INSIGHT

This Issue's Feature:

ARCHIMEDES
INITIATIVE

DECEMBER 2022
VOLUME 25 / ISSUE 4

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

INCOSE®

CSER 2023

conference on systems engineering research

20 YEARS

Systems Engineering Toward a Smart and Sustainable World

March 16-17, 2023

KEYNOTE SPEAKERS

Dr. Victoria Coleman

Chief Scientist, United States Air Force

Dr. Rick Kazman

Danny and Elsa Lui Distinguished Professor of Information Technology Management, University of Hawaii

Ms. Emily Kagan Trenchard

Vice President, Digital and Innovation Strategy, Northwell Health

Dr. Merve Unuvar

Director, AI Platforms and Automation Thomas J. Watson Research Center, IBM Research

Dr. Kathryn W. Jablokow

Program Director, National Science Foundation & Professor of Engineering Design and Mechanical Engineering, Penn State

The Conference on Systems Engineering Research celebrates its 20th year where the first conference was held: Hoboken, NJ. This premier conference focused on research across the systems engineering community brings together hundreds of faculty, researchers, and students to share results and ideas.

CSER.INFO/CSER2023

Visit the CSER website to learn more about the conference.

HOSTED BY

Stevens Institute of Technology

1 Castle Point Terrace, Hoboken, NJ 07030 USA

STEVENS

USC

SYSTEMS ENGINEERING RESEARCH CENTER

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

DECEMBER 2022 VOLUME 25 / ISSUE 4

Inside this issue

About This Publication

INFORMATION ABOUT INCOSE

INCOSE's membership extends to over 19,000 individual members and more than 200 corporations, government entities, and academic institutions. Its mission is to share, promote, and advance the best of systems engineering from across the globe for the benefit of humanity and the planet. INCOSE charters chapters worldwide, includes a corporate advisory board, and is led by elected officers and directors.

For more information, click here:
The International Council on Systems Engineering
(www.incose.org)
INSIGHT is the magazine of the International Council on Systems Engineering. It is published four times per year and

OVERVIEW

features informative articles dedicated to advancing the state of practice in systems engineering and to close the gap with the state of the art. ***INSIGHT*** delivers practical information on current hot topics, implementations, and best practices, written in applications-driven style. There is an emphasis on practical applications, tutorials, guides, and case studies that result in successful outcomes. Explicitly identified opinion pieces, book reviews, and technology roadmapping complement articles to stimulate advancing the state of practice. ***INSIGHT*** is dedicated to advancing the INCOSE objectives of impactful products and accelerating the transformation of systems engineering to a model-based discipline.
Topics to be covered include resilient systems, model-based systems engineering, commercial-driven transformational systems engineering, 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 and Conditions and download your license
- For any technical queries please contact customercare@copyright.com
- For further information and to view a Rightslink® demo please visit www.wiley.com and select Rights and Permissions.

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

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

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

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

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

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

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

ARTICLE SUBMISSION

insight@incose.net

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

- March 2023 issue – 2 January 2023
- June 2023 issue – 1 April 20023
- September 2023 issue – 1 July 2023
- January 2023 issue – 1 October 2023

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 20,000 individual members and uncounted employees and students of more than 100 CAB organizations worldwide. Readership includes engineers, manufacturers/purchasers, scientists, research and development professionals, presidents and chief executive officers, students, and other professionals in systems engineering.

Issuance	Circulation
2022, Vol 25, 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 and Australia corporatesalesaustralia@wiley.com
- Europe, Middle East and Africa (EMEA) corporatesaleseurope@wiley.com
- Japan corporatesalesjapan@wiley.com
- Korea corporatesaleskorea@wiley.com

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

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

Or please contact:
Marcom@incose.net

CONTACT

Questions or comments concerning:

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

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

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

ADVERTISER INDEX December Volume 25-4

CORPORATE ADVISORY BOARD — MEMBER COMPANIES

Aerospace Corporation, The
Airbus
AM General LLC
Analog Devices, Inc.
ARAS Corp
Arcfield
Australian National University
AVIAGE SYSTEMS
Aviation Industry Corporation of China
BAE Systems
Ball Aerospace
Bechtel
Becton Dickinson
Belcan Engineering Group LLC
Boeing Company, The
Bombardier Transportation
Booz Allen Hamilton Inc.
C.S. Draper Laboratory, Inc.
California State University Dominguez Hills
Carnegie Mellon University Software Engineering Institute
Change Vision, Inc.
Colorado State University Systems Engineering Programs
Cornell University
Cranfield University
Cubic
Cummins Inc.
Cybernet MBSE Co, Ltd
Dassault Systèmes
Defense Acquisition University
Deloitte Consulting, LLC
Denso Create Inc
Drexel University
Eindhoven University of Technology
EMBRAER
Federal Aviation Administration (U.S.)
Ford Motor Company
Fundacao Ezute
General Dynamics
General Electric Aviation
General Motors
George Mason University
Georgia Institute of Technology
IBM
Idaho National Laboratory
ISAE - Supaero
ISDEFE
ITID, Ltd
Jacobs
Jama Software
Jet Propulsion Laboratory
John Deere
Johns Hopkins University
KBR
KEIO University
L3Harris Technologies
Lawrence Livermore National Laboratory
Leidos
Lockheed Martin Corporation
Los Alamos National Laboratory
Loyola Marymount University
Mahindra University
ManTech International Corporation
Maplesoft
Marquette University
Massachusetts Institute of Technology
MBDA (UK) Ltd
MetaTech Consulting Inc.
Missouri University of Science & Technology
MITRE Corporation, The
Mitsubishi Heavy Industries, Ltd
National Aeronautics and Space Administration (NASA)
National Reconnaissance Office (NRO)
National Security Agency Enterprise Systems
Naval Postgraduate School
Nissan Motor Co, Ltd
Northrop Grumman Corporation
Pacific Northwest National Laboratory
Pennsylvania State University
Peraton
Petronas Nasional Berhad
Prime Solutions Group, Inc
Project Performance International (PPI)
Purdue University
QRA Corp
Raytheon Corporation
Rolls-Royce
Saab AB
SAIC
Sandia National Laboratories
Saudi Railway Company
Siemens
Sierra Nevada Corporation
Singapore Institute of Technology
SPEC Innovations
Stellar Solutions
Stevens Institute of Technology
Strategic Technical Services LLC
Swedish Defence Materiel Administration (FMV)
Systems Planning and Analysis
Tata Consultancy Services
Thales
The REUSE Company
The University of Arizona
Torch Technologies
TOSHIBA Corporation
Trane Technologies
Tsinghua University
TUS Solution LLC
UC San Diego
UK MoD
University of Alabama in Huntsville
University of Arkansas
University of Connecticut
University of Maryland
University of Maryland, Baltimore County
University of Michigan, Ann Arbor
University of New South Wales, The, Canberra
University of Southern California
University of Texas at El Paso (UTEP)
US Department of Defense
Veoneer
VG2PLAY
Virginia Tech
Vitech
Volvo Cars Corporation
Volvo Construction Equipment
Wabtec Corporation
Woodward Inc
Worcester Polytechnic Institute- WPI
Zuken Inc

DECEMBER 2022
VOLUME 25/ ISSUE 4

FROM THE EDITOR-IN-CHIEF

William Miller, insight@incose.net

We are pleased to announce the December 2022 *INSIGHT* issue published cooperatively with John Wiley & Sons as the systems engineering practitioners' magazine. The *INSIGHT* mission is to provide informative articles on advancing the practice of systems engineering and to close the gap between practice and the state of the art as advanced by *Systems Engineering*, the Journal of INCOSE also published by Wiley.

The issue theme is the *Archimedes Initiative*, a global systems engineering research network, to conduct applied research to evolve systems engineering and architecting principles, practices, methods, and methodologies for practicing engineers and scientists to address the complexity, dynamic behavior, evolution, and the underlying uncertainty in modern systems and system of systems.

Archimedes is within the scope of the global systems community addressing the challenges we face as expressed in the Systems Engineering Vision 2035 published by INCOSE in early 2022, freely accessible at https://www.incose.org/about-systems-engineering/se-vision-2035. The vision lays out the global context for systems engineering including human and societal needs, global megatrends, technology trends, stakeholder expectations, and enterprise environment. The vision goes on to describe the current and future states of systems engineering and a path to realize the vision with challenges, recommendations, and a roadmap of goals for 2025, 2030, and 2035 across five categories of applications, practices, tools and environment, research, and competencies. The future of systems engineering (FuSE) is charged with mobilizing global resources to realize the vision (www.incose.org/FuSE).

The *Archimedes Initiative* four founding research centers are the German Space Center (DLR) Institute of Systems Engineering for Future Mobility (https://www.dlr.de/content/en/institutes/institute-of-systems-engineering-for-future-mobility.html), the Netherlands Organization for Applied Scientific Research (TNO) Embedded Systems Innovation (ESI) center (https://esi.nl/), the Systems Engineering Research Center (SERC) of 22 universities operated by Stevens Institute of Technology for the US Department of Defense (https://sercuarc.org/), and the Center for Trustworthy Edge Computing Systems and Applications (TECoSA) at the Royal Institute of Technology (KTH) in Sweden (https://www.tecosa.center.kth.se/).

We thank theme editors Dinesh Verma, SERC executive director, and Wouter Leibbrandt, TNO-ESI science and operations director, and the contributing authors from the four research centers. Wouter and Dinesh abstract the DLR, ESI-TNO, SERC, and TECoSA research roadmaps in the four articles following their overview. The four research roadmap articles are followed by articles loosely organized around four themes: digital engineering and model-based systems engineering (MBSE), artificial intelligence and machine learning (AI/ML), systems engineering and agile development, and system security and resilience.

We hope you find *INSIGHT*, the practitioners' magazine for systems engineers, informative and relevant. Feedback from readers is critical to *INSIGHT*'s quality. We encourage letters to the editor at insight@incose.net. Please include "letter to the editor" in the subject line. *INSIGHT* also continues to solicit 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. For information about sponsoring *INSIGHT*, please contact the INCOSE marketing and communications director at marcom@incose.net. ■

DECEMBER 2022
VOLUME 25/ ISSUE 4

Editorial of *INSIGHT* Special Feature

Nurturing a Global Systems Engineering Research Network – The Archimedes Initiative

Wouter Leibbrandt, wouter.leibbrandt@tno.nl; and **Dinesh Verma,** dverma@stevens.edu

Systems engineering is widely practiced and taught across corporations, institutes, and universities that deal with the successful conception, realization, use, and retirement of complex engineered systems. Many of the principles, practices, and methods in use today, date back to the 1940s and 1950s, have been around for decades, and were developed to deal with largely mechanical or electro-mechanical systems. These methods continue to be applied in a productive manner today. However, given the influence of technological developments over the past decade or more (such as ubiquitous software, distributed and networked systems, and workflows, increasingly complex control regimes, cloud/edge computing), the complexity of modern systems are exaggerating the limits of classical systems engineering methods and practices. Modern societies depend on such complex systems and system of systems for commerce, healthcare, urban living, and transportation – with increasing dependence on developments in computational technologies, AI/ML, and human-machine teaming. Therein lies the focus of the research centers featured in this special issue of *INSIGHT* – to conduct applied research to evolve the systems engineering and architecting toolkit to help the practicing engineers and scientists address the complexity, dynamic behavior and evolution, and the underlying uncertainty in modern systems and system-of-systems.

While many, if not most, engineering organizations practice systems engineering, only a limited number of institutes conduct research into the modernization of systems engineering principles, practices, methods, and methodologies. Four leading centers that conduct such research have recently joined forces by founding the Archimedes Initiative and present themselves in this special issue of *INSIGHT*. These four centers are the German Space Center (DLR) Institute of Systems Engineering for Future Mobility, the Netherlands Organization for Applied Scientific Research (TNO) Embedded Systems Innovation (ESI) center, the Systems Engineering Research Center (SERC) of 22 universities operated by Stevens Institute of Technology for the US Department of Defense, and the Center for Trustworthy Edge Computing Systems and Applications (TECoSA) at the Royal Institute of Technology (KTH) in Sweden.

Given the practice-oriented nature of systems engineering, research in the development of modern systems engineering methods and tools cannot happen in isolation within an academic setting, but rather requires deep collaboration, engagement, and validation with the practitioner community, complete with a view towards the application context. The four research centers, therefore, all operate in the context of a diverse ecosystem with nodes in industry, government agencies, and academia. Furthermore, the four research centers are complementary in that they each target a different application domain. DLR works in the field of automotive and maritime mobility, ESI targets the high-tech equipment industry, SERC the defense industry, and TECoSA works with the automotive, truck, and aircraft industry.

The aim of the Archimedes Initiative is to accelerate the necessary collaboration and innovation in systems engineering research by learning from each other's best practices. Each application domain is facing similar challenges, but often in different orders of priority. It is therefore not a surprise that the four research roadmaps are complimentary, and there is ample opportunity for one center to generalize and build upon innovations developed at one of the other peer centers.

In this special issue of *INSIGHT*, we are sharing the research priorities and roadmaps of the four Archimedes Initiative members. These roadmaps are being shared in a synergistic manner for the first time – in the spirit of engaging the broader systems engineering research and practice community for their review and insights. It is certainly interesting to note the diversity in road mapping approaches, reflecting specific characteristics of the application domain of each of the research centers.

The starting point for the DLR roadmap is a societal problem statement, namely the challenge to limit climate change. This calls for various technology solutions,

out of which, from a systems engineering perspective, DLR is providing priority to autonomous driving, which in turn calls particularly for trustworthiness in cyber-physical systems.

ESI starts from a set of aspirations, challenges, and needs reflective of the priorities of the high-tech industry in the Netherlands, facing system complexity resulting from digitalization. This leads to the following priorities: dependability, architecting for diversity, reliable updates, verification and validation (V&V) of product families, architecting systems-of-systems, and democratization.

The SERC roadmap builds on the collective understanding of the participating groups and researchers of the future needs and ambitions of the US Department of Defense (DoD), as built up through long standing collaboration and interactions with them. The top priorities are summarized in the key words and phrases: digital engineering, mission engineering, velocity, security, and AI and autonomy.

TECoSA takes a different approach to its roadmap and builds it around a technology development, namely edge computing, and then elaborates on the impact and resulting needs in three use cases: augmented reality (AR)/virtual reality (VR)/cognitive assistance, cyber-physical systems, and distributed machine learning.

The diversity of these roadmaps provides a richness and, in our opinion, taken together these roadmaps provide a broad and comprehensive view of the innovation direction in systems engineering the coming decade. Several common themes and concerns run through these roadmaps. As an example, there is a clear and enduring need for novel principles and methods to ensure trustworthiness of future systems, across all application domains. Further, all roadmaps mention the need for automation or automated assistance with complex engineering tasks, both driven by increasing expertise levels required and by the increasing shortage of engineers in general. Finally, the advent of powerful AI/ML techniques holds both a promise as well as a concern for all application domains, and novel methodologies are called for to deal with that.

After the first four papers that present the research priorities and roadmaps from the four research centers, this special edition of *INSIGHT* includes several topical papers from the research centers, loosely organized around four themes: digital engineering and MBSE, AI/ML, systems engineering and agile development, and system security and resilience.

Bringing together the roadmaps in this issue of *INSIGHT* represents a first step of the Archimedes initiative towards creating a collaborative network of (multi-university and multi-sponsor) research centers focused on systems engineering research. We hope that this provides us a global platform for deeper and more meaningful research collaboration among the centers. This will hopefully include providing each other access to data and information from sponsors and organizations from other geographies to calibrate local findings, bringing partners from the various ecosystems together, for example around focused and topical research workshops, joint studies into topics of common interest, along with simply exchanging views, experiences, and best practices.

Finally, through this global collaboration we also aspire to raise the profile of systems engineering as a key enabling discipline within a broader community of decision-makers and policy makers. ■

ABOUT THE THEME EDITORS

Wouter Leibbrandt is the science and operations director of TNO-ESI. Before joining ESI in 2016, Wouter was with NXP for ten years, managing the Advanced Applications Lab. Until 2006 he was with Philips Research labs for 14 years, managing various projects and departments in The Netherlands and abroad. Wouter holds a PhD in physics from Utrecht University.

Dinesh Verma is a 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 the 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 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 a PhD in industrial and systems engineering.

DLR Institute of Systems Engineering for Future Mobility – Technical Trustworthiness as a Basis for Highly Automated and Autonomous Systems

André Bolles, andre.bolles@dlr.de; **Willem Hagemann,** willem.hagemann@dlr.de; **Axel Hahn,** axel.hahn@dlr.de; and **Martin Fränzle,** martin.fraenzle@uol.de

■ **ABSTRACT**
The newly established Institute of Systems Engineering for Future Mobility within the German Aerospace Center opened its doors at the beginning of 2022. Emerging from the former OFFIS Division Transportation after a two-year transition phase, the new institute can draw on more than thirty years of experience in the research field of safety-critical systems. With the transition to the DLR, the institute's new research roadmap focuses on technical trustworthiness for highly automated and autonomous systems. Within this field, the institute will develop new concepts, methods, and tools to support the integration and assurance of technical trustworthiness for automated and autonomous systems during their whole lifecycle – from the early development through verification, validation, and operation to updates of the systems in the field.
■ **KEYWORDS:** autonomous systems; technical trustworthiness; verification; validation; artificial intelligence; safety; dependability; autonomics

THE ADVENT OF AUTONOMOUS SYSTEMS

The transition to a sustainable transport system constitutes an important pillar among the measures addressing climate change. Besides the greenhouse gas emissions of the transport sector, we can see additional challenges, like congestion, lots of space reserved for transportation (streets and parking spaces), noise, and safety issues. While the electrification of vehicle drive trains directly addresses greenhouse gas emission reduction, automation of vehicles can contribute to overcome also many of the other challenges.

During the last decade, we have already seen an increasing number of applications using more or less intelligent and self-acting systems. Smartphones, software agents, and artificial intelligence, sometimes in the consumer market having names like Alexa and Siri, assist us in decision-making. They provide us with well-defined advertisements, help recruiters to identify suitable job candidates, and even help qualify loan applications for bank employees. With the progressive use of artificial intelligence and automation technology in safety-critical cyber-physical systems such as autonomous vehicles, new classes of systems are emerging (cf. SafeTRANS 2021). (This also holds for other safety critical areas like health, energy, industry, farming, etc. Due to the fact that the new DLR institute is focusing on transportation, this paper is focused on autonomous driving.) These systems will be deployed into highly dynamic environments, first to understand their impact, then to implement their decisions autonomously using their actuators in the physical world. The advent of autonomously acting cyber-physical systems capable of cooperation in frequently changing contexts and no longer subject to direct human control places novel and high demands on developing methodologies that ensure their trustworthiness.

Since today computing power allows sophisticated artificial intelligence models to recognize complex patterns in the real world and derive suitable actions from such percepts, from a functional perspective, the goal of autonomous driving appears imminently achievable. However, this technology then directly links to the real world. Decisions made by vehicles may directly harm humans and may cause catastrophic failures. Thus, it is imperative to ensure these systems' safety and additional properties, as described in the following sections.

The SafeTRANS roadmap on "safety,

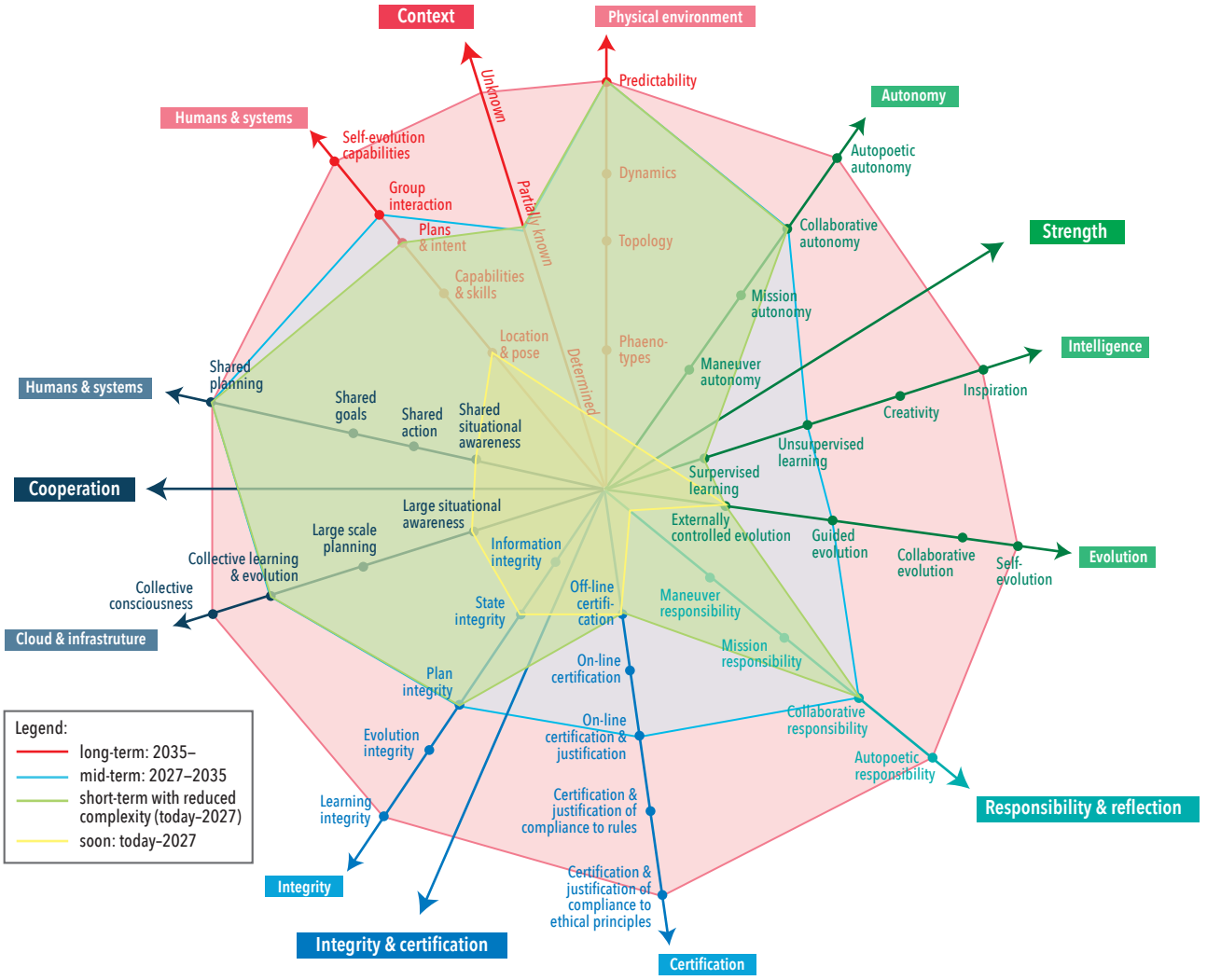


Figure 1. Roadmap for future man-machine systems (SafeTRANS 2021); The authors have translated the legend within that diagram

security, and certifiability of future man-machine systems" (SafeTRANS 2021) dares a look into the future of autonomous systems. It sketches several dimensions of the complexity of these systems, as shown in the following figure taken from the roadmap, and forecasts how autonomous systems will cover these dimensions within the years to come.

Based on this SafeTRANS roadmap, the DLR Institute of Systems Engineering for Future Mobility defined its research roadmap in an internal unpublished concept paper (DLR 2020) explored in further detail below.

From the diagram above, we can see five main axes "cooperation," "context," "strength," "responsibility & reflection," and "integrity & certification." Besides these main axes, additional secondary axes refining the related concepts can be found. As a newly founded DLR institute, we identified for our research roadmap that we can distinguish two types of complexity dimensions from this figure: functional dimensions and extra-functional dimensions. The axes "cooperation," "context," and "strength" reflect functional dimensions, sketching functional capabilities of autonomous systems. On the other hand, "responsibility & reflection" and "integrity & certification" reflect extra-functional dimensions, sketching necessary mechanisms and properties to be fulfilled by autonomous systems to consider them trustworthy. We are conscious that different viewpoints are possible here. However, this distinction helps us to define our roadmap as sketched in the following paragraphs (A slightly modified view on the axes was developed in the internal paper (DLR 2020) but will be omitted here for space reasons.)

SafeTRANS considers the dimension "Cooperation" to describe the future cooperation capabilities of systems, systems-of-systems, a comprehensive group of systems of a specific domain, or even cross-domain groups of systems (SafeTRANS 2021, p 61). Cooperation in this context means cooperation between the systems but also between systems and humans (cf. SafeTRANS 2021, p. 61).

The dimension "context" describes the complexity of the environment, which the systems considered need to be able to handle (see SafeTRANS 2021, p. 43). From the diagram and the description in the roadmap, one can see that the degree of uncertainty increases along the axes shown and that the degree of controllability decreases correspondingly (cf. SafeTRANS 2021, p. 43-ff). Future human-machine systems, assuming a development along the defined axes, will be able to act in more complex environments with much more uncertainty and less controllability of environmental parameters. By this, they will be able to handle many more situations and realize increasingly complex tasks.

The dimension "strength" represents the capability of systems to successfully accomplish application-specific objectives in a self-determined manner (see SafeTRANS p. 27) (As this dimension refers to certain inherent capabilities of a system, we also call it the "System Capabilities" dimension in our institute.). This requires systems to understand and analyze even unfamiliar contexts

(subdimension "intelligence"), adapt to those contexts in order to increase the scope of possible actions (subdimension "evolution"), and finally achieve the desired goal autonomously through a complex sequence of individual or cooperative actions (subdimension "autonomy")(cf. SafeTRANS 2021, p. 27-ff.).

In our understanding, these dimensions sketch a roadmap on how, in which context, and with which complexity, uncertainty, and uncontrollability human-machine systems will be able to achieve goals in the future. However, assuming that all these capabilities will become true, the following and similar questions still need to be raised:

Will these systems then automatically be trustworthy enough, for example, to put our children into an autonomous vehicle and have them be driven by it to their grandparents in the neighboring city without the possibility of intervening? How can we prove such trustworthiness?

The central term to be discussed here is **trust** (The definition of this term and its implications for the research questions of the new DLR institute is ongoing work.) We are convinced that the question above is covered by the other two dimensions of the roadmap, "Integrity & certification" and "Responsibility."

(SafeTRANS 2021) describes "integrity & certification" as a dimension covering mechanisms to ensure consistency and trustworthiness in decision-making and to enable recovery of system integrity after integrity violations (cf. SafeTRANS 2021, p. 85). This is the primary dimension that the DLR institute has focused its research on. All methods, concepts, processes, and tools, including formal verification, model-based systems engineering, contract-based design, virtual certification, monitoring, and diagnosis, can be partially captured under this dimension. However, whatever is needed in the future to ensure system integrity strongly depends on what system capabilities will look like along the dimensions of "cooperation", "context" and "ctrength" (cf. SafeTRANS 2021, p. 85-ff.). Therefore, the DLR Institute of Systems Engineering for Future Mobility is embedded into a network of other institutes (internal and external to DLR) discussing future capabilities of autonomous systems to ensure trustworthiness from the very beginning.

In addition to the dimensions above, the final dimension, "responsibility & reflection," widens the perspective on the extra-functional properties of autonomous systems. The more autonomy and capabilities future human-machine systems acquire, the stronger the need to answer additional questions only marginally explored today. Some of these questions are already sketched in (SafeTRANS 2021, p. 77, translation made with deepl.com and partially edited by the authors):

- "What can machines be responsible for?"
- "What may or can machines decide?"
- "Will machines in a future 'human machine society' be partners of humans or will they even decide over them?"
- "How much autonomy do we want to grant machines?"

These are predominantly ethical and societal questions rather than technical ones. However, answers to these questions strongly depend on the degree of trust we place in machines. Thus, to increase autonomy in an accepted way, we need to increase their trustworthiness. Aside from integrity, there are additional questions that need to be answered to increase trustworthiness with respect to machine autonomy. The concept of responsibility described by SafeTRANS sketches additional challenges to address, which we summarize into the following exemplary questions (SafeTRANS 2021, p. 77-ff. (Additional and similar questions have been defined in the internal institute's concept paper (DLR 2021))

- How can ethical and societal values be implemented into autonomous systems?
- How can compliance with ethical and societal values be ensured during operation and how can this be made transparent?
- How can we enable machines to evaluate consequences of their action in advance?
- How can we enable machines to build trust in other machines and humans?

