get, “Sorry—I don’t understand” instead of a more helpful guess like, “Should I add ice cream to your shopping list?” Most current conversational systems cannot do as well with unexpected input as a two-year-old.

A similar, but distinct characteristic we also associate with intelligence is input flexibility. A system, or human, seems more intelligent if it can accept our input any way we give it. When requesting a cash withdrawal, we expect to make the request however we wish from a caveman-like, “Cash: $100,” to an over-explained, “Hello and good day. My sister is coming to town and we need some cash to shop at the antique market—you know the one on Broad Street? Do you think $350 will be enough? No wait, let’s do $500. Thanks.” When interacting with a bank teller, or intelligent system, it would be strange indeed to interact as we do with the much-less-intelligent ATM outside, wordlessly handing the teller the ATM card first, then waiting for the teller to provide a choice menu, identifying the choice in exact terms, then providing the amount, and more. One could imagine a more intelligent ATM with which we can interact in a more flexible way.

Think about how one enters an address into an online form. Virtually all systems ask the user to enter address, city, state/province, zip/postal code and perhaps country into separate fields, in that order. A more intelligent system would simply ask the zip/postal code first, and then find a valid address as the user types the street address character by character, then fill in the city, state/province, and country automatically (allowing changes if necessary). An even more intelligent system would ask the user, “What is your address?” and interpret whatever the user supplied to derive the correct address. “123 Main in Vienna” might be enough to uniquely specify an address, but what if there is more than one Vienna in the world (there are at least a few)?

The city-name example leads us to another intelligent systems’ characteristic and, like the other characteristics, humans do it instinctively. When information provided is incomplete, an intelligent system will ask for the missing information, using what it has as a guide. In the example above the system might ask, “Do mean the Vienna in Austria, Virginia, West Virginia or New York?” which is what we expect of a very geographically literate human. Less intelligent systems (what we have today it seems) would respond with, “invalid address, try again,” (what the system should say is “try something different,” but we should not argue with unintelligent systems).

Moving beyond intelligent user interfaces, intelligent systems also keep watch over boundary conditions. Ask an Uber driver to go 100 mph in a residential area, and he or she will (hopefully) refuse. Simple hard-wired thermostats have, for decades, had a failsafe function keeping the home heated to a minimum of 50°F/10°C no matter the user’s settings, to avoid frozen pipes and property damage. This missing failsafe in a Nest thermostat caused a stir as home temperatures plunged after a failed software update (Bilton 2016). At the other end of the temperature scale, if a thermostat detects the home has risen in temperature from 70°F/21°C to 200°F/93°C in the last hour, it should do more than log this interesting fact in the historical database—it should call the fire department! Self-driving cars should avoid collisions with anything, regardless of the GPS-powered navigation system and its optimized route. An inspirational self-management and overriding constraint conditions example can be seen in Asimov’s *three laws of robotics*.

Systems engineers will notice many intelligent capabilities require multiple intelligent subsystems and higher-level intelligent capabilities to coordinate and arbitrate between them, making intelligent system design a systems challenge, suited to system engineering skills. A watchdog subsystem should oversee a deep neural network with training-data based decisions, preventing ridiculous decisions resulting from neural network structure errors, or from faulty, incomplete, or intentionally misleading data fed to the system after deployment.

These intelligent system characteristics (and many more) can give us an approach to specifying and assessing intelligence in systems. Assessing intelligence has never been easy or uncontroversial, even with the most familiar intelligent systems we know—human beings.

## ASSESSING HUMAN INTELLIGENCE

When assessing human being intelligence, the range of characteristics to consider is wide and probably far beyond the systems we can imagine building. Science fiction makes it easy to visualize systems with these human intelligence characteristics, but they do it by clothing a human personality in mechanical garb and adding flashing lights. When assessing human intelligence, even in children, we find characteristics surpassing what we know how to build, such as those listed in the table below. These characteristics can however, act as inspiration for where we might look to increase system intelligence.

***What Human Beings Do When They Behave Intelligently*** *(Costa 1988)*

1. Persistence: persevering when the solution to a problem is not immediately apparent
2. Decreasing impulsiveness: Often students blurt the first answer that comes to mind rather than considering alternatives
3. Listening to others–with understanding and empathy
4. Flexibility in thinking
5. Metacognition: awareness of our own thinking
6. Checking for accuracy and precision
7. Questioning and problem posing
8. Drawing on past knowledge and applying it to new situations
9. Precision of language and thought
10. Using all the senses
11. Ingenuity, originality, insightfulness: creativity
12. Wonderment, inquisitiveness, curiosity, and the enjoyment of problem solving

It is natural for humans to assess the intelligence of other humans using language and conversation. At this point in intelligent system evolution, conversation is not a feasible way to assess a system's intelligence. Conversational systems, including chatbots, are available and have become popular, but conversational interfaces to intelligent systems are rare, and do not give us access to assess the system's intelligence, so the ways we assess human intelligence (IQ-tests, interviews, writing assignments) do not apply to intelligent systems.

Conversational systems, designed purely to provide conversation, have claimed to pass the Turing test, fooling human conversational partners about whether they were talking with a human or a computer. In one celebrated claimed success, however, characterizing the conversational system as a 13-year old boy whose first language was not English reduced expectations for highly intelligent conversation (BBC 2014).

We can imagine a future where a general conversational interface or voice API might be available on virtually all intelligent systems, allowing users to question them in plain language about their behavior, knowledge, and intelligence. Systems engineers will design how conversational interfaces, offered by multiple intelligent subsystems in an environment such as a vehicle cockpit, will combine and integrate to provide a smooth and intelligent user interface. Complex conversational systems may need to meet human language requirements such vocabulary size, comprehension, and personality and sentiment, then measured and evaluated to see if they meet the required standards. A simple system might require only a 10-year-old's language level and vocabulary, with a friendly, helpful personality. Another might need to operate at a college level with a more assertive personality. Systems engineers must find ways to specify these new requirements.

In the future when a general voice interface is available, it might be conceivable to assess system intelligence through an interview or written IQ test, much as with human beings. These tests have their limitations, but the natural language flexibility allows a wide-ranging evaluation of intelligence aspects. Until we have access to HAL (*2001: A Space Odyssey*), Cmdr. Data (*Star Trek: The Next Generation*) or C3PO (*Star Wars movies*), we need methods other than natural language interaction to assess system intelligence.

## WHEN INTERVIEWS ARE IMPOSSIBLE

Psychologists, neurologists, and those who study animal behavior often need to assess intelligence or functional ability levels for pre-verbal or non-verbal children, non-verbal or disabled adults, and animals. Since a natural language interface is not available for these natural systems, they must use other means, which might be applicable to evaluating engineered intelligent systems.

As an example, consider the Wisconsin Card Sorting Test, used to measure child development. A child chooses a card from a deck and must match it to one of four other cards, matching number of symbols, their color, or their shape. The child must guess which matching to use, and the matching type changes, unannounced, throughout the test. The test measures how quickly the child can adapt and how many avoidable mistakes he or she makes. This method can test systems designed to recognize objects.

It is interesting to consider building a system to take this test. While it would be straightforward to design it so its only function is taking this test, it would be more interesting to build such an intelligent system using subfunctional units, mirroring the way we believe children learn. The system would need to learn modular subfunctions like identifying objects (cards), reading, counting and classifying sub-objects (the card symbols), reinforcement learning of the correct matching rule, and adaptation to a rule change. If designed correctly, these subfunctions should work in other situations, such as a system classifying traffic light images by color or number, or even one choosing the best move in a game, based on changing game conditions.

**The Wisconsin Card Sorting Test**

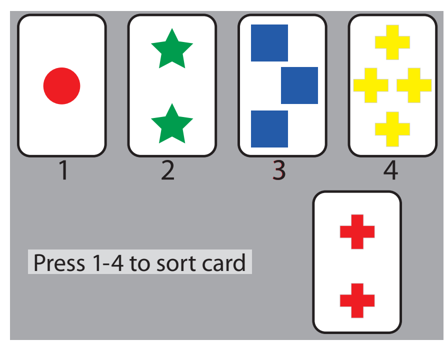



*Image credit:* http://pebl.sourceforge.net/screens/cardsort.jpg *(public domain)*

Human intelligence testing relies on progressive intelligence—human intelligence grows with education, time, and practice, using building-block subfunctions (moving a knight correctly in chess) to create higher-level functions (formulating and launching an attack in chess). This building-block approach is familiar to systems engineers who seek to develop modular architectures, rather than "hard-wired" monolithic single-function systems. The modular approach is likely to lead to

**Examples of Characteristics of Intelligent Systems**

| Intelligent System Characteristic | Requirement / Specification | Verification Methods |
|---|---|---|
| Integration of multiple intelligent components, in various combinations or sequences to accomplish system scenarios | Scenarios and use cases of sufficient breadth and diversity to exercise system and explore limitations | Traditional system testing of key scenarios; simulation of additional scenarios and/or generation of scenarios via Monte Carlo or other statistical methods (systems testing systems) |
| Ability to acquire and apply new knowledge and skills | What will the system need to learn after its initial deployment?<br>Will the system learn as it operates (adaptive learning) and if so, how will the system be ongoingly verified?<br>Will the system be re-trained (and re-verified) periodically or based upon conditions? | Inspection/analysis of training data/ examples for correct scope, diversity, negative examples<br>Inspection/analysis of synthetic data used to train system |
| Ability to perform subfunctions (derived from main system functions) in an intelligent manner | Recursive decomposition of main system functions into subfunctions and determination of intelligence requirements for each subfunction | Recursive verification of each subfunction as an intelligent system function |
| Ability to handle incomplete information | Specification of system needed information is, what information can the system obtain, and what information can the system assume or supply, and what information can the system accept or ignore without hampering performance | Verification of multiple scenarios of incomplete, unforeseen or contradictory information.<br>Verification of safe system performance even in the presence of no information or extreme information |
| Ability to self-manage and limit extreme behavior | What are the fail-safe conditions, limits to system action, and ability for system to override its "normal" operation? | Testing or simulation of system under normal and extreme conditions |
| Conversational interaction | Specification of conversational capabilities either by explicit semantics or by large set of examples; vocabulary size; complexity of language, for example, grade level | Explicit test cases supplemented by wide ranging conversational input, generated by another system, itself validated for accurate conversational synthesis |

flexibility and adaptability making more intelligent systems.

Animal intelligence requires testing without using language. These tests reward the animal for correctly performing the task. In the AI and machine learning field this approach is known as reinforcement learning, and has led to notable successes such as AlphaZero, which taught itself to play chess, shogi, and go (Oullette 2018). Assessing an animal's intelligence takes the same approach as reinforcement learning—run many repeated task trials, rewarding when results are good. Think rats running mazes for a cheese reward at the end. How quickly it learns a maze or how well it does at remembering multiple mazes assesses the rat's intelligence.

Assessing system intelligence of even a complex system can use a similar approach by designing a maze or game for the system to run repeatedly, and assessing its performance each time. This method could evaluate a system comprising multiple intelligent subsystems or agents, much as it evaluates a single animal (a system of systems itself). Systems engineers can formulate such mazes and performance scenarios along with measures for how intelligently the system performs.

## SPECIFYING AND VERIFYING INTELLIGENCE

Intelligence can act as a requirement category for an engineered system, though not a singular, monolithic one. We cannot specify, "the system shall be intelligent," in the same way we cannot require, "the system shall be easy to use." More work by systems engineers is required to determine the aspects of intelligence required. Little is free in the systems engineering, and when we add or increase the level of an intelligence characteristic, we increase the engineering and system development cost. At the same time, new technologies can reduce these costs. Deep neural networks are both superior in performance and less costly to build for various applications when compared to previous artificial intelligence methods.

Intelligence may be like obscenity, tempting us to fall back on Justice Potter Stewart's perspective that a complete definition may be impossible, but, "I know it when I see it." Nevertheless, as systems engineers, we must work to understand and define intelligence as it applies to systems. The chart below shows a beginning framework, drawn from information presented so far in this article. The chart can function as a guide to thinking through intelligent system characteristics and evaluating each characteristic's need, want and cost in a system's design. Stakeholders including system acquirers, customers, users, designers, and testers should consider these characteristics and develop feasible and verifiable requirements from them.

## EASIER ON OTHER PLANETS?

Our planet, owing to fortune and evolution, has produced one clear intelligence champion—human beings—with no close second place. The large gap between humans and anything else makes it more difficult to imagine a world with various ranges of intelligence. While philosophers, scientists, and science fiction writers may help us imagine such a world occurring naturally (Asimov 1979), through AI, we are gradually building a world with intelligence ranges, not-always-clear boundaries, and overlaps. A computer can play chess like a grandmaster, but cannot match a 6-year old in conversation. A robot can perform parkour moves, but cannot find its way out of a simple cage the way a cat can. A somewhat autonomous car can steer and make lane changes, but has trouble negotiating a 4-way stop sign as smoothly as a 15-year-old student driver.

As intelligent components combine into systems and systems integrate into complex systems interacting with other complex systems, the systems engineering profession has an opportunity to provide leadership, by fully understanding new capabilities and limitations, and developing successful ways of architecting, implementing, and verifying this new breed of intelligent system. ■

## REFERENCES


- Asimov, I. 1979. *Extraterrestrial Civilizations*. New York, US-NY: Crown Publishers.BBC News. 2014. "Computer AI Passes Turing Test in 'World First.'" https://www.bbc.com/news/technology-27762088.
- Bilton, N. 2016. "Nest Thermostat Glitch Leaves Users in the Cold." *New York Times*. January 13.
- Blything, L.P. and K. Cain. 2016. "Children's Processing and Comprehension of Complex Sentences Containing Temporal Connective: The Influence of Memory on the Time Course of Accurate Responses." *Developmental Psychology* 52(10): 1517-1529. doi:10.1037/dev0000201.
- Checkland, P.B. 1999. *Systems Thinking, Systems Practice: Includes a 30-Year Restrospective*. Chichester, UK: Wiley & Sons, Ltd.
- Costa, A.L. 1988. "Teaching for Intelligence: Recognizing and Encouraging Skillful Thinking and Behavior." Context Institute. https://www.context.org/iclib/ic18/costa/.
- Johnson, H. 2018. "We Tried the Kiosks that Analysts Say Could Help McDonald's Win Back $2.7 Billion in Sales. Here Is the Verdict." *Business Insider*, July 26.
- Kasparov, G. 2017. *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins*. Public Affairs. Lexico. 2020. "Intelligence." https://www.lexico.com/definition/intelligence.
- Li, L., J.K. Riling, T.M. Preuss, M.F. Glasser, F.W. Damen, and X. Hu. 2012. "Quantitative Assessment of a Framework for Creating Anatomical Brain Networks Via Global Tractography." *NeuroImage* 61(4): 1017-1030. doi: https://doi.org/10.1016/j.neuroimage.2012.03.071.
- Ouellette, J. 2018. "Move Over AlphGo: AlphaZero Taught Itself to Play Three Different Games." *Ars Technica*. December 6. https://arstechnica.com/science/2018/12/move-over-alphago-alphazero-taught-itself-to-play-three-different-games/.
- Shah, A. 2011. "Brain-Like Chips Tested." CIO 24(18).
- Yoshida, S., M. Kato-Negishi, and S. Takeuchi. 2018. "Assembly and Connection of Micropatterned Single Neurons for Neuronal Network Formation." Micromachines 9(5) 235. https://doi.org/10.3390/mi9050235.



## ABOUT THE AUTHOR

**Barclay R. Brown** is an engineering fellow for Raytheon Company, focused on model based systems engineering and the application of artificial intelligence and machine learning in systems engineering. Before joining Raytheon, he was the global solution executive for the aerospace and defense industry at IBM and was the lead systems engineer for some of IBM's largest development projects. Dr. Brown has been a practitioner, consultant and speaker on systems engineering, intelligent systems and software development methods for over 25 years and holds a bachelor's degree in electrical engineering, master's degrees in psychology and business and a PhD in industrial and systems engineering. He teaches systems engineering and systems thinking at several universities, and is a certified expert systems engineering professional (ESEP), certified systems engineering quality manager, and the former INCOSE director for the Americas.

# Test and Evaluation for Artificial Intelligence

**Laura Freeman,** laura.freeman@vt.edu


**ABSTRACT**
Incorporating artificial intelligence (AI) leveraging statistical machine learning (ML) into complex systems poses numerous challenges to traditional test and evaluation (T&E) methods. As AI handles varying decision levels, the underlying ML needs confidence to ensure testable, repeatable, and auditable decisions. Additionally, we need to understand failure modes and failure mitigation techniques. We need AI assurance–certifying ML and/or AI algorithms function as intended and are vulnerability free, either intentionally or unintentionally designed or inserted as data/algorithm parts. T&E provides a process for AI assurance. This article highlights existing test and evaluation methods, the key challenges embedded-AI exacerbates, and themes based for how T&E will evolve to provide AI system assurance.

**KEYWORDS:** test and evaluation, complex system validation, artificial intelligence, statistical machine learning

## INTRODUCTION

Test and evaluation (T&E) provides information used to inform system design and development, characterize existing systems capabilities, and inform decision makers on the benefits and risks of using a system. A canonical T&E community question is "how much testing is enough?" Examination through the years revealed various answers to this question. Numerous standards exist across numerous test objectives (performance, security, safety) and organizations have adopted their own standards to define adequate testing.

Statistical methods, including design of experiments, have provided foundational T&E tools for years. The National Research Council recommended these methods in 1998 in their publication on *Statistics, Testing, and Defense Acquisition: New Approaches and Methodological Improvements* (National Research Center 1998). DoD testing has widely implemented these methods. However, some have noted they are failing to provide the necessary tools for complex systems with embedded software.

System changes resulting from increased embedded software require new T&E methods to keep pace. Figure 1 from the Defense Science Board (DSB) shows the embedded software evolution in terms fighter aircraft code lines(Defense Science Board 2018). The figure shows a dramatic increase in recent years. The DSB highlights the need for new T&E process for software, notably testing software changes requires a "software factory." A key challenge is, embedded software enables capabilities in complex systems and test design approaches have revolved around modeling those capabilities across an operational space. When the software, human operators, and complex environments interact the result is a stochastic system with a large operational space to cover.

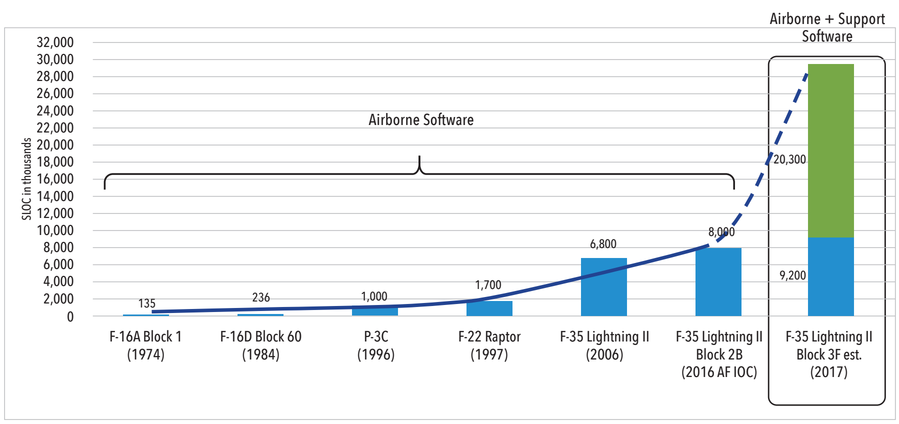



*Figure 1. Software lines of code (SLOC) in fighter aircraft from the Defense Science Board's report on Design and Acquisition of Software for Defense Systems. (Defense Science Board 2018)*

Incorporating AI algorithms into complex systems exacerbates the T&E challenge by adding complexity via stochastic algorithms. Test and evaluation for AI algorithms based on statistical learning is still in its nascent phases. Both expert system enabled AI and statistical machine learning enabled AI pose new challenges for T&E. Expert systems based on explicit models covering all models, which can expand test scope. However, the newest wave in AI – machine learning algorithms learning from experience and improving by analyzing large data quantities and identifying patterns – adds a complication in changing systems outcomes based on new data inputs or the algorithms themselves changing overtime.

New test methods need to provide these AI algorithms' assurance even as they continue to learn. In a report on operational testing of autonomy, the Institute for Defense Analysis (IDA) notes "a fundamental challenge in autonomy is developing trust in its decision-making capacity across all situations and environments it potentially will encounter" (Wojton Porter Pinelis Bieber McAnally and Freeman 2019). They also note while we can make assumptions about human decision making they are not sure for machine decision making. For example, just because a human can walk on varied surface types (sand, snow) we cannot assume a robot leveraging AI can do the same. This clearly illustrates the increased operational space magnitude an AI-enabled system must test. Conventional test methods reasonably assume how results from one environment transfer to another, this may not be a straightforward assumption for systems leveraging AI.

## THE NEED FOR T&E FOR AI

The "National Artificial Intelligence R&D Strategic Plan: 2019 Update" ( National Science and Technology Council 2019) highlights enhancements needed in AI verification and validation. The plan calls for methods measuring and evaluating AI technologies via standards and benchmarks. It also notes the need for testbeds for AI. In validation terms, they note AI systems "may need to possess capabilities for self-assessment, self-diagnosis, and self-repair to be robust and reliable."

Addressing rigorous AI T&E must happen before we can use AI in situations where it has the most potential. Those situations include environments where AI enables missions where humans cannot go (deep space, remote harsh locations, and more) and situations where the timescale is too fast for human decision making. For example, the National Security Commission on Artificial Intelligence Interim Report (National Security Commission on Artificial Intelligence 2019) notes the AI for accelerated cyberattacks value – when malware "can move through multiple systems at superhuman speed."

Systems are starting to incorporate machine learning (ML) and artificial intelligence (AI) algorithms to support varying autonomy levels, so the challenge is pressing (See for example General Dynamics Knifefish Unmanned Undersea Vehicle: https://gdmissionsystems.com/en/underwater-vehicles/knifefish-unmanned-undersea-vehicle.)

Many academic disciplines have elements providing key insights and methods for developing an AI T&E framework. Key disciplines include: system engineering, human factors, statistics, software testing, and emerging research in machine learning and artificial intelligence assurance. Some examples include:

- Design of Experiments (DOE) provides a statistical process for T&E. This methodology allows test and evaluation practitioners to consider trade-offs in risk, sample size, environment coverage, and information required for decision making in a dynamic fashion. Freeman and Warner (2018) show this methodology has proven valuable in testing complex systems integrating technologies, humans, and missions where outputs are stochastic (Freeman and Warner 2018).
- Human factors testing is experimental design subset where test techniques focus on controlling order effects (humans learning over time) and ensuring generalizable conclusions not specific to a particular individual. These methods might suggest techniques for dealing with algorithms learning overtime similar to humans.
- Software testing generally covers three test levels: unit, integration, system, and acceptance. As the software changes, regression testing can perform at any level. Integration testing considers interactions among software components. Rapid test techniques including SecDevOps (Security Development Operations) provide frameworks for thinking about testing AI software.

## CHALLENGES FOR TESTING AI

The DoD and intelligence communities take this challenge seriously. They have published numerous strategic initiatives including the 2018 DoD Artificial Intelligence Strategy (Department of Defense 2019) and The AIM initiative: A Strategy for Augmenting Intelligence Using Machines (Coats and Gordon 2019). The DoD has stood up the algorithmic warfare cross-functional team (Project Maven) and the Joint Artificial Intelligence Center (JAIC). These initiatives have driven numerous working groups, panel discussions, and forums over the past year, I have attended many and noted several themes emerging. These themes address primary challenges T&E must address to adequately test AI including:

- Systems leveraging AI struggle to enumerate requirements and testing all cases is impossible. Traditional human intuition about generalizing which cases fail to simplify the test strategy.
- AI systems are complex and exhibit emergent behavior. System decomposability assumptions may not be valid.
- System is not "final" during the deployment. It changes dynamically as algorithms learn and the environment changes.

## THEMES FOR ENHANCING T&E TO ADDRESS AI CHALLENGES

This initial theme list is far from comprehensive, but provides a starting point moving towards a T&E methodology for AI assuring autonomous systems will function as intended when called upon in operations. The themes highlight where T&E processes need to evolve. Many themes could also improve general T&E practices today, they are not AI exclusive, but incorporating AI in systems provides an opportunity to develop the scientific methods behind T&E.

### *Theme 1 – Testing and Evaluation is a Continuum*

Current test and evaluation processes are isolated test events answering a specific decision, like "should I start production of my system?" or "can I field this system to operational users?" In operational testing of systems with autonomy, IDA highlights reforms T&E needs for testing autonomy:

> "It is infeasible to create a brute force test that covers the entire operational space of autonomy. We must instead build a "body of evidence" over time, pulling from multiple data sources to answer our questions. We must use sequential testing, leverage modeling and simulation (while overcoming the unique challenges autonomy will pose), and eliminate the hard-line distinction between developmental testing and operational testing" (Wojton et al. 2019).

It is already infeasible to create test events at any given point in time covering the operational space where a system will be used. Adding AI to a system expands operational space. Test events must move beyond points in time to inform specific decisions to a T&E paradigm as a continuum where information accumulates over time across varying venues (computer models and simulations, software in the loop, hardware in the loop). Iterative testing should also imply iterative learning across a continuum, where a prior event informs each test point/event.

### *Theme 2 – The Continuum Does Not End Until the System Retires*

For fielded systems not leveraging AI, follow-on testing or regression testing occurs only if the system receives any new improvement and then only if the improvement is substantial enough to impact performance. Integrating AI into a system continually evolves our knowledge of the system's ability to accomplish tasks. For systems learning off-line there is the chance they will encounter new data in the field producing an unexpected or new outcome. Additionally, systems allowed to learn online continually evolve as the algorithms process new data. Therefore, T&E must be continuous. As the IDA report notes "systems must be designed to record data about themselves, by themselves." The test continuum needs built-in test modules and the ability to provide an independent observer (human or software, separate from the system) the ability to do independent test result checks.

Additionally, continuously learning systems require occasional correction and/or recertification – test processes need to determine when to apply corrective action. Quality control concepts and control charts provide useful frameworks on how to implement such a system on an AI system.

***Theme 3 – Integrating information from disparate data sources requires methods***

Models, simulations, digital environments, hardware in loop testbeds, and more, all have the potential to provide valuable information about the evolving AI enabled system's performance. Using these different test venues is not new, many different system types have used them for years. However, we currently lack the ability to integrate different data types under one model to draw conclusions at the appropriate inference level. Bayesian hierarchical models and increased computing power have addressed this challenge in defense systems and the national labs to combine information from different data types. Dickinson shows how to combine reliability data from different testing types using both classical regression models and Bayesian hierarchical models. (Dickinson et al. 2015) Researchers from Los Alamos have developed methods for combining quality assurance data and system failure data (Anderson-Cook et al. 2008) and combining component level reliability data and full system reliability data (Anderson-Cook et al. 2007). These methods require both system and mathematical expertise to implement, but provide the opportunity to develop analyses quantifying capability and uncertainty as a reflection of all possible information sources.

***Theme 4 – Data management is foundational***

Better data management is foundational using information from across a continuum in integrated models. The National Security Commission on Artificial Intelligence Interim Report (National Security Commission on Artificial Intelligence 2019) noted:

"AI is only as good as the infrastructure behind it. Within DoD in particular this infrastructure is severely underdeveloped." They go on to note AI is supported by computing power, data storage, and communication networks.

A key data management aspect is understanding the data's quality and pedigree. Lawrence developed a scale for data readiness levels summarized below (Lawrence 2017):

- Level C – Conceive –machine readable data, accessible
- Level B – Believe – the data have known and correct data types, data pedigree known (data history), and documented missing values
- Level A — Analyze — known meta-data and meaningful values to the problem at hand

Since 2017, Lawrence et al. have built on level A data.

- Level AA – Auto-Analyze – data support the automation of bias detection, feature selection, normalization, and more
- Level AAA – Safe – data are issue free, integrity checks, complete provenance information

Understanding the data quality provides context for what data can integrate to various models across the testing continuum. Better data management includes a clear documentation of meta-data and data pedigree, foundational for future AI testing.

***Theme 5 – AI systems require a risk-based test approach***

While statistical methods have always provided a risk-based approach to Test and Evaluation, AI-enabled systems drive risk-based approach necessity and add new dimensions to risk characterization. Test risk needs to consider documented system and algorithm knowledge, operational space expanse, AI type, the decision authority granted to the algorithm, and consequence severity for making an incorrect decision.

***Theme 6 – Previous test metrics still apply, but they may have different interpretations***

The general concepts behind test and evaluation. A system needs to remain effective when operationally realistic operators (representative human user) call to accomplish tasks in the intended environment. This implies the following high level metrics remain relevant:

- Performance: Task- and mission-level
- The 'ilities: Reliability, flexibility, agility, stability, availability, scalability, maintainability
- Security, safety, and human factors considerations

However, the metrics definitions may expand/change. For example, maintainability previously captured the probability of performing a successful repair in an acceptable time period. Hardware repairs and computer restarts have used it. However, in AI-enabled systems, maintainability could reflect algorithm maintenance as operational environment shifts over time.

***Theme 6b – We need new metrics focusing on risks for AI systems***

New metrics needing consideration reflect the AI algorithm's human cognition functions including assessing the algorithm's perception, comprehension, learning, reasoning, and decision making. Additionally, trust and/or reliance metrics can measure human adoption of AI.

***Theme 7 – An expanded definition of threat is necessary***

The threat surface expands for AI-enabled systems. Threats historically have been kinetic (mortars, rocket-propelled grenades, man-portable air defense missile, weapons of mass destruction, ballistic and cruise missiles) or cyber-threats, via internet protocol (IP) and non-IP mediums. Threats to AI-enabled systems are broader and reflect the close intersection of the environment, data collected, algorithms, and tasks the system must perform.

The adversarial AI community has identified new threat categories reflecting this connectivity. They include 1) data poisoning –an adversary attempts to impact the training data to shift the algorithm decision making process in their favor; 2) evasion attacks –the adversary intentionally feeds an already trained algorithm inputs engineered to cause a miss-classification (adding noise to); and 3) model inversion attacks — the adversary attempts to probe the AI algorithms inputs/outputs enough to generate an emulation of the algorithm enabling other attacks. This also provides the adversary with insight into the algorithm's decision making process. These new and expanding attack vectors now must factor into the T&E program.

These attack methods also suggest the owners of systems using AI need to start thinking about the training data security.

*Theme 8 – Operational relevance is essential*

The close interactions between AI-enabled software making decisions, the environments they operate in, and the task they accomplish mean the training (development) environment should closely reflect the true operational environment. This may be a big challenge for systems designed for use in military applications where data and the actual operational environment may only occur very rarely.

*Theme 9 – All AI areas need Testbeds.*

Both development and training purposes need testbeds closely reflecting the operational environment. For AI, testbeds provide the opportunity to conduct experiments on algorithms using actual operational data. The National Artificial Intelligence R&D Strategic Plan ( National Science and Technology Council 2019) highlights the unique opportunity AI testbeds provide, noting "The government has massive amounts of mission sensitive data unique to government, but much of this data cannot be distributed to the outside research community. Appropriate programs could be established for academic and industrial researchers to conduct research within a secured and curated testbed environments." Note this recommendation provides data security and the opportunity to explore expanding operationally relevant threats], key themes highlighted here. The report goes on to note this would create "unique research opportunities" for students.

This unique research opportunity engages students in experiential learning projects and connects them to organizations needing AI talent – bringing us to the final theme: the need for pipeline programs to develop the future workforce.

*Theme 10 – T&E workforce*

The future T&E workforce needs to leverage new, rapidly evolving skillsets. Every report in this article highlights the need for computer science, machine learning, and artificial intelligence skills. However, that does not go far enough. AI adoption is fundamentally about human decision making. Trust, fairness, and reliance on these systems need social science research. We need a T&E workforce skilled in computer science, statistics, and systems engineering, but we also need human sciences.

The National Security Commission on Artificial Intelligence Interim Report (National Security Commission on Artificial Intelligence 2019) notes the current challenges:

"DoD and the IC are failing to capitalize on existing technical talent because they do not have effective ways to identify AI-relevant skills already present in their workforce."

"Agencies need to make better use of pipelines for people with STEM training."

Tapping systems engineering, human factors, phycology, statistics, and computer science backgrounds will help build the workforce pipelines needed for the future. ■

## REFERENCES


- Anderson-Cook, C.M., T. Graves, M. Hamada, N. Hengartner, V.E. Johnson, C.S. Reese, and A.G. Wilson. 2007. "Bayesian Stockpile Reliability Methodology for Complex Systems with Application to a Munitions Stockpile." *Journal of the Military Operations Research Society* 12(2): 25-38.
- Anderson-Cook, C.M., T. Graves, N. Hengartner, R. Klamann, A.K. Wiedlea, A.G. Wilson, G. Anderson, and G. Lopez. 2008. "Reliability Modeling Using Both System Test and Quality Assurance Data." *Journal of the Military Operations Research Society* 13: 5-18.
- Coats, D. and S. Gordon. 2019. "The AIM Initiative: A Strategy for Augmenting Intelligence Using Machines." Office of the Director of National Intelligence. January 16. https://www.dni.gov/files/ODNI/documents/AIM-Strategy.pdf
- Defense Science Board. 2018. "Design and Acquisition of Software for Defense Systems." Department of Defense report. https://dsb.cto.mil/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf.
- Depatment of Defense. 2019. "Summary of the 2018 Department of Defense Artificial Intelligence Strategy." https://media.defense.gov/2019/Feb/12/2002088963/-1/-1/1/SUMMARY-OF-DOD-AI-STRATEGY.PDF.
- Dickinson, R., L.J. Freeman, B.A. Simpson, and A.G. Wilson. 2015. "Statistical Methods for Combining Information: Stryker Family of Vehicles Reliability Case Study." *Journal of Quality Technology* 47(4): 400-415.
- Freeman, L.J. and C. Warner. 2018. "Informing the Warfighter-Why Statistical Methods Matter in Defense Testing." *Chance*. 31(2): 4-11.
- Lawrence, N.D. 2017. "Data Readiness Levels." Cornell University. doi: arXiv:1705.02245
- National Science and Technology Council. 2019. "The National Artificial Intelligence Research and Development Strategic Plan: 2019 Update." Select Committee on Artificial Intelligence Report. June. https://www.nitrd.gov/pubs/National-AI-RD-Strategy-2019.pdf
- National Security Commission on Artificial Intelligence. 2019. "Interim Report." November. https://www.epic.org/foia/epic-v-ai-commission/AI-Commission-Interim-Report-Nov-2019.pdf.
- National Research Council. 1998. *Statistics, Testing, and Defense Acquisition: New Approaches and Methodological Improvements*. Washington, US-DC: The National Academies Press. https://doi.org/10.17226/6037.
- Wojton, H.M., D.J. Porter, Y.K. Pinelis, C.M. Bieber, M.O. McAnally, and L.J. Freeman. 2019. "Operational Testing of Systems with Autonomy." Institute for Defense Analyses. March.


## ABOUT THE AUTHOR

**Dr. Laura Freeman** a research associate professor and the associate director of the Intelligent Systems Lab at the Virginia Tech Hume Center. Her research leverages experimental methods for conducting research that looks at the intersections of cyber-physical systems, data science, artificial intelligence, and machine learning to address critical challenges in national security. She develops new methods for test and evaluation focusing on these emerging system technologies.

Previously, Dr. Freeman was the assistant director of the Operational Evaluation Division at the Institute for Defense Analyses (IDA). During her tenure at IDA, Dr. Freeman participated in test planning activities for over 60 different US DoD systems. She established and developed an interdisciplinary analytical team of statisticians, psychologists, and engineers to advance scientific approaches to DoD test and evaluation. During 2018, Dr. Freeman served as that acting senior technical advisor for Director Operational Test and Evaluation (DOT&E). As the senior technical advisor, Dr. Freeman provided leadership, advice, and counsel to all personnel on technical aspects of testing military systems. She reviewed test strategies, plans, and reports from all systems on DOT&E oversight. Dr. Freeman has a BS in aerospace engineering, a MS in statistics and a PhD in statistics.

MARCH 2020 VOLUME 23/ ISSUE 1

# Exploiting Augmented Intelligence in Systems Engineering and Engineered Systems

**Azad M. Madni,** azad.madni@usc.edu


**ABSTRACT**
With AI resurgence paced by recent machine learning advances, several engineering disciplines including systems engineering turn to AI to improve system model accuracy, process flexibility, content exploration and search, and team productivity. More recently, AI has become a means to augment rather than replace human capability. This perspective alters AI's role from autonomous intelligence to augmented intelligence (AugI). Inherent in this view, recognizing AI and human together can perform certain tasks better than either could alone. This paper presents a methodological framework for effectively exploiting AugI in systems engineering and in engineered intelligent human-machine systems.

**KEYWORDS:** augmented intelligence; artificial intelligence; systems engineering; human-AI collaboration; ontology engineering

## INTRODUCTION

Three themes motivate Systems Engineering's transformation: *development of formal underpinnings* placing systems engineering on the same footing as other engineering disciplines in scientific rigor; *accelerating convergence with digital engineering* to strengthen system verification and testing while spanning the full system life cycle; and *leveraging ongoing convergence* with other disciplines and technologies filling gaps or enhancing performance in problem-solving (Madni 2015). The latter is this paper's topic.

AI today is a promising technology to enhance systems engineering. This requires AI as an associate or an aid to the systems engineer, and as a partner in engineered human-machine systems. Augmented intelligence (AugI) centers around this relationship. AugI's current definition is: AI-augmented human, where AI performs as a human's partner or assistant. However, augmentation can be a two-way proposition: AI-augmented human, and human-aided AI (Madni 1988). In other words, while AI can augment human engineer capabilities, the human engineer can also enhance AI capabilities, especially in planning and decision-making tasks in systems engineering or in engineered human-machine systems (Madni Sievers and Madni 2018).

*Systems Engineering Example*: The engineer(s) define one system's objectives and constraints, AI can refine (decompose) the objectives, recall known options, examine the trade-off space, evaluate candidate options, and make recommendations to human engineers. This way, AI can augment human decision-making performance. Conversely, it is difficult for AI to contextualize a problem situation and keep up with changing context. Therefore, it is difficult for AI to define objectives, and to generate novel options satisfying objectives. In these circumstances, human engineers can define objectives, set parameter ranges, and generate novel alternatives relying on their innate ability to contextualize and create (Madni 2014).

*Engineered System Example*: Consider automatic target recognition (ATR) in an operational combat mission. For this scenario, hilly terrain partially hides a vehicle from view. The ATR system needs to rapidly locate and identify this vehicle. To this end, it needs to determine where to look, locate the vehicle, identify the vehicle type (friend, foe, neutral), and report to the commander. In this problem, it is difficult for the AI to determine where to look (it requires rapid contextualization, something AI is not good at). However, the AI, equipped with a vehicle and recognition technique library can rapidly identify the vehicle even if partially hidden from view. AugI comes in here. The human, being fast in problem contextualization, can tell the AI where to look for potential threats, allowing the AI to locate the vehicle; identify it as a threat, neutral, or friend; and notify the commander. This function allocation between human and machine is "shared perception" (Madni 1994). Today, it serves as an excellent AugI example, where the human augments the AI system capabilities.

Based on the foregoing examples, AugI can work as two-way augmentation, AI-augmented human and human-augmented AI.

Systems engineering (Bird 2017)(Kaminski 2019), and more specifically model based systems engineering (MBSE) can exploit AugI. MBSE is the systems engineering community's response to managing complexity while supporting interdisciplinary collaborative groups in systems development using systems engineering principles, heuristics, and methods with the system model constituting an enduring, authoritative source of truth. Today, MBSE is more mainstream as system modeling languages continue to mature and add new semantics needed for spanning the full systems engineering life cycle.

AI savants would agree we are far from artificial general intelligence (or common-sense reasoning). We need progress on a few important fronts before building systems to independently process, reason, and create at a level comparable to the human brain. But is replicating human intelligence our goal as systems engineers (even though it continues to be a worthwhile goal for AI researchers)? My straightforward and simple answer is no. I believe our goal should be to augment and amplify human intelligence through AI technologies, augmented intelligence (AugI). AugI focuses on complementing human intelligence and supporting humans in performing tasks not performable by the human or AI alone, but possible through human-AI collaboration (Madni and Madni 2018). Importantly, with AugI, the human remains the ultimate decision-maker allaying concerns about "AI taking over the world."