This list is incomplete, but it demonstrates the future need for broad research initiatives. It seems clear that a deep understanding of ethical and societal values that influence today's social coexistence between humans will have to be implemented into machines in the future to generate technical trustworthiness. Unlike between humans however, trustworthiness of machines will probably not be generated by long-lasting cooperation between humans and machines or by simple test mechanisms like a short driver's license exam. In the case of machines, we expect that each brick for generating trustworthiness needs to be verified and validated. Thus, we assume that in the future, besides classical verification and validation approaches, we also need more and more advanced approaches covering not only technical characteristics but also technical implementations of non-technical concepts.

In (Liggesmeyer 2017), Peter Liggesmeyer, as the director of the "Fraunhofer Institute for Experimental Software Engineering IESE" (https://www.iese.fraunhofer.de/, last visit: July 14, 2022), underlined that technical as well as ethical and legal questions are demanding answers with respect to autonomous systems. Though the term "autonomik" actually is considerably older (for example, compare the research programme on "Autonomics – Autonomous and simulation-based systems for medium-sized companies" that ran from 2008 – 2014 (BMWK 2022)). Also others propose interdisciplinary research in this field (Koopmann and Wagner 2017), it was Liggesmeyer who in (Liggesmeyer 2017) publicly proposed a discipline of "autonomik." The authors agree with (BMWK 2022) to translate this term as "Autonomics" for building reliable and trustworthy autonomous systems in an overarching interdisciplinary way. We picked up this idea during the founding phase of the new DLR Institute of Systems Engineering for Future Mobility. We agree with Liggesmeyer's proposals and think that besides computer science aspects, also perspectives from other technical and non-technical disciplines need to be considered in an integrated way, such as for example mechanics/robotics, social sciences, natural sciences, philosophy, ethics, law, neurosciences, psychology, and biology. Our vision is that for the development and operation of autonomous systems, we will need entirely new systems engineering methods, development approaches, tools, and concepts. We strongly believe that the disciplines referenced above can learn from each other by transferring methods and tools from non-technical sciences to technical sciences and vice versa. This will generate completely new approaches for systems engineering and for the other disciplines. In line with this, the detailed consideration of the complexity facets for future cyber-physical systems development in (Törngren and Sellgren 2018) shows that there is an additional need for education and awareness raising on complexity as well as research into efficient overarching organizational structures and processes. As an illustration, one example could be to integrate social science models for generating trust between humans with formal methods from computer science to support the generation of trust between humans and autonomous systems. We expect that these kinds of synergies will be lifted by strongly integrating the disciplines along the whole life-cycle of an autonomous system, including the development and operational phases. Based on this scope, we place the focus of our new institute on the development of technical methods, tools, processes,

and concepts, that enable and ensure the generation of trust in autonomous systems. By doing this, we are expanding our area of expertise, which in the past mainly focused on safety or dependability, and include the non-technical aspects that will require extended technical support in the future within autonomous systems.

TECHNICAL TRUSTWORTHINESS AS AN ESSENTIAL BASIS FOR AUTONOMOUS SYSTEMS

The division Transportation of OFFIS, as the predecessor of the recently founded DLR institute, had a strong focus on methods and tools guaranteeing safety respectively dependability for human-cyber-physical systems (HCPS – An overview about this term can be found in (Zhiming and Wang 2020)). With respect to this, the term dependability has been understood as defined in Avizienis et al. 2004. With the founding of the new DLR Institute and the definition of its roadmap (DLR 2020), the focus on dependability was broadened as explained in the following.

Since we believe that artificial intelligence will play a significant role in the trustworthiness of autonomous systems, we had a look into research on ethical principles for artificial intelligence. Jobin and her colleagues identified that "there is an emerging convergence around the following principles: transparency, justice and fairness, non-maleficence, responsibility and privacy (Jobin et al. 2019 p. 391)." Although Jobin and her colleagues identify the diversity in interpretations of these terms, within the several works analyzed in their study (Jobin et al. 2019, pp. 391-ff.), we agree that these five principles will become essential considerations for autonomous systems. Within the unpublished internal DLR roadmap paper (DLR 2020), the combination of these five principles, together with dependability, is defined as the comprehensive concept of technical trustworthiness (Assuming that privacy and confidentiality are meaning the same, dependability and privacy are also covering all attributes security is covering in (Avizienis et al. 2004) and by this our definition of technical trustworthiness also covers security.) We believe that this definition and the combination of technical and non-technical issues are compatible with the requirements on trustworthy AI given in the Ethics Guidelines for Trustworthy AI by the Independent High-Level Expert Group on Artificial Intelligence set up by the European Commission (AI HLEG 2019). Notwithstanding this, we will probably refine the definition of technical trustworthiness and related research questions in the future.

In our opinion, it will be essential that at least the aforementioned ethical principles will be considered from the very beginning of the development of autonomous systems to avoid generating mistrust against such systems. We need to start doing so now, and therefore we need to involve technical and non-technical sciences as detailed above.

The first step in this direction will be further to refine the definition and implications of technical trustworthiness (We are especially open for including even more principles to be considered here.). These implications, in addition to their technical nature, may lead to research questions within other disciplines. The new DLR institute, in this context, aims to establish collaborations between researchers from several technical and non-technical disciplines and to intensify existing ones in order to address these questions from the very beginning. This is done, for example, through the Research Center on Human-Cyber-Physical Systems at the University of Oldenburg (https://uol.de/fzhcps, last visit: June 17, 2022).

METHODOLOGICAL CONTRIBUTION FROM THE NEW DLR INSTITUTE

The DLR Institute of Systems Engineering for Future Mobility is aligning its research and development activities along the DevOps approach (Overviews can be found for example in (Ebert et al. 2016 and (Mayank and Singh 2021)). It focuses on the development of methods, tools, processes, and concepts for the assurance of technical trustworthiness for autonomous systems through the whole life-cycle – meaning from the beginning design, development and build phases, through the verification and validation phases and incorporating the operational phases covering deployment, operation, monitoring and updates of the systems in the field. (Decommissioning of autonomous systems will also need to be addressed, although it is not always covered when talking about DevOps.). This is, on the one hand, reflected in the organizational structure as follows:

- "The department Systems Theory and Design (THD) considers all phases of design, development, verification and validation of highly automated and autonomous traffic systems. Methods and tools are developed that enable mechanisms for technical trustworthiness and responsibility, ensure integrity and demonstrate appropriate properties already at the design stage of a system (DLR 2022)."
- "The department System Evolution and Monitoring (EVO) considers all phases during the operation of highly automated and autonomous traffic systems. The focus of this department is on the development of methods and tools that enable a trustworthy evolution of systems and that monitor and ensure compliance with integrity, responsibility and trust measures at system runtime (DLR 2022)."
- "The department Application and Evaluation (ANE) identifies application-specific requirements for integrating and ensuring technical trustworthiness, responsibility, and integrity. At the same time, this department provides platforms to evaluate the methods and tools developed in the THD and EVO departments and integrates them into industry-relevant processes. The ANE department contributes these results to standardization and regulatory activities (DLR 2022)."

On the other hand, this is also reflected in the thematic organization. Within the DLR Institute of Systems Engineering for Future Mobility, thematic clusters — so-called assets — summarize and integrate all activities that are related to specific topics. These thematic assets either have a methodological approach, application-, or technology-driven focus. The methodology-driven assets (Application driven and technology driven assets focus on maritime traffic simulation, testbeds and digital twins.) focus on the topics of:

- scenario-based verification and validation,
- continuous timing assurance,
- human modeling,
- automation risks, and
- online updates and upgrades.

All these assets, sketched below, integrate research done in several projects. This allows building on earlier project results, developing synergies between projects, and professionalizing research prototype development and demonstrations in industrially relevant use cases. Furthermore, complete toolchains can be set up in order to evaluate the research results within seamless processes and to identify gaps that demand further research to become closed. The following paragraphs give a short overview about the thematic orientation of the assets mentioned above.

"*The first asset* [scenario-based verification and validation] *is concerned with developing and prototyping methods and tools that can be used in scenario-based verification and validation approaches for automated transportation systems. Our main focus is formally*

specifying relevant abstract scenarios that are readable by humans while also being machine readable. This allows us to automatize the verification and validation which increases confidence in e.g., safety of the systems due to a dramatically increased number of executed tests while reducing the manual effort from humans (Birte Neurohr, project lead of the asset on scenario-based verification and validation, DLR)."

"*Ensuring timing properties is a crucial aspect in safety-critical systems at both design time and run-time. For example, safe operation of a highly automated vehicle includes the ability to react on appearing obstacles in a specified maximal time span. Asset 2 'continuous timing assurance' provides methods and tools enabling specifying, verifying and monitoring of timing properties along the system lifecycle (from specification, to implementation and test in the development phase, over monitoring, to diagnosis and feedback to the developers in the operations phase). The asset also establishes expertise on the underlying DevOps processes in which these methods and tools are applied, as they (1) are integral parts of many safety standards that must be followed in industry, and (2) should match the requirements and state-of-the-art of industrial practice. The capability of a continuous timing assurance is of crucial importance for all manufacturers and suppliers of future highly automated learning systems, because they are especially challenged by regular software updates and the repetitive real-time proof* (Kim Grüttner, Head of Department System Evolution and Operation, DLR)."

"*Our asset 'human modelling' provides human models that can be used as so-called virtual test drivers or as virtual co-drivers. We research techniques and formalisms to model how humans interact with machines in complex traffic situations. These models are able to recognize and predict human behaviour. As virtual test drivers they are used to test design variants of human-machine interaction for safety critical systems. Such virtual tests can be done very early in the system development process before testing with real humans. As virtual co-drivers they are used to recognize the state of the driver and to predict her/his actions in order to initiate interventions in hazardous situations. We research not only driver models but also models of seafarers and aircraft pilots* (Andreas Lüdtke, Group Leader Human-Centered Engineering, DLR)."

"*The asset 'automation risks' deals with the question how to identify and analyze hazards and triggering scenario properties that arise from the introduction of automated and automatic systems. Therefore, it focuses on the development of methods and tools to find relevant factors influencing the criticality for system classes as well as identify and quantitatively assess newly occurring sources of harm within a specific system* (Lina Putze, project lead of the asset on automation risks, DLR)."

"*The fifth asset [online updates and upgrades] deals with software updates for individual modules of safety-critical systems. Tools are being developed to evaluate the correctness of a new software version in the overall system during development (virtual integration testing). For the safety-critical system, methods are developed to replace individual software modules separately with new versions without endangering the safety of the overall system. For this purpose, methods and tools are developed to secure the update process itself as well as to monitor the system properties after the update. This is of particular importance for suppliers to the automotive industry because it ensures that the increasingly complex automotive software can be continuously tested and further developed* (Domenik Helms, Group Leader Deployment and Updates, DLR)."

Covering the lifecycle of autonomous systems is important for the DLR Institute of Systems Engineering for Future Mobility since we believe that trust between humans and autonomous systems is something that will — similar to that between humans — evolve. Additionally, we think it will not be possible to design, develop and certify a system once without iterations between development and operation — at least due to changing environments.

THE FUTURE DEVELOPMENT OF TRUSTWORTHINESS

Finally, let us look at the future development of technical trustworthiness as foreseen in the institute's research roadmap (DLR 2020). The institute's roadmap is based on the estimated developments in the SafeTRANS roadmap (SafeTRANS 2021), depicted in Figure 1. It mainly addresses scientific goals along the complexity dimensions of "integrity & certification" and "responsibility & reflection". For the time to 2027, this covers mainly the yellow area and with respect to specific aspects like "maneuver responsibility" also the green area depicted Figure 1. The research and development for the time after 2027 will be analogous to (SafeTRANS 2021) covering the green, blue, and light red areas. ■

ACKNOWLEDGMENTS

Acknowledgments go to the whole team involved in defining the research roadmap for the new DLR institute presented in this paper. This team consisted of the management team – namely Kim Grüttner, Eckard Böde, Andreas Lüdtke, Eike Möhlmann, Domenik Helms, Bernd Westphal, Sebastian Feuerstack, Arne Lamm, Günter Ehmen, Michael Siegel, and authors Axel Hahn and André Bolles. Furthermore, we also thank the senior researchers involved of the division Transportation from OFFIS — namely Ingo Stierand, Jan-Patrick Osterloh, and Gregor Nitsche. Additional thanks go to Werner Damm for his strategic and scientific support during the founding phase of the new DLR institute. We are also thankful to the program management transportation of the DLR and the partner institutes for helping us to align the roadmap with existing DLR strategies. Furthermore, special thanks also go to SafeTRANS, namely Jürgen Niehaus, who supported us with his expertise on their roadmap.

REFERENCES

- Avizienis A., J.-C. Laprie, B. Randell, and C. Landwehr. 2004. "Basic Concepts and Taxonomy of Dependable and Secure Computing." In *IEEE Transactions on dependable and secure computing* 1 (1): 11 – 33. doi: 10.1109/TDSC.2004.2.
- Deutsches Zentrum für Luft- und Raumfahrt e. V. 2021 (DLR). "Institute of Systems Engineering for Future Mobility – Description of the new DLR research institute in Oldenburg." unpublished internal paper.
- Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR). 2022. "Institute of Systems Engineering for Future Mobility – Brief description of the three departments." https://www.dlr.de/se/en/desktopdefault.aspx/tabid-15540/, last visit: June 14, 2022.
- Ebert, C., G. Gallardo, J. Hernantes, and N. Serrano. 2016, "DevOps." In *IEEE Software* 33 (3): 94-100. doi: 10.1109/MS.2016.68.

- Federal Ministry for Economic Affairs and Climate Action (BMWK). 2022. "Autonomics - Pioneer for Industry 4.0." https://www.digitale-technologien.de/DT/Navigation/EN/ProgrammeProjekte/AbgeschlosseneProgrammeProjekte/Autonomik/autonomik.html, last visit: July 14th, 2022.
- Independent High-Level Expert Group on Artificial Intelligence set up by the European Commission. 2019. "Ethics Guidelines for Trustworthy AI." https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai, last visit: October 27th, 2022.
- Jobin, A., M. Ienca, and E. Vayena. 2019. "The global landscape of AI ethics guidelines." In *Nature Machine Intelligence* 1: 389 – 399. doi: 10.1038/s42256-019-0088-2.
- Koopmann, P., M. Wagner. 2017. "Autonomous Vehicle Safety: An Interdisciplinary Challenge." In *IEEE Intelligent Transportation Systems Magazine* 9 (1): 90 – 96. doi: 10.1109/MITS.2016.2583491.
- Liggesmeyer, P. 2017. "Autonome Systeme – Editorial." In *Informatik Spektrum* 40 (5): 399. doi: 10.1007/s00287-017-1046-1.
- Mayank G., R. Singh. 2021. "DevOps: A Historical Review and Future Works." In *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*: 366 – 371. doi: 10.1109/ICCCIS51004.2021.9397235
- SafeTRANS e. V. 2021. "Safety, Security, and Certifiability of Future Man-Machine Systems." https://www.safetrans-de.org/de/Uploads/AK_2018_RLE_CPS/SafeTRANS_RM_SSC_FMMS_Roadmap_V2.pdf?m=1611136486, last visit: May 27, 2022.
- Törngren, M., U. Sellgren. 2018. "Complexity challenges in development of cyber-physical systems." In Marten Lohstroh, Patricia Derler, Marjan Sirjani (ed.), *Principles of modeling: Essays dedicated to Edward A. Lee on the occasion of his 60th birthday*: 478 - 503. doi: 10.1007/978-3-319-95246-8_27.
- Zhiming, L., and J. Wang. 2020. "Human-Cyber-Physical Systems: Concepts, Challenges, and Research Opportunities." In *Frontiers of Information Technology & Electronic Engineering* 21 (11):1535–1553. doi: 10.1631/FITEE.2000537.

ABOUT THE AUTHORS

André Bolles is head of the department "Application and Evaluation" and temporary head of the department "System Theory and Design" within the DLR Institute of Systems Engineering for Future Mobility. His background is in computer science. From 2011 to 2021, he was group manager (2011–2018) and director (2018–2021) within the division "Transportation" of OFFIS, where focused on autonomous shipping and highly automated driving. He was involved in the founding of the new DLR Institute and within this process participated in the definition of the new research roadmap of the institute. Today his research focuses on the technical trustworthiness of autonomous systems.

Willem Hagemann is a researcher at the DLR Institute of Systems Engineering for Future Mobility. His background is in mathematics and computer science. He joined the new institute in early 2022, working in the group Evidence for Trustworthiness. Before that, he was a research associate at the University of Oldenburg. His research interests are explainability for autonomous systems and formal verification of cyber-physical systems.

Axel Hahn is the director of the DLR Institute of Systems Engineering for Future Mobility. Before that, he was a professor at the University of Oldenburg for System Analysis and Optimization and a board member of OFFIS. He coordinated the founding phase of the new DLR institute and was strongly involved in defining its research roadmap.

Martin Fränzle is a professor for Foundations and Applications of Systems of Cyber-Physical Systems at the University of Oldenburg. He was the scientific director within the division Transportation of OFFIS and is continuing this activity within the newly founded DLR institute. In this position, he was involved in developing the research roadmap for this institute.

TNO-ESI – Systems Engineering Methodologies for Managing Complexity in the High-Tech Equipment Industry: Our Roadmap

Wouter Leibbrandt, wouter.leibbrandt@tno.nl; **Jacco Wesselius,** jacco.wesselius@tno.nl; and **Frans Beenker,** frans.beenker@tno.nl

■ **ABSTRACT**
The high-tech equipment industry brings complex industrial products to the market with high speed, enhanced functionality, a better cost-performance ratio, and greater integration into customer workflows. Driven by digitalization, the complexity of these systems continues to grow steeply. To manage this complexity, continuous innovation in systems engineering methodologies is needed. TNO-ESI targets to 1) create impactful and industrially applicable systems engineering methodologies and 2) provide innovation support to the industry to get these applied in an industrial context. The ESI research program is defined through a roadmapping process that follows two tracks: a roadmap that maps industry needs and related research and development requirements and a roadmap that describes the developments in the expertise areas necessary for addressing these industry needs. In this paper, we describe the ESI mission, our way of working and activities, and explain the roadmapping process and the roadmaps.

INTRODUCTION TO ESI – WHO ARE WE

TNO-ESI (Embedded Systems Innovation, ESI for short) (www.esi.nl) is an open innovation research center with strong partnerships with industry-leading high-tech equipment companies and strong associations with fundamental research of academia (both nationally and internationally). As ESI, we are part of the Netherlands Organisation for Applied Scientific Research, referred to as TNO or the TNO-ESI. By developing new systems design and engineering methodologies, we address the ever-increasing complexity the high-tech equipment industry faces in the systems it creates and maintains throughout the entire lifecycle. We are about managing complexity. Our research program aims to advance the high-tech equipment industry by improving the lead times and effectiveness of their product innovation processes and their products' functionality, quality, and societal impact. We contribute through a robust research program, dedicated innovation support, a focused competence development program, and various knowledge- and experience-sharing activities. We create impact by turning the latest insights in systems engineering methodologies into practice in the harsh reality of the industry.

THE DUTCH HIGH-TECH EQUIPMENT INDUSTRY

The Dutch high-tech equipment industry is developing world-class systems for diverse business markets. Along with other application domains, they focus on systems and equipment for the semiconductor industry, healthcare imaging, professional production printing, electron microscopes, warehouse automation, and combat management systems. These companies have much in common despite their apparent differences in markets and application domains. They all target the high end of their respective markets, serving an international customer base. They all make highly complex systems in relatively low numbers – typically hundreds per year and sometimes even fewer. And all systems operate in the field for a long time – twenty years is no exception. Finally, these companies also share a business driver: to digitalize their products and solutions.

This industry also commonly recognizes the advantage of joint innovation, particu-

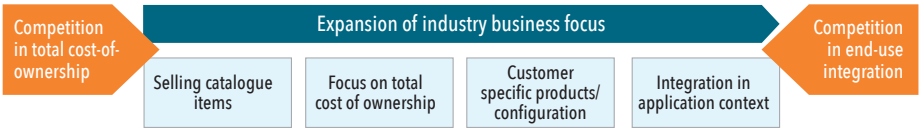


Figure 1. Expansion of industry business focus

larly in research and development, to manage the ever-increasing complexity. Being non-competitors, they are very much open to working together. ESI's open-innovation model enables them to learn from each other and stay on top of market developments.

Over the years, the business focus of the industry has gradually expanded. Their initial focus was only on stand-alone (often high-performance) products (catalog items). This has become less and less a viable business proposition. As a basis, products must provide the required functionality, performance, and a competitive total-cost-of-ownership (TCO) with strict product quality and reliability requirements. However, they must increasingly be adaptable to individualized customer needs, provide flawless integration and cooperation with other products and end-customer processes and applications, and preferably be sufficiently future-proof to accommodate continuously changing operating requirements during their lifecycle; see Figure 1. Digitalization is the driver and enabler, bringing new complexity and system dynamics challenges.

DIGITALIZATION CHANGES BUSINESS

Product innovations are pre-dominantly realized as a complex multi-disciplinary interplay of software and physical components (– cyber-physical systems). Value, cost, and complexity have shifted increasingly into the software. Leveraging digital technologies has become a key engineering competence for the high-tech equipment industry (without reducing the need to be competent in managing complex physical technologies). Digitalization has brought new opportunities, challenges, and market expectations to the industry, including a demand for regular updates and upgrades.

CREATING IMPACT AND OUR WAY OF WORKING

The mission of ESI is to impact the Dutch high-tech equipment industry by embedding cutting-edge systems engineering methodologies to cope with their products' ever-increasing complexity. This mission defines our activities and our way of working. Methodologies here are meant as consisting of formalisms, techniques, methods and tools as explained in (Heemels 2007).With newly developed knowledge of methods, we target individual products or applications, foster synergies, and share and exchange methodologies and knowledge in an open-innovation setting. For successful innovation and value take-up by the industry, systematic attention must be given to all required elements of the knowledge chain. This consists of the following:

- agenda setting and programming (translating industrial challenges into a research program),
- applied research (executing the research program in cooperation with industry and academia),
- consolidation (knowledge base for general use, professionalized tooling),
- dissemination (presenting, sharing, discussing, demonstrating results, enabling service providers to apply our results), and competence development.

We have found that for our applied research to be impactful, it is vital to work on real-world, business-critical industrial challenges. To access the often company confidential information, on-site presence is required. Therefore, we conduct our work at the premises of our industry partners, a way of working called *industry-as-a- lab*. This requires trust that we have developed and nurtured over the years with our industry partners, with whom we execute applied research projects which may span several years.

POSITIONING ESI'S RESEARCH

The positioning of our research is depicted in Figure 2. We have carefully chosen the meta-2 level. This means that ESI does not create products (meta-0); this is the work of our industrial partners by applying, among others, systems engineering and system architecting methodologies (meta-1). Our focus is on delivering innovations in the methodologies required for industrial systems engineering practices in their product realization. This creates focus and opportunities to fully exploit the collaboration with our industry partner, their suppliers, and our academic partners. Now and then, with our peers, like the centers presenting themselves in this volume, we reflect on how to organize and conduct such research (meta-3).

Another axis is to look at technology-ready levels (TRL). Our emphasis is on TRL 4-7. Higher TRLs are addressed by industry, service providers, and tooling companies, while we leave fundamental research at TRL

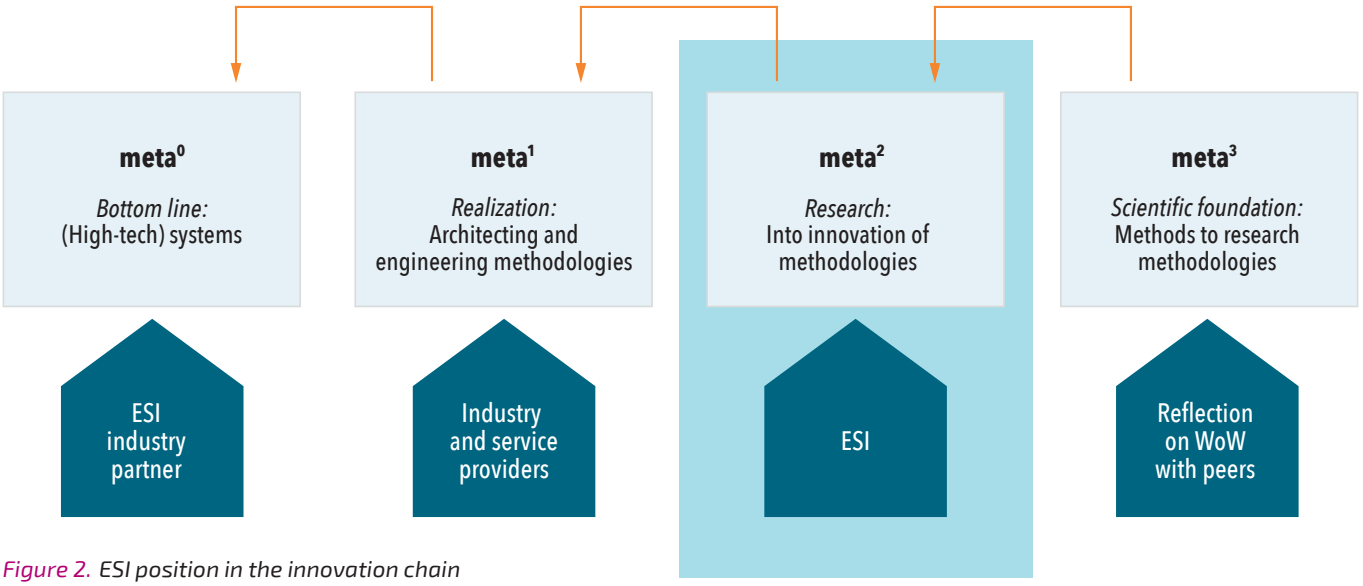


Figure 2. ESI position in the innovation chain

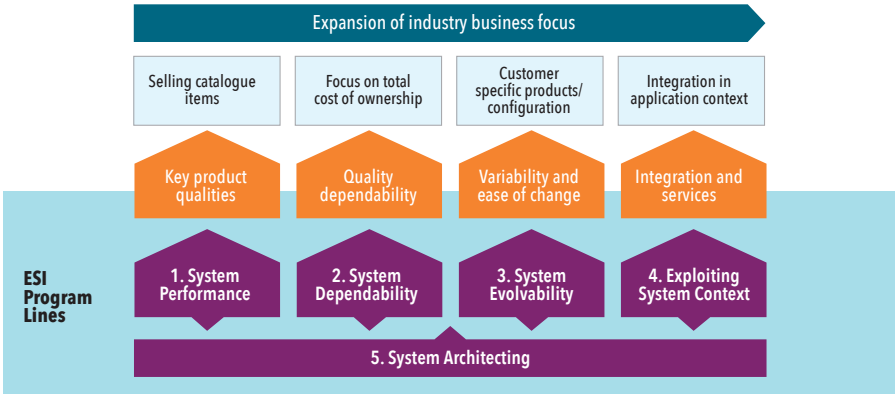


Figure 3. ESI program lines

1-3 to our academic partners.

PROGRAM LINES

During the 20-year existence of ESI, the focus of our research evolved, following the needs of the industry. This resulted in 5 research program lines, as depicted in Figure 3 (see https://esi.nl/research/program-lines for project results). Four program lines can be coupled to the life-cycle phase of products and businesses, and the fifth support these with methodologies for systems architecting:

- Program Line 1 – **System Performance**
  This program line focuses on the performance of systems in a broad sense: the system provides functions; how well does the system do this? The program line targets methodologies that enable engineers to get control of those system qualities that give value to their systems in the market. Since trade-offs are common in these cases (enhanced throughput might result in reduced output quality), methods are needed to relate the many system qualities and to provide methods to perform trade-off analysis to optimize design choices.

- Program Line 2 – **System Dependability**
  As systems are increasingly business/mission-critical, it is not enough to deliver the correct functionality and performance; it is critical that they also dependably deliver this: unforeseen system failure or performance degradation has to be avoided.

  ESI develops methodologies to address this challenge in the Systems Dependability program line. It also addresses the various lifecycle phases of systems, currently centering around two topics: (i) verification and validation and (ii) system diagnostics.

- Program Line 3 – **System Evolvability**
  Newly released high-tech equipment systems are usually evolutionary improvements of existing systems, in which components evolve at different speeds, software often having the highest evolution and obsolescence speed. There is a need for regular system updates during service in the field that can last multiple decades. That is a challenge and an opportunity: on the one hand, many versions need to be serviced, while the update business can be significant and profitable.

  This program line addresses that in several ways: methods for defining and managing component interfaces to improve system methods to deal with legacy software and systems, and methods for configuration management and architectural modeling of system diversity.

- Program Line 4 – **Exploiting Systems Context**
  Today's high-tech equipment is hardly ever performing its function in isolation. The systems will be integrated into the customer context regarding customer processes/workflows, data exchange, or physical integration. The equipment becomes part of (customer-specific) systems-of-systems.

  This program line focuses on: (i) integrating equipment into systems-of-systems; (ii) systems that are intelligent and that adapt to their operational environment, and (iii) optimizing the performance of equipment integrated into a system-of-systems and optimizing the performance and dependability of systems-of-systems as a whole.

- Program Line 5 – **System Architecting**
  Finally, the System Architecting program line supports the other program lines with research on system architecting methodologies. The program line delivers model-based system architecting methodologies with a strong focus on creating models that link customer and business value to architecting and engineering decisions.

  A special track in this program line is on the value and adoption of MBSE.

ROADMAPPING PROCESS

To ensure that the research of ESI is relevant for the Dutch high-tech equipment industry, we regularly conduct a process to take stock of the needs of the industry. We recently completed this biannual process in the summer of 2022. The approach we took this time was structured as sketched in Figure 4.

In a series of meetings, we discussed two topics with each industry partner of ESI, eight in total, individually:

- Strategic business directions, opportunities, and challenges;
- The key capabilities that are needed to achieve these.

After each of these discussions, we analyzed the outcome, and per partner, we mapped the business and capability needs to the area of (systems) engineering methodologies. A one-page summary was composed per partner containing an

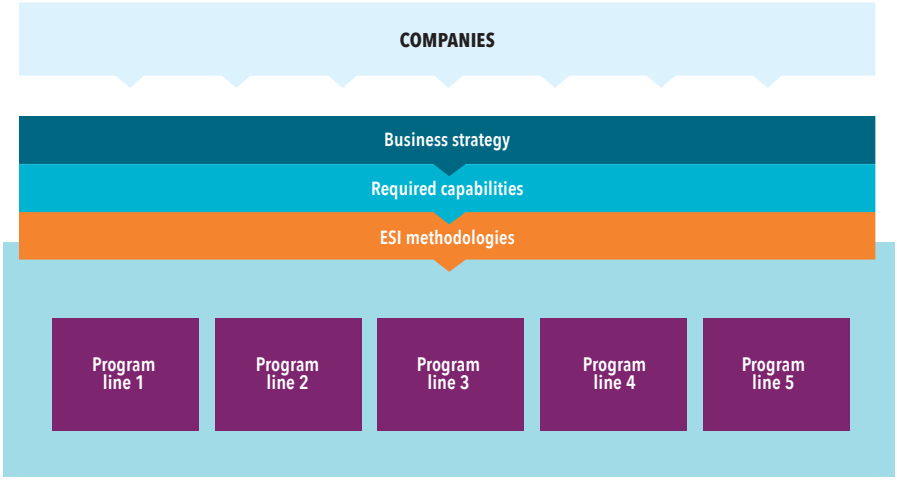


Figure 4. Roadmapping process

overview of (i) business needs; (ii) required capabilities, and (iii) ESI opportunities to support the business with developing and embedding innovative methodologies.

The aggregated picture was created in a subsequent workshop with all industry and academic partners, and we identified common challenges, needs, and priorities.

Thus, we identified ample methodological opportunities for research in the next five years, as the basis for the demand-driven roadmap and as input for the 2023 research program.

ROADMAP

The process sketched above resulted in an updated industry-driven research roadmap of ESI. It combines the continuation of running program lines, shifts of focus in running program lines, and initiatives for new program lines. Having consulted a broad range of leading industries in the Dutch high-tech equipment domain, we are confident that it covers well the needs of the high-tech equipment industries.

Trends and Characteristics

Resulting from this process, we have identified several critical trends and characteristics and needs that follow from that:

i. **Enhanced criticality**: Systems are increasingly critical, and methodologies to assure dependability (also in systems of systems) are key.
ii. **Enhanced diversity:** Customer-specific systems diversity is growing, combined with deeper integration into customer processes and systems of systems, asking for improved methodologies for diversity management and efficient and effective verification and validation of diversified product families.
iii. **Continuous innovation and updating:** The market expects products that are kept up to date. Therefore, methods are needed to ensure easy, dependable updates, guaranteeing system functionality and performance for diversified installed bases. A more agile system engineering method is asked to support such fast innovation.
iv. **Climbing the value chain:** Equipment manufacturers climb up in the value chain, providing higher-level services to customers. By doing this, the scope of their architecting expands quickly beyond their equipment. They need to understand systems beyond their equipment, and they need broader domain knowledge.
v. **High demand for engineering experts:** A general observation is that experts are hard to find. New team members take a long time to maximize productivity and quality. This calls for "democratization."
   - *Democratization of the systems*: the industry seeks ways to make the operation of its systems less complex by leveraging artificial intelligence, machine learning, and other smart algorithms. This raises a new, related challenge: how to optimally design their systems for AI integration combined with context-sensitive, adaptive system behavior?
   - *Democratization of research and development R&D*: making it easier to develop the equipment. The industry seeks methodologies and tools that support the complex tasks of engineering the equipment, for example, bringing AI-based algorithms and other advanced algorithms to the engineers' fingertips.

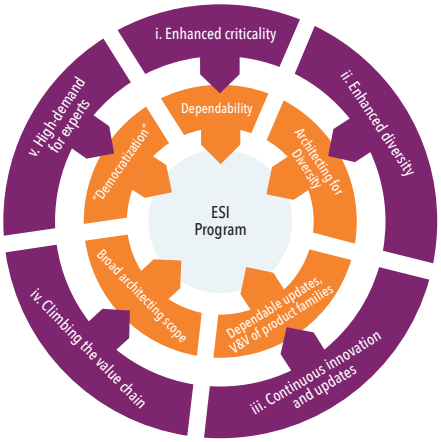


Figure 5. Trends and characteristics in the high-tech equipment industry and the resulting needs

Roadmap Priorities

Given these trends and characteristics, ESI discussed with its partners the priorities for their research, which led to the following high-priority topics:

- Architectural methods to create more modular systems; methods to design, describe, and enforce system and module interfaces, including inference of legacy interfaces.
- Methods to enhance the efficiency and effectiveness of verifying and validating highly diversified systems and systems-of-systems. This applies for V&V of new system releases but, maybe even more importantly, for system and software updates and upgrades.
- Methods for smart diagnostics and for architecting systems with optimal diagnosability.
- Methods to architect systems for optimal integration of AI/ML in critical high-tech equipment (engineering for AI).
- Methods applying AI to optimize the efficiency and effectiveness of R&D teams (hyper-automation for R&D, AI for Engineering).

Updates to the Program Lines?

Based on these priorities, we have reconsidered our existing program lines:

- The existing five program lines remain relevant for the high-tech equipment industry.
- In these program lines, we will address the opportunities and challenges of AI/ML.
- We expect to see more research in the evolving systems and systems in context programs, focusing on system diversity and systems-of-systems aspects, respectively.
- In the exploiting systems context program line, we expect to work on system adaptivity using AI and ML: how do we realize system adaptivity in high-tech equipment and combine this with the required dependability of critical equipment?
- In the systems architecting program line, we expect to combine these challenges in methods for systems architecting. A special focus is expected for MBSE, digital engineering, and agile (model-based) systems engineering.

Next to these shifts, we see new topics emerging. For these, we will start dedicated studies with our partners to define new research initiatives (in the existing program lines or potentially in a new program line): *system democratization* and *R&D democratization*.

EXPERTISE TEAMS AND ROADMAPS

The research at ESI draws on the knowledge and insights built up at ESI and our partners over the years and is clustered in seven expertise areas:

1. **System performance engineering:** methods and tools to address the performance challenges in system design, which are cross-cutting and often require a holistic view. The approach is primarily model-based, and the current practice is that models support design decisions. Based on a recent field survey conducted by ESI (Van der Sanden 2021), the vision is that models will act as authoritative sources of truth and form the basis for automated synthesis of implementation artifacts supporting optimal system performance over the lifecycle.
2. **Software legacy and rejuvenation:** methods and tools to support

ESI Expertise	Short term (~2 years)	Mid term (2–5 years)	Long term (5–10 years)
System performance	Model and analyze product families	Automatic design-space exploration	Performance by construction
	Automatic inference of performance models	Diagnose and resolve performance issues with AI and domain knowledge	Runtime optimization for autonomous and adaptive systems
	Analysis of microservice/ cloud-based systems	Analysis of system-of-systems	
System and software testing	Behaviour-based traceability	Specifications as tests	Error free, or fully verified and validated software
	Test generation and selection based on evolving Product Line specs	Evidence-based Product Line testing	
		System and test quality quantification (When to stop testing?)	
SW legacy and rejuvenation	Software understanding and checking	Controlled software redesign	Computer-aided software maintenance
Intelligent diagnostics	Generalist service engineer	First time right	Zero unscheduled down-time
	Guided reactive system-level root cause analysis		
Software and system behavior	Modeling non-functional aspects	Modelling variability and configurations	Code generation and supervisor synthesis
	Checking model correctness		First time right development
	Behavioral comparison	Change impact analysis	Computer guided evolution
Adaptivity and machine reasoning	Decision support	Self adaptation and supervised autonomy	Intelligent systems
	Digital Twin inside		
System architecting systematics	Value-based Reference Architecture	Architecting for data and AI	Effective systems architecting in a digital environment
	MBSE for business value		
	Effective platform development	Flexible composition using platforms	Well-founded product innovation
			Effective and well-managed product lifecycle

Figure 6. Summary view of the ESI expertise roadmaps

the developers of industrial software in understanding their codebases and in (semi-)automatically improving them and reducing the accidental complexity. Legacy code can be scrutinized and rejuvenated using static analysis (Mooij 2020) and dynamic code analysis (Aslam 2020). The vision is to establish continuous computer-aided software maintenance, so code never gets old.

3. **System and software testing:** the objective is to guarantee system quality. This expertise focuses mainly on improving the effectiveness and efficiency of software testing at the system level (see Hendriks 2020). The strategy is to leverage the availability of models to automate testing as much as possible, thus speeding up the test process and enriching the test suites, for example, using model-based approaches (Tretmans 2019). The vision is to achieve error-free or fully verified and validated software and systems.
4. **Intelligent diagnostics:** identifying the root cause of system performance degradation and complete system failure. The aim is to speed up this process by automatically providing correct, timely information, thus reducing the knowledge required for troubleshooting (Barbini 2021). The vision is to achieve first-time-right diagnostics and, ultimately, zero unscheduled downtime.
5. **Software behavior:** the aim is to move beyond the traditional structural description of software and systems and to describe the behavior in an actionable way. This is done by capturing how the building blocks affect their surroundings, as described in (Schuts 2018). This includes static and dynamic behavior of a single interface, multiple interfaces, an entire component, multiple components, and the complete architecture. This must make it possible early in development to comprehensively analyze the behavior and detect issues. The vision is to establish an authoritative source of information with complete descriptions of the behavior and to achieve correctness by design.
6. **Adaptive systems and machine reasoning:** addresses the need for systems to autonomously change and adapt over their lifecycle, for instance, using AI techniques. An important topic is the perfection of knowledge-based and data-driven digital twins and their concurrent in-system use

(Pil 2022). While it is still early for these developments, the vision is to establish self-adaptation and supervised autonomy of systems and, ultimately, truly intelligent, possibly even self-aware systems.

7. **System architecting systematics:** approaches, methods, and tools to advance the art of architecting and help R&D departments and system engineers deal with the ever-increasing complexity of high-tech systems (Wesselius 2022). This complexity owes to the systems being increasingly software and data-intensive and integrated into systems-of-systems. At the same time, development faces trends like continuous value delivery and growing demand for customization. The vision is to develop an effective, scalable, and deployable practice in system architecting to meet the needs of highly digitalized systems rich in data and AI content.

ESI experts in each of the above areas assemble in teams that develop a view of future developments within the area in a global sense. This view is captured in a detailed roadmap per expertise area. Each roadmap identifies trends and objectives on short (<2 years), mid- (2-5 years), and long terms (5-10 years). These trends are underpinned by the expected or needed solutions and capabilities, with the foreseen innovations in formalisms, techniques, and methods supporting these objectives. Detailed descriptions of each roadmap are beyond the scope of this paper. Figure 6 summarizes all the roadmaps, offering a comprehensive view of all the ESI expertise areas.

BRINGING THE INSIDE-OUT AND OUTSIDE-IN ROADMAPS TOGETHER

At ESI, we have thus created two roadmaps of different natures. First, we have presented the industry, demand-driven roadmap, leading to the definition of five program lines, each addressing specific problems and challenges the industry faces. We call this the outside-in roadmap. Subsequently, we discussed the expertise roadmaps focused on expected and desired methodological developments in each domain. We call this the inside-out roadmap. Though not identical, these roadmaps are by no means independent and uncoupled:

- the outside-in roadmap translates industry needs into solution directions worth exploring and applying, drawing on relevant expertise;
- the inside-out roadmaps describe the projected developments within the discipline, where these developments are, among others, driven by requirements from the industry.

The two roadmaps come together in the definition and execution of the research projects that constitute the ESI research program, as depicted in Figure 7.

Research projects are initiated to address an industrial challenge, focusing on a specific industrial use case. In projects, one or more methodologies will be developed and validated typically against the specific industrial use case. This way, the industry challenges identified in the outside-in roadmaps shape the ESI research program.

The execution of a project draws on one or (usually) multiple ESI expertise areas. The insights gained in a project contribute to extending the expertise of ESI. During the project definition phase, the relevant expertise roadmaps are consulted to check which roadmap items are covered by the set of projects. The expertise teams will indicate which roadmap items have a high priority to be addressed by the research program. Project plans will be amended accordingly, ensuring the roadmaps are executed. Thus, the inside-out roadmaps help define the program.

Soon we intend to bring the two roadmaps together in a more holistic view. Here, we will acknowledge that although the roadmaps are correlated. There also is and should remain, some tension. On the one hand, there will be industry needs where the potential solutions remain beyond the current possibilities. On the other hand, some methodological developments logically follow from pursuing a trend that may not yet address a clear industry need.

Our roadmap also feeds directly, through active participation, into national and internal research agendas, such as the Dutch HTSM systems engineering roadmap and the EU electronic components and systems roadmap, ECS-SRIA.

COMPETENCE DEVELOPMENT PROGRAM (CDP)

Methodologies only have value when organizations can put them into practice. This requires stepping into education and developing professional capabilities for designing high-tech systems. ESI has created several programs and learning tracks to support companies in developing and exploiting such competencies, focusing on long-lasting results. We recognize that to be effective theoretical classroom training needs to be applied in industry practice. Our learning tracks typically center around an executive-sponsored industrial use case brought in by the company of the participants. Our competence development program, the ESI academy, aims to cover our knowledge base across all expertise areas.

CDP ROADMAP

Our CDP program covers expertise areas 2, 5, and 7, as listed above. Contents-wise, the goal for the coming years is to cover all expertise areas.

From an educational point of view, the CDP activities have changed over the years from dominantly technical stand-alone courses to tailor-made learning interventions coupled with business and personal development needs. The impact of this move has proven to be very significant. We also moved from executing such tailor-made programs within a single company to settings where multiple companies team together while maintaining the same high quality and personalization. This has resulted in a further increase in impact.

We aim to further develop our approach by creating and intensifying multi-company

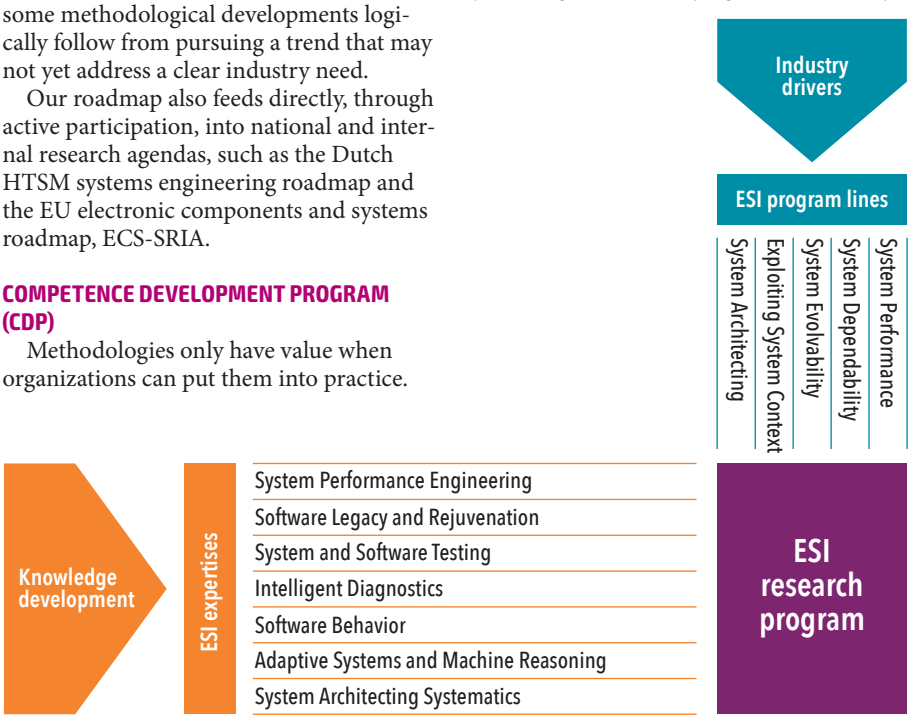


Figure 7. The ESI program lines, ESI expertise areas, and their respective roadmaps define and drive the ESI research program

thematic programs in which competence development and running research projects come together more directly.

In systems engineering, at least in The Netherlands, a lack of connection or alignment exists between mid-career education, as in the ESI career development program, and university and college education. ESI has co-founded an initiative to create a consistent program aligning concepts, mindsets, and methods across these different levels and stages of education. Going beyond the training of systems engineering as a process only, it aims to teach a systems engineering mindset in the context of business, domain, technical expertise, and leadership.

CONCLUSION

TNO-ESI leverages its strong and intimate partnerships with industrial companies to create and regularly update an overview of the high-tech equipment industrial needs and challenges in systems engineering. This leads to a demand-driven or outside-in roadmap organized in five program lines. Next, seven expertise roadmaps are created and regularly updated to lay out the expected and desired methodological developments. Together these define the ESI research program executed in collaboration with its university and industry partners. Similarly, these roadmaps steer the ESI competence development program to guarantee that knowledge transfer to industry reaches its full potential. ■

ACKNOWLEDGMENTS

Our thanks go to all involved in the ESI roadmapping process. This includes ESI research fellows and project managers, the members of the ESI Partner Board, and many co-workers at our partners' organizations. We also thank the ESI Partners and the Netherlands Organisation for Applied Scientific Research TNO for their financial support, management support, and collaborative efforts to conduct the research and reach the results described here. Further, we acknowledge the Netherlands Ministry of Economic Affairs and TKI-HTSM, the Netherlands Enterprise Agency (RVO), and the European ECSEL JU for financial support.

REFERENCES

- Heemels, W.P.M.H., E,H, van de Waal, G.J, Muller. 2007. "A design methodology for high-tech systems." In: *Boderc: Model-based design of high-tech systems*, edited by Maurice Heemels and Gerrit Muller: 11-26. Embedded Systems Institute, Eindhoven, The Netherlands.
- Van der Sanden, B., Y. Li, J. van den Aker, B. Akesson, T. Bijlsma, M. Hendriks, K. Triantafyllidis, J. Verriet, J. Voeten, T. Basten. 2021. "Model-Driven System-Performance Engineering for Cyber-Physical Systems." EMSOFT '21: *Proceedings of the 2021 International Conference on Embedded Software September 2021*: Pages 11–22. doi: 10.1145/3477244.3477985
- Mooij, A.J., J. Ketema, S. Klusener and M. Schuts. 2020. "Reducing Code Complexity through Code Refactoring and Model-Based Rejuvenation." *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*: 617-621, doi: 10.1109/SANER48275.2020.9054823.
- Aslam, K., L. Cleophas, R. Schiffelers and M. van de Brand. 2020. "Interface protocol inference to aid understanding legacy software components." *Softw Syst Model 19*: 1519–1540. doi: 10.1007/s10270-020-00809-2.
- Hendriks, T., K. Triantafyllidis, R. Mathijssen, J. Wesselius, P. van de Laar. 2020. "A Virtual Test Platform for the Health Domain." In: *Validation and Verification of Automated Systems*, edited by A. Leitner, D. Watzenig, J. Ibanez-Guzman, 297-320. Springer, Cham, CH. doi: 10.1007/978-3-030-14628-3_21
- Tretmans, G.J., and P. van der Laar. 2019. "Model-Based Testing with TorXakis: The Mysteries of Dropbox Revisited." In: CECIIS: 30th Central European Conference on Information and Intelligent Systems, HR. Proceedings, edited by V. Strahonja, 247-258, Varazdin, HR: October 2-4. Faculty of Organization and Informatics, University of Zagreb, Zagreb, HR. http://archive.ceciis.foi.hr/app/public/conferences/2019/Proceedings/QSS/QSS3.pdf.
- Barbini, L., C. Bratosin, and T. Nägele. 2021. "Embedding Diagnosability of Complex Industrial Systems into the Design Process Using a Model-Based Methodology." *PHM Society European Conference* 6 (1):9. doi: 10.36001/phme.2021.v6i1.2806.
- Schuts,M., J. Hooman, I. Kurtev and D. Swagerman. 2018. "Reverse Engineering of Legacy Software Interfaces to a Model-Based Approach." *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018: pages 867-876. doi: 10.15439/2018F64
- Pil, A. 2022. "AI trains itself match fit on a digital twin." Bits&Chips, 2 June. https://bits-chips.nl/artikel/ai-trains-itself-match-fit-on-a-digital-twin/.
- Wesselius, J., J. van den Aker, R. Doornbos, T. Hendriks, J. Marincic, W. T. Suermondt. 2022. "MBSE in the High-Tech Equipment Industry, MBSE-Study of ESI and Partners – Observations and Conclusions." White paper, ESI (TNO). https://publications.tno.nl/publication/34639873/jgHNmz/TNO-R2022-R11504.pdf. TNO 2022 R11505/ESI 2022-10029

ABOUT THE AUTHORS

Wouter Leibbrandt is the science and operations director of TNO-ESI. Before joining ESI in 2016, Wouter was with NXP for ten years, managing the Advanced Applications Lab. Until 2006 he was with Philips Research labs for 14 years, managing various projects and departments in The Netherlands and abroad. Wouter holds a PhD in physics from Utrecht University.

Jacco Wesselius has been the business director of TNO-ESI since July 1, 2022. He joined ESI as a senior project manager in 2018. Before joining ESI, Jacco was with Technolution, where he was project manager, technology director, and business unit director from 2012 until 2018. Until 2012, Jacco was with Philips Healthcare for seventeen years, where he had various functions such as software engineer, systems architect, technology manager, and R&D director. Jacco holds a PhD in software engineering from the Delft University of Technology.

Frans Beenker was the ESI business director responsible for managing programs and activities within the high-tech industry. Before joining ESI at its start in 2001, Frans was with Philips, where he held several technical and management positions. Starting in 1982 at Philips Research, he moved to Philips Medical Systems in 1994, where he worked as a project manager on several large-scale product development projects. Frans holds a PhD in electrical engineering. In the summer of 2022, Frans transferred his responsibilities to Jacco Wesselius.

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.

DECEMBER 2022
VOLUME 25/ ISSUE 4

Guiding Systems Engineering Research for Enhanced Impact in the Development of Increasingly Complex Cyber-Physical Systems

Tom McDermott, tmcdermo@stevens.edu; and **Dinesh Verma,** dverma@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 set of roadmaps (SERC 2019) structuring and guiding four areas of systems engineering research: digital engineering, velocity, security, and artificial intelligence (AI) and autonomy. This paper presents the development of these roadmaps and the key underlying transformation aspects.

INTRODUCTION

The Systems Engineering Research Center (SERC) is a university affiliated research center of the US Department of Defense and leverages the research and expertise of senior lead researchers from 22 collaborator universities (Figure 1, next page) throughout the United States. Begun in 2008 and led by Stevens Institute of Technology, the SERC is a national resource providing a critical mass of systems engineering researchers — a community of broad experience, deep knowledge, and diverse interests. The SERC is unprecedented in the depth and breadth of its reach, leadership, and citizenship in systems engineering through its conduct of vitally important research and the education of future systems engineering leaders.

As part of its UARC designation, the SERC operates to a 5-year technical and research plan. The research strategy aligns to four core research areas as agreed upon by our research sponsor. These four areas have been relatively stable over our history. In the last 5-year planning activity we adopted four additional cross-cutting research strategies that would integrate across the core research areas. While the core research areas link to fundamental aspects of systems engineering, the cross-cutting research strategies address specific application challenges of our sponsors. In its strategic research planning, the SERC employs a research council composed of senior faculty across our university collaboration network. Our research roadmaps were developed collectively across the research council in a 6-month period in 2019. The use of a graphical roadmap (as opposed to a text document) was selected to make our research strategy more accessible to our sponsors and the larger systems engineering community. The SERC's current research strategy now aligns to the four core research areas supported by the four cross-cutting research strategies, as shown in Figure 2 (next page). The research areas are enterprises and innovation, models and data, digital transformation, and human capital development. The cross-cutting research areas that are supported by these roadmaps are:

- Digital engineering: systems engineering is in a transformation process based on the data use and collaboration using models. Digital engineering has become the basis for all three SERC crosscutting missions and resulting research roadmaps.
- 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.

Each roadmap has a set of research "vectors" (arrows in the roadmap diagrams) leading to a visionary outcome or set of outcomes, and a set of capabilities (dots in the roadmap diagrams) we believe are needed to meet those long-term outcomes. The listed capabilities in these roadmaps reflect not only SERC research, but other areas of research either known to be active or prioritized by our sponsors and the systems engineering community in general and our sponsors. It is our hope by sharing this work we will guide not only SERC research but also the transformation of the systems engineering discipline in general. The following sections start with a description of the digital engineering roadmap then follow with descriptions of each of the other roadmaps.