It is important to realize the underlying technologies powering AI and AugI are the same, but the goals, usage, and application context are fundamentally different. AI creates systems operating without the need for humans (dispense with humans), whereas AugI uses AI to augment human performance (which implies maximizing human-AI performance), which for many activities will be superior to the individual human or AI performance. It is also important to realize AugI is not a new technology category; rather, it is a different way to think about AI technology's purpose. While researchers pursue AI to see how far it can go is an admirable goal to realize key technologies proving invaluable in the future, the real world today is better served by AugI.

This paper argues for a shift in how we see humans and AI based on how we architect human-machine systems and how we view AI and systems engineering. Rather than view humans as liabilities needing replaced by AI, humans should be potential assets to exploit when the problem situation calls for ingenuity and creativity. Madni (2010a; 2010b) describes this as "exploiting the human's capability to adapt without exceeding the human's capacity to adapt." This insight is at AugI's core.

Section 2 discusses AI misperceptions along with AI pros and cons. Section 3 argues for AugI in systems engineering. Section 4 presents several key systems engineering-augmented AI concepts. Specifically, it presents AugI applicability and the activities benefiting from AugI. Section 5 presents AugI architectural implications. Section 6 presents a conceptual framework for developing AugI systems. Section 7 summarizes the key concepts presented in this paper and discusses broader implications of AugI.

## SEPARATING MISPERCEPTIONS OF AI FROM ACTUAL PROS AND CONS

People have been uneasy about AI since its inception. This is partially due to fear of the unknown and the distorted view of AI presented by Hollywood movies such as "2001: A Space Odyssey," "Terminator," and "The Matrix" casting AI as some malevolent intelligence. However, there are legitimate concerns about AI. The late, great physicist, Stephen Hawking, cautioned while humans and AI could co-exist, "a rogue AI" could be difficult to stop without appropriate safeguards. Technology leaders such as Space-X's Elon Musk, have made even more dire pronouncements. These concerns have drawn attention to how we intend to use AI, which in turn, requires understanding today's AI's pros and cons.

As with any technology, AI is not inherently good or evil. It all depends on what role AI fills and the attention paid to its implementation. Like any other technology, AI, when hijacked, can be nefarious. This should not deter pursuing AI for societal good and for the country's security and safety. There is no denying AI technology will profoundly impact the job market. Even as it decimates routine and mundane jobs, it expects to create several new jobs for knowledge workers.

Incorporating AI more deeply into our societal fabric requires understanding AI pros and cons. The pros include the ability to offload humans in routine and structured decision-making tasks, facilitate faster decision-making, derive insights from machine learning, perform error-free processing, perform hazardous tasks, and make limited predictions (Madni Sievers and Madni 2018), discussed next.

*Offload Humans in Mundane Tasks*: humans bore easily and lack motivation when performing routine or mundane tasks. They also tire when performing repetitive tasks over extended periods. However, machines do not. Therefore, it makes sense to assign such tasks to machines, and more specifically to AI if they involve contingency handling and decision-making. In fact, AI has successful uses in adaptive process automation, increasing process flexibility and resource utilization efficiency, while offloading humans allowing them to perform more creative tasks.

*Faster Decisions and Actions*: AI and cognitive technologies facilitate rapid decision-making and faster action execution. For example, dynamic planning and scheduling and target detection and identification are problems in which AI excels.

*Derive Insights from Machine Learning (ML)*: In today's world, data continues to grow exponentially with ever-increasing connectivity, instrumentation, and access. Big data, for example, has come to mean datasets in the petabytes. Assimilating such data far exceeds human cognitive bandwidth. This is where AI in ML and specifically unsupervised learning can plough through data at blinding speed to provide timely insights for informed human decision-making.

*Error-Free Processing*: AI algorithms are infallible for routine tasks if properly architected. AI processing can ensure error-free data processing, regardless of dataset size. However, problems calling for judgment are best left to humans.

*Hazardous Task Performance*: AI can perform jobs posing grave risks to humans. Explosive ordnance disposal and space exploration are two such tasks. An autonomous or semi-autonomous robot is ideal for disposing explosive ordnance. Similarly, the Curiosity Mars rover, can autonomously examine Mars surface and dynamically determine the best path to take while continuously "learning" its environment.

*Outcome Prediction*: AI technologies, such as computer vision, can help achieve better outcomes through improved prediction. This capability is crucial for demand forecasting, oil exploration, and medical diagnosis.

AI also has many cons. To begin with, it can engender distrust as a gut response to mitigate through consistent, repeatable performance. It is worth recognizing the human does not have to understand how the AI works to develop trust in the AI given the AI responses are consistent and match human's expectations (Madni 2011). Viewed solely as a replacement for humans, AI poses economic challenges in losing low-skilled jobs, inequitable AI generated wealth distribution, and disproportionate power increase in a few hands. But beyond the economic challenge, AI has other drawbacks such as poor judgment, lack of creativity to deal with "broken plays," inability to keep up with changing contexts, and inability to adapt quickly to changing goals and plans. According to Facebook's Jerome Pesenti, deep learning and current AI have many limitations. In other words, we are far from human intelligence. Today's AI can propagate human biases, is not easy to explain, and does not have common sense. It is more on pattern matching level than robust semantic understanding. However, progress is happening on these fronts, and the field progresses fast.

Poor "judgment" was evident in the 2014 downtown Sydney shooting and hostage incident aftermath. Panicked pedestrians began calling Uber to vacate the area. Because the surge in demand remained in a limited area, Uber's algorithms resorted to supply-and-demand heuristics. As a result, ride rates skyrocketed. It turns out Uber's algorithms had not accounted for such crises. While impatient pedestrians did not care at the time, they were beside themselves when they discovered the exorbitant ride fare that Uber charged them in the crisis. This incident forced Uber to reevaluate its ride fare calculation algorithms for such emergencies.

Human intelligence, a delicate balance of knowledge, emotions, and skills, is always in perpetually adjusting. This characteristic enables human judgment to reflect shades of grey, while stimulus from the real world continues to shape human behavior. Therefore, replacing adaptive human behavior with rigid AI can lead to irrational behavior within ecosystems comprising people and human-made artifacts.

The Uber scenario clearly suggests unless AI algorithms anticipate all contingencies in advance, AI can reach simplistic conclusions and act on them to the detriment of both humans and the environment. In other words, AI programs designed to pursue benign goals may, in the interest of expediency and using simplistic logic, implement a highly unrealistic and unacceptable solution. One example solution is "if there is a problem with over-population in cities, an AI solution may call for reducing the population by any means available rather than proposing more reasonable solutions such as developing adjacent areas and villages for some fraction of the population to move to."

## THE CASE FOR AUGI IN SYSTEMS ENGINEERING

AI is already beginning to impact various systems engineering aspects. These include systems modeling and verification, process management, search and information retrieval, dynamic context management, and human-systems integration. In each case, the AI is AugI. AugI can provide the best of AI and human intelligence while circumventing and/or compensating for their respective limitations. Figure 1 presents the various ways AugI can contribute to systems engineering activities using different AI technologies. As shown in, AI-augmented systems engineering encompasses the system, the infrastructure support facilities, and processes.

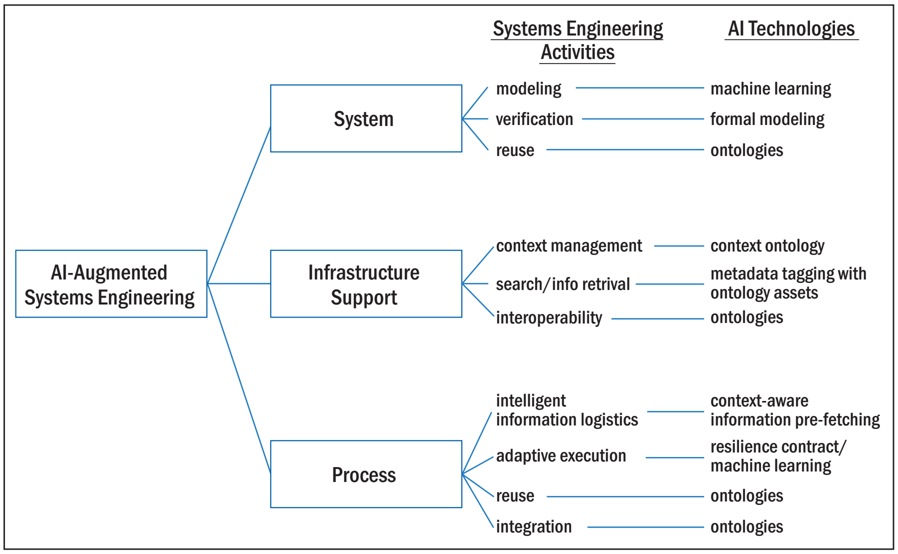



*Figure 1. AugI can contribute to systems engineering activities using AI technologies*

At the *system level*, system modeling, verification, and reuse exploit AI-augmented systems engineering. In system modeling, AugI can exploit ML in reinforcement learning to collect real-time evidence to update an incomplete initial system model, called "closed-loop system modeling." AugI can employ formal logic in contracts to ensure static model correctness. And, system and domain ontologies help identify model fragment reuse opportunities.

At the *infrastructure support level*, AugI can: facilitate dynamic context management where context ontologies define context; accelerate information search and retrieval by metadata tagging content with ontology elements; and facilitate interoperability by exploiting underlying ontology.

At the *process level*, AugI can enable intelligent information logistics (information pre-fetching); adaptive execution by exploiting flexibility afforded by resilience contracts and machine learning; and reuse and integration by leveraging underlying system and domain ontologies.

AugI can take various forms including memory joggers, active prompters and prodders, assistants, associates, and team orchestrators (Madni 1988). However, all these forms involve some human-AI collaboration. Garry Kasparov (2017) in Deep Thinking argues for human-AI collaboration rather than human replacement by AI: "Many jobs will continue to be lost to intelligent automation. But if you're looking for a field that will be booming for many years, get into human-machine collaboration."

Licklider (1960) was prescient in presenting his vision for cognitive assistance. He predicted "in not too many years, human brains and computing machines will be coupled together very tightly, and the resulting partnership will think as no human brain has thought." My own view is "the resulting partnership will think and do as no human brain or AI system in isolation could ever think or do." In other words, I advocate AugI.

AugI is all about human-AI collaboration. I define AugI as teaming human and AI to capitalize on their strengths while circumventing their respective limitations (Madni 2010a). A common refrain applying in this context is: a team of experts does not make an expert team. We know this is true in sports. Think about the Celtics led by Bill Russell, or the Patriots led by Tom Brady. Despite most players not being superstars on these teams, they were championships teams. When Bill Belichick, the Patriots coach, urges his players to, "Do your job," he means more than just do your own tasks (taskwork). He is demanding they be cognizant of interdependencies with teammates so they can excel at teamwork (learn to play as a team). Learning to play as a team means understanding teammate roles, developing mutual expectation awareness, and being cognizant of the interdependencies with teammates (DeChurch and Mesmer-Magnus 2010) (Madni and Madni 2018) (Bansal Nushi Kamar Weld Lasecki and Horvitz 2019). The true competitive team advantage resides in this knowledge of mutual interdependencies and the commitments implied by them (Johnson and Vera 2019).

However, the AI-human teaming problem is more complicated than human-teaming (Kamar 2016). While we know about human teams, we know much less about AI-human teams. AI and humans have different strengths and limitations. Human are superior at rapid contextualization, considered and snap judgments, creative problem-solving, and handling unknowns (known and unknown). However, humans are poor at monitoring infrequent events because they tend to experience a drop in alertness or vigilance. At the same time, they cannot keep up with rapidly occurring events, because they suffer cognitive overload, then fatigue, and eventual drop in motivation. AI is far superior in recalling past events and cases, has faster reaction times, and excels at rare event monitoring and computation. However, AI is not as good as humans in creative problem solving, novel option generation, information aggregation, and responding to previously unseen disruptions.

There are particular tasks neither the human nor AI can do by themselves. These tasks require both the human and AI to work in concert. This is AugI's fertile regime. AugI requires shared contextual awareness, mutual expectation awareness, respective strengths and limitations knowledge, and interdependencies and other constraints understanding. In actual operation, AugI requires knowing which interdependencies and constraints apply in which context.

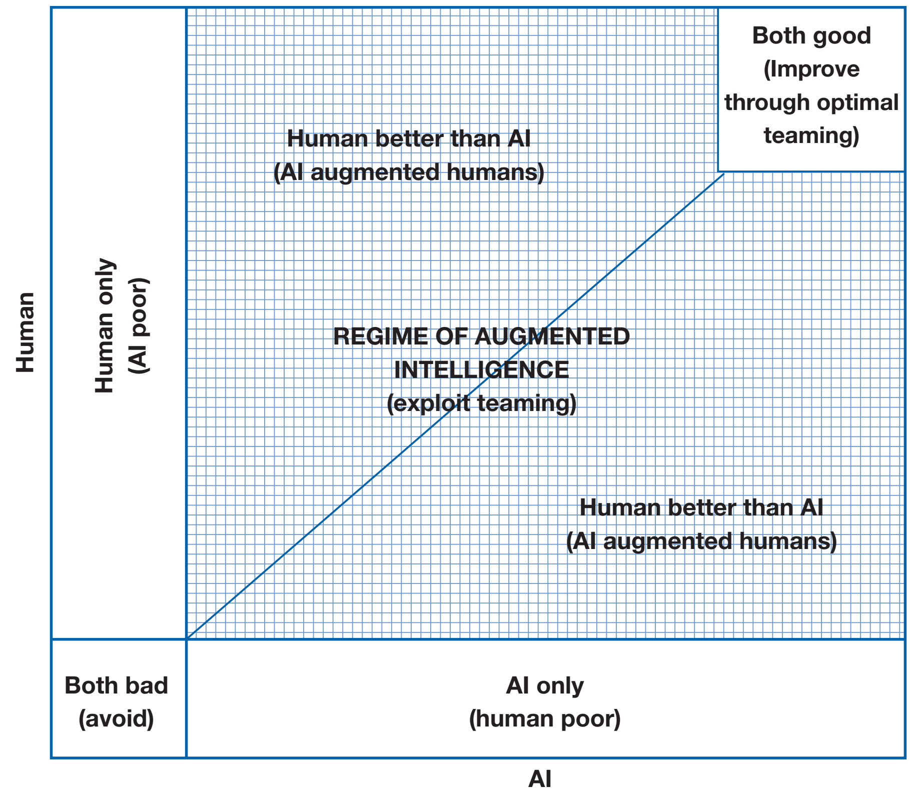



*Figure 2. Task regime for AugI (Adapted from Madni and Madni 2018)*

## ARCHITECTURAL IMPLICATIONS OF AUGI

AugI has several architectural implications. The most obvious implication is since the human will always be part of the system, the system architecture needs to follow human-centered architecting principles. These principles (Madni 2009) are:

- Inspectable, explainable, potentially suboptimal algorithms preferred to opaque/black-box optimization techniques
- Flexible human-AI collaboration at multiple levels based on problem context, tasks
- Shared contextual awareness especially during adaptive collaborative response
- Selective information aggregation to maximize situation and contextual awareness (context-sensitive declutter) without cognitive overload
- Mutual augmentation to maximize joint performance
- Mutual learning of priorities/ preferences in different engineering/ operational contexts
- Shift from traditional human-AI function allocation to human-AI synergy exploitation across various contexts

*Inspectable, Explainable Algorithms*: allow human to intervene in AI operation if needed without making needless assumptions. Consider route planning to a destination with specific waypoints needing traversed. AugI would offload the human by controlling low-level navigation and obstacle avoidance tasks thereby offloading the commander to concentrate on tactics. Unlike an optimization algorithm, AugI would be transparent to the commander allowing the commander to insert additional waypoints, override AugI recommendation, or take over manual control. As important, AI must articulate (be able to explain) its decisions and recommendations when called upon. This is a debate area. What constitutes an explanation? For our purposes, we limit explanation to presenting the rules followed in response to triggering events.

*Flexibility in Human-AI Collaboration Interface*: for AugI to aid the human at different task hierarchy levels, the architecture needs to provide human-AI interactions at multiple levels. In this regard, a context-sensitive, multi-level dashboard could ensure sustained situation awareness in the AI-human team.

*Context-Sensitive Search and Query*: facilitate recall thereby compressing development cycle time. Realizing this capability requires a context ontology (Madni and Madni 2018). Multi-perspective, multi-level visualization requires a domain ontology, class hierarchies for various concepts, and the necessary taxonomies.

*Selective Information Aggregation*: essential for maximizing situation awareness without cognitive overload. Agregating entities the human does not interact with in specific contexts helps achieve it. This capability is dynamic; if context changes certain aggregated elements may need disaggregated while other elements may need aggregated.

*Shift from Human-AI function Allocation to Exploiting Synergy in Human-AI teams*: recognizes, even in those tasks where AI is superior, there are areas human intervention can help to maximize performance. The same is true for tasks where the human is superior, but AI can still provide opportunistic useful intervention.

## CONCEPTUAL FRAMEWORK FOR DEVELOPING AUGI SYSTEMS

Figure 2 presents a conceptual framework informing the methodology for developing AugI systems.

As shown in this figure, there are six partitions associated with human and AI capabilities. The square at the bottom left corner represents tasks both AI and humans do not do well. Examples are time-stressed decision-making with partial information and rapid exhaustive parallel searches. Good design practices avoid this region. The vertical rectangle to the left represents the region humans perform exceedingly well and therefore tasks in this region are best left to humans. The horizontal rectangle at the base represents a region best performed by AI without human intervention. The square at the top right corner represents tasks that AI and humans do well, with. assignment to one or the other based on respective availability. The two trapezoids in the middle represent task regimes where both AI and human can perform, except in the top trapezoid the human is better than AI, and in the bottom one AI is better than the human. These two regimes are fertile grounds for AugI.

The top trapezoid represents the tasks regime where humans are superior to AI. However, AI can augment the human. Examples are recalling past successful and unsuccessful options (designs), visually presenting multi-dimensional tradeoffs, rapidly evaluating options, managing and sharing dynamic context changes, flagging anomalous situations, and learning from both the human and data. The bottom trapezoid represents the task regime where AI is superior to humans. However, the human can augment AI. Examples are goal and parameter setting for a particular mission or problem; rapidly contextualizing a previously unseen or unknown situation; generating novel options to augment the options recalled by AI using metadata tagging; making judgments in uncertainty or ambiguity; and intervening at AI's request, when AI fails to respond in allotted time, or when AI has not previously experienced the situation.

## ROLE OF SYSTEMS ENGINEERING IN AUGMENTED AI

Thus far, this paper has addressed AI and systems engineering, and how AugI can have a major impact on both systems engineering and engineered intelligent human-machine systems. Now this paper discusses how systems engineering can benefit by developing AugI systems. This regard takes a broader systems engineering view to include systems theory, control systems theory (Madni 2018), and systems thinking. In particular, using a control-theoretic construct, we can leverage the analogy between the plant in a traditional control system and the problem domain in AI systems (Alpcan Erfani and Leckie 2017) (Madni 1999)(Passino and Antsaklis 1989). Then, it becomes possible to create concepts for the AI system equivalent to properties such as controllability, observability, stability, system rate, and feedback.

*Observability and controllability* are fundamental concepts needing defining and understanding within an AI framework (Passino and Antsaklis 1989)(Madni, 1999). For example, if a certain event occurs within the system but the information contained in the observations is inadequate for event detection and assessment, then the system (the way we choose to model it and the observations we obtain) is unobservable. Similarly, system states unaffected by any commands we apply could be uncontrollable.

*Stability* is another important concept needing defining and understanding within an AI framework. Stability can fit several views. First, set acceptable states of AI system symbolic variables can define stability. Thus, when a variable in question is outside the acceptable set, appropriate processing brings the system states back into the acceptable set. Second, stability can act as a "string" of acceptable system behaviors. This definition is less restrictive; the system can go through unacceptable states defined above as long as the system continues to exhibit overall desired behavior. For a warfighting system, this could mean there may be temporary setbacks, such as damage to assets, but as long as recovery remains possible and enemy defeat stays achievable, the overall system behavior can be acceptable.

*Performance*, a crucial metric for a control system, can extended to AI systems where performance would imply the rate at which the AI system can achieve the end objective (Madni 1999). In this case, the performance metric must tie to the desired end goal and context.