Figure 1. SERC university collaboration network

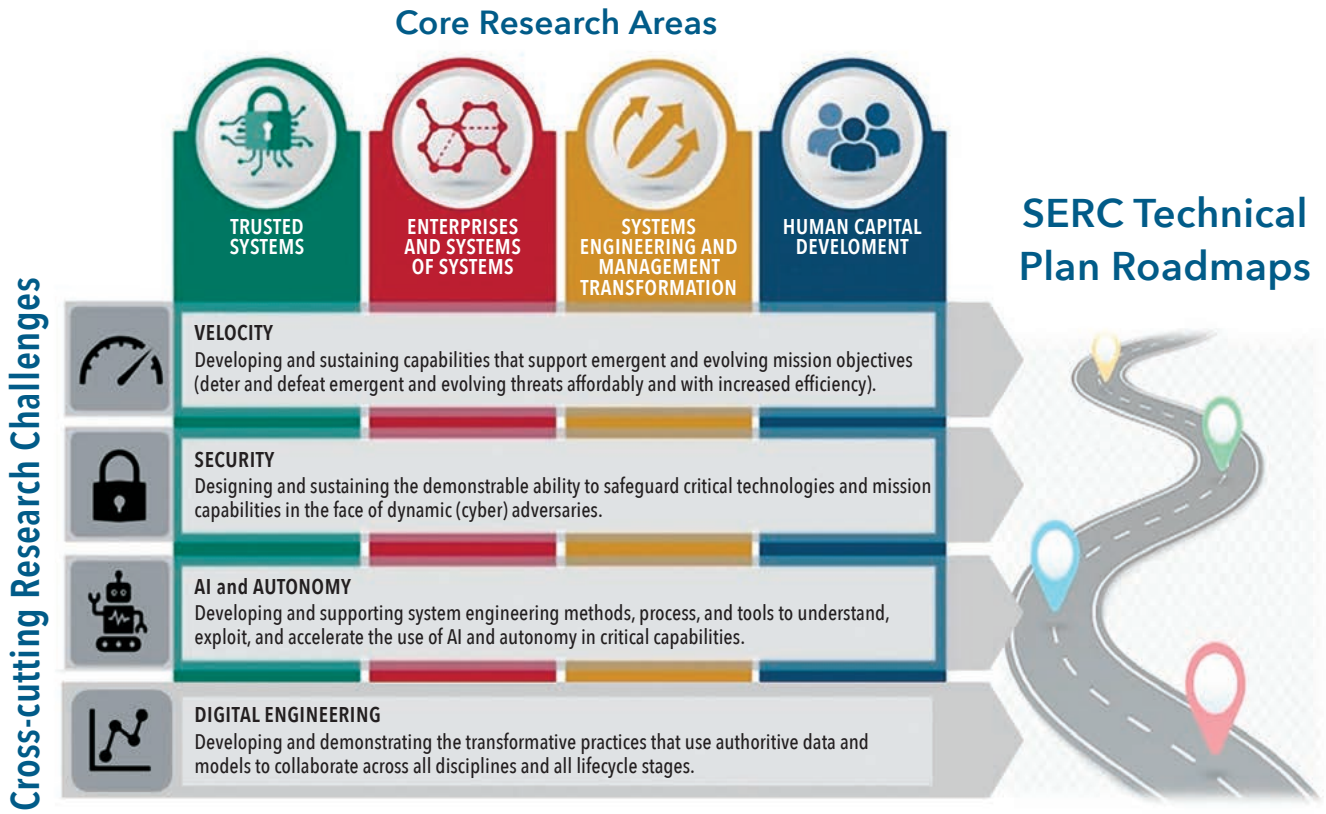


Figure 2. SERC research areas and roadmaps

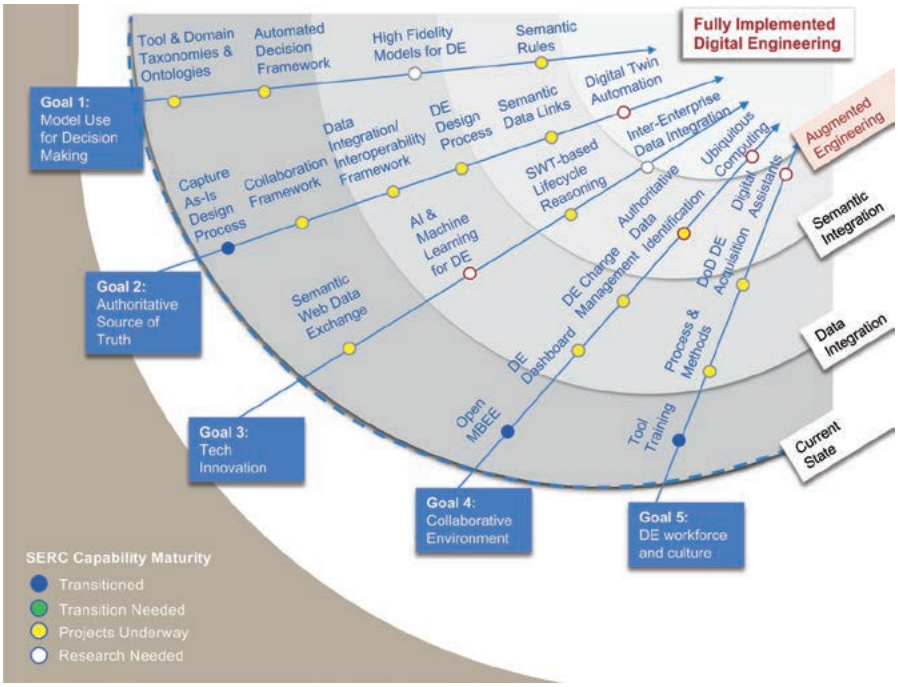


Figure 3. Digital engineering research roadmap

DIGITAL ENGINEERING ROADMAP

The digital engineering roadmap is shown in Figure 3. In these diagrams the color codes associated with each research area reference state of progress. In this diagram specifically the red outlined circles reflect areas associated with augmented engineering, or "Artificial Intelligence for Systems Engineering (AI4SE)."

Digital engineering forms the basis for all three of the other SERC crosscutting research challenges and resulting research roadmaps. We are leading a systems engineering transformation process that is based on the use of data and collaboration using models. The digital engineering research roadmap vectors align with the five stated goals of our DoD sponsor's strategy: (Goal 1) model use for decision making; (Goal 2) the authoritative source of truth (AST); (Goal 3) technological innovation; (Goal 4) collaborative environments; and (Goal 5) workforce and cultural evolution (DoD 2018). The progression in digital engineering is expected to begin with data integration followed by the semantic integration of models. We expect to soon see advances in augmented intelligence – the use of models and "big data", that bring automation to engineering processes and system quality and certification. The primary research needs associated with each goal are described here.

Goal 1: Formalize the development, integration, and use of models to inform enterprise and program decision-making.

The SERC looks to interoperability through ontologies in the future – graph databases for linked data are becoming more prominent; taxonomies provide the starting point for building ontologies, ultimately enabling AI-based reasoning. The combination of ontologies, SysML (descriptive models), and analytics provide a more automated framework for decision making related to analysis of alternatives across any type of decision, characterized by an objective hierarchy (basis for decision). Having the appropriate fidelity model is important for addressing the needed information; our research includes looking at different optimization architectures, and another research challenge is moving back to the parametric space after moving to higher fidelity models. Semantic rules based on knowledge representations such as ontologies will provide the basis for reasoning about completeness and consistency using AI and machine learning (ML) based tools.

Goal 2: Provide an enduring, authoritative source of truth (AST).

This goal is primarily focused on capturing the as-is design process in a digital collaboration framework. This will provide a new operational paradigm for program insight and oversight as well as more seamless collaboration between various disciplines and stakeholders. Challenges include protection of data rights, intellectual property (IP), and security as system designs are given access through collaboration tools. A means to analyze data/information seamlessly and efficiently across domains and disciplines, and from mission to systems, and downwards to components across the lifecycle is the desired future state. This future state is initially reflected by some examples demonstrating the art-of-the-possible by doing "everything" in models, simulations, data, and more, including subsuming processes enabled by managing data models in an authoritative framework and workflow that is linked to the program's systems engineering and development process. Semantics such as the use of ontologies will provide the basis for more meaningful interrelationships of information and will provide the basis for applying AI and ML to finding and managing data. The end game is digital twin automation – fully dynamic virtual and parallel representations of physical systems that evolve over time with real-world feedback.

Goal 3: Incorporate technological innovation to improve the engineering practice.

Ontology-based and associated semantic web based technology data exchange infrastructure will enable more seamless and efficient data/information exchange. Systems engineering is a combination of semantic and mathematical reasoning. Semantic web interfaces to ontologies-based knowledge representation will enable reasoning about mission and systems engineering to enable augmented intelligence. To realize this, we need continuing evolution of high-performance computing and other technologies. Data/information will become seamlessly updated/exchanged continuously in "real-time" cutting across the entire enterprise (technical, manufacturing, cost, risk).

Goal 4: Establish a supporting infrastructure and environments to perform activities, collaborate, and communicate across stakeholders.

Information technology "stacks" for the engineering disciplines are evolving from single disciplinary environments to large interdisciplinary infrastructures. This evolution has proceeded relatively independently in the engineering and product line management communities versus the software and data management communities. In the future these will become integrated along with other enterprise data-driven disciplines like project management and supply chain management. The digital engineering dashboard, real-time communication on continuous flows of data across all engineering and management activities, is a much needed area of research. This requires new ways to visualize multi-parametric and multi-objective information to support decision making, personalized based on stakeholder needs. A challenge is extending change management to consider data management and model management, which is much more "object-based" and also more aligned to competencies and roles of

stakeholders. Automation is needed to find "authoritative data," assisted by AI/ML tools that understand what the user is looking for. In the future we won't even think about the underlying computation or where it is stored.

Goal 5: Transform the culture and workforce to adopt and support digital engineering across the lifecycle.

As is always the case in a digital transformation, much of the change will be cultural. Tool training is needed specific to roles. A challenge is having relevantly complex examples to use in association with learning the tools. There is a need to focus more on the methods that characterize the information that must be captured and the associated process that provides guidance in capturing the relevant information, to build right the system and build the system right. New data and software environments will continue to challenge currency of skills, much like software disciplines today. New policies that align with the new operational models are required. We will see an increase in "digital assistants" in systems engineering activities, and need methods to trust AI guidance in engineering and decision making.

VELOCITY ROADMAP

Velocity and agility are critical characteristics of future systems, both for the system that is being deployed and the system that is 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 is perhaps the most difficult to articulate as it is rooted in current organizational implementation of these practices and methodologies. With our defense and other government sponsors, velocity centers on three goals:

1. architecting systems for continuous development and deployment,
2. leading an agile transition across large government and contractor systems, and
3. the role of digital engineering as an enabler.

Overall, our vision is to enable the transformation of systems engineering 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. The research verticals in this area strive for application into two areas: improved mission engineering processes and creation of more adaptive systems. Research areas include rapid development of systems as platforms, architecting these platforms for DevOps enabled systems and environments, and execution of DevOps practices in our sponsor organization.

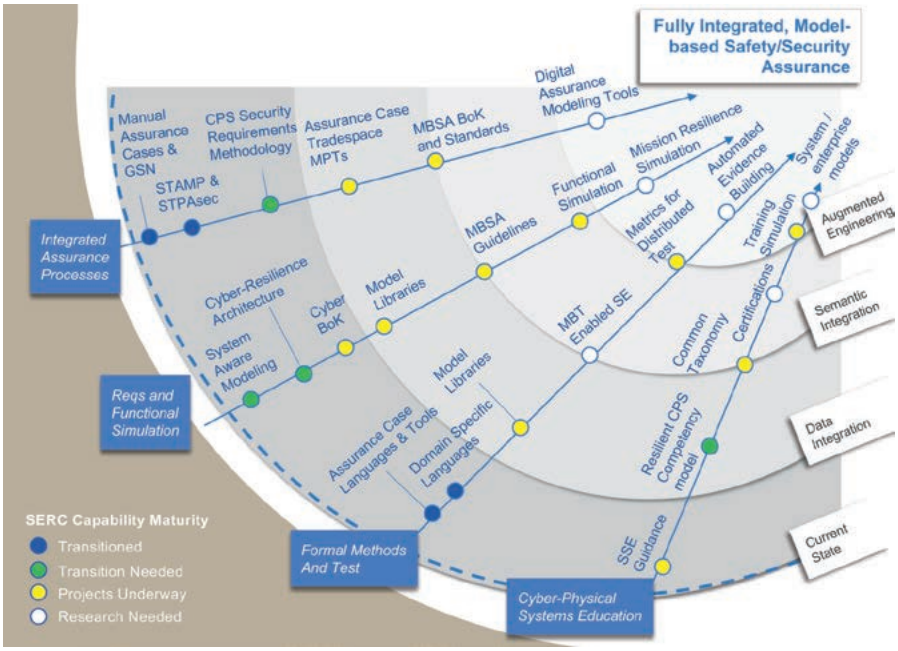


Figure 4. Security research roadmap

SECURITY ROADMAP

The SERC security roadmap is shown in Figure 4. This roadmap focuses on critical engineered systems such as cyber-physical systems, embedded systems, and weapon systems. These are often 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. Research is underway in four vectors: integrated assurance processes, which address the system design space in a way that integrates security/safety/reliability and advances practices across all three disciplines; requirements and functional simulation, which focuses on early stage design practices and security patterns (build the right system); formal methods and test, which hopes to advance research in proof driven validation and evidence (build the system right); and cyber physical systems education, addressing the current shortfall of security related education in engineering programs.

Traditional assurance case design uses goal-structured notation or similar arguments, there has been limited adoption of assurance cases for cybersecurity at the system level, particularly for cyber resilience. System theoretic process assessment-security (STPA-Sec), developed by Nancy Levison at MIT, moves from causal chain-based assurance to control loop analyses and is more effective at the system level. The research challenge is to capture this into system modeling tools. The SERC developed the cyber security requirements methodology: a systematic process for behavioral analysis of security threats and associated risk assessment and is transitioning these methods into modeling tools. Still needed are quantifiable measures of safety/security assurance, via economic studies and criticality models, to examine and formally trade development from a safety and security view. Digital assurance modeling tools are an active research area.

The SERC also developed the system aware modeling approach to capture and model combined system, threat, and countermeasure behaviors. Research continues with development and demonstration of cyber-physical system architecture patterns that support behavioral models of cyber threats and assurance cases. Guides and standards for model development and model quality to support functional assurance are still needed. Reusable libraries of system, threat, and countermeasure functional components and patterns are needed as complexity of the analysis increases. Simulation of system functions to evaluate threat/ countermeasure effectiveness, simulation of missions and operations in cyber-threat environments linked to quantifiable measures, and related visualization tools are still needed.

This will lead to improved assurance case formalisms and tools: standard and domain specific assurance case languages linked to design tools. DARPA programs prototyped

an assurance case language that has seen limited use. This work includes domain specific languages that support modeling of cyber-physical system architectures and characteristics to support automated design and code generation. Further research is needed on model libraries allowing reuse and aggregation of component models to support design and test buildup. Also, metrics for distributed test: measurement models and AI/ML based prediction of coverage for distributed testing. Future AI/ML research will lead to automated evidence building – automation of test and certification processes via models and quality assurance (QA).

In the education domain, the community lacks a lexicon/taxonomy to adequately describe the cyber-physical system security domain. There remains a need to develop formal taxonomies to link the computing and military cybersecurity domains. Competency models need to extend existing software and IT security focused frameworks into engineering competencies, specializations, and roles. System Security Engineering guides are needed for this domain. There is also a need for educational simulations: cyberspace-realistic virtual reality simulation for relevant systems (aircraft, missile, trucks, power plants, and so forth) in an unclassified domain. In the longer term, formal security certifications for engineering professionals need to be matured.

AI AND AUTONOMY ROADMAP

The AI/autonomy roadmap is shown in Figure 5. The envisioned long-term outcome of the SERC AI and automation roadmap is "human-machine co-learning." This outcome captures a future where both humans and machines will adapt their behavior over time by learning from each other or alongside each other. More importantly for systems engineering, this is a lifecycle model that is not envisioned and supported by most of the current-day systems engineering practices. To achieve this end state, one might consider there is a need for both the AI and systems engineering disciplines to pass through a set of "waves" or eras. The first of these includes sets of technologies and approaches that make the decisions produced by AI systems more transparent to the human developers and users. The second wave is to produce systems that learn but are also appropriately robust and predictable in the type of critical applications normal to systems engineering. The third wave involves systems that adapt and learn dynamically from their environments. The vectors of this notional roadmap span five categories. The first of these vectors recognizes that the technological implementation of AI systems will evolve and will need to evolve in directions relevant to systems engineering. Most of these can be related to the development of transparency and trust in technology. The second vector recognizes that the purpose of AI in systems is generally to provide automation of human tasks and decisions. The third vector recognizes that AI technologies will gradually be used more and more to augment the work of engineering and the fourth vector recognizes that the current digital engineering transformation will be enabler for that. The final vector recognizes a transformation will need to be accomplished in the systems engineering workforce, with significantly more integration of software and human behavioral sciences at the forefront.

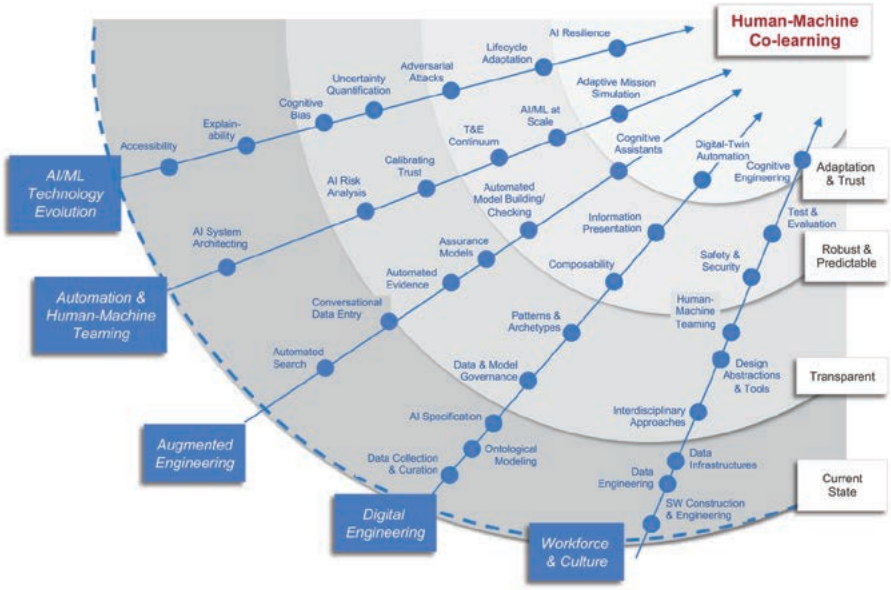


Figure 5. AI/autonomy roadmap

AI AND MACHINE LEARNING TECHNOLOGY

We foresee that AI algorithms and methods will continue to become more available in tools that can be used by multiple disciplines. As systems engineers we are always interested in the rigor with which these will be developed and tested. There are a set of continuing technology related challenges that remain significant areas of research. These include:

- Explainability: developing sets of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance (prediction accuracy); and enable human users to understand, appropriately trust, and effectively manage the resulting automation.
- Cognitive bias: reducing errors induced in sampled data or algorithms that cause the expected results of the system to be inappropriate for use.
- Uncertainty quantification: representing the uncertainty of AI predictions as well as the sources of uncertainty.
- Adversarial attacks: use of adversarial samples to fool machine learning algorithms; defensive techniques for detection/classification of adversarial samples.
- Lifecycle adaptation: evolution of AI performance over the lifecycle of a system as the system changes/evolves.
- AI resilience: operational resilience of the system and its users incorporating AI, particularly involving the characteristics of ML systems.

AUTOMATION AND HUMAN-MACHINE TEAMING

There is a lack of testbed environments in the research community to explore the end-effects of human machine interactive teaming. Many of the collaborative behaviors are not well understood when first developed and learning is required on both the engineering and user side to evolve effectiveness. Building appropriate data and live and virtual system architectures to support learning and adaptation is a critical research area. More agile change processes are also critical. Methods, processes, and tools are needed to connect system risk analysis results with AI software modules related to those risks. This is very similar to the cyber resilience research area. AI systems that self-adapt while maintaining rigorous safety, security, and policy constraints are not widespread today, so this is a significant research area. Methods for addressing AI-related system test and evaluation, particularly when these systems' ability to adapt and learn from changing deployment contexts improves. One largely

unexplored area is AI/ML at scale: appreciation for the dependence of an AI's outputs on its inputs. Scale in AI-based systems will increasingly lead to more general intelligence and an inability to relegate AI to a particular subsystem or component — in other words the problem becomes difficult to decompose. Computer-based simulation and training supporting non-static objectives and/or goals (games, course of action analysis) necessary to provide contextual learning environments for these systems.

AUGMENTED ENGINEERING

AI and ML have significant potential to help engineers and especially systems engineers do their work. We call this augmented engineering. Automated search algorithms will be very beneficial, applying ML to historical data and relationships in the engineering domains. Human/computer interaction processes to convert natural language and other media to formal models should follow. Research is growing on automated construction of models from features in semantic data, used in both creation of new models and correctness of developed models. Automation of certification and accreditation processes via models and automation of quality assurance data will improve the reliability of future systems. This includes automation of evidence-based models for assuring correctness and completeness of system requirements and design. Research is underway on cognitive digital assistants — conversational systems automating many mundane data entry, exploration, and engineering calculation tasks, and many workflows.

DIGITAL ENGINEERING TO SUPPORT AI AND AUTONOMY

Digital engineering will be a great enabler for use of AI/ML into engineering functions. Many of these research areas were mentioned in the digital engineering roadmap. Systems engineering will need to manage specific activities to build infrastructure and collect and manage data needed for engineering and programmatic activities in system development and support. As mentioned earlier, ontological modeling of engineering and programmatic data providing interoperability through standard, and domain specific ontologies will be critical. Lifecycle management, control, preservation, and enhancement of models and associated data will be a core systems engineering activity to ensure value for current and future use, as well as repurposing beyond initial purpose and context. In the long-term AI should enable digital twin automation: 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.

WORKFORCE AND CULTURE

AI and autonomy in the engineering domain, particularly the physics-based disciplines, requires much more interdisciplinary learning of data infrastructures, data engineering and software construction and engineering than is typically taught today. AI systems are highly interdisciplinary. Human-machine teaming is also very interdisciplinary, requiring knowledge across disciplines of machine control, cognitive science, and human learning. The traditional systems engineering specialty disciplines such as safety and security must adapt their practices to non-deterministic processes and systems. Test and evaluation must be integral to development and continually evaluating the system. All these dimensions will create workforce challenges for system developers and multidisciplinary challenges for educators.

SUMMARY

The SERC research roadmaps have evolved to be not just a tool to guide our research, but also a tool to help the systems engineering community understand the trends that will drive our discipline over time. They will continue to be updated on a regular basis. The SERC has a bias in these roadmaps towards our primary sponsor, the US Department of Defense. Comparing projections like these across all organizations focused on systems engineering research in all application domains will help create a much better picture of needs, funding priorities, and educational outcomes. ■

ACKNOWLEDGEMENTS

Although the listed authors wrote this article, the roadmaps themselves were created by several members of the SERC research council. The primary developer of the digital engineering roadmap was Mark Blackburn at Stevens Institute of Technology. The primary authors of the velocity roadmap were Paul Collopy at the University of Alabama-Huntsville and Barry Boehm at the University of Southern California. The primary author of the security roadmap was Peter Beling at Virginia Tech. The primary author of the AI and autonomy roadmap was Tom McDermott at Stevens Institute of Technology. Other significant contributors were Daniel DeLaurentis at Purdue (current chair of the SERC research council), Valerie Sitterle at Georgia Tech Research Institute, Barry Horowitz at the University of Virginia, Jon Wade (now at University of California San Diego), Bill Rouse at Georgetown University, and Kara Pepe at Stevens Institute of Technology.

REFERENCES

- Department of Defense (DoD). 2018. Digital Engineering Strategy. https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf
- Systems Engineering Research Center (SERC). 2019. Research Roadmaps 2019-2020. https://sercuarc.org/wp-content/uploads/2021/08/ROADMAPS_3.5.pdf

AUTHOR BIOGRAPHIES

Tom McDermott serves as the deputy director and chief technology officer of the Systems Engineering Research Center (SERC) at Stevens Institute of Technology in Hoboken, NJ. The SERC is a university affiliated research center sponsored by the Office of the Secretary of Defense for Research and Engineering. With the SERC he develops new research strategies and is leading 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 consults with several organizations on enterprise modeling for transformational change, and often serves as a systems engineering expert on government major program reviews. He has served on the INCOSE Board of Directors as director of strategic integration.

Dr. Dinesh Verma is a 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 the 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 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 a PhD in industrial and systems engineering.

TECoSA – Trends, Drivers, and Strategic Directions for Trustworthy Edge Computing in Industrial Applications

James Gross, jamesgr@kth.se; **Martin Törngren,** martint@kth.se; **György Dán,** gyuri@kth.se; **David Broman,** dbro@kth.se; **Erik Herzog,** erik.herzog@saabgroup.com; **Iolanda Leite,** iolanda@kth.se; **Raksha Ramakrishna,** rakshar@kth.se; **Rebecca Stower,** stower@kth.se; and **Haydn Thompson,** haydn.thompson@thhink.com

■ **ABSTRACT**
TECoSA – a university-based research center in collaboration with industry – was established early in 2020, focusing on Trustworthy Edge Computing Systems and Applications. This article summarizes and assesses the current trends and drivers regarding edge computing. In our analysis, edge computing provided by mobile network operators will be the initial dominating form of this new computing paradigm for the coming decade. These insights form the basis for the research agenda of the TECoSA center, highlighting more advanced use cases, including AR/VR/Cognitive Assistance, cyber-physical systems, and distributed machine learning. The article further elaborates on the identified strategic directions given these trends, emphasizing testbeds and collaborative multidisciplinary research.
■ **KEYWORDS:** edge computing, cyber-physical systems, trustworthiness, systems engineering, innovation eco-systems

INTRODUCTION

Several trends and drivers interact in the digitalization shift, including edge computing, connectivity, artificial intelligence, and big data loops, where field data are gathered to update software systems continuously. This transformation offers unprecedented innovation and product development opportunities and enables industrial companies to meet their targets for sustainable development goals. The need to address all dimensions of sustainability is highlighted by the recent European Commission initiative on Industry 5.0, emphasizing that previous efforts, such as Industry 4.0, have predominantly focused on productivity (EC Industry 5.0, 2022). A concrete example of what CPS can do for sustainability is the "tools" available to facilitate circularity, as an example, with traceability and predictive capabilities to support decisions regarding maintenance and recycling. However, digital transformation also increases system complexity. It introduces challenges of a socio-technical nature, such as risks related to technical systems acting in open environments, including ethical considerations related to fairness and personal integrity, INCOSE (2021), Törngren (2021). Specifically, our future societies will depend on increasingly sophisticated infrastructures where **edge computing** will act as a new tier, complementing the cloud and embedded systems. TECoSA, a research center on trustworthy edge computing systems and applications, was formed in 2020 to address the corresponding key challenges (TECoSA 2022; Törngren et al. 2021). The center brings together multiple research teams at KTH Royal Institute of Technology and (currently) 15 industrial partners spanning several industrial domains. The discussions among the center partners form the basis for the results presented in this paper.

TECoSA is active in industrial digitalization with a focus on edge computing systems. The aim is to provide methods, tools, and theories for building trustworthy systems relying on edge computing. The emphasis during the initial phase of the center—in the context of trustworthiness—has been on safety, cyber-security, and predictability (see Figure 1). Trustworthiness has traditionally been associated with human-machine interactions and security, referring to how we (humans) perceive trust in services and machines.

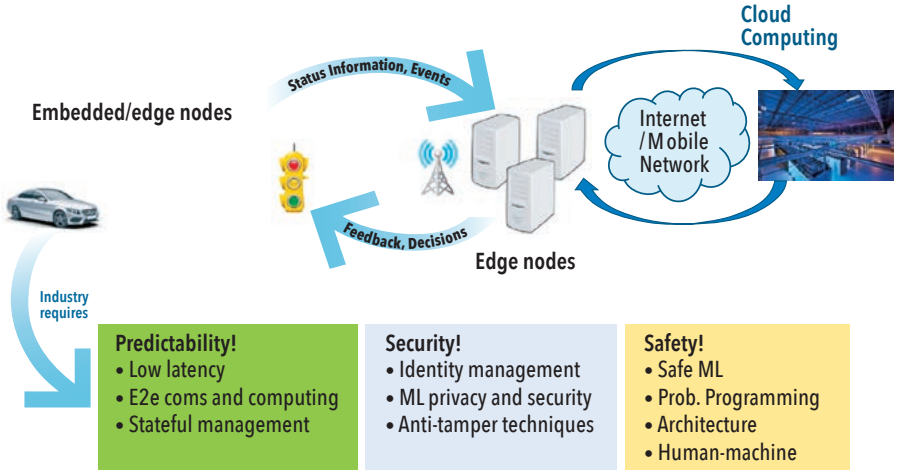


Figure 1. Edge computing as a new tier complementing embedded systems (device edge) and the cloud, illustrating initial trustworthiness properties and challenges addressed by the TECoSA center

Trustworthiness has evolved as an umbrella term encompassing the concept of dependability, associated with properties like reliability, availability, maintainability, safety, and security, and properties associated with artificial intelligence, such as transparency, explainability, and fairness (AI HLEG 2021).