*Robustness* is another important concept in control systems. This creates a control system designed to withstand changes in the controlled "plant" and external disturbances or disruptions. In the AI realm and more specifically in AugI, the situation is similar. Changes within the system, due to unpredictable decisions or enemy moves, are unknown model changes. Unexpected or "unmeasurable" events affecting the system's response are external disturbances. In this case, the command and control inputs should minimize the outcome changes affected by model changes and/or external disturbances (binding the outcome).

*Time delay* is always a crucial factor in real-time dynamic systems. For AI systems, the collection, assessment, and next input processing time interval is vitally important. Significant time delays can affect performance and often lead to instabilities. For an AI system or AugI system, one can estimate the bounds for allowable delays to guarantee stability and performance. The key hypothesis here an observe-assess-respond cycle time less than the duration between key external events creates a stable system (experience no oscillations).

In sum, formalizing the above-defined control system properties for AI systems will be a major systems engineering contribution to AI as many important AI system properties contributing to their behavior will arise.

## CONCLUDING REMARKS

This paper has advocated augmented intelligence (AugI) to connote the desired symbiosis between humans and AI. AugI engenders greater acceptance than AI does. It does not imagine intelligent robots running amuck while reducing humans to helpless bystanders. More importantly, AugI has crucial implications for the resulting system architecture, testing approach, and metrics. One might say AugI is a construct responding to Lange's (2015) famous refrain "Technology is a useful servant but a dangerous master." AugI also responds to Galbraith's (1967) caution "We are becoming the servants in thought as in action, of the machines we have created to serve us." AugI is a solution to this quagmire. It holds hope without fearing AI taking over. As important, it dispels the misconceptions and hyperbole surrounding AI. Ultimately, AugI is crucial to AI's future understanding and broader acceptance (Bird 2017).

The paper also addresses the role systems engineering, broadly defined, can play in AugI system development. Relying on the classical feedback control systems construct as the guiding pattern, the paper advances ways in which this construct can expand for symbol-processing systems, AugI systems. Specifically, by generalizing concepts such as observability, controllability, and stability, it becomes possible for AugI systems to exhibit predictability and performance lacking in current AI systems. ■

## REFERENCES


- Alpcan, T., S.M. Erfani, and C. Leckie. 2017. "Toward the Starting Line: A Systems Engineering Approach to Strong AI." Cornell University. arXiv:1707.09095v2.
- Bansal, G., B. Nushi, E. Kamar, D.S. Weld, W.S. Lasecki, and E. Horvitz. 2019. "Updates in Human-AI Teams: Understanding and Addressing the Performance/ Compatibility Tradeoff." Paper presented at the 3rd Association for the Advancement of Atificial Intelligence (AAAI) Conference on Artificial Intelligence Honolulu, US_ HI January 27-February 1.
- Bird, S. 2017. "Why AI Must Be Redefined as 'Augmented Intelligence.'" *Venturebeat*. January 9.

- Chakraborti, T. and S. Kambhampati. 2018. "Algorithms for the Greater Good! On Mental Modeling and Acceptable Symbiosis in Human-AI Collaboration." Cornell University. arXiv:1801.09854v1.
- DeChurch, L.A. and J.R. Mesmer-Magnus. 2010. "The Cognitive Underpinnings of Effective Teamwork: A Meta-Analysis." Journal of Applied Psychology 95(1): 32-53. doi: 10.1037/a0017328.
- Galbraith, J.K. 1967. The New Industrial State. Princeton, US-NJ: Princeton University Press.
- Horvitz, E.J. 2007. "Reflections on challenges and promises of mixed-initiative interactions." *AI Magazine*, 28: 13-22.
- Johnson, M. and A. Vera. 2019. "No AI is an Island: The Case for Teaming Intelligence." *AI Magazine* 40 (1), 16-28. https://doi.org/10.1609/aimag.v40i1.2842
- Kamar, E. 2016. "Directions in Hybrid Intelligence: Complementing AI systems with Human Intelligence." Paper presented at the 25th International Joint Conference on Artificial Intelligence (IJCAI), New York, US-NY, July 9-15.
- Kaminski, P.2019. SERC SSRR Keynote, Nov 19.
- Kasparov, G. 2017. Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins. New Yok, US-NY: Hachette Book Group.
- Licklider, J.C.R. 1960. "Man-Computer Symbiosis." *IRE Transactions on Human Factors in Electronics* (1):4-11.
- Madni, A.M. 1988. "The Role of Human Factors in Expert Systems Design and Acceptance." *Human Factors Journal* 30 (4): 395-414.
- ——. "Shared Perception: Capitalizing on the Synergy Between Humans and Machines in Maximizing Situation Awareness in Dynamic Environments." ISTI-WP-10-94-1.
- ——. 1999. "Real-time Symbolic Control of a C2 Enterprise," Proceedings of DARPA's Advances in Enterprise Control Symposium San Diego, US-CA, November 15-16.
- ——. 2010. "Integrating Humans with Software and Systems: Technical Challenges and a Research Agenda." *Systems Engineering* 13 (3): 232-245, Autumn (Fall), 2010a.
- ——. "Integrating Humans with Software and Systems: Technical Challenges and a Research Agenda," Keynote Presentation, 22nd Annual Systems and Software Technology Conference, Salt Lake City, Utah, April 27, 2010b.
- ——. 2014. "Generating Novel Options During Systems Architecting: Psychological Principles, Systems Thinking, and Computer-Based Aiding," *Systems Engineering*, 17(1): 1-9.
- ——. 2015. "Expanding Stakeholder Participation in Upfront System Engineering Through Storytelling in Virtual Worlds." *Systems Engineering* 18 (1):16-27.
- ——. 2018. "Formal Methods for Intelligent Systems Design and Control." Paper presented at AIAA SciTech Forum, 2018 AIAA Information Systems, AIAA InfoTech@Aerospace Kissimmee, US-FL, January 8-12.
- Madni, A.M. and C.C. Madni. 2018. "Architectural Framework for Exploring Adaptive Human-Machine Teaming Options in Simulated Dynamic Environments." MDPI Systems, special issue on *Model-Based Systems Engineering* December 9.
- Madni, A.M., M. Sievers, and C.C. Madni. 2018. "Adaptive Cyber-Physical-Human Systems: Exploiting Cognitive Modeling and Machine Learning in the Control Loop." *INSIGHT* 21 (3): 87-93.
- Passino, K.M. and P.G. Antsaklis. 1989. "A System and Control Theoretic Perspective on Artificial Intelligence Planning Systems." *Applied Artificial Intelligence* 3: 1-32, 1989.

## ABOUT THE AUTHOR

**Azad Madni,** SERC principal investigator, is a professor of astronautical engineering and executive director of the systems architecting and engineering program at the University of Southern California. He is the founder and CEO of Intelligent Systems Technology, Inc., a high-tech company specializing in complex systems engineering research and education for defense, aerospace and automotive industries. He conducts research in distributed autonomy and intelligent systems with emphasis on formal and transdisciplinary model-based methods for engineering adaptive cyber-physical-human systems and system-of-systems. He is an INCOSE life fellow and recipient of INCOSE's pioneer award, founders award, and outstanding service award. He is a life fellow of IEEE, SDPS, and IETE, and fellow of AAAS and AIAA. He is also the recipient of AIAA/ASEE John Leland Atwood award for excellence in aerospace research and education, the ASME CIE leadership award for advancing the use of computers in engineering practice, and the Society of Modeling and Simulation International's presidential award for advancing the modeling and simulation technology. He is also the principal systems engineering advisor to The Aerospace Corporation. He is the author of *Transdisciplinary Systems Engineering: Exploiting Convergence in a Hyperconnected World* (Springer, 2017) and co-author of *Tradeoff Decisions in System Design* (Springer, 2016). He received his BS, MS, and PhD degrees in engineering from UCLA with specialization in engineering systems, computational methods, and AI.

# Motivating a Systems Theory of AI

**Tyler Cody,** tmc4dk@virginia.edu; **Stephen Adams,** sca2c@virginia.edu; and **Peter Beling,** beling@virginia.edu


**ABSTRACT**
Artificial intelligence is an emerging technology with few principled engineering frameworks guiding its application, particularly theoretical frameworks for understanding the interrelationships between systems and their learning processes. We propose addressing this gap by using systems theory as a mathematical superstructure for learning theory. Such a framework would connect learning theory and machine learning directly to model-based systems engineering practices. A general systems learning theory would show how general learning processes relate to general systems and, when applied to particular systems, could reveal how particular learning processes relate to particular systems. Condition-based maintenance and actuators for context to discuss these concepts.

## INTRODUCTION

Artificial intelligence (AI) has moved beyond a research field to a workable approach for building autonomous functions into systems. However, a principled systems engineering discipline for AI-dependent systems has yet to emerge. Although learning algorithms offer statistical performance improvements and allow for novel functionality, interactions with embedded systems raise concerns. There are missing robust test and evaluation practices, deployment sustainment tradecraft, and failure modes in AI systems lack understanding.

Frameworks studying the interrelationships between systems and their learning algorithms are underdeveloped. Although cybernetics and general systems theory research has explored these issues, its findings rely on metaphorical abstractions bringing the frameworks away from the specific nature of the underlying learning processes. Conversely, machine learning and learning theoretic approaches focus on the learning processes in isolation, thereby neglecting the broader systems context.

From this observation, it appears desirable to find a middle ground. Sacrificing some systems theory generality, and some learning theory specificity, a set-theoretic systems learning theory closely knits systems and learning theory together. Set-theoretic systems theory is a mathematical framework for studying the general system natures. Using systems theory as a superstructure for learning, learning algorithms can formally study the systems within which they operate.

## BACKGROUND

General systems theory studies general systems, and Ludwig von Bertalanffy, a founding father of the field, describes its motivation in an observation (Von Bertalanffy 1968):

> "… there exist models, principles, and laws that apply to generalized systems or their subclasses, irrespective of their particular kind, the nature of their component elements, and the relationships or 'forces' between them. It seems legitimate to ask for a theory, not of systems of a more or less special kind, but of universal principles applying to systems in general."

Bertalanffy's efforts began in the 1950's and gave a banner to countless researchers in philosophy, engineering, mathematics, and science whose contributions helped shape and proliferate systems thinking and supported establishing systems engineering as a discipline.

Shortly thereafter, mathematical approaches to general systems theory emerged, replacing metaphors with axioms, and forming the basis of model-based systems engineering. Set-theoretic general systems theories, in particular, were mathematical superstructures allowing for generalizing and integrating more specialized theory (Mesarovic and Takahara 1975)(Wymore 1976). Following the formalization approach, which adds assumptions and mathematical structure in order of their generality, set-theoretic systems theories provided formal, top-down theories leading to principled, top-down methodologies.

AI development closely relates to cybernetics (Weiner 1948), and through cybernetics, to mathematical general systems theory. Learning theory and machine learning are AI subfields concerned with learning to perform tasks from data (Vapnik 2013)(Bishop 2006). At an intuitive level, systems and learning theory closely relate in pursuing general understandings and general methods. This relation persists at a technical level where systems theory is largely a theory of sets and learning theory is a theory of probability, or a theory of measures on those sets.

A mathematical systems learning theory, if developed following M.D. Mesarovic and Wayne Wymore's visions, would give an approach to formally model learning algorithms in systems within which they

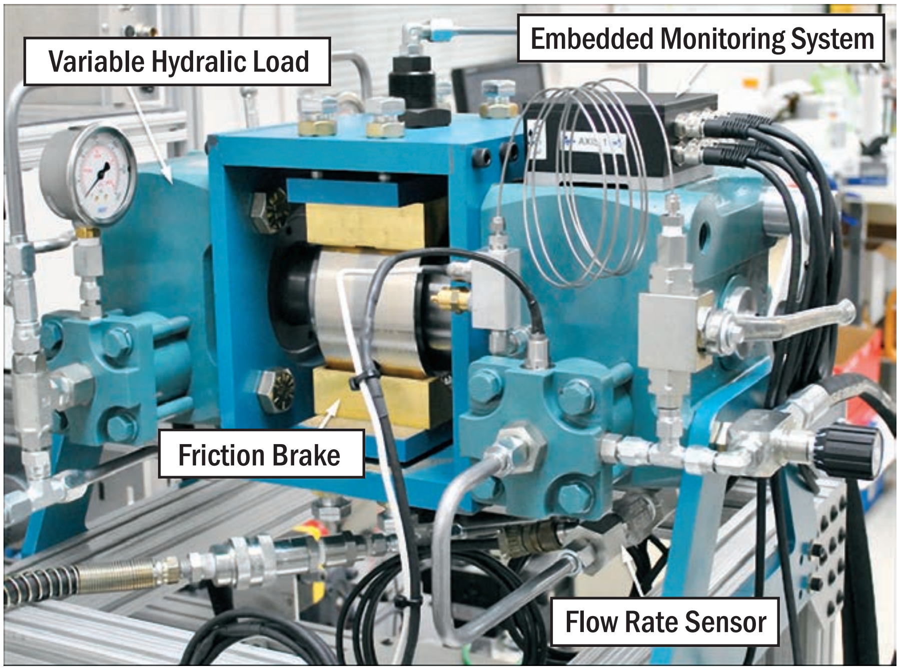



*Figure 1: Hydraulic actuator test stand (Adams Beling Farinholt Brown Polter and Dong 2016)*

operate, a framework studying the logical consequences of using learning algorithms to satisfy particular system functions, and interdisciplinary communication and organization in AI engineering efforts. Importantly, it would do so naturally connecting learning theory and machine learning directly to model-based systems engineering practices. A general systems learning theory would reveal how general learning processes relate to general systems and, when applied to particular systems, could reveal how particular learning processes relate to particular systems.

## MOTIVATING EXAMPLE

Sensorizing machinery has created data-driven approaches to condition-based maintenance. Condition-based maintenance (CBM) uses information about current and future health states to inform maintenance decisions (Jardine Lin and Banjevic 2006)(Peng Dong and Zuo 2010). Numerous fields including wind turbines (Hameed et al. 2009), rotary machines (Lee et al. 2014), electric motors (Nandi Toliyat and Li 2005), and lithium ion batteries (Zhang and Lee 2011) have applied CBM. Machine learning approaches to CBM use data and learned algorithms to predict such health states (Si Wang Hu and Zhou 2011). We discuss a system-theoretic approach to AI in a CBM process with a specific motivating example of condition monitoring for a hydraulic actuator.

Industry and the U.S. military widely use hydraulic actuators. U.S. Navy surface ships utilize hundreds of hydraulic actuators performing numerous functions. Currently, these hydraulic actuators receive service on a time-based maintenance schedule, but moving to a condition-based maintenance routine will reduce cost and improve utility. Data on healthy and faulty conditions collect in a constructed test bed with embedded sensors. Figure 1 depicts the test bed. Standard machine learning classifiers have demonstrated accurate health state estimates (Adams et al. 2016). A hierarchical formulation of the condition-monitoring problem has prediction problem aspects including adding resource constraints (Adams et al. 2019). Other studies have investigated feature selection (Adams et al. 2017) (Meekins et al. 2018) and methods for conserving energy consumption on embedded hardware (Farinholt et al. 2018).

Figure 2 shows a block-diagram of a CBM process, wherein a learning system receives data from a machine and sends a health state prediction to a decision system, which, in turn, results in a maintenance decision for the machine.

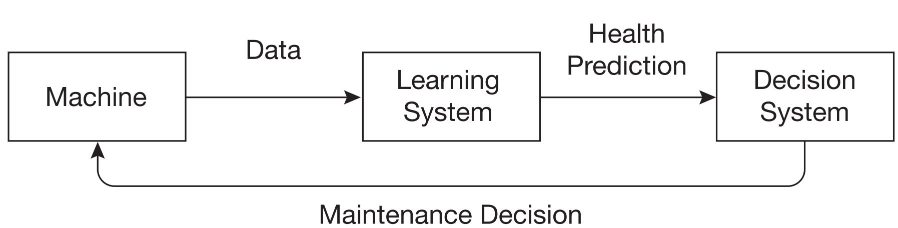



*Figure 2: Block-diagram of condition-based maintenance*

CBM process structures, as depicted in Figure 2, present a fundamental challenge to machine learning for condition monitoring. If the maintenance decision changes the machine, such as part repair or replacement, it also changes the data distribution witnessed by the learning system, and a degradation in the learning system's predictive performance. In previous research, we found such a phenomenon exists when rebuilding hydraulic actuators. Using the previously described test bed, we collected healthy and simulated failure data, deconstructed then reconstructed the actuator, and recollected the data. We found a binary healthy-failed classifier's accuracy dropped from 99% to 77% due to system rebuild associated distribution changes (Cody et al. 2019).

An important and general machine learning approach to addressing distributional changes is transfer learning (Pan and Yang 2009). Transfer learning describes the system's ability to use knowledge learned in previous tasks to help learn novel tasks and is intimately related to propagating knowledge forward through a system's life cycle. The machine learning approach to transfer learning involves transferring knowledge in data, features, and parameters between learning tasks.

Using feature selection and random sampling, we recovered classifier performance of 85% (Cody et al. 2019). Such an approach, however, neglects systems knowledge and, by using a system model, in addition to the transfer learning methods, we can perform better. We modeled the actuator rebuild procedure as a random process and used the random process to provide a probabilistic model of the rebuilt-actuator's behavior to the transfer learning algorithm. With this additional systems knowledge, the performance climbed to 88%, a significant increase from the original 77% considering the inaccuracy consequences.

By modeling part of the process shown in Figure 2, we brought some system perspective benefits to bear on the transfer learning problem. However, such an approach still treats transfer learning as an algorithm-centric discipline. If we assume

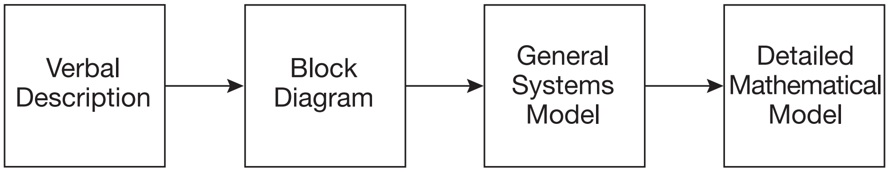



*Figure 3: Mesarovic's systems approach to modeling*

learning algorithms couple to the systems within which they operate then, from a systems theoretic perspective, transfer learning expands to consider not just data and algorithms, but also their systems (Cody Adams and Beling 2019). Under such a perspective, one connecting learning algorithms to their systems, system design emerges as an important mechanism for successful knowledge transfer.

In actuator rebuilds, it is apparent the rebuild procedure's design determines the associated distribution changes witnessed by the learning system. For example, consider completing the rebuild process carefully, but not tediously. If, for example, we measure and replicate the pre-rebuild actuator's fastener tensions during the rebuild, we would expect a smaller distribution change. Similarly, if the rebuild was haphazard we would expect a larger distribution change and a more difficult generalization problem.

System designs and their life cycles clearly relate to generalization problems faced by component learning systems, but formal methods for analyzing and exploiting such relationships are underdeveloped. A systems and learning theory synthesis may help uncover their theoretical nature and, in doing so, reveal important, related properties and measures.

## SYSTEMS THEORETIC MODELING OF LEARNING SYSTEMS

In an effort to demonstrate such a synthesis, we will show a systems theoretic modeling of a learning-based CBM process. Modeling is fundamental to synthesis and analysis and, in engineering contexts, is crucial to system design and operation. The systems approach to modeling is top-down, using a verbal description to create a block diagram, and subsequently a detailed mathematical model.

The problem with such an approach, Mesarovic advocates (Mesarovic and Takahara 1975), is block-diagrams are imprecise and vague, yet detailed mathematics specificity can be inflexible, constraining, and resource-intensive. General systems models bridge this gulf by bringing precise, mathematical formalism to the block-diagram without losing its generality. Figure 3 shows Mesarovic's approach to modeling using general systems theory.

General systems theory may be especially useful in learning, because the gap between general systems learning models and detailed mathematical learning models is comparatively small with respect to other domains, such as biology. Thus, a modeler can iteratively add mathematical structure to general systems models following the formalization approach, considering modeling questions as they arise in their generality order, arriving at a detailed mathematical learning model. We demonstrate this in the following example.

Consider a case where a learning system monitors a machine's health. The modeling problem starts with a verbal description from a hypothetical stakeholder.

> "We would like to model how a learning system can use observations from a machine to predict its health for making maintenance decisions. The model will help design the learning system."

We show how general systems theory can create such a model by adding mathematical structure in successive iterations.

From the verbal description, we can create the block-diagram in Figure 2. At this abstraction level, high-level questions emerge. Is this a reasonable model structure? Does it capture the desired features? Is it granular enough? Because we care most about modeling for learning system design, the detail in Figure 2 seems sufficient, particularly in an iterative formalization process.

We can create an abstract systems model of the block-diagram. Figure 4 shows how a machine $M$ with health state $C$ passes a hypothesis $h$ with input $X$. Hypothesis $h$ outputs a health prediction $Y$ to a decision process $D$, which sends a maintenance action $W$ to $M$. With the added input-output structure choices the $X \times Y$ space emerges. What $Y$ are we trying to predict?

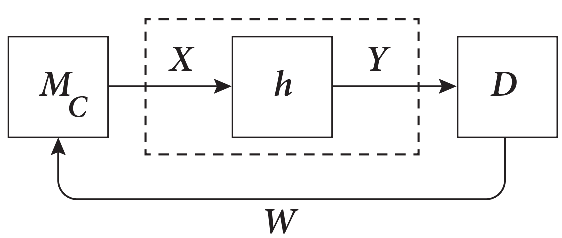



*Figure 4: A Mesarovician general systems model of condition-based maintenance*

Are we predicting discrete health states, and if so, binary or multi-categorical? What are possible $X$ inputs, perhaps considering instrumentation, cost, and security constraints? We can also begin investigating properties like the stationarity of the joint distribution $P(x,y)$, given we narrowed down to a set of particular $X \times Y$ spaces.

Narrowing our scope, we can focus on the learning system's learning theoretic aspects. We can add learning theoretic mathematical structure by adding a learning algorithm $A$ using a sample $S$ to select a hypothesis $h$ from a hypotheses set $\mathcal{H}$, depicted in Figure 5.

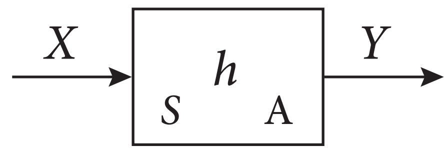



*Figure 5: Adding learning theoretic structure to the learning system*

Questions about the constraints on $A$ and $\mathcal{H}$, for example, due to hardware, or the data size, independence, and distributions in $S$, present themselves. Less quantitative questions emerge at this level abstraction. For example, do predictions made by hypotheses from $\mathcal{H}$ need explaining, and if so, to experts and non-experts alike? How does this impact the specification of $\mathcal{H}$?

We can continue adding mathematical detail. What loss functions should we consider? Do we want theoretical guarantees about the learning process? Should we adopt the empirical risk minimization framework from statistical learning theory (Vapnik 2013) for its computable bounds and convergence properties? Further yet, we can add time into our model, and consider the case where $A$, $S$, and $\mathcal{H}$ are dynamic, perhaps due to iterative development or policy changes.

This case study is merely a thought experiment, and the reader must answer these modeling questions. The broader point we emphasize in this exercise is general systems models, and particularly general systems frameworks for learning systems, facilitate creating top-down, successively more specific modeling considerations. Mesarovic's modeling process, Figure 3, allows for a natural descent from general, verbal system descriptions into specific, mathematically explicit, learning theoretic nuances of a system's component learning systems.

## CONCLUSION

Systems theory has an important role to play in the design and operation of AI systems. The motivating example suggested systems-level knowledge and thinking are integral to developing and deploying machine learning solutions. We discussed its role in learned algorithms' robustness and transfer learning success to motivate more general, theoretical systems research. With this motivation, we showed how learning theoretic mathematical details might join conventional general systems modeling approaches, and the mathematical abstractions related to doing so.

AI and machine learning technologies are moving from laboratories to fielded systems; however, they are far from maturation. Basic and fundamental concerns about these technologies' function in systems are under-addressed. Set-theoretic systems theory offers a formal framework for connecting the general systems knowledge to learning theory particulars. Such a synthesized theory, synthesized in foundational model-based systems engineering frameworks, would provide a base for developing digital engineering methods for AI engineering. Although much AI systems engineering tradecraft will focus on heuristics, empiricism, and lessons learned; systems theory offers a research path for expanding the extent mathematically grounding such tradecraft and, thus, a path towards principled methodologies for AI systems engineering. ■

## REFERENCES


- Adams, S., P.A. Beling, K. Farinholt, N. Brown, S. Polter, and Q. Dong. 2016. "Condition Based Monitoring for a Hydraulic Actuator." Paper presented at the Annual Conference of the Prognostics and Health Management Society 2016, Denver, US-CO. October 2-8.
- Adams, S., R. Meekins, P.A. Beling, K. Farinholt, N. Brown, S. Polter, and Q. Dong. 2017. "A Comparison of Feature Selection and Feature Extraction Techniques for Condition Monitoring of a Hydraulic Actuator." Paper presented at the Annual Conference of the Prognostics and Health Management Society 2017, St. Petersburg, US-FL, October 2-5.
- ——. 2019. "Hierarchical Fault Classification for Resource Constrained Systems." *Mechanical Systems and Signal Processing* (134) doi:101016/j.ymssp.2019.106266
- Bishop, C. 2006. *Pattern Recognition and Machine Learning*. New York, US-NY. Springer.
- Cody, T., S. Adams, and P.A. Beling. 2019. "A Systems Theoretic Perspective on Transfer Learning." Paper presented at 2019 IEEE International Systems Conference (SysCon), Orlando, US-FL, April 8-11.
- Cody, T., S. Adams, P.A. Beling, S. Polter, K. Farinholt, N. Hipwell, A. Chaudhry, K. Castillo, and R. Meekins. 2019. "Transferring Random Samples in Actuator Systems for Binary Damage Detection." Paper presented at 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, US-CA, June 17-20.
- Farinholt, K.M., A. Chaudhry, M. Kim, E. Thompson, N. Hipwell, R. Meekins, S. Adams, P. Beling, and S. Polter. 2018. "Developing Health Management Strategies Using Power Constrained Hardware." Proceedings of the the Annual Conference ofthe PHM Society, 10(1) https://doi.org/10.36001/phmconf.2018.v10i1.584.a.
- Hameed, Z., Y.S. Hong, Y.M. Cho, S.H. Ahon, and C.K. Song. 2009. "Condition Monitoring and Fault Detection of Wind Turbines and Related Algorithms: A Review." *Renewable and Sustainable Energy Reviews* 13(1): 1-39.
- Jardine, A.K.S., D. Lin, and D. Banjevic. 2006. "A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance." Mechanical Systems and Signal Processing, 20(7): 1483-1510.
- Lee, J., F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel. 2014. "Prognostics and Health Management Design for Rotary Machinery Systems-Reviews, Methodology, and Applications." Mechanical Systems and Signal Processing 42(1-2): 314-334.
- Meekins, R., S. Adams, P.A. Beling, K. Farinholt, N. Hipwell, A. Chaudhry, S. Polter, and Q. Dong. 2018. "Cost-Sensitive Classifier Selection When There is Additional Cost Information." Paper presented at International Workshop on Cost-Sensitive Learning, San Diego, US-CA, May 3-5.
- Mesarovic, M.D. and Y. Takahara. 1975. *General Systems Theory: Mathemeatical Foundations Vol. 113*. Cambridge, US-MA: Academic Press.
- Nandi, S., H.A. Toliyat, and X. Li. 2005. "Condition Monitoring and Fault Diagnosis of Electrical Motors-A Review." *IEEE Transactions of Energy Conversion* 20(4): 719-729.
- Pan, S.J. and Q. Yang. 2009. "A Survey on Transfer Learning." *IEEE Transactions on Knowledge and Data Engineering* 22(10): 1345-1359.
- Peng, Y., M. Dong, and M.J. Zuo. 2010. "Current Status of Machine Prognostics in Condition-Based Maintenance: A Review." *The International Journal of Advanced Manufacturing Technology* 50(1-4): 297-313.
- Si, X-S., W. Wang, C-H. Hu, and D-H Zhou. 2011. "Remaining Useful Life Estimation-A Review on the Statistical Data Driven Approaches." *European Journal of Operational Research* 23(1): 1-14.
- Vapnik, V. 2013. The Nature of Statistical Learning Theory. Berlin, Germany: Springer Science & Business Media.
- Von Bertalanffy, L. 1968. *General System Theory*. New York, US-NY: G. Braziller.
- Wiener, N. 1948. Cybernetics: Or Control and Communication in the Animal and the Machine. Cambridge, US-MA: MIT Press.
- Wymore, A.W. 1976. Systems Engineering Methodology for Interdisciplinary Teams. Hoboken, US-NJ: Wiley.
- Zhang, J. and J. Lee. 2011. "A Review on Prognostics and Health Monitoring of Li-ion Batter." *Journal of Power Sources* 196(15): 6007-6014.


## ABOUT THE AUTHORS

**Tyler Cody** received a BS in systems engineering from the University of Virginia (UVa) in spring 2018, and, as part of an early research program, transitioned into UVa's systems engineering doctoral program in fall 2018. He currently studies systems theory, machine learning, and probability under Professor Peter Beling. Tyler's research seeks to connect the foundations of systems theory and learning theory, in an effort to contribute to the broader research community's interest in developing a principled discipline of AI systems engineering. His doctoral work is scoped to investigate relationships between systems design, transfer learning, and generalization, as they relate to system life cycles. Tyler has worked in applied machine learning, including areas such as rotorcraft, satellite communications, and industrial machinery, in experiences stemming from research at UVa and NASA Glenn Research Center.

> continued on page 47

# A Systems Engineering Approach for Artificial Intelligence: Inspired by the VLSI Revolution of Mead & Conway

**Jon Wade,** Jon.Wade@alum.mit.edu; **Jorge Buenfil,** JBuenfil@stevens.edu; and **Paul Collopy,** Paul.Collopy@uah.edu


**ABSTRACT**
Artificial intelligence is at a tipping point where the technical capabilities it enables are outstripping systems engineers' ability to employ it in viable systems. This paper reviews the very large scale integration (VLSI) revolution enabled by Mead and Conway in the late 1970's and the lessons learned from this experience. It applies these principles to engineering artificial intelligence (AI) centric systems and formulates approaches to systems engineering and education to successfully meet this critical challenge.

**KEYWORDS:** systems engineering, artificial intelligence, deep learning, OODA loop, VLSI revolution, AI systems framework, Mead and Conway

## INTRODUCTION

"History doesn't repeat itself, but it often rhymes." —attributed to Samuel Clemens (aka Mark Twain)

The self-assembling organization of complex systems is comprises relatively long slow evolution periods, punctuated by infrequent synergistic paradigm-shifting change bursts (Eldredge and Gould 1972). It is during these rapid change periods, due to numerous coinciding critical factors in the system, it reaches a tipping point enabling new emergent behaviors. Often the necessary seeds of change precede the state change by a significant time period, providing an enabler when all required conditions are ready.

So it is true for technological systems which, when including the vast supply chains, economic and knowledge networks supporting them, are the most complex ones in human history. One such tipping point was the very large scale integration (VSLI) revolution, sparked by inventing the transistor in 1947, funded by the 1960's space programs, and reaching commercial economic scale in the early 1970's. While there was great potential in the technology, human engineering capabilities became the limiting factor for its use. The VLSI revolution, enabled by the engineering processes pioneered by Carver Mead and Lynn Conway in the late 1970's, helped alleviate these engineering limitations. Artificial intelligence is now at a tipping point where human systems engineers' ability to employ it in viable systems limits the technological potential.

This paper first reviews the challenges faced by engineering VLSI systems and the Mead and Conway revolution addressing these issues. It then describes the current situation with AI-centric systems and the similar challenges it presents to systems engineering. Next are the lessons learned from the VLSI revolution which may apply AI-centric systems engineering. Finally, this knowledge helps formulate approaches to successfully address the critical AI challenge.

## MEAD & CONWAY VLSI REVOLUTION

### *Moore's Law and its Impact*

On December 23, 1947 at Bell Laboratories, William Shockley, John Bardeen, and Walter Brattain demonstrated the first solid state germanium transistor. This preceded numerous innovations including using lower-cost silicon (1954), planar fabrication processes (1959), Metal-oxide-semiconductors field effect transistors (MOSFETs) (1959), and self-aligning gates (1967) which led to the first silicon-gate MOS integrated circuits (1968). Gordon Moore, Fairchild Semiconductor's co-founder and Intel's CEO, described observing the number of components per integrated circuit doubled every year and projected this growth would continue for at least another decade (Moore 1965). In 1975, he revised this to transistors doubling every two years. Amazingly 'Moore's law' has arguably held up for the past fifty years.

Investors in new technologies look for a 'killer-app' increasing the odds for rapid market acceptance and profitability. For the semiconductor industry it was the Apollo space program. Integrated circuits had no technological competition for the space program's critical needs for computing power, low-weight, durability, power, and volume. The MIT Instrumentation Lab tried to design the Apollo computer using individual transistors, but after about a year into the effort it became obvious they would not be sufficient. In November 1962, MIT's engineers got NASA's permission to use a new technology: integrated circuits (Fishman 2019). The Apollo computers were their time's most sophisticated general-purpose computers, designed to fly the entire mission on autopilot, while informing the astronauts of the craft's status in real time.

This early development work did two things, it provided critical funding from a reliable source, the US government, and it rapidly drove down cost and dramatically improved quality. Chips costing $1,000 per piece at the beginning of the program dropped to $15 each in 1963 and were $1.58 in 1969. In 1962 the US government bought 100% of the US's integrated circuit production. In 1963 and 1964 the government consumed 85%, reduced to 72% in 1965 at a volume twenty times greater than in 1962. The commercial market realized this new technology potential and made rapid strides in this area. The first integrated circuit microprocessor's design, the Intel 4004 initiated in 1969, and launched in late 1971 with the prophetic ad "Announcing a new era in integrated electronics."

With the rapid development in technology, it became extremely difficult to master all technologies necessary to create a design. Only an expert designer with deep MOS process technology knowledge could hand craft the Intel 4004 and comparable designs. Federico Faggin, hired at Intel in April 1970 from Fairchild Semiconductor as the 4004's project leader and designer, brought his silicon gate transistors mastery he had invented at Fairchild in 1968 and used it to develop his novel methodology for random chip design, key to making all the early Intel microprocessors. While architecturally simple, the 4004 design implementation and fabrication details were obscure to most computer scientists. According to Lynn Conway, the question was "whether the design of VLSI systems would be possible outside Intel moving forward" (Conway 2012).

Until 1979, specialists who understood every technology aspect, from semiconductor fabrication and transistor characteristics to functional blocks of approximately a thousand gates, the limit of chip fabrication at the time, designed IC. In the late 1970s, this approach started to break down. Design was too complex for people who understood the individual technical processes to do it, and the development process itself was complex enough it became the realm of its own specialists. In time, roles would stratify into architects, front-end designers, verification engineers, logic designers, circuit designers, physical layout experts, and physical packaging experts. As a result, most computer architects, the systems engineers' equivalent, lacked knowledge about the rapidly advancing integrated circuit technology, and could not envision how to fully exploit it. Nor could expert circuit designers provide architects with the necessary circuit level hooks to resolve intractable computer architecture problems. Integrated circuit development became limited by the architects and systems-level engineers' ability to effectively design.

| |
|---|
| Applications – Software Engineering |
| Binary Code – Computer Science |
| Architecture – Computer Science |
| Logic – Computer Science |
| Circuits – Electrical Engineering |
| Device Models – Electrical Engineering |
| Device Properties – Device Physics |
| Material Properties – Material Science |

*Figure 1. Abstraction layers in VLSI*

### *Mead and Conway Design Approach*

In 1975, Ivan Sutherland, Carver Mead, and Tom Everhart conducted a major Advanced Research Projects Agency (ARPA) study of microelectronics fabrication's basic limitations. Their ARPA report urgently recommended research into "very-large-scale integrated circuits" system design implications in light of coming advances in scaling, emphasizing no methods existed for handling such complexity and no approaches underway held potential solutions (Sutherland Mead and Everhart 1976). As a result, Bert and Ivan Sutherland formalized a collaborative research project between Xerox Palo Alto Research Center (PARC) and Caltech to explore how to create entire systems in silicon. Among others, Carver Mead joined the team from Caltech while Lynn Conway came in from PARC. The research followed an intensive hands-on learning process, incorporating the latest technologies from Silicon Valley in tool building and prototyping.

By late 1976, Lynn Conway "sensed in our work a parallel to Steinmetz's time – a time when DC (direct current) technology was well established but was running out of steam – while the emerging AC (alternating current) concepts seemed mysterious, even to expert practitioners, who as yet had no formal theories to develop AC technology. Steinmetz had broken the logjam by coalescing mathematical methods and design examples that enabled practicing engineers to routinely design AC electrical systems with predictable results. This starter set of knowledge was sufficient to launch the AC revolution. By applying Steinmetz's principles, practicing engineers spawned a whole new industry" (Conway 1981). Conway believed this was the right approach to take on the VLSI complexity problem. "Instead of visualizing an ever more complex future into which all current and evolving developments were projected, why not begin by simplifying, simplifying, simplifying? Would that not spawn something starkly simple and eminently practical instead?" Conway notes "this wasn't about engineering new things; it was about the engineering of new knowledge. My key idea was to sidestep tons of accumulated vestigial practices in system architecture, logic design, circuit design and circuit layout, and replace them with a coherent but minimalist set of methods sufficient to do any digital design – restructuring the levels of abstraction themselves to be appropriate for MOS-LSI (metal oxide semiconductor – large scale integration)." Figure 1 shows many critical VLSI based system abstractions based on this work. Thus, this approach would support a "simplified methodology for designing whole systems in silicon, not just circuits – and aim it specifically at computer architects and system designers" (Conway 2012).

### *Mead & Conway VLSI Revolution*

The new method's clarity and conciseness enabled architects to design systems from top-to-bottom, as they previously did with vacuum tubes and mechanical switches using simple rules of thumb and heuristics. Long, complicated calculation and simulation sequences were no longer necessary for this process. While this work was still necessary at the lower layers in the process, system design avoided these complications.

If the designers stayed within the abstraction layer limitations, then everything would work as designed.

Given these capabilities, the challenge was how to best transition this knowledge into general practice. In early June 1977, during an evening team-brainstorming meeting at PARC, Conway launched the writing and self-publishing a book idea, which was a novel idea at the time. The book, eventually named "Introduction to VLSI Systems" (Mead and Conway 1980), helped design courses at Caltech and U.C. Berkeley. Lynn Conway received another challenge, to teach a senior/masters-level course at MIT covering the material. Conway developed teaching concepts and methods in the semester's first half, and the students completed a design project in the second half which, after fabrication, would return to the students for testing after completing the course. She "felt that unless students could learn by doing, and make things that worked, they would have merely learned a theory of design." In addition, not only did the project work inform the students, but the students' results and feedback informed the entire process from the design methodology, to the textbooks and notes, the courses, design environments, and implementation methodologies (Conway 1981).

Word spread quickly on the ARPANET about the MIT course the completed advanced projects. As a result, many professors contacted Conway about offering similar courses including the fabricated design projects. In response, Conway's group at PARC began training instructors in teaching VLSI design. In addition, her lecture notes created an "Instructor's Guide to VLSI System Design" (Conway 1979) and those interested in teaching the course received printed copies (Conway 2012). These notes were a critical element containg the full description of the new design methods not fully described in the book. These notes, rather than the textbook alone, provided a complete description of concise, clear, and accessible design to the students, sufficient for them to do VLSI system design.

Another critical fact enabling rapid scale-up was Mead's earlier work in VLSI instruction which started in 1971. This first VLSI course included each student designing a simple project, placed onto a single mask-set, fabricated and then tested by the students. This course repeated each year until 1977 when Mead and Conway wrote the book. The result was the entire design process, the course material , and the fabrication interface evolved on a small scale before any attempt to scale it up. This ensured creating a success driven vehicle for multiple universities to participate in this process using the a "silicon foundry."