The primary purpose of this article is to initially summarize and assess the current trends and drivers regarding edge computing. These insights form the basis for the research agenda of the TECoSA center. As a second purpose, the article elaborates and discusses the identified strategic directions given these trends.

EDGE COMPUTING STATE-OF-THE-ART

Edge computing is best understood in contrast to cloud computing. In the 2000s, the client-server approach dominated the first-generation internet architecture. Most clients were desktop PCs on private or corporate premises, connected via the Internet to web servers. A private or corporate entity intending to offer information or services on the Internet had to acquire server hardware and software, install and maintain it on corresponding premises, and set up a matching Internet connection. By 2010, this division had changed dramatically.

On the one hand, an increasing fraction of the clients were mobile devices, connecting through mobile networks like 3.5G and the upcoming 4G (LTE) to the Internet. On the other hand, web service offerings moved more and more to cloud providers, where very large pools of server hardware were brought together, allowing a scalable and efficient operation of web services from an installation, maintenance, and connectivity point of view. Web service operators moved from hosting and maintaining servers (with the content) locally on-premise to only curating content while renting the hardware and software for the web service from cloud providers. As a result, cloud computing centers of corresponding providers often ended up in locations where physical space, energy supply, and backbone connectivity were cheap, resulting in relatively remote locations. By and large, this is the dominating service model of the Internet as of today.

In this context, edge computing is primarily defined as computing services in "closer physical proximity" to clients compared to cloud computing, that is, offering computing services towards the "edge" of the Internet / wide-area networks. Given the dominant presence of 4G and 5G mobile networks as primary access networks of most clients in today's Internet, edge computing is realized by placing corresponding compute resources within the mobile network core or even within a radio access network, depending on the preferred proximity. In this line of thinking, proximity is traded with scale and cost: The higher the desired proximity of edge compute resources to the mobile clients, the more physical locations for placements of edge compute resources will be required, typically leading to fewer computational resources available per edge compute location.

Visions associated with edge computing have in various academic/industrial communities been given different names, including, for instance, multi-access edge computing (MEC) (related to telecommunications and 5G, earlier referred to as mobile edge computing), (Abbas et al. 2018), fog computing (with localized computations through communication devices such as routers and gateways in collaboration with the cloud), (Bonomi et al. 2012), and cloudlets (small scale localized data centers), (Satyanarayanan 2017). In the current discourse, edge computing has been associated with either locality, computing technologies, or both (Varghese et al. 2021). While the many projections for edge computing may appear confusing, this situation is not surprising since we are in the early stages of edge computing with an ongoing market positioning.

Our analysis from a commercial point-of-view is that edge computing provided by mobile network operators will be the initial dominating form of this new computing paradigm for the upcoming decade. Beyond that, new concepts might arise that exploit a continuum of available compute points from mobile clients to cloud centers (Duranton et al. 2021). *In the following, we refer to edge computing as the provisioning of additional computing resources through mobile networks.* Edge computing could be introduced to decrease hardware costs in mobile devices, such as industry robots and civilian or military surveillance systems while meeting latency and predictability demands. In this sense, edge computing adds computational resources that complement the existing capabilities of devices (embedded systems) and the cloud, belonging to a tier of a digitalized infrastructure. For a presentation of more detailed use cases, see the following discussion below.

Edge computing was arguably introduced roughly twenty years ago under the synonym "cyber foraging" (Balan et al. 2002). Since then, a set of various arguments have been brought up highlighting the potential benefits of edge computing:

- The original cyber-foraging research was motivated to **improve energy efficiency** if compute-intensive jobs could be offloaded from battery-powered mobile clients to stationary but close-by cloudlets, decreasing network-wide energy consumption. Either code or input data is offloaded through a mobile network to cloudlets, sending the computation back to the client. Cloud computing is seen in this context as having too long latency and unreliable, necessitating edge computing.
- A second argument, related to the above, can be made about the relative distance of cloud computing centers and, therefore, a much **lower access delay** in the case of edge computing. For compute tasks that are either too complex for mobile clients or require input from multiple mobile clients while being latency-sensitive, edge computing provides a clear advantage in providing lower round-trip delays. This case is, for instance, often made in the context of augmented or extended reality applications.

- Edge computing can also drastically **reduce the bandwidth** required for certain analysis services that run in the cloud. In this case, cloudlets are used as primary processing units, for instance, with respect to video analytics in detecting certain events or states in the video stream. Instead of conveying the entire stream to a cloud center, leading to a large bandwidth requirement as more and more endpoints are included in the service, only indices of the video frames and the detected objects are provided upstream to the cloud center. The corresponding video frames are nevertheless stored at the edge and can be retrieved by the cloud center. Similar cases can be made for predictive maintenance, IoT systems, and distributed machine learning applications.
- Finally, edge computing systems come with **different security and privacy features**. While typical concerns of security and privacy regarding cloud computing centers do not carry over to edge computing, new aspects such as physical access and manipulation become more relevant in the case of edge computing. Related to this shift towards more "local" aspects of security and privacy are also advantageous of edge computing with respect to **regulatory frameworks**. Due to the geographical proximity of deployed cloudlets and corresponding clients, edge computing offerings might guarantee the manipulation and storage of data within a specific regulatory framework, which a general-purpose cloud provider might not be able or willing to guarantee (in contrast to sovereign cloud offerings).

From these diverse drivers and advantages discussed in the academic/industrial community over the last ten years, for the first wave of commercial edge computing offerings foreseeable today, the regulatory and bandwidth-saving aspects are likely the main drivers. Concerning B2B customers, edge computing offerings of mobile network providers, referred to as Telco edge, as well as cloud providers, referred to as the regional cloud, will offer guarantees for the computing and storage location and, therefore, the regulatory conditions under which data is manipulated and stored. In addition, hybrid edge-cloud solutions are emerging that push the bulk of the processing to edge cloudlets while integrating the results of local cloudlet-based computing with cloud services. In both cases, "best effort" service level agreements (SLAs) between the service provider and customer are sufficient for successful commercialization. Beyond these B2B offerings, in the B2C space, a prominent commercialization case for edge computing appears to be online multi-player gaming, where depending on the location of the players and the placement of the game backend process, significantly higher quality of experience can be achieved. Still, corresponding offerings in the gaming domain will be run under best-effort SLAs.

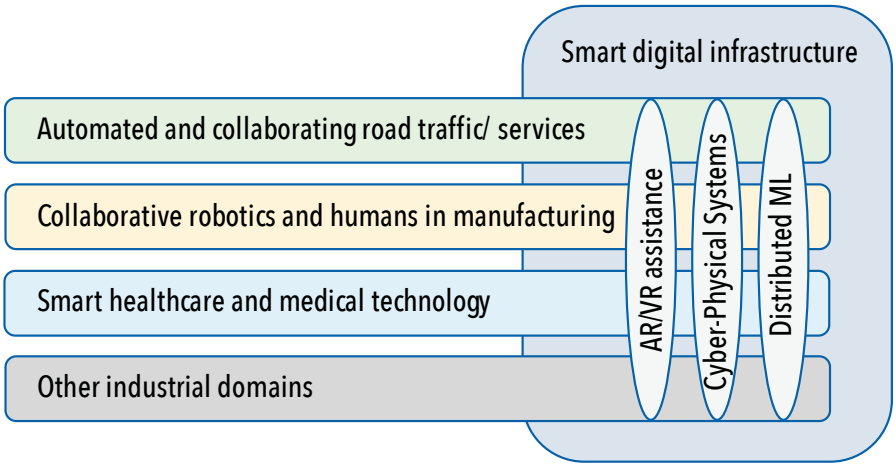


Figure 2. Various application domains, use-cases of cross-domain relevance, and interactions with a digital infrastructure (providing edge computing, communication, and other capabilities such as positioning)

BEYOND "BEST EFFORT"

More advanced use cases exist that could benefit from edge computing but where different challenges exist today, including both technical/scientific as well as related to business models. The commercial viability of these opportunities thus remains uncertain, and the TECoSA center has identified three types of use cases as particularly interesting where more research is needed. These use cases all demand more localized computing power, providing incentives for edge computing. The use cases are also relevant in several application domains, driving setups in which a digitalized edge computing-based infrastructure promises added value (see Figure 2). In manufacturing, for example, many ongoing field tests involve using private 5G networks and edge computing, representing such digitalized infrastructures. We first elaborate on these use cases and then discuss approaches to address them.

- **Use case 1: Mobile AR/VR/Cognitive Assistance**: The first use case concerns the advantages of future edge computing deployments in human-in-the-loop applications like virtual reality (VR) and augmented reality (AR). These are closed-loop systems where different "status" information is conveyed upstream to the point of computation (that is, the cloudlet). The provided status information is used for generating feedback at the backend, which is then transmitted back to the application client. AR and VR applications are generally characterized by (1) high data rate requirements upstream and/or downstream, (2) complex backend processing taking place at the cloudlet, and (3) quality-of-experience (QoE) of the application is directly related to the responsiveness of the entire loop (upstream communication, compute, and downstream communication). Subtle differences exist concerning the workloads and QoE requirements for AR systems versus VR systems, where VR systems require higher bandwidths in the downlink. Generally speaking, the latency requirements are also higher due to the level of immersion. The specific challenges for both application types relate to the following:
  1. **Efficient application support:** Due to the interplay between communication and compute elements over the offloading loop, many trade-offs exist to manage end-to-end delays at runtime dynamically. These trade-offs are largely unexplored, particularly about quality-of-experience implications in the short- and long-term. Managing end-to-end delays with respect to QoE over a heterogeneous set of active AR/VR applications is a further challenge, as is the question of optimal placement of the compute backend or efficient and reliable mobility support for such applications. To a large extent, the efficient support of such applications also hinges on the degree of control the application will be able to execute over the mobile network. In the past, mobile network systems have offered only very limited APIs (application

programming interfaces) as QoE requirements for voice, video, or web applications have been similar and hence easy to manage. However, for AR or VR applications, more complex trade-offs are likely to be only known to the application at runtime. Hence, a more powerful API for resource control enables a significantly more efficient operation.

2. **Scalable life-cycle support of applications and end system acceptance:** While several SDKs exist for AR and VR systems, devising a new application over programming, deployment, and updates is highly complex and requires deep software engineering and platform knowledge. This contrasts with the corresponding life-cycle support of smartphone apps of various ecosystems currently in the market. From the perspective of the supply side of future AR/VR applications, a significant simplification of the life-cycle support is likely to be established over the following years. Due to the above limitations, AR technology commercialization has been limited. Advanced designs, combined with a changing sentiment in the group of early adopters, might lead over the following years to a breakthrough in these applications. A scalable provisioning of backend compute capabilities via edge computing paired with near-ubiquitous mobile network access will undoubtedly lift the technological bottlenecks for widespread adoption.

▪ **Use case 2: Cyber-physical systems (CPSs):** CPSs represent the "integration of computation, networking, and physical processes." While CPSs have been around since the 1970s with the integration of microprocessors with physical systems, these systems now see unprecedented potential in their capabilities (Thompson and Reimann 2018). Representative examples include automated vehicles and future manufacturing systems. In such CPSs, additional sensors, communications, and collaboration can enhance context awareness and planning. The role of edge computing comes into play to provide the needed computational and analytics support, providing the potential for handling large amounts of data for real-time applications and supporting CPS collaboration. TECoSA has identified many applications in domains such as those depicted in Fig. 2, supporting enhanced quality and new functionalities, for example, by ensuring that the right assembly tools are used for the right parts in a manufacturing process. For CPSs, we identify the following challenges:

1. **Holistic management of computing and communication resources:** Industrial applications have demanding requirements on real-time (predictable and short enough) latencies, availability, and error detection and handling. This requires novel end-to-end resource management capabilities, including exploiting an interplay between applications and infrastructure and considering energy consumption as a key metric. With such considerations, edge computing promises to minimize/reduce the overall energy consumption of applications.
2. **Trustworthy applications based on edge computing:** As already introduced, trustworthiness has evolved to become an umbrella term. Given the evolution of CPS, most of the trustworthiness properties will be relevant for future CPS. Incorporating edge computing into future CPS poses new challenges, given new failure modes and cyber-security risks (vulnerabilities) of edge computing-based infrastructures and applications. The dependencies and trade-offs between trustworthiness properties require specific attention, especially for open and collaborative CPS with potential conflicts between cyber-security, safety, availability, and data sharing. The uncertainty involved in such open further CPS requires run-time risk assessment and handling/adaptation to balance safety and availability/performance appropriately. Certification and re-certification of (evolving and adapting) edge-based CPS also represent an open challenge.
3. **Collaborating systems and scalability**: Collaborating systems, often referred to as systems of systems (SoS), lack a central authority responsible for systems integration and where the constituent units evolve independently (for example, in the domain of roads, actors such as vehicles and the physical and digital infrastructures of the roads) (Maier 1998). This leads to challenges regarding the overall design and responsibilities of such SoS and strongly relates to the business model(s) and liability if something goes wrong. The "intelligent transport systems" example has shown the difficulty of establishing such SoS. We believe that the introduction of 5G and beyond as a digital infrastructure, with its provision for low latency and quality of service, may help to create the momentum needed to establish the required models for collaboration.

▪ **Use Case 3: Distributed ML:** Machine learning (ML) is widely considered an efficient tool for optimization, prediction, and classification tasks found in various industrial and consumer applications, among others in AR/VR systems (UC1) and CPSs (UC2). The use of ML in these systems could be limited to applying the pre-trained model for performing a certain task on data received from end devices, referred to as inference. More generally, it can entail periodic training of the model to adapt it to changing environmental conditions. The use of ML for inference usually involves upstream traffic and may involve downstream traffic also if/when the inference leads to decisions that, in turn, affect devices. Training of a model may also involve downstream traffic if the updated model is to be distributed to end-user equipment. ML algorithms are often represented as execution graphs and can be deployed on various devices spanning the edge-to-cloud continuum. Such distribution of ML primitives enables capabilities previously unattainable in energy and computationally-constrained environments. For example, by placing parts of the execution graph having challenging real-time requirements and low computation complexity on end-user equipment and computationally intensive parts in the edge cloud, one can obtain low-latency ML algorithms with limited computational resources. At the same time, distributed ML comes with a variety of challenges, in particular:

1. **Interoperability:** Interfaces for interconnection are needed to enable interoperability between components from different vendors and to make system integration more cost-efficient. Since ML algorithm development is in its early stages, it is challenging to establish interfaces that will last years or decades.
2. **Systems architecting:** Systems architecting aspects and algorithmic issues will become key in ever more complex installations. It needs to be clarified how to formulate architectural and design principles for complex, ML-enabled systems to ensure functional and non-functional requirements and simultaneously allow for efficient life-cycle management. Sustainability in terms of energy consumption and the environmental footprint of the computing and communications infrastructure needed for ML integration is a closely related issue.

3. **Robustness and cybersecurity.** Robustness to adversarial environments and the lack of privacy guarantees could also hinder the wide-scale adoption of ML-enabled systems. ML algorithms are vulnerable to adversarial inputs, for example minor perturbation of the data, unnoticeable manipulations of algorithm parameters, and trained ML models may also reveal confidential information about the data set used for creating them (Ramakrishna 2022). These issues related to trustworthiness remain to be solved.

The TECoSA center approach to address these challenges: Successful research centers have been reported to exhibit characteristics including collaborative multidisciplinary research involving multiple domains, use of testbeds/demonstrators, and having a strong connection to education (Patterson 2014). We agree that these characteristics are important. TECoSA has emphasized creating a knowledge ecosystem with the involved stakeholders and aims to develop testbeds as experimental and open infrastructures in automated and connected road traffic and collaborative robotics in the coming period. These testbeds will be used to support collaborative research and education. A critical aspect of the testbeds is stimulating the interplay between applications – potentially involving all the mentioned use cases – and digital infrastructures (Figure 2). This interplay corresponds to interactions between different research teams and organizations/companies, places the focus on platforms and services (the interfaces between applications and the infrastructures), and has the potential to be used in education and stimulate open debate on the socio-technical implications.

CONCLUSIONS AND FUTURE WORK (CONVERGENCE OF THE USE CASES AND MORE)

We have discussed trends and drivers related to edge computing as a new computing tier overcoming limitations of and complementing the cloud and resource-constrained embedded systems. The multitude of concepts such as MEC, cloudlets, fog computing, "near/far/nano/enterprise edge," and "distributed cloud" (Heinen 2021), while partly confusing, is natural considering that we are in the early stages of edge computing with an ongoing market positioning. In our analysis, edge computing provided by mobile network operators will be the initial dominating form of this new computing paradigm for the coming decade. In this sense, edge computing adds computational resources that complement the existing capabilities of devices (embedded systems) and the cloud, belonging to a new tier of a digitalized infrastructure. Regulatory aspects, bandwidth saving, and soft real-time interactions such as gaming will likely drive the first wave of commercial edge computing offerings. We highlight that in these cases, "best effort" SLAs between the service provider and customer are sufficient for successful commercialization.

We have also discussed more advanced use cases, including AR/VR/Cognitive Assistance, CPSs, and distributed ML, and corresponding challenges that require further research. The three presented use cases will, in many ways, be part of the same system, for example, with humans in the loop (such as "cobots"—humans and robots collaborating) in the context of CPS and with data gathering and distributed machine learning taking place in parallel with the other use cases.

In addressing these use cases and challenges, the identified key role of a university-led research center is to maintain and grow a knowledge ecosystem to support innovation, research, and education in trustworthy edge-based CPS and to develop corresponding technological foundations and methodologies. ■

REFERENCES

- Abbas N., Y. Zhang, A. Taherkordi, and T. Skeie. 2018. *Mobile Edge Computing: A Survey. IEEE Internet of Things Journal* 5 (1): 450–465.
- AI HLEG 2021. *High-Level Expert Group on AI of European Commission. Overview of deliverables from the AI HLEG*. Web page reference: https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai (accessed 2022-08-17).
- Balan R., J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H. Yang. 2002. *The case for cyber foraging. Proc. 10th workshop on ACM SIGOPS European workshop (EW 10)*. Association for Computing Machinery, New York, US-NY: 87–92. https://doi.org/10.1145/1133373.1133390
- Bonomi F., R. Milito, J. Zhu, and S. Addepalli. 2012. *Fog Computing and Its Role in the Internet of Things. Proceedings 1st Edition MCC Workshop on Mobile Cloud Computing* (Helsinki, Finland) (MCC '12). ACM, New York, US-NY: 13–16. https://doi.org/10.1145/2342509.2342513
- Duranton M., M. Malms, and M. Ostasz. 2021. *The continuum of computing*. Hipeac Vision 2021. https://doi.org/10.5281/zenodo.4719341
- EC Industry 5.0. 2022. Web page reference: https://research-and-innovation.ec.europa.eu/research-area/industry/industry-50_en (accessed 2022-08-17).
- INCOSE. 2021. *Systems Engineering Vision 2035*. https://www.incose.org/about-systems-engineering/se-vision-2035
- Heijnen A. et al. 2021. *IoT and Edge Computing: opportunities for Europe*. Report by the NGIoT project (Next Generation Internet of Things). Retrieved from https://www.ngiot.eu/
- Maier M. 1998. Architecting principles for systems-of-systems. *Systems Engineering Journal*. 1 (4): 267–284, 1998.
- Patterson D. 2014. *How to build a bad research center.* Commun. ACM 57(3): 33–36. https://doi.org/10.1145/2566969
- Satyanarayanan M. 2017. The Emergence of Edge Computing. *IEEE Computer* 50 (1).
- Törngren M. 2021. *Cyber-physical systems have far-reaching implications*. Hipeac Vision 2021. https://doi.org/10.5281/zenodo.4710500
- TECoSA, 2022. Web page reference: https://www.tecosa.center.kth.se/ (accessed 2022-08-17).
- Ramakrishna R. and G. Dán. 2022. *Inferring Class-Label Distribution in Federated Learning.* ACM Workshop on Artificial Intelligence and Security (AISec).
- Ramli R. and M. Törngren. 2022. *Towards an Architectural Framework and Method for Realizing Trustworthy Complex Cyber-Physical Systems.* Joint Proceedings of RCIS 2022 Workshops and Research Projects Track, Barcelona, ES: May 17-20, 2022.
- Thompson H. and M. Reimann. 2018. *Platforms4CPS Key Outcomes and Recommendations*. https://www. platforms4cps.eu
- Törngren M., H. Thompson, E. Herzog, R. Inam, J. Gross, and G. Dán. 2021. *Industrial Edge-based Cyber-Physical Systems – application needs and concerns for realization*. Proc. of ACM Symp. on Edge Computing Workshop on Trustworthy Edge Computing.
- Varghese B. et al. 2021. *Revisiting the Arguments for Edge Computing Research*. IEEE Internet Computing, doi: 10.1109/MIC.2021.3093924.

ABOUT THE AUTHORS

James Gross received his PhD degree from TU Berlin in 2006. Since November 2012, he has been with the Electrical Engineering and Computer Science School, KTH Royal Institute of Technology, Stockholm, where he is professor for machine-to-machine communications. At KTH, James is currently associate director of the Digital Futures Research Center, as well as co-director of the VINNOVA competence center on Trustworthy Edge Computing Systems and Applications (TECoSA). His research interests are in mobile systems and networks. He has authored over 150 (peer-reviewed) papers in international journals and conferences.

Martin Törngren is a professor in embedded control systems at the Mechatronics division at KTH since 2002, and with PhD in machine design/mechatronics also from KTH. Prior to becoming a professor at KTH he co-founded the company Fengco Real-time Control AB, specializing in advanced tools for developers of embedded control systems and related consultancy, and also did a postdoc period at the EU-JRC, Institute for Systems, Informatics and Safety, Ispra, Italy. He has particular interest in trustworthiness and dependability of cyber-physical systems and their design methodologies. Networking and multidisciplinary research have been characteristic throughout his career. He is the principal initiator of the Innovative Centre for Embedded Systems (www.ices.kth.se), launched in 2008 (and served as its director until 2020) – with close interactions with the Swedish chapter of INCOSE. He is the director of the TECoSA Swedish national competence center on Trustworthy Edge Computing Systems and Applications (initiated in 2020). In 2011/2012 he was visiting scholar at UC Berkeley (2011/12) and in 2018 at Stevens Institute of Technology (Hoboken, New Jersey, 2 months).

György Dán (M'07, SM'17) is professor of teletraffic systems at KTH Royal Institute of Technology, Stockholm, Sweden. He received the MSc in computer engineering from the Budapest University of Technology and Economics, Hungary in 1999, the MSc in business administration from the Corvinus University of Budapest, Hungary in 2003, and the PhD in telecommunications from KTH in 2006. He worked as a consultant in the field of access networks, streaming media, and videoconferencing 1999-2001. He was a visiting researcher at the Swedish Institute of Computer Science in 2008, a Fulbright research scholar at University of Illinois at Urbana-Champaign in 2012-2013, and an invited professor at EPFL in 2014-2015. He served as area editor of Computer Communications 2014-2021, and has been editor of IEEE Transactions on Mobile Computing since 2019. His research interests include the design and analysis of content management and computing systems, game theoretical models of networked systems, and cyber-physical system security and resilience.

David Broman is a professor at the Department of Computer Science, KTH Royal Institute of Technology and an associate director faculty for digital futures. He received his PhD in computer science in 2010 from Linköping University, Sweden. Between 2012 and 2014, he was a visiting scholar at the University of California, Berkeley, where he also was employed as a part-time researcher until 2016. His research focuses on the intersection of (i) programming languages and compilers, (ii) real-time and cyber-physical systems, and (iii) probabilistic machine learning. He has worked several years within the software industry, co-founded the EOOLT workshop series, and is a member of IFIP WG 2.4, Modelica Association, a senior member of IEEE, and a board member of Forskning och Framsteg.

Iolanda Leite is an associate professor at the School of Electrical Engineering and Computer Science at KTH Royal Institute of Technology. She holds a PhD in information systems and computer engineering from IST, University of Lisbon. Prior to joining KTH, she had postdoctoral appointments at Yale University and Disney Research. Her goal is to develop social robots that can perceive, learn from, and respond appropriately to people in real-world situations, allowing for truly efficient and engaging long-term interactions with people.

Raksha Ramakrishna received the BE degree in electronics and communications engineering from the Rashtreeya Vidyalaya College of Engineering, Bangalore, India, in 2014, and the MS and PhD degrees in electrical engineering from Arizona State University, in 2017 and 2020, respectively. She is currently a postdoctoral researcher with the Division of Network and System Technology, KTH Royal Institute of Technology, Stockholm, Sweden. Her research interests include the domains of statistical signal processing, data analytics for power systems, and security and privacy in federated machine learning systems.

Rebecca Stower is a postdoctoral researcher at KTH Royal Institute of Technology. She holds a PhD in psychology from Jacobs University, Bremen, Germany and a BSc in psychology from the University of Queensland in Australia. She is passionate about the intersection of psychology and technology and how psychological research methods can be applied to digital industries. Dr. Stower is working on swarm robotics and is interested in the conceptualization of social intelligence in robots and the design of social robot behavior.

Professor Haydn Thompson, BSc, PhD CEng has over 35 years experience working in a mixture of senior industrial research and development roles in flight control systems, space programmes and radar signal processing applications for leading companies. From 1993- Feb. 2013 he was the program manager of the Rolls-Royce Control and Systems University Technology Centre. He is managing director and founder of the THHINK Group of companies. He is recognised and used by the European Commission as an expert in many fields, CPS, IIoT, AI, aerospace, automotive, autonomous vehicles, smart agriculture, advanced electronics, and more. Dr. Thompson is a consultant to a range of companies and government bodies. He defines strategic technology roadmaps across Europe and for companies such as Rolls-Royce. He has over 100 publications, has written two books and contributed to several others.

DECEMBER 2022
VOLUME 25/ ISSUE 4

Creating Value with MBSE in the High-Tech Equipment Industry

Teun Hendriks, teun.hendriks@tno.nl; **Joris van den Aker,** joris.vandenaker@tno.nl; **Wouter Tabingh Suermondt,** wouter.tabinghsuermondt@tno.nl; and **Jacco Wesselius,** jacco.wesselius@tno.nl

■ **ABSTRACT**
The Netherlands has a strong presence in the high-tech equipment industry sector with world-wide renowned organizations. Systems engineering is a key capability that is well-established in this sector. The industry now sees model-based systems engineering (MBSE) as indispensable to bringing systems engineering capabilities to the next level. Despite this, MBSE is not a fully established practice in this sector as in other industries. ESI has initiated a study to understand the background of the sector's interest in MBSE, the challenges to address with MBSE, experiences with, and fit of current MBSE methodologies versus the characteristics of this sector. This article reports on the results of this study. It highlights innovation in MBSE to address the needs and characteristics of the high-tech equipment industry.

■ **KEYWORDS:** MBSE, high-tech equipment industry, digital transformation, brownfield development, systems engineering

SYSTEMS ENGINEERING FOR THE HIGH-TECH EQUIPMENT INDUSTRY

The Netherlands has a strong presence in the high-tech equipment industry sector with world-wide renowned organizations. Innovations now take these systems (for example, nanometer-accurate lithography systems, angstrom resolution electron microscopes, minimally invasive medical equipment, commercial printing equipment, and advanced warehousing systems) towards unprecedented levels of features and functions, increasing complexity every day. Consequently, R&D organizations have grown, with (business-) critical issues needing to be addressed in the ecosystems of partners (supply chain partners, field service partners, innovation partners).

Some characteristics of this industry sector are the annual production quantities, typically 10s to 1000s, while offering an extensive product portfolio with product variants, options, and sometimes customer-specific features. This diversity makes almost any delivered product unique. The organizations operate in commercial markets; some industries experience strong competition, while others are unique suppliers or market leaders. To deal with this business context, these industries apply an evolutionary way of working for their R&D. Solutions are developed incrementally, using the previous product as a baseline:

- The industry has established product families supported by product platforms to deal with the variety. Product innovations should comply with platform thinking.
- The products add market value to system qualities such as performance, throughput, or uptime, which requires a robust system-level architecting approach.
- The industry has widely adopted an agile way of working, which fits well with the evolutionary way of working.
- The products and solutions offered to the market have a long lifetime (often well over twenty years), resulting in a large installed base. It has a key business value because it shows its leading position in the market and provides opportunities for upgrade and replacement sales for a service business.
- The industries are traditionally strong physics and electro-mechanics oriented. However, in the past decades, they have experienced an ever-increasing effort in software development.
- Further, the solutions used to be standalone applications but require integration into a larger system of systems.
- Design knowledge is captured in documents, which are sometimes generated from databases.

Many R&D employees are employed for a lengthy period; sometimes, they work their whole professional career at a single company. They have in-depth knowledge about current developments as well as the installed base. Although this knowledge is essential, keeping it up to date is expensive.

The technical and business complexity forces these industries to grow, which means an influx of new people — who do not have the complete design history in their minds. Also, the retirement of senior employees working on crucial expertise is a source of loss of know-how. A solution for retaining critical know-how is vital to maintain its market position.

The above drivers cause this industry sector to have an increasing interest in replacing its classical systems engineering approach with a (more) model-based approach. ESI and the sector hence started a study (Wesselius, van den Aker, et al. 2022) to see what MBSE can bring to the sector, the challenges to address with MBSE, experiences with, and to gauge the fit of current MBSE methodologies versus the characteristics of the high-tech equipment industry.

MBSE VERSUS MODELS IN SYSTEMS ENGINEERING

Systems engineering is both about technical engineering aspects and engineering management: ensuring that all engineering is done for system effectiveness in a controlled way. Models abound in systems engineering these days. Are all system engineers doing MBSE, at least to some extent? The answer is "no." MBSE is not about "using models while systems engineering;" instead, the fundamental MBSE tenet is that models become the primary asset, the authoritative source of information for everyone. In MBSE, models are not add-ons to (authoritative) documents. Instead, models replace those documents. If documents are needed, they are generated from the models. In case of doubt, the models are authoritative and overrule the documents.

In the Dutch high-tech equipment industry, models abound, but they are not yet the authoritative source of information. Most systems engineering related models have a single purpose and are disconnected. The sector, however, looks to MBSE to improve its systems engineering practice. A Sandia report (Carroll and Malins 2016) and the MBSE Survey of Stevens Institute of Technology (Cloutier and Bone 2015) show positive results in terms of cost savings and quality improvements for organizations that adopted MBSE. To introduce MBSE successfully, the Sandia report also names several prerequisites that need to be in place: related to systems engineering processes, training of system engineers, and investment in full-scale MBSE tools. Also, the organization needs to commit to processes, resources, people, and infrastructure to support model management throughout the system design life-cycle.

Both these reports strongly focus on the aerospace and defense industry. Commercial, high-tech equipment manufacturers have a different business context, influencing the role and value of systems engineering and MBSE. Therefore, the MBSE experiences described in these reports cannot be carried over one-on-one to the commercial sector. Understood must be which are the main drivers for MBSE in this sector.

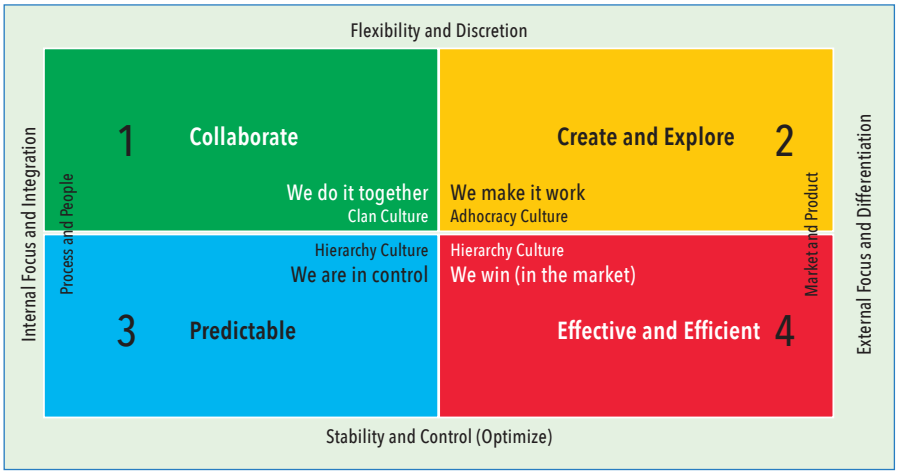


Figure 1. Quadrants to map motivations for MBSE introduction (inspired by Quinn and Cameron and Insights Discovery)

DRIVERS FOR MBSE ADOPTION IN THE HIGH-TECH EQUIPMENT INDUSTRY

Motivations for MBSE have typically been expressed in organization-specific terminology. To analyze these drivers uniformly, we used a four-quadrant matrix inspired by the competing values culture model (Cameron en Quinn 2006, Quinn Association 2022) and the insights discovery color coding (The Insights Group Limited 2021).

Each quadrant captures a class of MBSE drivers which may be related to the culture and business strategies of the organizations. These quadrants (see Figure 1) are created by combining two axes: (1) internal focus and integration or external focus and differentiation, (2) stability and control or flexibility and discretion. The first axis looks at how an organization operates in the market: based on an internal focus and integration or an external focus and differentiation versus the competition. The second axis looks at how an organization approaches the effectiveness of its workforce and work processes: via control and stability (and strict processes) or flexibility and discretion. This yields four quadrants as follows:

1. **Collaborate [Internal Focus and Integration | Flexibility and Discretion]**
   This quadrant focuses on drivers for collaboration aspects. Motivators are positioned as follows:
   - Having independent teams doing concurrent engineering, effective collaboration ensures that their design deliverables integrate into a product.
   - The need to spread system knowledge throughout the organization.
   - The desire to enhance collaboration across disciplines in self-managing teams.
2. **Create and Explore [External Focus and Differentiation | Flexibility and Discretion]**
   This quadrant focuses on drivers for conceptualization, exploration, and trade studies. Motivators are positioned as follows:
   - Exploring design options and simulating their consequences at a system level.
   - Performing trade-space analysis and make trade-off decisions, taking the system-wide scope into account.
3. **Predictable [Internal Focus and Integration | Stability and Control]**
   This quadrant focuses on drivers for keeping control and consistency. Motivators are positioned as follows:
   - Assuring that all requirements are met and verified.
   - Standardizing the way of working throughout the organization.
4. **Effective and Efficient [External Focus and Differentiation | Stability and Control]**
   This quadrant focuses on drivers for bringing products fast to market. Motivators are positioned as follows:
   - Being able to quickly compose customer-specific systems from platforms of pre-designed and pre-released components.
   - Reduce the time from ordering to delivering a system to the customer.
   - Being able to ensure properties of multiple system configurations.

MAPPING MBSE DRIVERS TO THE NEEDS OF THE HIGH-TECH EQUIPMENT INDUSTRY

In the MBSE study of ESI and Partners, (Wesselius et al. 2022), the high-tech equipment industry sector put forward four main drivers to explore the introduction of MBSE:

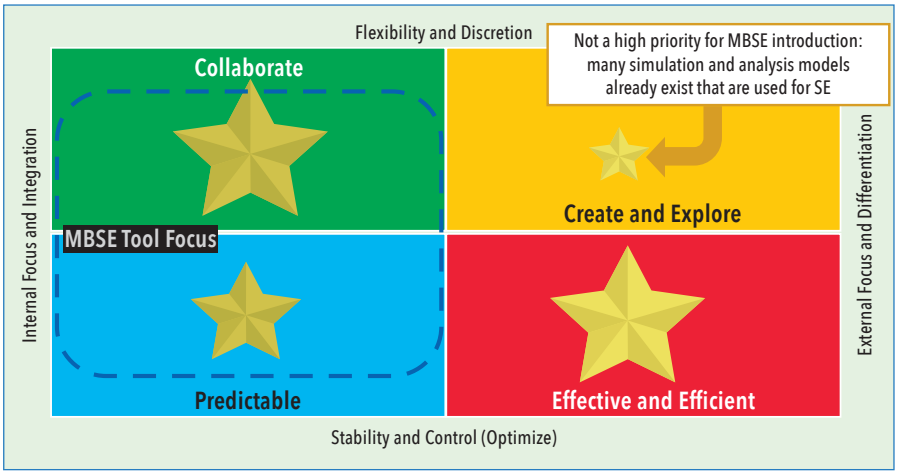


Figure 2. Relative importance of motivations for MBSE in the Dutch high-tech industry

1. Enhancing cooperation and knowledge exchange (quadrant 1).
2. Consolidating knowledge from legacy systems for new generations of engineers (quadrant 1).
3. Leveraging platforms to accelerate system/solution design and delivery (quadrant 4).
4. Assuring the properties/qualities of system variants shipped to customers (quadrant 4).

Figure 2 shows the ranking of MBSE motivations expressed by the Dutch ecosystem, overlaid on the focus of current MBSE tools: quadrant 3 (predictable) while supporting quadrant 1 (collaborate). quadrant 2 (create and explore) surprisingly scored significantly lower with the Dutch ecosystem. This could create the impression that their interests lay outside of using models to speed up innovation. This conclusion, however, is incorrect as they already use models and tools for this purpose. These (engineering) models are typically not connected and do not establish an "authoritative source of truth." However, they expect MBSE to enable smoother communication and knowledge exchange about those innovations, which is part of the drivers in quadrant 1 (collaborate).

Furthermore, all industries expect the core benefits of MBSE: the authoritative, consistent, and easily accessible system-wide information (quadrant 3). To them, this is the enabler that MBSE is expected to provide.

GREENFIELD VERSUS BROWNFIELD SYSTEMS ENGINEERING

The evolutionary way of working, that is the *brownfield* business context of the Dutch high-tech equipment industry, is important. "Brownfield" can be characterized by: (i) this year's system is an incremental, evolutionary innovation of last year's system; (ii) a substantial part of the business is related to legacy systems in their installed base (iii) therefore, a large portion of their R&D capacity is used to sustain their installed base, including the delivery of upgrades to systems that are twenty to thirty years old. The brownfield nature of this sector's business context gives specific challenges that also drive their motivations for exploring MBSE:

- Organizations must efficiently and effectively sustain their installed base for long periods, so they need teams knowledgeable of those systems. Developing system-level domain knowledge for new engineers requires considerable time and a steep learning curve. Therefore, consolidating systems engineering knowledge of legacy systems and making it easily accessible to new engineers is crucial for their business. This is a motivator for quadrant 1.
- As they have delivered systems in many configurations and generations, consolidating the knowledge required to sustain the installed base is complex. MBSE is hoped to bring an innovative way to structure the required systems engineering information and assure consistency for systems in the field.
- As next year's system is, in most cases, an evolutionary innovation from this year's system, in-depth knowledge about this year's system is crucial for successful innovation. This also puts a strong focus on knowledge consolidation and knowledge exchange.

What complicates the introduction of MBSE in such *brownfield* organizations is that there are no authoritative models for current or past systems. Instead, large sets of documents are available, typically neither fully up-to-date nor entirely consistent. Next to these documents, the knowledge is typically captured in organizations; processes; databases; people (the experienced "local heroes"). Introducing MBSE requires leap-frogging this chasm. A complex transition path is needed in which hybrid approaches (document-based and model-based systems engineering) will co-exist.

POSITIONING METHODS AND TOOLS ON INDUSTRY MBSE MOTIVATORS

The MBSE study of ESI and partners (Wesselius et al. 2022) was conducted as a series of workshops with the high-tech equipment industry, where in turn, every partner explained their MBSE ongoing work as the basis for discussion. In addition, interviews with leading MBSE tool vendors were conducted to explore the capabilities and strategies of contemporary MBSE methods and tools. Mapping our quadrant observations leads to the following overview (Figure 3).

- Quadrant 1: MBSE tools to collaborate [Green = Internal | Flexible]
  Several tools provide advanced collaboration environments supporting distributed teams. In this quadrant, we did not find methods or tools that actively

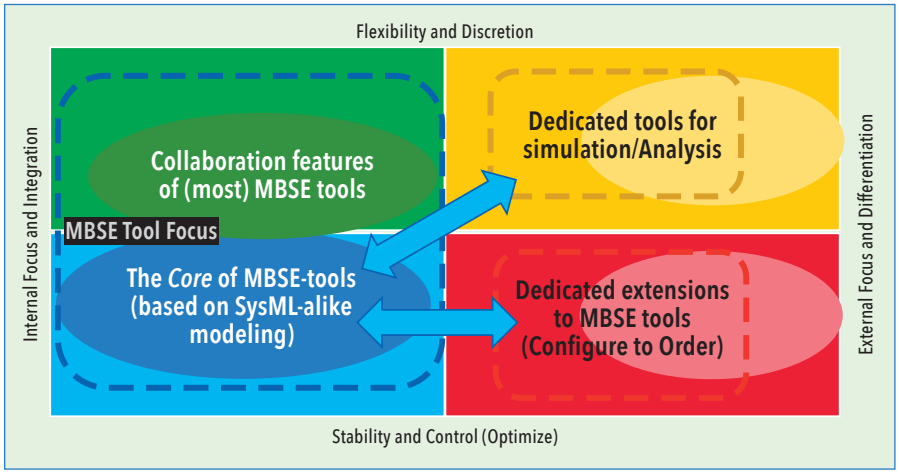


Figure 3. Positioning methods and tools relative to MBSE motivations.

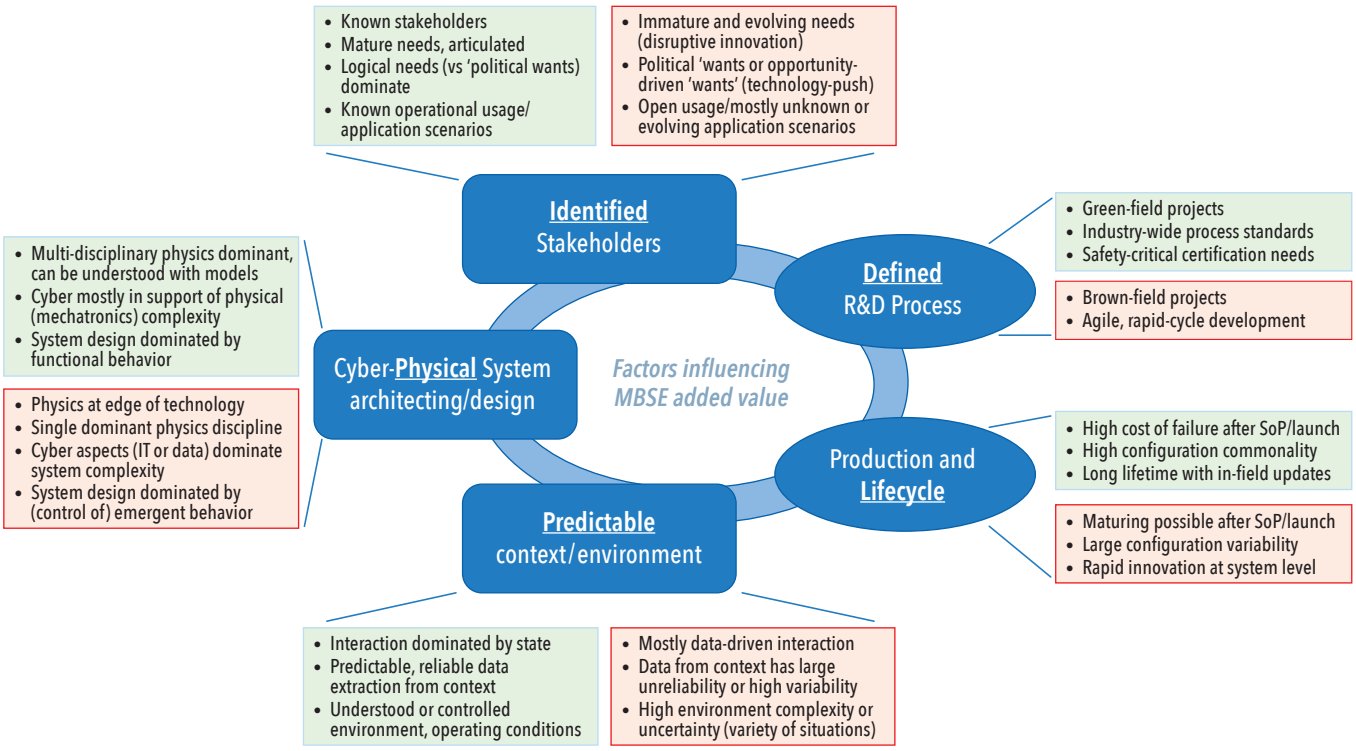


Figure 4. Factors influencing the added value of MBSE (positive factors in green, negative factors in red)

support working in an evolutionary brownfield environment and capturing legacy and current systems' design rationale or intent.

- Quadrant 2: MBSE tools to create and explore [Yellow = External |Flexible]
  Modeling is a key aspect of design exploration and analysis. Many dedicated methods, tools, and techniques are available to analyze and simulate specific system aspects. Strong interfaces between dedicated tools and MBSE-core tools are needed at a system level to maintain a consistent, authoritative source of systems engineering information.
- Quadrant 3: Predictable core of MBSE-tools [Blue = Internal | Control]
  Most MBSE tools focus on quadrant three by providing SysML-alike modeling languages and techniques to create RFLP-alike structures: In this quadrant, MBSE tools primarily offer techniques for function decomposition and allocation but less to system qualities and system properties (a next-generation SysML V2 (OMG 2022) is in the works to address these issues).
- Quadrant 4: Tools to be efficient and effective in the market [Red = External | Control]
  Several MBSE vendors have the strategy to deliver complete and integrated tool suites, including product lifecycle management and manufacturing. Others focus on MBSE-core functionality and have the strategy to connect to dedicated PLM, simulation, and analysis tools through the open interfaces provided by these.

During the study, several more generic observations and attention points were identified. Firstly, modeling is a crucial aspect of design exploration and analysis (quadrant 2). This needs strong interfaces between simulation and analysis tools and the MBSE-core tools (quadrant 3) to assure consistency, cohesion, and authoritativeness to support collaboration and concurrent engineering (quadrant 1). Secondly, suppose the models become the authoritative source of systems engineering information. In that case, they should also capture design rationale and intent (today, architects spend much time talking with design teams to convey these). Thirdly, MBSE methods and tools are needed to create models from legacy design artifacts (documents, Excel sheets, Visio diagrams, CAD files) in a brownfield environment. Lastly, given the sector's MBSE motivations, integrating key aspects of platform-thinking and product line engineering into the MBSE-core methods and tools is needed for MBSE to be effective in this sector, including reasoning about system variants/diversity and across legacy.

CONSIDERATIONS FOR INTRODUCING MBSE IN THE HIGH-TECH EQUIPMENT INDUSTRY

MBSE has been first applied in aero-space and defense in long-running "engineer-to-order" type projects. From then on, other domains have adopted or experimented with MBSE. How do these organizations and experiments relate to the characteristics of the high-tech equipment industry?

Factors Influencing the Added Value of MBSE Introduction

To support the high-tech equipment industry in introducing MBSE, influencing factors were identified for the likelihood of added value for MBSE over "just" doing systems engineering. These factors were inferred from success reports of the application of MBSE in various domains and complemented with insight into the nature and strengths of MBSE methods and tools. Figure 4 presents an overview of these (generic) influencing factors.

As shown in Figure 4 (on the left), the nature of the systems may have a considerable influence on whether MBSE could add significant value. MBSE thrives when the design challenge is balancing multi-disciplinary physics (hence the underlined physical). When cyber aspects or management of emergent behavior dominates complexity, MBSE is less suitable for managing such aspects.

Stakeholders should be known and able to articulate their needs (hence the underlined identified in Figure 4). MBSE needs well-articulated system requirements, which form the basis of traceability into the

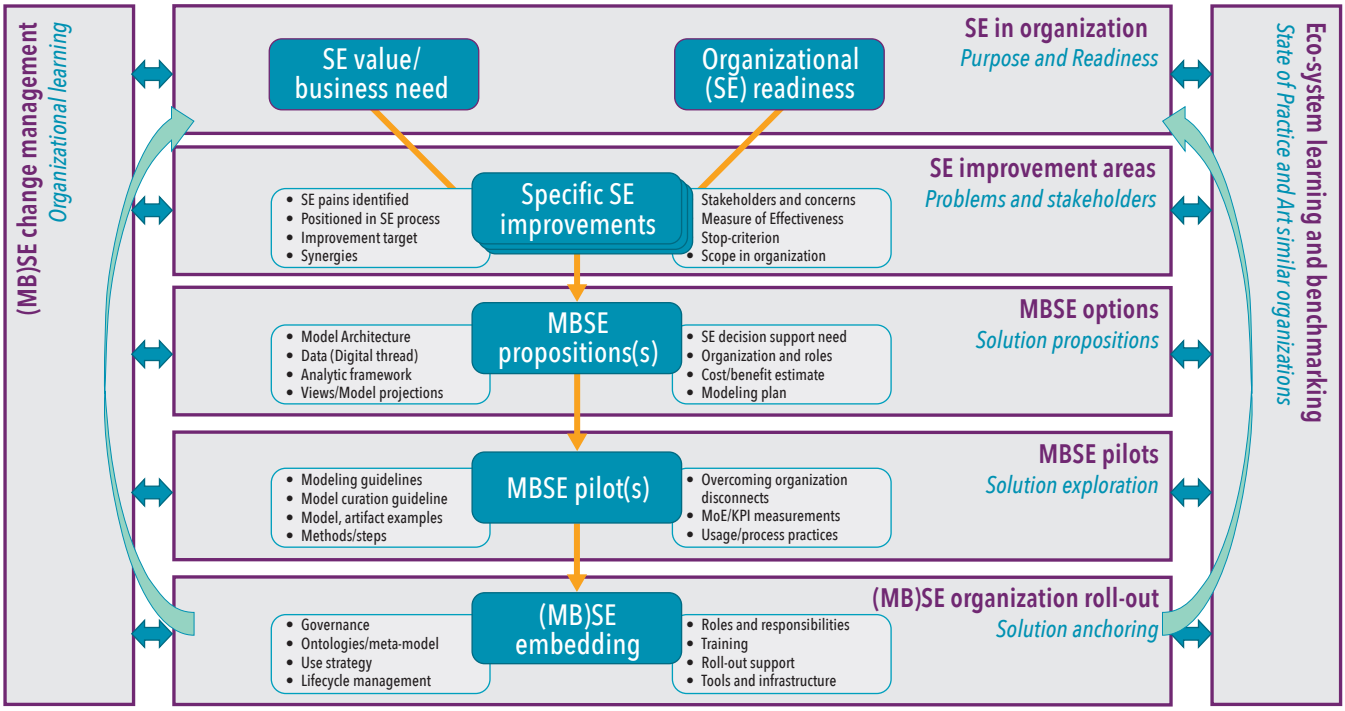


Figure 5. Reasoning line to guide the introduction of MBSE in an organization

design and verification of its decomposition and properties.

A system's context (Figure 4 bottom) should be understood and captured in models (hence the underlined predictable). When this context sees considerable uncertainty or mostly data-driven interaction, then this presents difficulties for (typically function-oriented) MBSE approaches. Specialized approaches are then called for (for example, capturing driving scenarios in automated driving).

Concerning the R&D process (Figure 4, right and top), greenfield projects allow the complete use of MBSE over the full design scope (emphasized by the underlined defined). In contrast, brownfield development for incremental design upgrades faces missing models and lost know-how. Recreating these for MBSE incurs high R&D overhead and long time-to-value.

Finally, concerning production and lifecycle (Figure 4, right and bottom), MBSE is particularly suited to minimize unacceptable risks when a high cost of failure could occur after start-of-production (SoP) (indicated by the underlined lifecycle) as is typically the case with safety-critical systems, such as road vehicles or commercial aircraft. For products that can be launched quickly as minimally viable products, MBSE overhead may be too much.

Most MBSE methods and tools implement a variant of the requirements-functional-logical-physical (RFLP) approach. This approach has been well-suited for particular domains and organizations. Figure 4 provides contrasting factors to consider where MBSE could add the most value or where the value could be less significant than just doing Systems Engineering with (disconnected) models.

A Reasoning Line for the Introduction of MBSE in the High-Tech Equipment Industry

The introduction of MBSE, as part of an organization's digital transformation, requires "an integrated digital approach that uses authoritative sources of system data and models as a continuum across disciplines to support lifecycle activities from concept through disposal" (Department of Defense – Office of the Deputy Assistant Secretary of Defense for Systems Engineering 2018). Besides introducing tools and methods, much attention and effort must be spent on creating organizational capabilities. To achieve the full benefits of digitizing (systems) engineering, just introducing MBSE tools and the associated methods does not suffice.

Thus, how to guide (and anchor) an introduction of MBSE in an organization? Achieving/perceiving value is far from trivial (Cloutier and Obiako, 2020). ESI created a reasoning line for this purpose (see Figure 5), which considers both technical and organizational aspects for MBSE to add value and needs to be embedded in a Systems Engineering way-of-working.

This reasoning line considers seven main viewpoints to guide, customize, and rationalize a value-add introduction to MBSE, and are annotated with relevant technical and organizational aspects to be considered.

The first viewpoint looks at the state of **systems engineering in the organization**: its systems engineering readiness and the business need for change. An organization must be capable of doing systems engineering before embarking on MBSE. Also, the intended change must be rooted in a clear business need/value improvement.

Secondly, **systems engineering improvement area**s should be understood and identified, the "problem space" (Noguchi, Minnichelli, and Wheaton 2019). What are the systems engineering pains? Which stakeholders experience these? What part of systems engineering needs to be improved, how much, when to stop (and when is good, good enough)? Also, which part(s) of the organization should be involved? Do the outcomes address the real needs of "outside" beneficiary stakeholders (sales, service, lifecycle)?

Thirdly, based on rationalized and scoped Systems Engineering improvements, a selection can be made where **MBSE options** could add value. For those, value propositions should be defined, including the systems engineering support targeted and how to achieve this with MBSE (which model(s) include analytic framework and data), but also how to organize and plan this, with a cost/benefit analysis. An overview of potential MBSE value options (benefits) is given in (McDermott et al. 2020).

Fourthly, **MBSE pilots** can explore these options' effectiveness and measure/assess benefits. Pilots also can refine methods and provide (input for) guidelines. Pilots may encounter organizational issues and disconnects exposed by a more formal way of working.

Fifthly, **(MB)SE organization roll-out** needs to ensure that the MBSE way of working is sustained by embedding it in the organization. This requires governance, ontologies/meta-models, tools, infrastructure, training, and definition of (new or changed) roles and responsibilities.

Two further supports for the activities include i) general **(MB)SE change management** to ensure that organizational learning and a change in Systems Engineering culture takes place, and ii) **ecosystem learning / benchmarking** to ensure lessons learned in similar organizations are incorporated, not duplicated.

The introduction of MBSE is a complex change process affecting many parts of an organization and how they collaborate. The reasoning line aims to guide MBSE introduction in the high-tech equipment industry but has wider applicability. Its purpose is a check for the rationalization of activities: not to be mistaken for a process (Parnas and Clements 1986). Having an articulated purpose and rationalization of MBSE activities is crucial to gain organizational support, achieving value, and for MBSE to be firmly embedded in Systems Engineering.

CONCLUSION AND OUTLOOK

In the high-tech equipment industry sector, invariably development of a new system starts from the design of the earlier generation of systems. These organizations typically support an installed base with systems up to fifteen to thirty years old. On the one hand, this installed base is a key business value; it shows their leading position in the market and provides opportunities for upgrade and replacement sales as a service business. On the other hand, supporting an aging installed base is costly and requires their teams to know past system generations. Furthermore, current development is increasingly software intensive and of increasing multi-disciplinary complexity.