The movement was a huge success in its first year. Twelved leading universities participated, allowing 124 designers to create 82 designs using the new Mead-Conway text, the new *Instructors Guidebook*, and the most recent *Guide to LSI Implementation*. Many designs were quite advanced for the time, pushing the limits of system architectures. Following this success, Caltech and MIT organized VLSI conferences, published a VLSI design magazine, and Defense Advanced Research Projects Agency (DARPA) funding launched a VLSI research program. Within two years, 120 universities around the world offered Mead-Conway VLSI courses, with the book translated into Japanese, Italian, French, and Russian. The result was thousands of students educated in the process, venture capital VLSI design startup company funding, EDA companies and foundry services. Concepts such as simplified design methods, new electronic representations of digital design data, scalable design rules, clean formalized digital interfaces between design and manufacturing, and widely accessible silicon foundries suddenly enabled chip designers to create tens of thousands of chip designs. A completely new way of creating VLSI systems on silicon developed. The most significant result was the ability for engineering design capabilities to support exploiting silicon capabilities driven by Moore's Law, effectively enabling a million fold increase in transistors from 1970 to the present.

## SYSTEMS ENGINEERING CHALLENGES WITH AI

### *Artificial Intelligence Background*

As with VLSI technologies and the resultant Moore's Law, artificial intelligence (AI) has a long and eventful history. To understand where AI is today, it is helpful to understand how it developed over the past sixty years. There has always been tension between two purposes for AI: one, understanding the way our minds work, and two, automating human thought. The second purpose leads to thinking robots and to familiar hopes and concerns for the future economy. However, historically, goal one's scientific pursuit has led to progress toward goal two. In the early 1960s, Herbert Simon and Marvin Minsky followed two very different paths to develop software experiments not only producing behavior resembling thinking, but did so with techniques imitating the human brain. The hope was, if a computer thinks like a person, the computer's software could help better understand human thought.

Simon developed Soar, an AI program modeling logical thinking and human memory. Researchers found practical thought required a great deal of logic, and eventually this path led to extremely complicated programs just to move blocks around on a table, and no one believed people's minds truly addressed such complex logical deductions in everyday tasks. However, after simplifying the logic down to a large rule set and a rule-processing engine, AI software performed impressively. By the early 1970s, these rule-based expert systems out-performed humans at diagnosing infectious diseases (MYCIN) and analyzing organic chemicals (DENDRAL). In the 1980s, expert systems functioned in industrial applications such as logistics optimization and call center phone response automation. These rule-based systems' success directly depends on the logic developed by humans who are expert in the particular situation. As a result, it is difficult for these systems to handle new situations and also are limited in their abstraction capabilities, taking insights derived from one situation and applying them to new problems.

Minsky took a much lower-level approach, structuring his software to imitate the way neurons connect in the brain. This software became known as neural networks or neural nets. Neural nets faced the opposite challenge to logic software. Where logic AI seemed far more complex than everyday human thought, neural nets seemed far too simple. Neuroscience moving from a 1960s brain model, containing millions of neurons with matching connections, to the 1990s model, a trillion neurons and countless connections, did not help. Even for modern computers, imitating such a network is intimidating. Also, deeply understanding operating much simpler neural nets with only millions of connections is nearly hopeless. So while the insight into human thought provided through this approach was limited, in the twenty-first century, neural nets are outperforming rule and logic based systems on cognitive tasks. These neural net systems do not rely on precise and exact rules, but rather use statistical models for certain problems, and then 'train' on many data samples to make them more precise and efficient. Neural nets detect patterns from large data sets, and use them to predict where a new data set sample will fall, either in a category (such as cat, dog) or in an approximate numerical value (such as selling prices for houses). Often, the training requires the networks to analyze tens of thousands of data sources to achieve a tiny improvement.

So far, neural net systems have managed to outdo humans at face recognition, speech transcription, object recognition, predicting

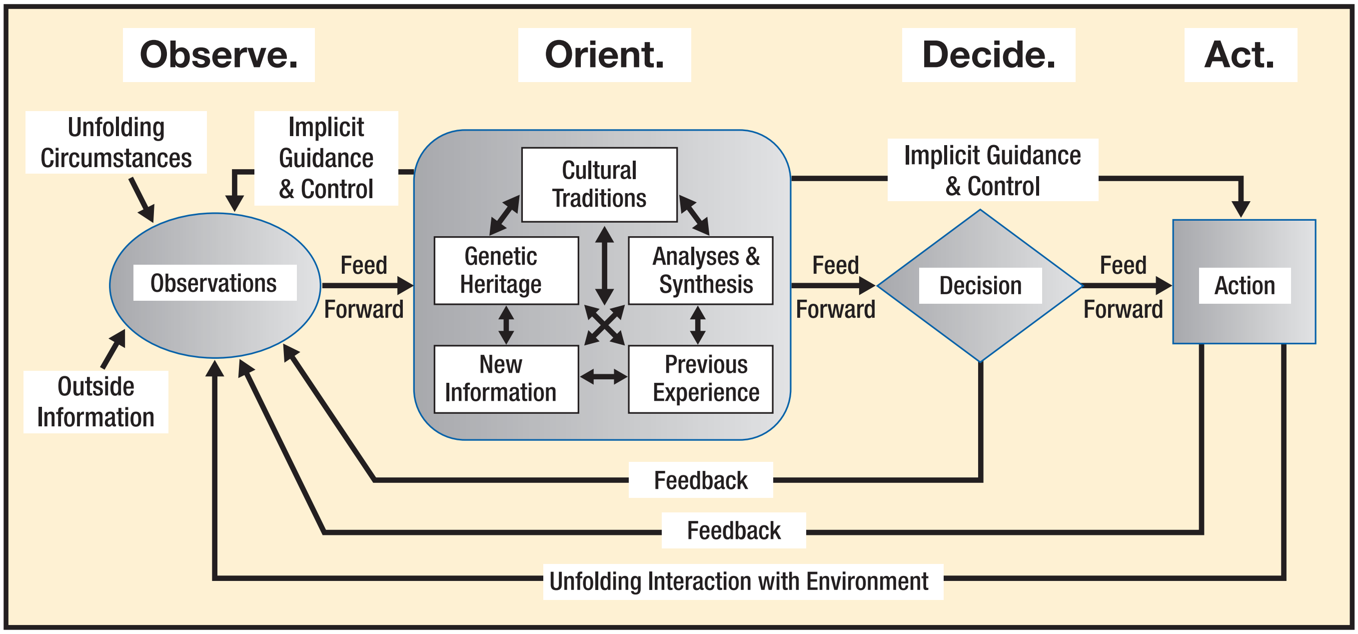



*Figure 2. John Boyd's OODA loop*

relationships, and even playing chess and more recently Go. They are making great leaps forward in translation, and are starting to control autonomous cars and aerial drones. The success of these systems at such complex tasks leave many non-experts baffled as they are not sure how or why they actually work. Major issues neural net systems face pertain to explainability, making them work with small data sets, and gneralizing other domains on their own.

*Success of AI-Intensive Systems*

While experts abilities to explain the 'how' of their decision-making processes severely limited rule-based system's hand crafted knowledge approach, neural net systems have migrated from classic machine learning forms, requiring significant human interaction through structured, labeled data, to deep learning forms which can operate with limited human oversight. This deep learning benefited from numerous external factors, namely computation driven by advancements in AI, Moore's Law and graphics processing units, and the availability of vast data amounts.

There have been advancements on multiple fronts in AI, primarily in more efficient machine-learning algorithms for neural nets and in areas such as graphical models, including Bayesian nets; causal models and Markov models; statistical models; and various ensemble methods.

Graphics processing units (GPUs) drive statistical learning AI systems' performance. In 2018, Nvidia CEO Jensen Huang noted their GPUs were 25 times faster than they were five years ago, versus the factor of 10 achieved by Moore's law, and the performance gain on another benchmark AlexNet, a neural network trained on 15 million images has undergone a speedup of 500x (Perry 2018).

Innovation excels by both hardware and the entire stack. While AI algorithms and software drive capabilities, big data expands the reach. Circa 1980, Robert Metcalfe, an engineer, entrepreneur, and co-inventor of the Ethernet posited a telecommunications network's effect is proportional to the square of connected users. While Moore's law drove the transistor numbers on an integrated circuit, Metcalfe's law has driven the internet and the internet of things' explosion, and both have driven data's hyper-explosion.

In this case, the government did not drive the killer app, but rather of the resulting commercial forces in determining customer preferences (Google, Amazon, Facebook, Netflix, Salesforce) and the financial market in risk and reward assessment. A study shows as early as 2017, over 60% of New York Stock Exchange (NYSE) trading happens algorithmically (Cheng 2017).

***AI Impact on Systems Engineering***

AI has not only had a tremendous impact our society, but also on the systems nature we are envisioning, developing, supporting, and sustaining. A framework useful in understanding this transformation is the *observe-orient-decide-act* (OODA) loop developed by military strategist and United States Air Force Colonel John Boyd as shown in Figure 2. Boyd applied the concept to the combat operations process, often at the operational level during military campaigns. It now applies to understanding commercial operations and learning processes. The approach explains how agility can overcome raw power in dealing with human opponents. Boyd believed whomever could traverse the OODA loop most quickly, would be able to bewilder and confound their opposition. He made a $40 bet he could transform a vulnerable position to dominance in a fighter plan within 40 seconds. He never lost the bet (Coram 2002). Digital engineering's current transformation within systems engineering has in its roots in the OODA loop's acceleration.

As defined by INCOSE, "Systems engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods." However, systems engineers have only designed the *act* system portion and perhaps occasionally the *observe* functionality, whereas the people using the system conduct the *observe-orient-decide* parts. So, while the systems engineers might design a machine to transform or move something, the determination on how to use the machine was usually in the hands of the human operator whose training and oversight were the another party's responsibility. Thus, even though the human's performance is critical to the system's successful operation, a systems engineer would rarely consider the humans' training and their performance's continual review to be their responsibility. Using artificial intelligence changes this equation as engineers increasingly bear responsibility for the *observe, orient*, and *decide* loop portions. This has deep and far reaching consequences.

Rather than blaming failures on human error, the most frequent error type, increasing responsibility falls to systems engineers for how the entire system operates in its environment. As systems engineers are rarely responsible for human training and performance monitoring, it is no wonder this presents extremely challenging responsibilities with technology replacing the human element. AI continually learning as the environment changes also means the systems engineer's job never ends, but rather the system design must successfully evolve. Indeed, if the systems engineer is responsible for these issues, then the system certainly has moved from being merely complicated, behavioral understanding, to where it is complex and fundamentally unpredictable. As such, systems engineering needs to move from an inherently open-loop system, to closed-loop with continual active engineering engagement. OODA loops series and hierarchy represent a closed-loop system.

For these reasons, systems engineering in many instances will need to transform as noted below:

- Systems built to last => systems built to evolve
- Satisfying the requirements => Constant experimentation and innovation
- Phase-based V&V => Continuous V&V
- which the following transitions enable:
- Opinion-based decision making => Data-driven decision making
- Paper-based documents => Model-based documentation
- Deeply integrated architectures => Modular architectures
- Hierarchical organizational model => Ecosystem of partners

As with VLSI in the 1970's, today there is a tremendous challenge in assembling a team with the necessary skills to develop AI intensive systems. Even if the engineering and systems sciences existed to support it, it is unclear where to find the engineers broadly supporting this work. Since 2000, the AI startups have increased by 1400 percent, and since 2013 the jobs requiring AI skills has increased by 450 percent. Based on 2018 LinkedIn data, data science roles have increased 500 percent since 2014 and machine learning roles have increased by 1200 percent. Those with the critical skills tend to go where they are most highly valued; Google and Facebook hire approximately 80 percent of the recent PhD graduates in machine learning (Siebel 2019). Clearly, it is not possible to have significant engineering staff understanding systems and the entire AI stack.

## SYSTEM ENGINEERING AI REVOLUTION

*Lessons from Mead and Conway*

The current situation with AI intensive system engineering has much in common with the environment in the late 1970's VLSI design. In both cases, the technical capabilities' rapid growth far outstripped the ability to engineer the technology, with respect to both complexity and the quantity of people understanding the entire technical stack. Given these dual needs, it is clear education has a major role in developing the engineering knowledge and in training those using it, both grounded in engineering real systems.

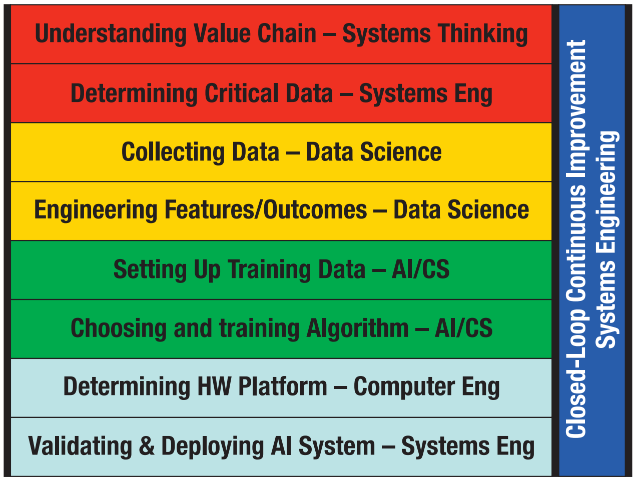



*Figure 3. Abstraction layers in AI enabled systems*

In 2014, a conference panel session and paper reflected on Mead and Conway's heritage (Casele-Rossi 2014). As Prof. Luca Carloni noted, "Thirty years later what has remained the same includes: (1) the importance for interdisciplinary approaches to research and development, (2) the continuous quest for new vertically-integrated scalable design methodologies, and (3) the need for open standards and interchange procedures that foster innovation by enabling collaborative engineering across institutions and beyond geographic constraints." Carloni adds, "the heritage of Mead & Conway lives on due to the continuous need to develop new vertically-integrated design methodologies, which requires reinventing the stack of levels of abstractions to tame design complexity while unleashing performance scalability."

Finally, Professor Alberto Sangiovanni-Vincentelli notes the "meta principles are the pillars of its success over the years, namely simplification, interdisciplinarity, collaboration and orthogonalization of concerns." This 'orthogonalization of concerns' is noteworthy as it reduces complexity by separating various design aspects facilitating independent optimization. There has been a major focus in electronic system design for reuse at all abstraction levels to cope with increasingly complicated integrated circuits and relentless time to market pressure.

*Path Forward for Systems Engineering in AI Intensive Systems*

What is the path forward for AI intensive systems engineering? Clearly, it must take a transdisciplinary approach. Systems engineering and computer science need to not only learn to speak the same language, but to have discussion and continuing dialog. Unfortunately, the silos are strong in academia, and it is the exception rather than the rule to engage in transdisciplinary research and education. Even the adjacent systems and software engineering disciplines almost never interact in academe halls. While this may be less true in industry, the barriers formed during education often carry forward. There is hope, however, as academic silos were also in place in the 1970's at a time when great collaboration took place. With the extreme market need for graduates possessing these skills, and university's competitive desire for the best students, there are rich rewards for institutions taking a leadership position in this area. Some, like MIT making a billion dollar plus bet with the Stephen A. Schwarzman College of Computing (MIT 2018), have not lost the opportunity.

Creating solid abstraction layers supporting the 'orthogonalization of concerns' is critically important. Such a possibility, shown in Figure 3, has similar construct to one for the VLSI stack and captures AI intensive systems' essence which implements OODA loops at all scales.

Topping the stack is value chain understanding within the system which is critical to making decisions producing the desired benefits. Without this understanding, one blindly generates solutions in the hope they will find an acceptable one. This work involves understanding context, relationships and dynamics, and often denotes the systems thinker. The next layer down involves determining the decisions critical to achieving the desired outcome in the value chain, and the data used to support these. This is not something usually done in opinion-based environments, but is increasingly important to systems engineering in a data-driven environment.

After identifying the data type, its sources need determining and the data needs collecting and aggregating to support the necessary decision making. It is likely this process needs automation as this will be an ongoing process. This is data scientist's realm, working with the systems engineer, as these interactions tend to be very iterative. Next, it may be necessary to analyze the data and craft signals data scientists and domain experts believe pertain to the decisions. In addition, the data may need augmenting with meta-data to assist learning.

Next, the actual training data needs to identifying and preparing for learning use. This is a complex process requiring substantial effort to complete. First, the data needs to 'cleaning'; purging incomplete, irrelevant, or incorrect data. Then the data needs formatting for compliance with the appropriate training architecture and implementation. Where there are adversaries, measures must prevent 'data poisoning'. AI expertise is essential in this process. Then, the actual algorithms need selected and trained using the prepared data. Clearly, there will be iterations between the training data and algorithms, both taking place in AI province.

Once this completes, it will be necessary to determine how to execute the algorithms. Given the rapid technology development pace, there will be interplay between the system requirements, the algorithm's needs, and AI software/hardware stack capabilities. The bottom two stack layers involve software and hardware capability interplay, similar to many embedded systems in which the line between software and hardware has become blurred, directed by the system design.

Finally, there is a need to design the system to continually evolve, learning from the data it accumulates. This overall system architecture is the systems engineer's province.

Supporting these layered abstractions are open standards and interchange procedures enabling continual innovation and investment. Defining concise, clear, and sufficient layered abstractions is quite challenging as there are deep semantic and syntactical technical issues. Rather than create a complete ontology, it is important to remember Conway's advice to "simplify, simplify, simplify." In addition, there are numerous technical and non-technical issues with standard creation. This ultimately involves bringing tool creators and vendors, making tradeoffs and decisions not necessarily to any one entity's own self-interests. It is critical the AI ecosystems function as one of a 'growing the pie' versus a zero-sum game so decisions can be for the collective good with prospering for all. However, where it is not possible for commercial organizations to come to a consensus, sometimes open-source work can create the defacto standards.

Again, the AI system stack shown in Figure 3 is only an abstraction layer set example for AI intensive systems. While various roles in the engineering processes already support many activities, some are new or at least adapted from current forms. The actual abstraction layers need developing, verifying, and validating through their implementation in actual designs, and supporting by the appropriate standards and tools communities.

*Role of Education*

As with the Mead and Conway revolution, academics play a critical role in the AI systems engineering revolution. First, an academic environment strongly informed by the art's state and practice challenges is essential as the place where transdisciplinary work can define the 'vertically-integrated scalable design methodologies' required to effectively engineer AI intensive systems. While this work is foundational, its development is an iterative process requiring continuous experimentation and updates. Academia, in real world system class projects, can provide the environment for tool creation and experimentation, and serve as the research laboratory to test these ideas. A one or two semester class focusing first on principles and methodologies, and then secondly on practice in real designs provides an excellent means to evolve design knowledge and student capabilities. This coincides with the role the classroom played in the Meade and Conway VLSI revolution.

## FUTURE STEPS

While there is much work to do, the economic and social forces strongly align to make this revolution happen. Transforming open-loop systems engineering to a closed-loop form is already underway. Systems engineers bearing responsibility for the entire system, including the human elements, occurs in the Boeing 737-MAX saga in which pilot training is one major issue in a system's safety (Langewiesche 2019). Being responsible for the entire system necessitates including the human element, and thus the OODA loop. Work needs to integrate the OODA loop into architectural design allowing the *Orient* and *Decide* capabilities to migrate between human and machine without forcing major changes.

A critical need to develop new vertically-integrated scalable design methodologies, validated and updated in the classroom, exists. As noted, the layered abstractions definition supporting orthogonalized concerns needs completed and integrated into an overall design methodology, and tested with real design flows. Open standards and interchange procedures need to foster innovation by enabling collaborative multi-disciplinary engineering across institutions and beyond geographic constraints.

Finally, some pragmatic tangible steps need taken. A transdisciplinary group needs to write the book, create the courses, and conduct the classes. Let the revolution begin. ■

## REFERENCES


- Casele-Rossi, M. 2014. "The Heritage of Mead & Conway: What Has Remained the Same, What has Changes, What was Missed, and What Lies Ahead." Proceedings of the IEEE 102(2) February.
- Cheng, E. 2017. "Just 10% of Trading is Regular Stock Picking, JPMorgan Estimates." *CNBC* June 13. https://www.cnbc.com/2017/06/13/death-of-the-human-investor-just-10-percent-of-trading-is-regular-stock-picking-jpmorgan-estimates.html.
- Conway, L. 1979. *The MIT '78 VLSI System Design Course: A Guidebook for the Instructor of VLSI System Design*. Palo Alto, US-CA: Xerox PARC.
- ——. 1981. "The MPC Adventures: Experiences with the Generation of VLSI Design and Implementation Methodologies." Xerox PARC Technical Report VLSI-81-2. January.
- ——. 2012. "Reminiscences of the VLSI Revolution: How a Series of Failures Triggered a Paradigm Shift in Digital Design." *IEEE Solid State Circuits Magazine* 4(4): 8-31.
- Coram, R. 2002. *Boyd: The Fighter Pilot Who Changed the Art of War*. New York, US-NY: Brown Little.
- Eldredge, N. and S.J. Gould. 1972. "Punctuated Equilibria: An Alternative to Phyletic Gradualism." In *Models in Paleobiology*, edited by T.J.M Schopf, 82-115. San Francisco, US-CA: Freeman, Cooper, and Company.
- Fishman, C. 2019. "How NASA Gave Birth to Modern Computing-And Gets No Credit for It." *Fast Company*. June 13. https://www.fastcompany.com/90362753/how-nasa-gave-birth-to-modern-computing-and-gets-no-credit-for-it.