Like the aerospace and defense industry, the high-tech equipment industry thus needs the consistency and completeness promised by MBSE ("authoritative source of truth") to battle this complexity and needs the cooperation/collaboration platforms that most MBSE tools offer. In addition, this industry also needs ways to leverage the value of MBSE in their business environment, characterized by dealing with brownfield R&D, evolutionary delivery requiring knowledge about past system generations, and leveraging investments in platforms.

These needs expose a gap in what MBSE currently offers and impose conditions for successful MBSE usage in the high-tech equipment industry (see also (Wesselius, MBSE for the High-Tech Equipment Industry - MBSE-study of ESI and partners 2021)):

- Solutions are needed for MBSE to pay off in a brownfield situation (legacy systems) Solutions are needed for MBSE to support platform-based R&D
- Solutions are needed for MBSE to support managing a large system of diversity
- Solutions are needed for MBSE to multi-disciplinarily model system qualities
- Solutions are needed to combine MBSE with agile R&D approaches

Most organizations now innovate by leveraging their network partners' added value (such as supply chain partners, innovation partners, and service partners). Thus, methods and tools should be able to share and use models as the authoritative source of systems engineering information *across* organizational boundaries, ecosystems, and supply chains. As parties are typically involved in multiple networks, they are expected to handle models originating from multiple MBSE methods and tools.

ESI and Dutch high-tech equipment industry partners collaborate to explore these needs together. This Dutch ecosystem has the unique advantage of not being competitors, yet still grappling with a similar type of systems engineering issues. The first phase of this study has identified needs, drivers, and influencing factors and created guidance as a reasoning line. Further in-depth workshops will elaborate on selected topics and deepen this reasoning line in the coming period.

ESI is also interested in organizing further exchanges and wider experience sharing, including across industry sectors, in elaborating and sharpening insights on adding value with the introduction of MBSE in organizations. ■

REFERENCES

- Cameron, K.S., and R.E. Quinn. 2006. *Diagnosing and Changing Organizational Culture – Based on the Competing Values Framework*. San Francisco CA: John Wiley & Sons.
- Carroll, E.R., and R.J. Malins. 2016. *Systematic Literature Review: How is Model-Based Systems Engineering Justified?* Albuquerque, New Mexico: Sandia National Laboratories. Accessed January 27, 2020. https://www.incose.org/docs/default-source/enchantment/161109-carrolled-howismodel-basedsystemsengineeringjustified-researchreport.pdf?sfvrsn=2&sfvrsn=2.
- Cloutier, R., and I. Obiako. 2020. "Model-Based Systems Engineering Adoption Trends 2009-2018." *Guide to the Systems Engineering Body of Knowledge (SEBoK)*. October 30. https://www.sebokwiki.org/w/images/sebokwiki-farm!w/a/a5/Guide_to_the_Systems_Engineering_Body_of_Knowledge_Part_8.pdf#page=28.
- Cloutier, R., and M. Bone. 2015. "MBSE Survey – Presented Januari 2015 INCOSE IW." January 24. Accessed December 10, 2019. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_survey_results_initial_report_2015_01_24.pdf.
- Department of Defense - Office of the Deputy Assistant Secretary of Defense for Systems Engineering. 2018. "Digital Engineering Strategy." June. Accessed January 17, 2022. https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf.
- McDermott, T., N. Hutchison, A. Salado, K. Henderson, and M Clifford. 2020. *Benchmarking the benefits and current maturity of model-based systems engineering across the enterprise: Results of the MBSE maturity survey*. Hoboken, NJ, USA: Systems Engineering Research Center (SERC).
- Noguchi, R. A., R. J. Minnichelli, and M. J. Wheaton. 2019. "Architecting Success in Model Based Systems Engineering Pilot Projects." *IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Bari, Italy.: IEEE. 755-760. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8914417.
- OMG. 2022. *SYSML V2: THE NEXT-GENERATION SYSTEMS MODELING LANGUAGE*. July 22. Accessed March 10, 2022. https://www.omgsysml.org/SysML-2.htm.
- Parnas, D.L., and P.C. Clements. 1986. "A rational design process: How and why to fake it." *IEEE transactions on software engineering* 251-257.

- Quinn Association. 2022. *Robert E. Quinn AND Kim S. Cameron's Culture Typology - Essence of culture typology.* Accessed January 31, 2022. https://www.quinnassociation.com/en/culture_typology.
- The Insights Group Limited. 2021. *Insights Discovery.* The Insights Group Limited. Accessed January 31, 2022. https://www.insights.com/products/insights-discovery/.
- Wesselius, J. 2021. "MBSE for the High-Tech Equipment Industry - MBSE-study of ESI and partners." April 16. Accessed January 3, 2022. https://a.storyblok.com/f/74249/x/e802069d0f/1604-jacco-wesselius-mbse-applied-final.pdf.
- Wesselius, J., J. van den Aker, R. Doornbos, J. Marincic, and T. Hendriks. 2022. *MBSE in the High-Tech Equipment Industry.* Eindhoven: TNO-ESI. https://publications.tno.nl/publication/34639873/jgHNmz/TNO-R2022-R11504.pdf.

ABOUT THE AUTHORS

Teun Hendriks is a system architect/research fellow at ESI (TNO) investigating how to transform systems engineering towards the use of models and digital twins and how to connect high-tech systems to the cloud, also for mission-critical functionality. He also is part-time self-employed as a consultant for automotive traffic and travel services in Europe and USA. Teun received a Master's in aerospace engineering from Delft University of Technology (1986).

Joris van den Aker is the program manager of competence development at ESI (TNO), responsible for developing and introducing innovative competence development concepts to accelerate learning in the high-tech industry. Blended courses, (in-house) competence development programs, special interest groups, and action learning communities are a few examples. Joris received a Master's degree in economics from Tilburg University (1994).

Wouter Tabingh Suermondt is a research fellow at ESI (TNO). He is involved in applying systems engineering principles and methods to the Dutch high-tech industry. After working in the industry in various positions as an engineer and architect, he joined ESI (TNO) in 2016. He participated in various projects with industrial partners about transitioning from traditional systems engineering towards MBSE.

Jacco Wesselius is currently the business director at ESI (TNO). As senior program manager at ESI, he guided the MBSE study, including collaborating with industry partners. Before joining ESI (TNO) in 2018, he held various positions in the high-tech equipment industry, including program manager and system architect. Jacco received a PhD degree in computer science/software engineering from Delft University of Technology (1992).

Conducting Design Reviews in a Digital Engineering Environment

Mark R. Blackburn, mblackbu@stevens.edu; and **Benjamin Kruse,** benjamin-kruse@t-online.de

■ **ABSTRACT**
This paper discusses how digital signoffs can enable new operational paradigms for business operations, digital engineering reviews, and contracts. The paper explains ***what*** digital signoffs are in the context of their use with model-based systems engineering methods, which is how this concept and construct has evolved. The paper discusses ***how*** they are created in the current toolset. This paper explains the benefits of ***why*** digital signoffs are valuable, in addition to ***where*** they can be placed within models, and ***when*** they might be used. Finally, we discuss how digital signoffs might evolve as add-on capabilities for digital engineering more broadly.

CONTEXT

For many years systems engineering has mostly produced documents, where the analysis and architectural characterization related to the interactions among different systems components were described in disparate documents. The requirements may have been managed in a requirements management tool like DOORS (https://www.ibm.com/products/requirements-management-doors-next) and interface definitions often in spreadsheets or some type of database. The operational context for how the system is used might be defined in some other document or some briefing charts. We have now referred to this operational context view as a mission or business enterprise view, because in today's world most systems are part of a larger system of systems, which often continually evolves due to ever-changing missions, threats, and technologies (Baldwin 2019). Many organizations are transitioning to the use of digital engineering (DE) (Blackburn et al. 2020a), including model-based systems engineering (MBSE), but they often have not heard about a concept we have introduced as a digital signoff mechanism that is now enabled by MBSE and tools in the associated DE environment (Kruse 2019 and Blackburn 2021c). A digital signoff is a means to capture an evaluating intent, an approval or rejection for example, in a dependable, and legally binding way that does not require paper or electronic documents, PDFs for example, but is instead part of the model information that is being assessed. A digital signoff is directly associated with a modeling artifact that requires assessment, such as completeness, correctness, or risk. We have demonstrated on a US Navy surrogate pilot called Skyzer MBSE practices using a collaborative environment with mission, system, subsystem, and discipline-specific models managed as an Authoritative Source of Information (ASOI) (Blackburn et al. 2021b). Skyzer is a notional acquisition concept developed to enable the development and application of digital engineering tools and methods. Skyzer uses surrogate data from publicly available sources to enable process "deep dives," and the team modeled everything to demonstrate the art-of-the-possible, including various examples of digital signoffs, and digital signoff measures and metrics discussed herein.

An example is shown in *Figure 1.* that highlights a few points about the elements related to a digital signoff. The example shows hazard/failure analysis that is modeled using a fault tree analysis (FTA). Briefly the FTA on the right of Figure 1 describes the analysis for the possibility of a hazard (such as vehicle accident or cyber-attack) if the basic event (bottom nodes) manifest without mitigation. The digital signoff is associated with the model of the FTA (that is, the evidence used in the decision). The digital signoff can be completed if a subject matter expert (SME) agrees that the FTA is consistent, complete, and correct. The FTA is the model artifact being assessed for a potential hazard, and assessment of *completeness*, *probability*, and *impact* is captured with *approval status* for digital signoff. An additional benefit of MBSE that is not usually possible in documents is the formalization of the requirement, shown at the bottom of the FTA that has been defined as a mitigation against the manifestation of the basic event. This points out how traditional requirement management can be improved when the new requirement is placed with the analysis within the model. Keeping the requirement with the analysis allows for traceability in case the analysis changes. Just like the digital signoff that is embedded in the model, so is the associated mitigation requirement. Finally, we have also demonstrated that we can computationally "reset" a digital signoff if the associated model artifact is modified and automatically notify the reviewers that they should review the changes and update the digital signoff (Blackburn et al. 2021c and 2020b)

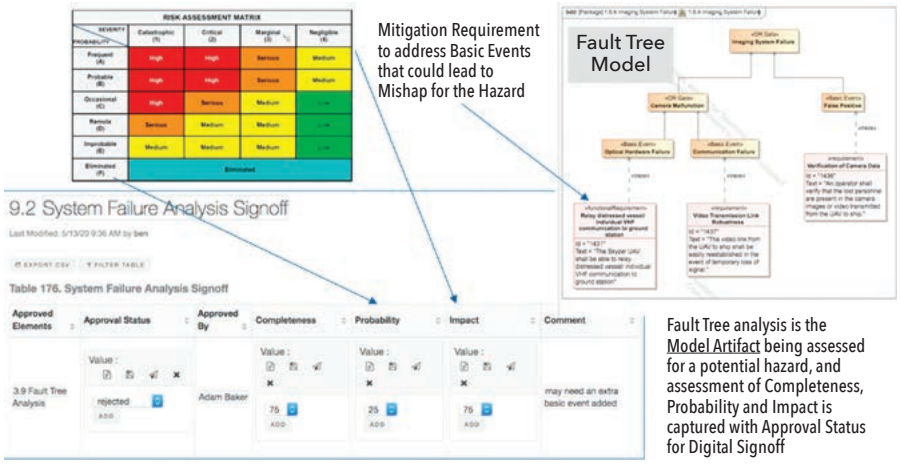


Figure 1. Example digital signoff for fault tree model and analysis

We used the open model-based engineering environment (OpenMBEE) (https://www.openmbee.org) software along with the MagicDraw (https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/) systems modeling language (SysML) model authoring tool to demonstrate how to expose model information for SMEs who have varying degrees of knowledge and concerns to review that model information in web browsers. We also demonstrated how a SME can digitally signoff model information in a web browser that was generated from the models in the ASOI (Blackburn et al. 2021c and Kruse and Blackburn 2019) as reflected in Figure 1. The OpenMBEE ASOI (https://www.openmbee.org) exists to enable multi-tool integration across disciplines using the OpenMBEE model management system (MMS) to store open and accessible model data while providing versioning, workflow management, and controlled access. OpenMBEE also consists of the model development kit (MDK), which is a plugin to the MagicDraw SysML modeling tool to generate and synchronize the model data with MMS. The MDK includes the DocGen (Delp et al. 2013) capability that uses a graphical modeling language for exposing model content as dynamic model-derived documents that are made accessible in the third part of OpenMBEE, the view editor web application (Kruse et al. 2020). The view editor runs in a standard web browser and offers a live web-based and light-weight access to the model data for agile virtual reviews and real-time collaboration. The signoffs are template-based as discussed in more detailed in Sections 2 ("what") and 3 ("how") and can include one or more roles performing the signoff and accompanying comments as discussed in Sections 5 ("where" to place digital signoffs).

WHAT ARE DIGITAL SIGNOFFS

A digital signoff is a template-based artifact that can have one or more signoffs with different types of criteria that are used to characterize the state of the artifact that is being "signed-off" in a model. A digital signoff currently has two parts. The first part is the model element to be signed off (for example, a use case, a diagram, a view, or a package) such as the fault tree analysis shown in Figure 1, which is part of the model and may contain further associated or owned elements for which the signoff applies, too. The second part is there to capture the status of the signoff and to make the signoff accessible in any web browser. The status of each signoff can hereby include additional information, for example, about who performed a signoff. It is tracked when the signoff occurs in the MMS, and who performed it.

The movement to use DE technologies for descriptive models with features such as: project usage (such as accessing model data from within other models), DocGen (that is, a language for generating documents from models) (Delp et al. 2013), view editor (that is, a web application that allows one to view and edit such model-derived documents) (Delp et al. 2013), and methods to accomplish a new operational paradigm by working directly and continuously in a collaborative DE environment is an enabler for digital signoffs, much like how we use DocuSign (https://www.docusign.com.) for electronic signatures of agreements. This new operational paradigm demonstrated that we could completely replace the use of static documents, which are subsumed into the modeling process using digital signoffs directly in the model through a collaborative DE environment for continuous and asynchronous reviews when the information is completed for any part of the system.

HOW TO CREATE DIGITAL SIGNOFFS

This is an area of opportunity for other ways to create embedded digital signoffs within a model, but the current/demonstrated approach leverages DocGen using view and viewpoints. The concept of view and viewpoints (Delp et al. 2013) has been around for more than a decade, but the specific implementation that comes with OpenMBEE MDK and DocGen provides a concrete mechanism to produce stakeholder-relevant views of models that are editable in the view editor that runs in a web browser. Views are defined as representations of a system that address stakeholder concerns. They are built using viewpoints,

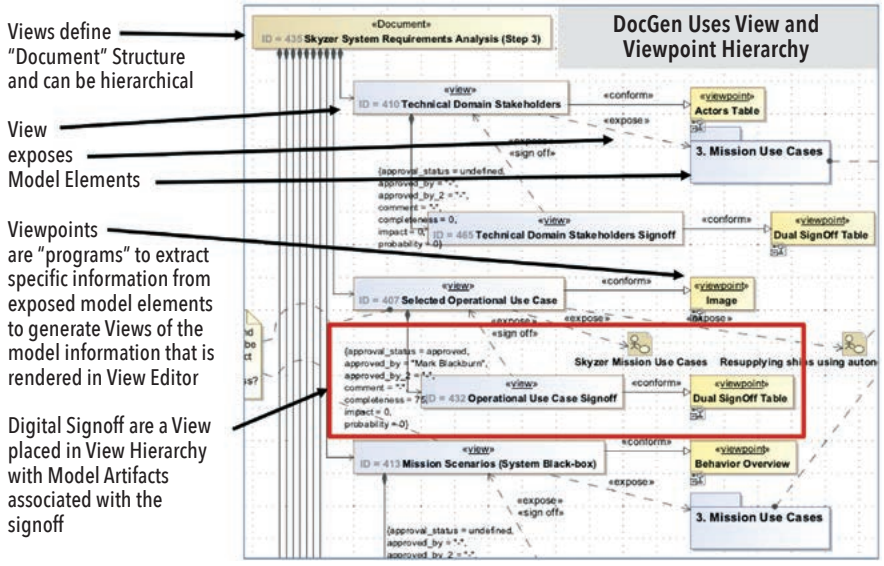


Figure 2. Digital signoffs are placed in view hierarchy with associated model artifacts to be signed off (NAVAIR Public Release 2019-443. Distribution Statement A – Approved for public release; distribution is unlimited)

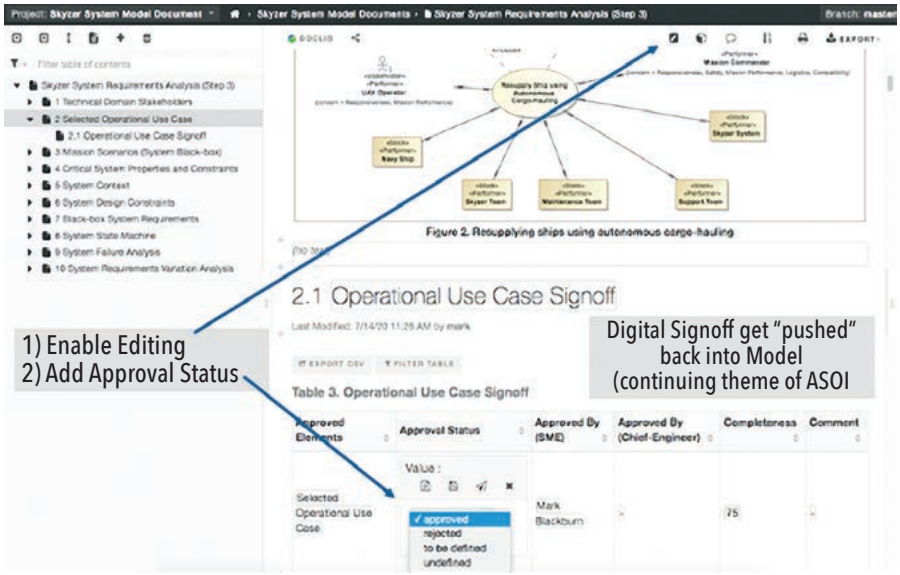


Figure 3. View editor showing digital signoff for operational use case for system why digital signoffs

Image_text removed

#	Date	Number Of Sign Offs	Number Of High Risk Sign Offs	Ratio Of Approved Sign Offs	Ratio Of Rejected Sign Offs	Average Risk
1	2020.04.17 13.44	5	1	0	0	55
2	2020.04.29 16.54	5	1	0.6	0.2	40

Figure 4. Digital signoff metric example

which specify the conventions and rules for constructing views out of the available model information. This capability of view and viewpoints to generate document-like views directly from model content, can provide stakeholder relevant information to be viewed in a web-browser or to be exported into a static document in Word or PDF.

An excerpt of a Navy standard template-based view and viewpoint hierarchy is shown in Figure 2 as part of developing the system model views. We have aligned it with the artifacts of the evolving NAVAIR systems engineering method (NAVSEM) (Blackburn et al. 2021b), which is process step 3.0 for the system requirements analysis. This approach demonstrates that modeling can be used to align modeling artifacts with existing standards that traditionally have been document-based. We have a set of view and viewpoint hierarchies that extract information from all the Skyzer models (for example, mission, system, and so forth) to "generate specifications." A portion of the system model view and viewpoint hierarchy includes the basic elements, such as views, viewpoints, and exposed model elements (for example, mission use case package, Skyzer mission use case diagram, and so forth) as shown in Figure 2.

Currently, we develop one or more view and viewpoint hierarchies as shown in Figure 2. Each view can hereby be understood as a section of the live document that is automatically generated from the exposed model element(s) based on the DocGen instructions of the viewpoint for which it conforms. For instance, for signoffs we place a view (for example, operational use case signoff), link the model artifact to be signed off (for example, selected operational use case) using an ***expose*** relation, and have the view ***conform*** to a fitting viewpoint (for example, dual signoff table) in order to create a table containing the signoff(s), which in this case requires two persons to sign off as shown in Figure 3, from an image of the view editor. Authorized users can ***enable edits*** and select the ***approval status*** and provide the name of the approver.

This type of technology enables and formalizes decision making, and directly associates each decision with specific model information related to that decision. Like DocuSign, the model and digital signoff facilitates a digital document for a contract. The digital signoff can (and should) be in one, and only one place to constitute decisions in an ASOI. Requesting specific model elements through digital signoffs that are part of standardized view hierarchies, can also enforce the use of specific modeling methods such as the NAVSEM (Blackburn et al. 2021b). This reduces time (Chen and Srinivasan 2019 and revised 2020), because the signoff can be performed as soon as the associated artifacts are "ready for review." The digital signoff is a DE construct, which means its state can be changed (computationally) if anything associated with the signed off artifacts (models) is changed. This can eliminate work and mistakes that occur when using documents, because it is often difficult to trace such relationships within one or more documents. This leads also to digital signoff metrics, as shown in Figure 4 that can be automatically generated to guide management and assessment of risk. These measures and metrics are automatically calculated and can again be viewed from a web browser.

We have developed this approach to enable a model for the request for proposal (RFP) response to become part of the ASOI by linking and tracing the contract RFP response directly to the mission and system model that formed the basis of the RFP. We demonstrated how to represent the technical source selection criteria as a digital signoff in the RFP response model. In government "talk," the digital signoffs in the ASOI provided an example for how to transform document-based contract data requirement lists (CDLRs) and data item deliverables (DIDs), and how to support asynchronous reviews enabled by collaborative information sharing. Involved government SMEs understand this new process and are interested in changing the

Approved Elements	Risk	Approval Status	Approved By	Comment
Air Vehicle Performance; Operational Radius	50	approved	Donald Polakovics	Evaluation Worksheet: Overall the aircraft far exceeds the operational radius KPP. Potential Strengths: Very significant margin for additional mission capability and versatility. Weaknesses: Aircraft may be larger and more expensive than necessary to do the mission. Deficiencies: None Uncertainty: Performance analysis could not be reviewed in its entirety due to some inconsistent data. Margins seem large enough to cover this however.
UAS Capability	0	undefined	N/A	N/A
Air Vehicle Performance; Endurance	50	approved	Donald Polakovics	Evaluation Worksheet: Overall the design appears to have sufficient endurance, with adequate development margin. Potential Strengths: Significant margin to KPP. Deficiencies: None Uncertainty: The endurance plots and table don't seem to agree. Some level of doubt as to actual endurance of the aircraft.

Figure 5. Digital signoff for source selection embedded in contractor RFP response

RFP process so that the contractors are required to include the source selection criteria as digital signoffs directly in their RFP response as shown in Figure 5.

Another benefit of the approach is that it can help identify missing model information. We start with an empty view and viewpoint template (Kruse and Blackburn 2019) that provides a way to represent what modeling artifacts should be created for a modeling method like NAVSEM. If model artifacts are not yet created and exposed in the view and viewpoint hierarchy as shown in Figure 2, the DocGen output indicates that there is missing model information. This is an important technology to increase consistency of "specifications/models" through automation, standardization, and reuse of curated Viewpoint libraries, and to provide information access in a web browser (or device) for those stakeholders that may not have access to tools, and it is a capability that enables the digital signoff mechanism. This further supports standardization and compliance with the modeling method, which is needed for digital signoffs. It also provides a way to create different views that are relevant to different stakeholders.

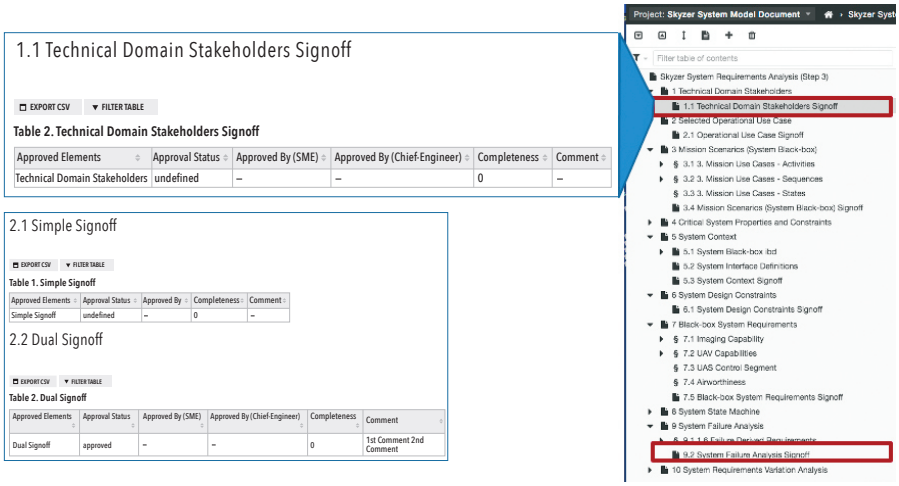


Figure 6. Template based examples as seen via view editor in a web browser

WHERE TO LOCATE DIGITAL SIGNOFFS

As described above, currently the digital signoffs are created in the SysML models, and we used OpenMBEE DocGen to generate web views of the model information with their digital signoffs. This permits the signoffs to be performed in a web browser using view editor. OpenMBEE MMS also tracks all changes to the model, including signoffs and who made the change as well as when it was made, and synchronizes the results back into the models and ASOI. It is important to be able to do these signoffs in a web browser because there may be many SMEs (that is, specialist and/or decision makers) who may not be able to navigate a model in a SysML authoring tool.

Being placed in view and viewpoint hierarchies, which are templates for automatic generation of stakeholder relevant views, signoffs can be rendered and edited in a web browser (or mobile device) and then synchronized (round tripped) back into the model from any part of the auto-generated live document. It is possible to include them directly together with their signed off artifacts, as shown in Figure 1 and Figure 3, or also in form of a cumulative table at the end or any other place. Such cumulative tables can also be used in addition to individual ones while showing and enabling the edit of the same signoffs in the ASOI at multiple places.

By having the view and viewpoint hierarchies in separate SysML projects as their exposed model content, it is possible to assign editing rights for only the views with the signoffs and not the model content or vice versa. This can for instance prevent modelers from signing off their own work or alternatively reviewers from accidentally changing the underlying mission or system model. In addition to the examples in Figure 1 and Figure 3, there are three examples shown in Figure 6, which reflects on how different digital signoff templates can be used depending on the needs of the particular signoff (for example, single person versus dual person signoff).

WHEN TO USE DIGITAL SIGNOFFS

The use of asynchronous reviews using digital signoffs through information access can be performed when the artifacts are ready for review, and this can (and should) be asynchronous allowing for more agile/interactive development scenarios during systems engineering. This is a dramatic movement away from traditional monolithic reviews such as preliminary design review (PDR) or critical design review (CDR) that have tended to be the norm for decades. In traditional approaches money and time may be wasted or lost for two reasons: 1) defects – that cause delay, or 2) inefficiencies, which cause delays as they wait for a "review." We can imagine that new contracting languages can be created to require the use of digital signoffs by contractors. These digital signoff measures could initiate a transformation away from traditional monolithic reviews enabling a new approach to continuous and asynchronous reviews when modeled artifacts are completed. Digital signoffs could also affect the way that earned value management (EVM) could be advanced; as the digital signoffs are executed, payments to the contractor could be made for the completed work. This could also help reduce the time and accelerate deployment of capabilities to the field.

CONCLUSION, CHALLENGES AND FUTURE THOUGHTS

While we introduced digital signoffs for descriptive models, the approach is currently implemented in SysML models (such as using MagicDraw, Magic System of Systems Architect) and could be more general and tool agnostic. We have not yet worked to propagate digital signoffs from other discipline-specific models back into the descriptive SysML models, but we do have research that can propagate information such as metadata from a discipline-specific simulation back to value properties and instances in a SysML model (Hagedorn et al. 2020). In addition, digital signoffs are methodology-specific, in that they are placed with specific types of artifacts that provide some concrete semantic meaning; this means that there are opportunities to teach people about the importance of modeling methods, as well as opportunities to develop digital assistance to aid people in complying with modeling methodologies.