- Langewiesche, W. 2019. “What Really Brought Down the Boeing 737 Max?” *New York Times*. September 18.
- Mead, C. and L. Conway. 1980. *Introduction to VLSI Systems*. Reading, US-MA: Addison-Wesley.
- MIT. 2018. “The Institute Launches a College.” MIT Technology Review. December 19. https://www.technologyreview.com/s/612580/the-institute-launches-a-college/.
- Moore, G.E. 1965. “Cramming More Components onto Integrated Cicuits.” Electronics 38(8): 114-117.
- Perry, T.S. 2018. “Move Over, Moore’s Law: Make Way for Huang’s Law.” *IEEE Spectrum*. April 2. https://spectrum.ieee.org/view-from-the-valley/computing/hardware/move-over-moores-law-make-way-for-huangs-law.
- Sieble, T.M. 2019. *Digital Transformation: Survive and Thrive in an Era of Mass Extinction*. New York, US-NY: RosettaBooks. Sutherland, I.E., C.A. Mead, and T.E. Everhart. 1976. “Basic Limitations in Microcicuit Fabrication Technology.” RAND Corporation Report. November.


## ABOUT THE AUTHORS

**Jon Wade, PhD**, is a professor of practice at the Jacobs School of Engineering at the University of California at San Diego where he is currently designing transdisciplinary education and research programs oriented around the fundamental principles of contemporary closed-loop systems engineering design. Previously, Dr. Wade was a research professor in the School of Systems and Enterprises at the Stevens Institute of Technology where he also served as the chief technology officer of the Systems Engineering Research Center (SERC) UARC. His industrial experience includes serving as executive vice president of Engineering at International Game Technology (IGT), senior director of Enterprise Server Development at Sun Microsystems and director of Advanced System Development at Thinking Machines Corporation. His research interests include complex systems, future directions in systems engineering research and using technology in systems engineering and STEM education. Dr. Wade received his SB, SM, EE and PhD degrees in electrical engineering and computer science from the Massachusetts Institute of Technology.

**Jorge Buenfil, PhD,** is a systems architect lead for the Automation, Augmentation, and Artificial Intelligence (AAA) Initiative of the US DoD’s National Geospatial-intelligence Agency. His research interests include ethics for artificial intelligence, federated artificial intelligence systems, artificial intelligence on mobile platforms, and the development of reference architectures for rapid development of intelligent systems. Dr. Buenfil received his BA in political science and public administration, and a MS in computer science from the National University of Mexico (UNAM), a ME in systems engineering from Iowa State University, and a PhD in systems engineering from Stevens Institute of Technology.

**Paul Collopy, PhD, PE**, is professor at the University of Alabama in Huntsville researching topics in systems engineering theory. He previously founded and led DFM Consulting and the Value-Driven Design Institute. He began his career as a manufacturing engineer at Procter & Gamble and an engineering manager at GE Aircraft Engines, where he designed the GE45 engine family. Dr. Collopy has written over 70 papers and cited over 1500 times. He earned his BS in electrical engineering and PhD in engineering-economic systems at Stanford University.

**Tyler Cody et al.** continued from page 40

**Dr. Stephen Adams** is a principal scientist in the Engineering Systems and Environment (ESE) department at the University of Virginia (UVA). He received his PhD from the University of Virginia (UVA) in systems engineering in December of 2015. Prior to joining the PhD program, he worked for UVA’s Environmental Health and Safety department and completed his master’s degree in statistics as a part-time student. His research interests include feature selection, machine learning with cost, sequential decision making, reinforcement learning, and probabilistic modeling of systems. He is currently part of the Adaptive Decision Systems Lab at UVA and his research is applied to several domains including activity recognition, prognostics and health management for manufacturing systems, psychology, cybersecurity, data trustworthiness, natural language processing, and predictive modeling of destination given user geo-information data.

**Peter A. Beling** is a professor of systems engineering at the University of Virginia. Dr. Beling’s research interests include decision-making in complex systems, emphasizing artificial intelligence, adaptive decision support systems, and model-based approaches to system-of-systems design and assessment. His research has found application in a variety of domains, including mission-focused cybersecurity, reconnaissance and surveillance, and prognostic and diagnostic systems. He directs the University of Virginia site of the Center for Visual and Decision Informatics, a National Science Foundation Industry/University Cooperative Research Center, and the Adaptive Decision Systems Laboratory, which focuses on data analytics and decision support in cyber-physical systems.

# Validation Testing of Autonomous Learning Systems

**Paul Collopy,** paul.collopy@uah.edu; **Valerie Sitterle,** valerie.sitterle@gtri.gatech.edu; and **Jennifer Petrillo,** jmp0060@uah.edu


**ABSTRACT**
Validating artificial intelligence systems is challenging. Systems continuing to learn new behaviors in the field are particularly difficult, because the designer does not know what the functions need proving. This article reports on a research project using the Toulmin augmentation method to structure validation test plans and validation testing for very difficult systems like field-learning AI.

**KEYWORDS:** validation; artificial intelligence; machine learning; testing; adaptive

## INTRODUCTION

We have the technological capability to make a system which learns behaviors in the world, acquiring field data to update its deep-learning engine. The goal for this capability is systems, once released into the world, taking actions which we never taught them, and possibly never imagined they would take. Such systems can adapt to a changing environment, outwit a tricky foe, and create novel tactical responses faster than people could. In a military engagement, facing a machine-learning weapon system, perhaps only another machine-learning weapon system can hope to keep up.

However, systems learning in the field present a fundamental challenge to systems engineering. Caught in a paradigm of specifying behavior then verifying that behavior, systems engineers have no way to test a system intended to behave in un-specified ways. The solution to this dilemma does not exist in verification testing, which tests only requirements, so we look to validation testing.

Example: Developing an adaptive pilot's assistant for an attack rotorcraft. For each pilot, the cockpit display is initially passive, but adapts to the views and instruments the pilot requests in various flight conditions and flight condition sequences. It also senses pilot motions and speech, incorporating this data into its learning. The result is a highly tailored interface different for every pilot. The pilot's assistant tests on 100 pilots, but every display configuration comes out differently, and no display configurations have stabilized or stopped adapting at test program's conclusion. While the results to date are encouraging, there is no sense the testing is complete, or ever could be complete.

The traditional approaches to verification and validation testing hoping to exhaustively evaluate the entire action space, inputs and outputs, have become hopelessly impractical even for conventional complex systems (Felder and Collopy 2012). These approaches are even more obviously inadequate for systems learing in the field and formulating their own actions after development is complete. Such learning enabled systems' behavior will be as dynamic as the field data they ingest. As highlighted in the 2010 Air Force Technology Horizon report (Dahm 2010), achieving gains "…from use of autonomous systems will require developing new methods to establish 'certifiable trust in autonomy' through verification and validation (V&V) of the near-infinite state systems that result from high levels of adaptability; the lack of suitable V&V methods today prevents all but relatively low levels of autonomy from being certified for use."

This challenge already exists in software validation. Traditional validation approaches used a decomposable tree to capture the relationship between decision states and algorithmic sequences. For even basic machine learning, however, this is not possible; the algorithm ingests data and then highly interrelated decisions take place within a virtual black box to produce the final result. DARPA's program on Explainable Artificial Intelligence, striving to develop approaches whereby artificial intelligence (AI) systems "have the ability to explain their rationale, characterize their strengths and weaknesses, and convey an understanding of how they will behave in the future" arose from this very problem (DARPA 2016). More recent forays such as DARPA's Assured Autonomy program (DARPA 2017) seek to bind the space for learning enabled systems by limiting operations to environments where a definable ground truth exists, such as autonomous navigation where there are discernible bounds of desirable versus undesirable state spaces. However, systems at the capability frontier will operate confidently in less rigidly defined environments, without predetermined constraints. The most powerful military AIs will learn in theater to identify and classify threats, determine firing solutions, and respond to evolving enemy tactics. For such systems, data in the field is dynamic, uncertain, and unlike training data.

Common sense, and several experienced systems engineers, say just like with an unpredictable learning human being, we will, with experience, come to trust these learning AI systems, and this constitutes system validation. However, it is hard to imagine signing off on releasing a complex billion-dollar system on only trust.

The authors have completed the feasibility research phase on an approach to validating learning AI systems formalizing this trust using a structured soft logic developed by Stephen Toulmin. Here we report the research results.

## WHAT IS VALIDATION?

The Systems Engineering Body of Knowledge (SEBoK) defines system validation as “a set of actions used to check the compliance of any element … with its purpose and functions” and focuses on checking conformance to system requirements. It stresses validation is a process occurring throughout the lifecycle, but too often validation combines with verification and performed as the last step prior to system delivery. When defining the intended system functions in advance, this approach to validation is successful. However, the whole reason for autonomous, adaptable, augmented-intelligence systems is to act in ways not specified by the designers. Thus, validating compliance with functions is too narrow to create trust in a learning system.

Validation relates to but is distinct from assurance, which originally meant the system is not broken (quality assurance), before migrating to a NASA-type view meaning the system is safe. This expanded to mission assurance, guaranteeing the system will not fail to execute the mission, which approaches the SEBoK validation meaning, except for the subtle distinction between system success and system failure avoidance. One can imagine a system defeating a peer enemy in four out of five engagements (valid) breaking in two out of ten engagements (not assured).

Validation standard definitions have regressed for the past 15 or 20 years toward an industry-preferred notion proving the system meets its requirements. Validation

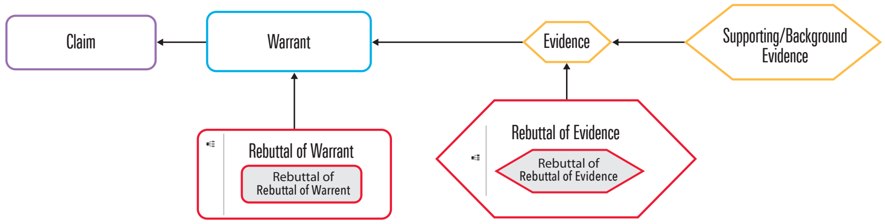



*Figure 1. The Toulmin model of soft logic*

originally intended to prove the system would be successful when deployed. Broadly defined like this, success entails many factors outside industry’s control, which explains contractor tendency to narrow the definition. However, it is still essential to guide system design and development toward success in the field, and validation in its original sense has a critical role.

## RESEARCH ON VALIDATION OF AI AND MACHINE LEARNING SYSTEMS

Many research efforts have addressed specific AI system testing aspects through stability theory. Zhang, Clark, Rattan, and Muse (2015) consider class verification of closed-loop systems containing a neural network controller for first-order, nonlinear systems with uncertainty. In their work, the closed-loop system may face faults in either the cyber or the physical system components. They developed an approach based on Lyapunov stability theory detecting unstable learning behaviors due to unanticipated faults within either system portion. The work focuses on a critical part of the overall problem space, using unstable learning behaviors as a proxy alert for undesirable dynamics, and hence compromising safety. It, however, inwardly focuses as system verification, not validation; the approach considers the system only and does not move toward a methodology providing confidence in system behavior under potentially highly-varying field conditions.

The commonsense approach to Roff and Danks (2018) discusses trust for autonomous systems, recognizing trust is a complex, multi-dimensional notion and not a binary concept. There are varying autonomy levels, each affecting trust development differently, and each posing enormous, albeit varying, challenges to testing, verification, and validation. Trust, while related to predictability and reliability, and characterized through behavior patterns, also contains a psychological aspect, particularly when considering trust across manned-unmanned teaming. The authors consequently advise against relying on binary “yes, trust” or “no, do not” frameworks and methods to discern actionable guidance. They similarly have little faith in purely technological solutions. Their research supports a methodology building confidence through answers to the question “What would it take to convince us X is true?”

Boehm (2006) discusses the parallel evolution of software engineering and systems engineering away from sequential, reductionist methods toward “softer” processes emphasizing a more spiral, continuous learning approach. Boehm observes software requirement emergence is incompatible with traditional, sequential, waterfall process models: “Fundamentally, the theory underlying software and systems engineering process models needs to evolve from purely reductionist “modern” world views (universal, general, timeless, written) to a synthesis of these and situational “postmodern” world views (particular, local, timely, oral) as discussed in (Toulmin 1992).”

## THE TOULMIN ARGUMENT METHOD

Stephen Toulmin was a British philosopher who began his working life during World War II in a role now called a radar systems engineer. After the war, at Cambridge, he studied under Ludwig Wittgenstein who strongly influenced Toulmin’s search for truth in a complex and contextual world. Toulmin found the standard for truth in philosophy was often Propositional Logic, which has been very successful as a truth standard in mathematics. However, this standard was seldom useful outside mathematics because Propositional Logic is too brittle and too restrictive.

Propositional logic focuses on the syllogism, summarized formally as (A and A implies B) proves B. An example might be “Bethesda is a city, and if something is a city, then it must contain buildings, therefore Bethesda contains buildings.” The positive aspect of the syllogism is, when usable, it is very reliable. The negative side is, in contextual reasoning about real-world situations, syllogisms can prove very little beyond what is obviously true in the first place.

To address this shortfall, Toulmin developed a softer and more robust deductive logic that has since found application in many disciplines (Toulmin 2003). Figure 1 shows Toulmin’s logic’s basic elements. He begins with a claim, a true or

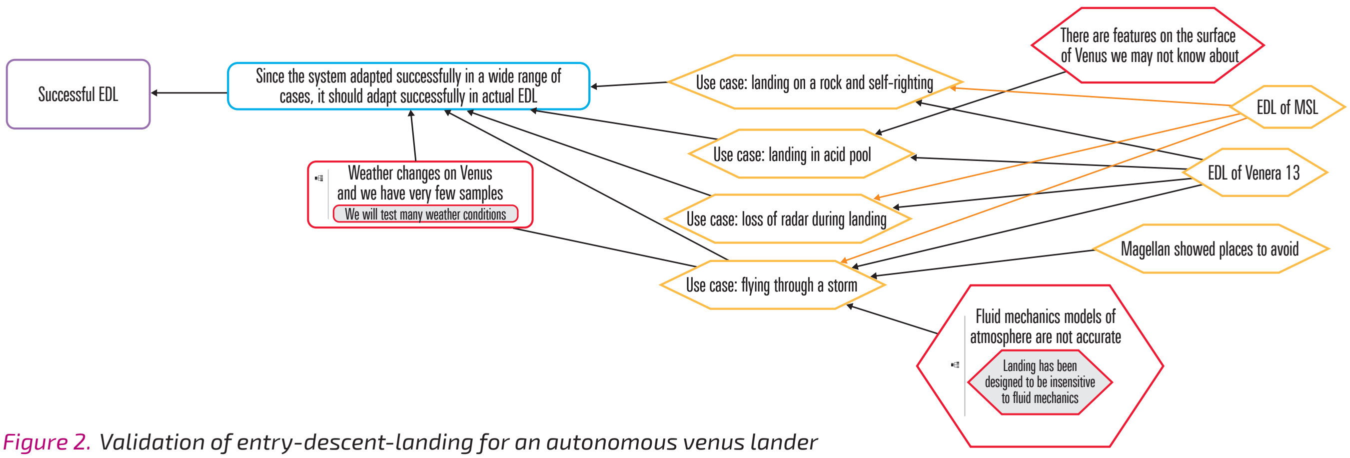



*Figure 2. Validation of entry-descent-landing for an autonomous venus lander*

false statement, and we would like to know whether it is true. Evidence, drawn from the world or from experience, supports the claim. So far, the claim replaces B in the syllogism, and the evidence replaces A. Toulmin creates a warrant to replace "A implies B." The warrant shows how the evidence leads to concluding the claim is true. The Toulmin system's strength is the warrant can employ practical reasoning and common sense. Sterile statements such as "A implies B" no longer restrict it. The warrant can be any acceptable argument, such as "in our experience, when things like A happen, they almost always bring about events like B."

Other elements support or clarify the relations between data, warrant, and claim. The backing is a way to reference a library, authority, or experience lending credibility to the warrant. Rebuttals restrict the data or warrants 'applicability – the rebuttal identifies areas where the data does not apply or the warrant is not legitimate.

Figure 2 shows this method's first fictitious application to validate a Venus autonomous lander system, illustrating our prototype tool, based on Vue.js, an open-source Java-based framework. The claim is the Entry-Descent-Landing phase of the lander will be successful. Note no reference to functional requirements support the claim. One goal was to show validation can occur without functional requirements, a necessary step to validate field-adapting systems, where the future function set is unknown at design time.

The blue warrant conveys the essential argument why the evidence supports the claim. The evidence is largely use cases, particularly the evidence directly supporting the warrant. The prevalence of use cases is not a surprising result, as testing complex transactional software, where exhaustive state checking is not feasible, commonly employs them. Use cases helped avoid direct individual function tests, such as deploying a parachute and observing whether

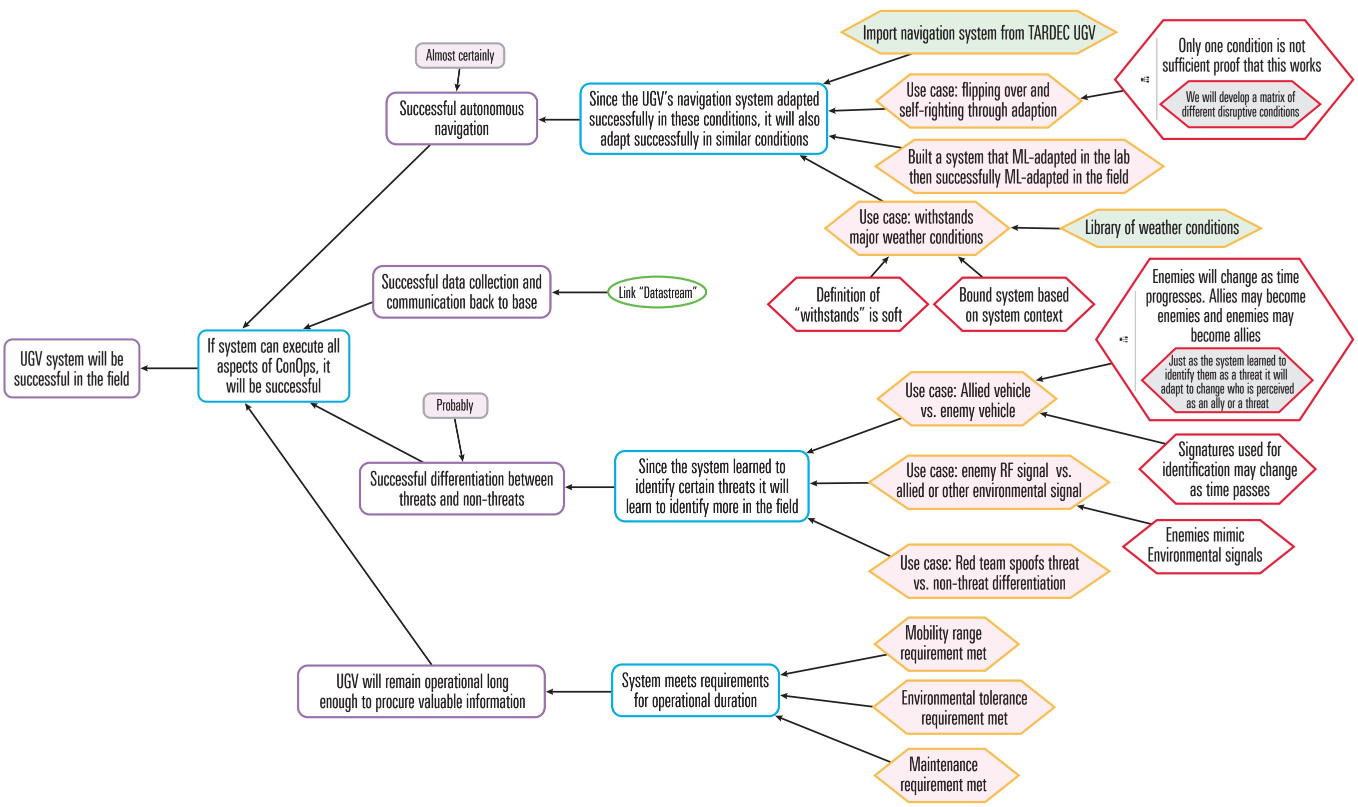



*Figure 3. A second application*

the parachute actually deployed. We had no objections to functional testing, but we knew if the new approach relied entirely on functional tests, we would have demonstrated a replacement for verification testing rather than a new approach to testing. It is our opinion verification testing does not need replacing, but adaptive systems seriously challenge validation testing.

The red polygons are rebuttals, which can show flaws or limits to warrants and evidence. Two rebuttals in Figure 2 contain gray rebuttals-to-the-rebuttal the rebuttal's impact.

Figure 3 shows our second prototype application, an unmanned ground vehicle (UGV) to perform reconnaissance. Again, this is a fictitious example to assess the method's basic feasibility.

In this graph, data shaded green means the data is currently available, and purple means there is a plan to collect the data or run the test, but nothing is available yet. We reserve yellow for a test or investigation currently in progress, but without complete results. In this way, the graph acts as a planning tool for validation as well as a record for when testing is complete.

This graph also includes qualifiers in gray borders applied to the blue warrants. Qualifiers are a major difference between propositional logic and the Toulmin model. In propositional logic, any statement can only possess one of three statuses with respect to truth: the statement is true, false, or undecided. The Toulmin model, in deference to actual language and the nature of fact in the real world, allows all qualifier levels. Statements can be possible, likely, almost surely true, or most definitely true. Qualifiers can apply to data, warrants, or claims to indicate the extent to which they are true. When Bayesian reasoning is available, qualifiers can be numerical probabilities. In some cases this will allow various data and warrant qualifiers to mathematically combine to form the claim's qualifier, an uncertainty quantification.

We also added the notion a warrant could show how claims might justify a more general claim. In order to preserve a hierarchical structure necessary to describing a complex validation, we devised a link connecting one graph to another.

In the end, validation rests on evidence collection comprising tests, analyses, data, and the usual validation components today. However our model compiles this evidence into the logic by which the evidence supports trust in the system's success. The model facilitates asking if evidence is sufficient, specifically where it needs shoring up. Impact on claims strength assists project management decisions to not execute or delay a test.

Our brief study, reported here, demonstrated the method's rough feasibility, and we learned much about applying it. A follow-on project could develop this validation method by applying it to actual systems and training systems engineers in using the method, then observing their performance to feed improvements back into the training. At this point, the method has shown it might be quite effective, and no showstoppers have appeared casting shadows on its feasibility for full-scale application to complex systems. Advantages we have observed for the Toulmin model include:

- Because the model is formal, it organizes validation test lists and shows an explicit and reasoned connection between tests, the specific aspects, and capabilities validated by the test.
- Warrants can capture the interaction between data from various tests and other sources. A surprisingly explicit and detailed result from one test may compensate for an unfortunately ambiguous result from another test.

We look forward to returning to *INSIGHT* with more information about this new approach as it further develops. ■

## REFERENCES


- Boehm, B. 2006. “Some future trends and implications for systems and software engineering processes.” *Systems Engineering* 9(1): 1-19.
- Dahm, W. J. 2010. *Technology Horizons: a Vision for Air Force Science & Technology During 2010-2030*. Arlington, US-VA: USAF HQ.
- DARPA 2016. *Explainable Artificial Intelligence* (XAI) DARPA-BAA-16-53 Broad Agency Announcement.
- DARPA 2017. “Assured Autonomy, I2O Program Description.” https://www.darpa.mil/program/assured-autonomy.
- Felder W. and P. D. Collopy 2012. “The Elephant in the Mist: What We Don't Know about the Design, Development, Test and Management of Complex Systems.” *Journal of Aerospace Operations* 1(4), 317–327, DOI10.3233.
- Roff, H. M. and D. Danks 2018. “Trust but Verify: The Difficulty of Trusting Autonomous Weapons Systems.” *Journal of Military Ethics* 17(1): 2-20.
- Toulmin, S. E. 2003. *The Uses of Argument*. Updated edition. Cambridge, UK: Cambridge University Press.
- ——. 1992. *Cosmopolis: The hidden agenda of modernity*. Chicago, US-IL: University of Chicago Press.
- Zhang, X., M. Clark, K. Rattan, and J. Muse 2015. “Controller Verification in Adaptive Learning Systems Towards Trusted Autonomy. “*Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*: 31-40. New York, US-NY: Association for Computing Machinery.


## ABOUT THE AUTHORS


**Paul Collopy** is a professor at the University of Alabama in Huntsville who researches theoretical aspects of systems engineering. He earned a BS in electrical engineering and a PhD in engineering-economic systems at Stanford University.

**Valerie Sitterle** is a principle research engineer at Georgia Tech Research Institute and a member of the research council of the Department of Defense's Systems Engineering Research Center. She holds BME and MS degrees in mechanical engineering from Auburn, a MS in aerospace engineering from Florida, and a PhD in mechanical engineering from Georgia Tech.

**Jennifer Petrillo** is an undergraduate student at the University of Alabama in Huntsville majoring in computer science.

SPECIAL FEATURE
MARCH 2020
VOLUME 23/ ISSUE 1

# AI as Systems Engineering Augmented Intelligence for Systems Engineers

**William B. Rouse**


**ABSTRACT**
AI technology's rapid development has led to various arguments for SE for AI, as well as AI for systems engineering. This article argues for and illustrates a third perspective — AI as systems engineering. A conceptual architecture articulates employing AI as an assistant systems engineering engineer. This cognitive assistant understands systems engineering concepts, principles, methods, and tools as well as the workflows and preferences of the particular supported engineer. The vision is a cognitive assistant understanding your domain, your current intentions, and you.

## INTRODUCTION

The first artificial intelligence (AI) wave emerged in the mid 1950s, with an emphasis on rule-based symbolic logic, some statistical modeling included. Overselling the AI prospects led to the *first AI winter* in the mid 1970s. Expert systems led to a second interest wave in the 1980s, but by the late 1980s, the *second AI winter* emerged. The third wave began in the 2000s with maturing "deep learning" via multi-layered statistical models. It remains unknown whether a *third AI winter* will emerge (Rouse and Spohrer 2018).

The hype associated with machine learning is extraordinary with various pundits projecting pending nirvana and others projecting human work elimination, economic chaos, and revolt. Brockman (2019) presents 25 different views by various experts. There is a middle ground recognizing AI as a technnology spectrum ranging from deep learning to rule-based symbolic logic. This spectrum can enable augmenting human intelligence to substantially enhance human-machine capabilities and performance.

Figure 1 portrays the AI spectrum from statistical models to symbolic logic. Machine learning can be superior for recognition and classification. Symbolic logic is better for problem solving, teaching, and advising. Decision making can benefit from both approaches, as illustrated later. Consider a system to teach algebra. I can imagine training machine learning with many problems and solutions so it can readily solve such problems.

However, if system needed to teach humans how to do algebra, this system would be useless because it could not explain how it was solving problems. In fact it is doubtful humans could employ the machine's approach even if it could explain it. This reminds me of a recent article reporting on how many Stop signs trained machine learning to recognize these signs. In contrast, when my three-year old grandson asked, "What is that sign called?" and I told him it was a Stop sign, he insisted on pointing out Stop signs for the rest of the day. He only needed one example.

## AUGMENTED INTELLIGENCE

Figure 2 provides an overall architecture for augmenting intelligence, drawing on Figure 1's AI spectrum, and building on a long evolving construct application series (Rouse 2019). The intelligent interface becomes a component in the broader augmentation. The overall logic is:

- Humans see displays and controls, and decide and act. Humans need not deal with anything other than these three architecture elements. The overall system frames human's roles and tasks and provides support accordingly.
- The intent inference function infers what task(s) humans intend to do. This function retrieves information and control needed for these task(s). The information management function determines displays and controls appropriate for meeting information and control needs
- The intelligent tutoring function infers humans' knowledge and skill deficits relative to these task(s). If humans cannot perform the task(s) acceptably, the information management function either provides just-in-time training or informs adaptive aiding (see below).

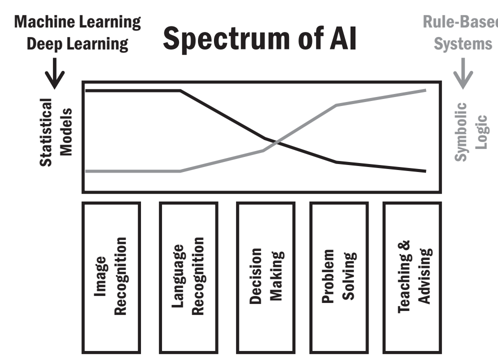


*Figure 1. The spectrum of AI*

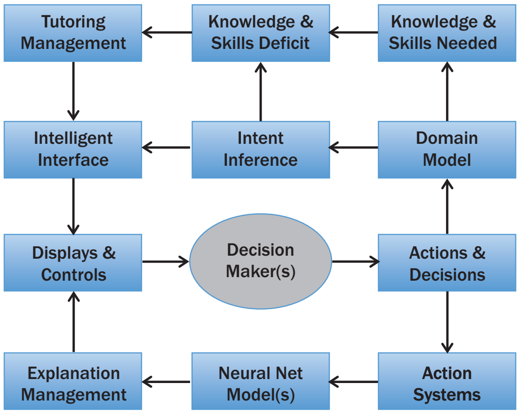



*Figure 2. Overall architecture of augmented intelligence*

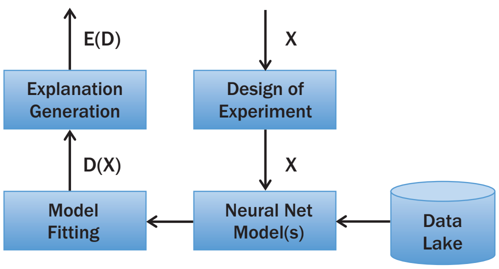



*Figure 3. Explanation management function*

- Deep learning neural nets provide recommended actions and decisions. The explanation management function explains these recommendations to the requested extent. The next section further elaborates on this function.
- The adaptive aiding function, within the intelligent interface, determines the human's execution role. This can range from manual to automatic control, with execution typically somewhere between these extremes. The error monitoring function, within the intelligent interface, detects, classifies, and remediates anomalies.

Note these functions influence each other. For example, if adaptive aiding determines humans should perform task(s), intelligent tutoring assesses necessary knowledge and skill availability, and determines training interventions needed, and information management provides the tutoring experiences to augment knowledge and skills. On the other hand, if adaptive aiding determines automation should perform task(s), intelligent tutoring assesses humans' abilities to monitor automation, assuming such monitoring is a requirement.

## EXPLANATION MANAGEMENT

As noted earlier, neural network models cannot explain their (recommended) decisions. This would seem to be a fundamental limitation. However, science has long addressed the need to understand systems unable to explain their own behaviors. Experimental methods develop statistical input-output relationship models. Applying these methods to neural network models can yield mathematical models enabling explaining the (recommended) decisions as shown in Figure 3.

Given independent variables **X**, a statistical experiment design, for example, a fractional factorial design, can determine the **X** value combinations to input to the neural net model(s). These models, typically multi-layered, have "learned" from exposure to massive data lakes with labeled true positive instances, possibly false positives, and false negatives. True negatives are the remaining instances.

The neural net models yield decisions, **D**, in response to the designed **X** combinations of **X**. A model **D(X)**, fits to these input-output data sets. Explanation generation then yields explanations **E(D)** based on the attributes and weights in the fitted model. The result is a first-order, non-deep, neural net (recommended) decision explanation.

As noted earlier, the paradigm underlying Figure 3 is the standard empirical natural science paradigm. Thus, it is clear it will work, yield rule-based explanations, but will it be sufficient to help decision makers understand and accept what the machine learning recommends? I imagine this will depend on the application.

As an example, consider control theory. Optimal stochastic control theory includes both optimal estimation and optimal control. Determining the optimal solution across both estimation and control involves rather sophisticated mathematics. We could apply the method in Figure 3 to the optimal control actions resulting from this stochastic control problem's solution.

We would not be able to infer the underlying sophisticated mathematics' nature. Instead, we would likely unearth something akin to classic PID controllers, where the acronym means proportional, integral, and derivative attributes of the errors between desired and actual states. This has provided a reasonable optimal control action explanation.

## AUGMENTING SYSTEMS ENGINEERING

The following scenario provides a vision for how AI might augment a systems engineer.

*Dave Sawyer has led the systems engineering SWAT team (SEST) for two months. SEST's role is to solve tough systems problems quickly. He inherited this role when his mentor retired. He had prepared for this role. He was happy to run with the ball.*

*Today, he feels more pressure than happiness. Their driverless car program's (Apollo) head noticed the advanced prototype they have tested at the Proving Grounds consistently makes a rather odd error. It occasionally makes a small, jerking movement to the right and then immediately recovers. It is rather disconcerting to passengers in the back seat.*

*Dave and his team have two days to provide insights and one week to recommend solutions before the blue ribbon oversight committee shows up for an evaluation of Apollo. Fortunately, they have the **systems engineering advisor** (SEA) to help them. Meeting these deadlines would be impossible without this AI-based platform.*

*Dave decides to explore the resources available before the first meeting with his team tomorrow morning. He logs into the secure SEA platform. Its intelligent interface immediately engages him.*

*"Welcome back, Dave. Do you want to continue the analysis you were working on?"*

*"No, Marie. I have a new problem."*

*Users of SEA can name their cognitive assistant. Marie was Dave's favorite aunt, almost his second mother.*

*"So, Marie, what do you know about control algorithms in driverless cars?"*

*"That's a broad question. Do you really want to know everything?"*

*"No, of course not. I will just assume you know everything and will help me get to exactly what we need."*

*"Let's just assume I know what can be known, but not everything."*

*"Great. Show me the trajectory data for the most recent Apollo tests."*

*"Do you want it aggregated across test runs, or just each individual run?"*

*"Actually, both would be a good idea."*

*A large interactive visualization appears almost instantly. It is quickly apparent the slight jerk to the right does not always occur at the same place. Thus, it has nothing to do with the track itself.*

*"I find it hard to believe these jerks are just random."*

*"Apollo is not just responding to the road. It is also responding to other vehicles," Marie offers.*

*"Good point. Do you have data on other nearby vehicles over time?"*

*"Yes. I thought you might want that, so I asked this data be transferred as well."*

*"Highlight in red the segments of the traces you just showed me whenever another oncoming vehicle is within 30 seconds of Apollo."*

*The highlighted traces appeared immediately. The slight jerks to the right always occurred when the segment was red, but it did not happen during all the red segments.*

*"Apollo is clearly reacting to the other vehicles, but not all of them."*

*"Do you want the trajectory traces of the other vehicles in each red segment?"*

*"Yes, that's a good idea."*

*"Should I also average across runs where Apollo reacts, with another average for when Apollo does not react?"*

*"Yes, great."*

*Another plot set quickly appeared.*

*"The differences between the two averages seem real, but quite subtle."*

*"Even if the differences are very small, keep in mind Apollo can sense things much better than you can," Marie responds.*

*"OK, but how do we explain these differences?"*

*"There is a suite of biomedical sensors used for all human-driven vehicles at the Proving Ground. It is a lot of data. Do you want to see it all?"*

*"Do you know which measures are the best predictors of drivers' loss of vigilance?"*

*"Yes, EEG is best."*

*"Great let's see it."*

*"These measurements are pretty noisy. Should I smooth them out a bit?"*

*"Yes, that will help."*

*"Two averages again?"*

*"Yes."*

*The plots were quite clear.*

*"Apollo is reacting to the oncoming drivers' fading vigilance."*

*"Yes, and Apollo is inferring this, without realizing it, from subtle movements of the oncoming vehicle."*

*"So, we know why the slight jerks to the right are happening, but what do we do about it?"*

*"Should I make sure the algorithm people are in tomorrow's meeting?" Marie asked.*

*"Absolutely. Put together a montage of everything we have done, with annotations for team members."*

*"Will do. It will broadcast in the next couple of minutes."*

*"Also, change the calendar to make tomorrow morning's meeting top priority. The Apollo program manager has to be there because we need an increased budget commitment from him."*

*"I am sure he will like that."*

*"Do I detect a bit of sarcasm, Marie?*

*"I am doing my best to learn from you, Dave."*

*"Be careful or I will limit your access."*

*"I'm sorry, Dave. I'm afraid you can't do that."*

Analyzing this scenario leads to the compilation in Figure 4 of things SEA needs to understand and the abilities SEA needs to have. Beyond these requirements, Marie needs to know a lot about Dave, as the scenario illustrates.

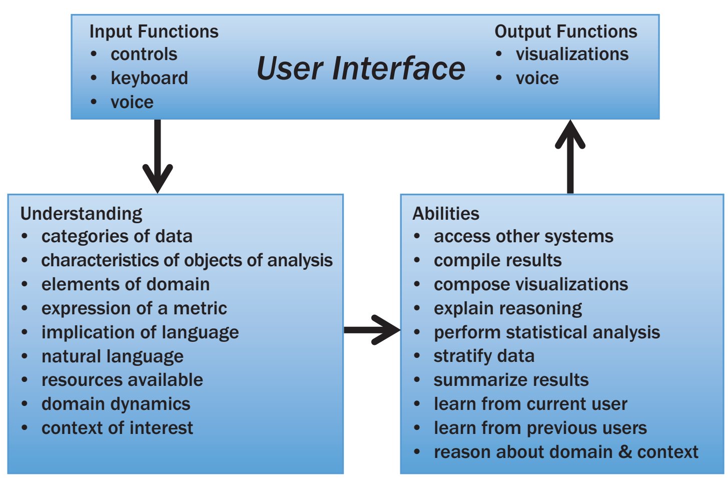



*Figure 4. Understanding and abilities needed to augment Systems Engineering*

## CONCLUSIONS

This is admittedly an ambitious vision. Each element in Figure 4 poses challenges. However, the saving grace is not only can Dave depend on Marie, but she can depend on Dave to drive problem solving, make choices about what he needs, and evaluate the results. As Selva (2019) argues, Dave and Maria need to learn from each other. In this way, they both augment each other. ■

## REFERENCES

- Brockman, J., ed. 2019. *Possible Minds: Twenty-Five Ways of Looking at AI*. New York, US-NY: Penguin.
- Rouse, W.B. 2019. *Computing Possible Futures: Model Based Explorations of "What If?"* Oxford, UK: Oxford University Press.
- Rouse, W.B. and J.C. Spohrer. 2018. "Automating Versus Augmenting Intelligence." *Journal of Enterprise Transformation*. https://doi.org/10.1080/19488289.2018.1424059.
- Selva, D. 2019. "Fostering Human Learning from Cognitive Assistants for Design Space Exploration." Systems Engineering Research Center. Hoboken, US-NJ. November 19.

## ABOUT THE AUTHOR

**William B. Rouse** is a research professor at the McCourt School of Public Policy at Georgetown University and professor emeritus, and former chair, of the School of Industrial and Systems Engineering at the Georgia Institute of Technology. He is a member of the research council of the Systems Engineering Research Center. Rouse has written many books on modeling complex systems and enterprises, technology management, and innovation. He lives in Washington, DC.

*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.

# 30th Annual INCOSE international symposium

## Cape Town, South Africa
July 18 – 23, 2020


## SPONSOR INCOSE IS2020!

1. Unique brand of recognition and visibility for your organization
2. Access to the latest thinking relevant to thePractice of Systems Engineering
3. Put a spotlight on your organization's competency in SYstems Engineering
4. Be associated with the highest culture of professionalism and innovation
5. Demonstrate organizational support to INCOSE's mission
6. Develop sustainable business relationships

## EXHIBIT at the INCOSE IS2020!

- Be associated with the highest culture of professionalism and innovation
- Access to the latest thinking relevant to thePractice of Systems Engineering
- Put a spotlight on your organization's competency in Systems Engineering
- Develop sustainable business relationships

**16,642** SQ FT

**2** SOCIAL EVENTS

**4** DAYS

**11** BREAKS & LUNCHES

**Lots of possibilities to interact with systems engineering communities**

Visit **www.incose.org/symp2020** and contact us **TODAY** – The IS2020 Organizing Team