Our Skyzer demonstrations have shown how this approach can transform CDRLs and source selection. Our demonstrations have informed our government sponsors, and we can anticipate that government acquisitions will be asking for this from contractors (again just like DocuSign has become a common way to digitally sign contracts). Digital signoffs in the ASOI provided an example for how to transform CDLRs supporting asynchronous reviews enabled by collaborative information sharing. Digital signoff link criteria often required in a CDRL that is needed at different program review points to be linked to model evidence. We determined an approach to use OpenMBEE view and

viewpoints as a means for placing a digital signoff directly with model information that provided the needed evidence. Digital signoffs can be updated in the view editor, with the signoff information (for example, approval, risk, approver, comments) added that get pushed back into the model through the ASOI. We also established a basis for automating digital signoff metrics. If a piece of information associated with a digital signoff is changed, the signoff can be automatically reset to an initial state. This should reduce cost by transforming/eliminating CDRLs that take on a new form in the model providing greater efficiency, consistency, automation, and standardization. This capability supports traceability for digital signoffs from high-level mission requirements to low-level discipline-specific design constraints as demonstrated in the surrogate pilot. ■

REFERENCES

- Baldwin, K. 2019. "Journal of Defense Modeling and Simulation (JDMS) special issue: Transforming the engineering enterprise—applications of Digital Engineering and Modular Open Systems Approach." *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology.* 16 (4): 323–324. DOI: 10.1177/1548512917751964.
- Blackburn, M., T. Hagedorn, D. Dunbar, Z. Yu, and B. Kruse. 2021a. "Ontologies for Engineering: A Pragmatic Perspective." Presented at NDIA Second Virtual Systems and Mission Engineering Conference, 6-8 December.
- Blackburn, M., D. P. Allan, T. Fields, and S. Cimtalay. 2021b. "Integrating Digital Engineering Technical Models with MBSE Cost Models." NDIA Second Virtual Systems and Mission Engineering Conference, 6-8 December.
- Blackburn, M. R., J. Dzielski, R. Peak, S. Cimtalay, T. Fields, W. Stock, S. Panchal, J. Sisavath, G. Rizzo, C. Noll, and S. Schmidt. 2021c. "Transforming Systems Engineering through Model-Centric Engineering." Final Technical Report SERC-2021-TR-012, WRT-1036 (NAVAIR), 3 August.
- Blackburn, M., T. McDermott, and M. Bone 2020a. "Digital Engineering Measures Correlated to Digital Engineering Lessons Learn from Systems Engineering Transformation Pilot." Presented at NDIA Virtual Systems and Mission Engineering Conference, 10 November – 13 December.
- Blackburn, M., B. Kruse, W. Stock, and M. Ballard. 2020b. "Digital Engineering Modeling Methods for Digital Signoffs." Presented at NDIA Virtual Systems and Mission Engineering Conference, 10 November – 13 December.
- Chen, W., and S. Srinivasan. 2019. "Going Digital: Implications for Firm Value and Performance." Harvard Business School Working Paper. No. 19-117, May. (Revised July 2020.)
- Delp, C., D. Lam, E. Fosse, and C.Y. Lee. 2013. "Model Based Document and Report Generation for Systems Engineering." Paper presented at IEEE Aerospace Conference, Big Sky, US-MT, 2-9 March.
- DocuSign. https://www.docusign.com.
- IBM. "Engineering Requirements Management DOORS Next." https://www.ibm.com/products/requirements-management-doors-next.
- Hagedorn, T., M. Bone, B. Kruse, I. Grosse, and M. Blackburn. 2020. "Knowledge Representation with Ontologies and Semantic Web Technologies to Promote Augmented and Artificial Intelligence in Systems Engineering." *INSIGHT* 23 (1): 15-20.
- Kruse, B., T. Hagedorn, M. A. Bone, and M. Blackburn. 2020. "Collaborative Management of Research Projects in SysML." Paper presented at virtual 18th Annual Conference on Systems Engineering Research (CSER), 8-10 October.
- Kruse, B., and M. R. Blackburn. 2019. View and Viewpoint based Digital Signoff using OpenMBEE as an Authoritative Source of Truth. Journal of Cyber Security & Information Systems, M&S Special Edition, December.
- MagicDraw, https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/
- OpenMBEE, https://www.openmbee.org.

ABOUT THE AUTHORS

Dr. Mark R. Blackburn is a member of the SERC research council, providing guidance with a focus in the systems engineering and systems management transformation research area. Dr. Blackburn is a member of OpenMBEE leadership team, a community of model-based engineering practitioners and software developers for the open-source model-based engineering environment or OpenMBEE. He is the principal investigator (PI) on several SERC research tasks for both NAVAIR and the US Army Armaments Center on systems engineering transformation through model-centric engineering. Dr. Blackburn is also a senior research scientist with Stevens Institute of Technology and principal at KnowledgeBytes. His research focuses on methods, models, and automated tools for reasoning about complex systems of systems. Dr. Blackburn holds a PhD from George Mason University, a MS in mathematics with an emphasis in computer science from Florida Atlantic University, and a BS in mathematics (computer science option) from Arizona State University.

Benjamin Kruse research is about model-based systems engineering with the general-purpose graphical systems modeling language SysML. At Stevens Institute of Technology, he worked as a researcher for SERC on research tasks about the systems engineering transformation through model-centric engineering. He contributed to related SERC research projects to better understand the relationships between systems engineering activities and methods in the context of a digital thread and to demonstrate and investigate the art-of-the-possible of model-centric engineering through a pilot study as part of the Department of Defense's digital engineering strategy. He has his ScD from the ETH Zurich in Switzerland, a diploma (university) for mechanical engineering in the field of aerospace at the Technical University Munich in Germany.

Scenario-based Verification and Validation of Automated Transportation Systems

Birte Neurohr, birte.neurohr@dlr.de; and **Eike Möhlmann,** eike.moehlmann@dlr.de

■ **ABSTRACT**
The research and development activities performed by the DLR Institute of Systems Engineering for Future Mobility (DLR-SE) are organized via so-called assets. We present a scenario-based verification and validation process and relate selected research activities.

Verification and validation approaches of automated transportation systems based on driving a certain number of kilometers are infeasible. Therefore, the DLR-SE asset "Scenario-based Verification and Validation of Automated Transportation Systems" investigates methods and prototyping tools for verifying and validating automated transportation systems employing scenarios as the main structuring element to capture complex traffic evolutions. While there are many different approaches, our focus is formally specifying relevant abstract scenarios that are readable by humans while also being machine-readable. This allows us to automatize the verification and validation process, which increases confidence in, for example, the safety of the systems due to a dramatically increased number of executed tests while reducing the manual effort from humans.
■ **KEYWORDS:** automated systems; verification; validation; scenario-based testing; automated transportation systems; automated driving; safety

INTRODUCTION

Automated and autonomous transportation systems are not only thought of as a way to make traveling more comfortable but also as a means to make it safer. To realize this and bring automated and autonomous transportation systems into the market, it is essential to guarantee their safe operation. This is a challenge as the systems as well as the input they receive (the environment) are highly complex and, further, depend on the targeted safety level. For instance, when human drivers are allowed to operate a vehicle, they have at least 17 years of experience with traffic, the expected behavior of other humans, and basic physical principles. Thus, the question arises of how to ensure a positive risk balance, including automated driving systems (ADS) causing fewer accidents than humans. Hence, for the validation and verification of automated transportation systems, it is not only necessary to develop them in a safe way but to test them extensively before rollout. These topics are addressed in the DLR-SE's asset "Scenario-based Verification and Validation of Automated Transportation Systems." The current focus of this asset is automated vehicles, but extension towards the maritime and the railway domain is ongoing.

Today's vehicles have been improved over decades, and human drivers can now drive relatively safely, thus, the average distance between accidents is very long. To demonstrate that a single automated driving system is safer than a human driver, the number of test kilometers necessary for statistical evidence amounts to several hundreds of millions of kilometers, depending on assumptions and the type of accident (Wachenfeld and Winner 2016, 442). To put this in perspective, all paved streets in the USA only form a network of 4.3 million km (World Factbook 2012). Even worse, — without further arguments — these tests would need to be performed with every newly developed or slightly modified automated driving system. Thus, an approach based on driving a distance to statistically show that an ADS is safer than a human-operated vehicle is infeasible in practice.

THE SCENARIO-BASED APPROACH

One possible solution for this dilemma is a scenario-based approach (Riedmeyer et al. 2020). A scenario describes a temporal evolution of traffic scenes, where a scene is a snapshot of the environment including its scenery (like lanes, obstacles and traffic signs) and dynamic objects (like cars, passengers and bicyclists) (Ulbrich et al. 2015).

Scenarios built the foundation of our verification and validation methods as they allow for structuring the complex environment consisting of an infinity of possibilities. They allow for reasoning the safe operation of an automated transportation system without relying solely on the num-

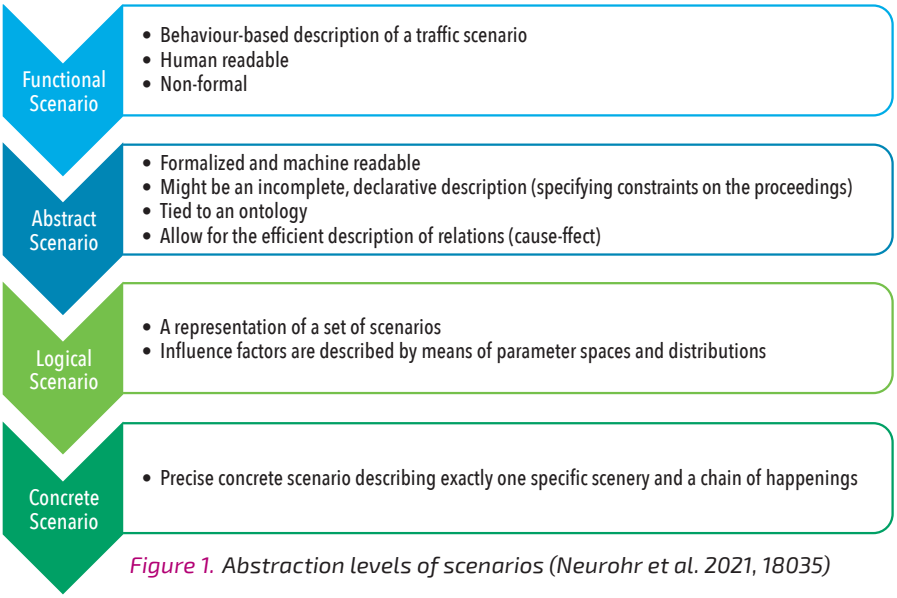


Figure 1. Abstraction levels of scenarios (Neurohr et al. 2021, 18035)

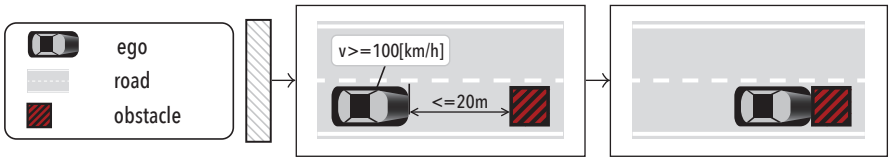


Figure 2. Car collides with an obstacle (specified as TSC) (Jan Steffen Becker, pers. Comm.)

ber of kilometers driven. Instead, they take advantage of the identification and understanding of which principles are essential for the safety of automated transportation systems. Thus, verification and validation methods can be structured and carried out in a more systematically than in a naïve distance-based approach with random test cases (Wachenfeld and Winner 2016, 442)

Scenarios can be described at different abstraction levels relevant at different stages in the V&V process (Menzel et al. 2018), (Becker et al. 2021, 3).

A *functional scenario* (Menzel et al. 2018) is human-readable and non-formal. It is a behavior-based description of a traffic scenario. Functional scenarios can be used in the very early phases of the verification and validation process.

Illustrative example: The ego vehicle is driving on the right lane of a two-lane highway below 100 km/h. There is one obstacle in front of the ego. Then the ego vehicle collides with the obstacle.

An *abstract scenario* (Neurohr et al. 2021) is formalized in a declarative way. Thus, it only specifies what is relevant to the described traffic scene and leaves out irrelevant aspects. It is always tied to an ontology and allows for describing alternatives and variance in objects and space. Abstract scenarios are used in the concept phase of the verification and validation process.

Illustrative example: As an example of an abstract scenario, we present a Traffic Sequence Chart (TSC) (Damm et al. 2017; Damm et al. Jan 2018; Damm et al. Jul 2018) in Figure 2. While it may look like a simple picture, it actually translates to a formula in a first-order multi-sorted real-time logic that machines can read and interpret. It should be noted that this TSC corresponds to a multitude of specific collisions.

The shown TSC captures only the relevant constraints and, hence, describes all traffic evolutions that (1) anything may happen, (2) a vehicle called ego with a velocity of 100 km/h (or higher) approaches an obstacle with a distance of at least 20m on a lane of a road with at least one more left lane, and (3) touches the obstacle. Note that aspects that are not constrained, such as the existence of other traffic participants, the shape of the road, the weather, the type of the vehicle, and the obstacle, are left open. Therefore, an infinity of concrete scenarios is described.

Logical scenarios (Menzel et al. 2018) have value ranges for parameters and parameter constraints that may also be given by specifying distributions. They may be used during system development.

In contrast to the example of an abstract scenario above, all parameters are specified (with a value range) here. For example, the width of the road is between 3m and 3,75m. This is not specified in the abstract scenario above.

Scenery, Concrete scenarios (Menzel et al. 2018) have concrete values instead of parameter ranges. Thus, they describe one specific *scenery* and chain of events.

These different abstraction levels of scenarios are used during verification and validation. The necessary level of abstraction depends on the phase of this process. Please note that the amount of described scenarios rise with the abstraction level.

A simplified framework of a scenario-based approach based on the work of the research projects ENABLE-S3 (www.enable-s3.eu) and PEGASUS (www.pegasusprojekt.de/en) can be seen in Figure 3 (Neurohr et al. 2020). The first step, scenario elicitation, consists of deriving adequate scenario classes to be tested. The requirement elicitation process equips the scenarios with the corresponding requirements. Testing will then be carried out virtually in simulations and physically

Table 1. Illustrative example of a logical scenario

Right lane width $[m]$	$[3,\ldots,3{,}75]$
Left lane width $[m]$	$[3,\ldots,3{,}75]$
Speed Ego vehicle $[\frac{km}{h}]$	$[100,\ldots,150]$
Long. position Ego vehicle $[m]$	$[80,\ldots,100]$
Long. position of obstacle $[m]$	$[80,\ldots,100]$
Long. position Ego vehicle < Long. position obstacle	

Table 2. Illustrative example of a concrete scenario

Right lane width $[m]$	[3,75]
Left lane width $[m]$	[3,75]
Speed Ego vehicle $[\frac{km}{h}]$	[125]
Long. position Ego vehicle $[m]$	[80]
Long. position of obstacle $[m]$	[92]
Long. position Ego vehicle < Long. position obstacle	

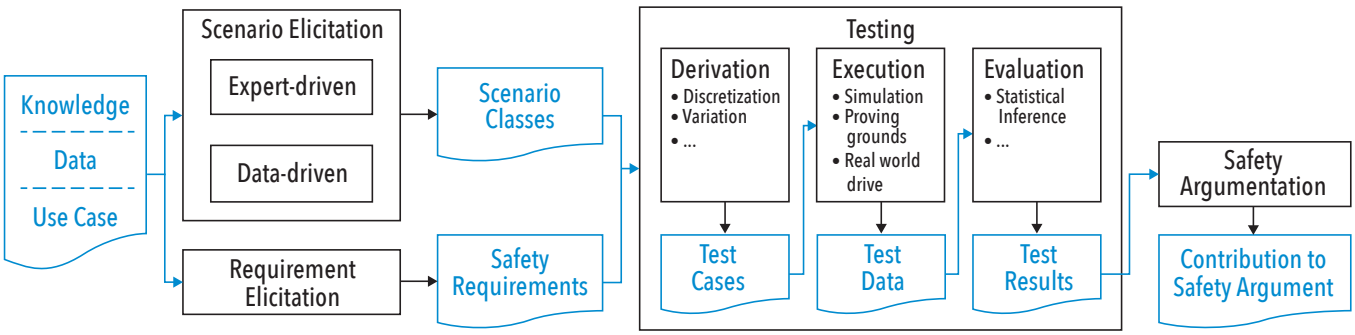


Figure 3. Simplified framework around scenario-based testing (Neurohr et al. 2020, 122)

on proving grounds and in the targeted environment. Finally, the results are integrated into an overarching safety argumentation (c.f. Koopmann et al. 2019), contributing to the safety case.

The focus of this asset lies in the scenario elicitation and execution part of this framework. Albeit the process and framework can incorporate different test techniques such as model-, software-, hardware-, and vehicle-in-the-loop, we, however, focus on computer simulations (MiL, SiL) as a virtual test bench.

CURRENT CHALLENGES THE ASSET IS ADDRESSING

While the idea of a scenario-based approach is well established and has already found its way into standardization organizations (ISO 21448; UL 4600), there are still many open questions about applying it.

On the one hand, knowing which scenarios are relevant is difficult. Using scenarios very similar to each other, like "driving on a highway with a yellow car in front of the ego" and "driving on a highway with a red car in front of the ego," does not add much value to the verification and validation process. Thus, it is imperative to identify scenarios that add value by identifying what makes a scenario relevant and critical (Zhang et al. 2022).

Scenarios are the foundation to reduce the search space for verification and validation approaches for automated transportation systems (Kalisvaart et al. 2020). This reduction is based on a fundamental principle: Myriad similar concrete scenarios can be described by one abstract or logical scenario. The process of determining abstract scenarios is called "Scenario-Mining." It can be approached either based on data, expert knowledge, or combining the best of both worlds. These approaches are addressed in the asset.

Closely related is the "Criticality-Analysis" (Neurohr et al. 2021), aiming to determine relevant phenomena and explain the underlying causality. This also contributes to determining which scenarios should be considered relevant for testing, however, from a different perspective. The criticality analysis strives to map the infinite-dimensional domain onto a finite and manageable set of artifacts that capture and explain the emergence of critical situations for automated vehicles. In the asset, we target a combined approach of expert-based and data-driven methods that leverages an ontology.

On the other hand, the question of how to correctly specify scenarios still needs to be fully answered because of the open context automated vehicles must operate. That means it is impossible to fully specify the operation environment at design time as it is highly complex and subject to constant change. Hence, human experts cannot carry out validation and verification methods for automated transportation systems alone, and methods for monitoring the satisfaction of requirements are needed. Additionally, monitors are needed to detect novelties and anomalies in order, for instance, to detect missing scenarios (addressing the open-world problem and, hence, incompleteness of any scenario set) and model inaccuracies as well as to activate fallback strategies like degraded operation modes and minimum risk maneuvers.

Here again, the abstract scenarios come into play. An abstract scenario covers infinite concrete scenarios. They focus on complex interrelations, especially cause-and-effect relationships, which are essential for a scenario-based approach. As the TSCs mentioned above are not only machine-interpretable but also easily interpretable by humans, they may build a solid basis to support humans in the verification and validation process and, hence, increase confidence in safety by being able to execute more tests while reducing the needed manual effort.

This asset's basis and the connecting element is the concept of abstract scenarios. Thus, a central goal is further developing and tailoring the TSC language. Currently, it has an automotive focus. Ongoing work is to extend the language towards the maritime and the rail domain and include more language features to increase the expressiveness.

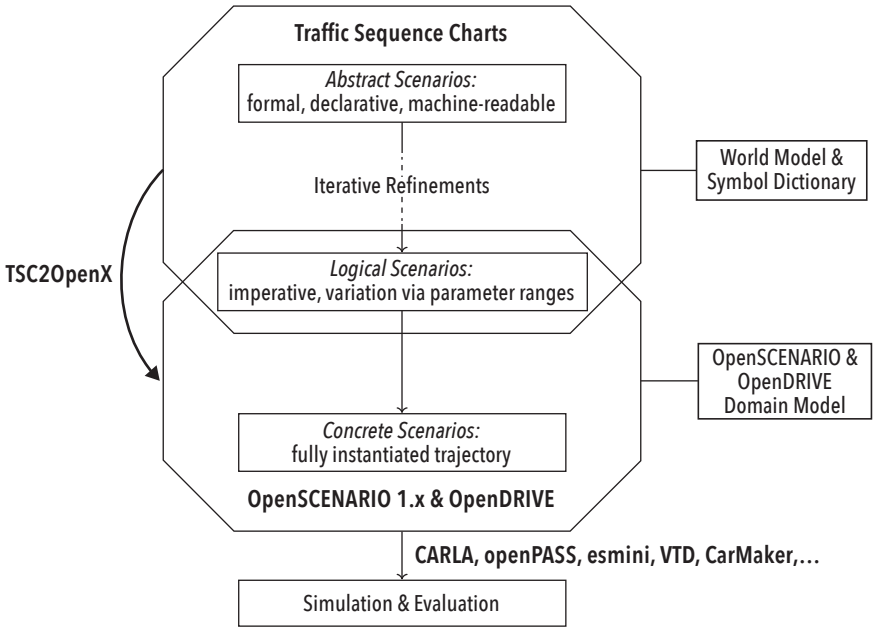


Figure 4. TSC Toolchain (Becker et al. 2020)

Furthermore, a prototypical tool for creating and evaluating TSCs is developed to do consistency analysis for TSCs and other automated reasoning. This prototypical tool specifies TSCs in a well-defined format that serves as an input for other TSC-related tools like TSC2OpenX. The aim of TSC2OpenX (see Figure 4 is to transfer abstract scenarios from TSCs into concrete scenarios in the industrially relevant formats OpenDRIVE (https://www.asam.net/standards/detail/opendrive/) and OpenSCENARIO (https://www.asam.net/standards/detail/openscenario/). These, in turn, can be simulated by most of the simulation platforms, thus reducing the manual effort of deriving concrete scenarios to be tested.

A test platform is needed to test or assess the safety of a given system. Within this asset, methods and different prototypes of scenario-based testing platforms for simulating the derived concrete OpenDRIVE and OpenSCENARIO are developed. These also guide the simulation into concrete scenarios with identified weaknesses, making them more meaningful for risk estimation.

Last but not least, when using simulation (relying, for example, on dynamic models) to assess the system's safety, we need to make sure that simulation results are transferrable to reality. Knowledge about this relation is a prerequisite for basing a safety argumentation for automated transportation systems on simulative tests within any verification and validation process. Thus, within the asset, we also investigate methods that help determine a simulation's validity, the used simulation models, and the obtained simulation runs.

OUTLOOK

For many of the challenges above, we are working on ideas, methods, and prototypical tools on how to tackle them. Automated transportation systems pose significant risks when they are not thoroughly verified and validated. This would put humans and our environment in danger and, consequently (and rightfully so), threaten their acceptance by society. Thus, we must develop methods and tools that help system and test engineers deal with the enormous complexity of traffic situations during design and assessment to obtain sufficient confidence in the safety of automated vehicles. While our research focuses on the mobility domain, we expect that gained insights (for instance, using scenarios for testing) can be transferred to other domains like health (https://enable-s3.eu). ■

ACKNOWLEDGMENTS

This paper reports on collaborative work by many colleagues from DLR and partners in the projects PEGASUS (https://www.pegasusprojekt.de/en/home), ENABLE-S3 (https://enable-s3.eu), VVMethoden (https://www.vvm-projekt.de/en/), SET Level (https://setlevel.de/en), KI Delta Learning (https://www.ki-deltalearning.de/en/) and KI Wissen (https://www.kiwissen.de/). This work received funding from the German Federal Ministry for Economic Affairs and Climate Action.

In particular, we want to thank our DLR (formerly OFFIS) colleagues Eckard Böde, Jan Steffen Becker, Günter Ehmen, Sebastian Gerwinn, Thies de Graaff, Daniel Grujic, Christian Neurohr, Lukas Westhofen, and Boris Wirtz, as well as the scientific leaders Werner Damm and Martin Fränzle.

REFERENCES

- Becker, J. S., C. Neurohr, and L. Westhofen. 2021. "TSC2OpenX – Constraint-based Concretization of Abstract Scenarios for Automated Driving." Presentation presented at *AVL Virtual International Simulation Conference* 2021. Online, June 22- July 2. doi: 10.13140/RG.2.2.18534.75841.
- Becker, J. S., T. Koopmann, B. Neurohr, C. Neurohr, L. Westhofen, B. Wirtz, E. Böde, and W. Damm. 2022. "Simulation of Abstract Scenarios: Towards Automated Tooling in Criticality Analysis." Paper presented at Workshop *Autonome Mobilität*, Zürich, Switzerland, January, doi: 10.5281/zenodo.5907154.
- Damm, W., S. Kemper, E. Möhlmann, T. Peikenkamp, and A. Rakow. 2017. "Traffic Sequence Charts – From Visualization to Semantics." Technical Report. AVACS Technical Report No. 117. http://www.avacs.org/fileadmin/Publikationen/Open/avacs_technical_report_117.pdf
- Damm, W., S. Kemper, E. Möhlmann, T. Peikenkamp, and A. Rakow. 2018. "Using Traffic Sequence Charts for the Development of HAVs." Paper presented at *ERTS* 2018. Toulouse, France, January 31- February 2. HAL Id: hal-01714060.
- Damm, W., E. Möhlmann, T. Peikenkamp, and A. Rakow. 2018. "A Formal Semantics for Traffic Sequence Charts" In *Principles of Modeling* 10760, 182-205, edited by Lohstroh, M., and Derler, P., and M. Sirjani. Springer, Cham
- Kalisvaart, S., Z. Slavik, and O. Op den Camp. 2020. "Using Scenarios in Safety Validation of Automated Systems." In *Validation and Verification of Automated Systems*, edited by Leitner, A., D. Watzenig, and J. Ibanez-Guzman: 27–44, Springer, Cham. doi: 10.1007/978-3-030-14628-3_5
- Koopman, P., A. Kane, and J. Black. 2019. "Credible Autonomy Safety Argumentation." In *27th Safety-Critical Sys. Symp. Safety-Critical Systems Club*, Bristol, UK
- ISO (International Organization for Standardization). 2022. ISO 21448:2022. "Road vehicles- Safety of the intended functionality." Geneva, CH
- Menzel, T., G. Bagschik, and M. Maurer. 2018. "Scenarios for Development, Test, and Validation of Automated Vehicles." Paper presented at *IEEE Intelligent Vehicles Symposium (IV)*, Changshu, Suzhou, CN, 26-30 June: 1821-1827, doi: 10.1109/IVS.2018.8500406.
- Neurohr, C., L. Westhofen, M. Butz, M.H. Bollmann, U. Eberle, and R. Galbas. 2021. "Criticality Analysis for the Verification and Validation of Automated Vehicles." *IEEE Access*. 9: 18016-18041. doi: 10.1109/ACCESS.2021.3053.
- Neurohr, C., L. Westhofen, T. Henning, T. de Graaff, E. Möhlmann, and E. Böde. 2020. "Fundamental Considerations around Scenario-Based Testing for Automated Driving," Paper presented at *IEEE Intelligent Vehicles Symposium (IV)*, online, 19 Oct.-13 Nov:121-127, doi: 10.1109/IV47402.2020.9304823
- Riedmaier, S., T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer. 2020. "Survey on Scenario-Based Safety Assessment of Automated Vehicles." *IEEE Access*. 8: 87456-87477. doi: 10.1109/ACCESS.2020.2993730
- Underwriters Laboratories (UL). 2022. UL Standard 4600 Edition 2 "Standard for Safety for the Evaluation of Autonomous Products." Illinois, USA.
- Ulbrich, S., T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. 2015. "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving." Paper presented at the *IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, Spain, 15-18 September. doi: 10.1109/ITSC.2015.164.
- Wachenfeld, W., and H. Winner. 2016. "The Release of Autonomous Vehicles." In *Autonomous Driving*, edited by Maurer, M., J. Gerdes, B. Lenz, and H. Winner: 452-449. Springer, Berlin, Heidelberg. doi:10.1007/978-3-662-48847-8_2.
- World Factbook. 2012. Roadways." https://www.cia.gov/the-world-factbook/field/roadways/.

> continued on page 64

Integrating System Failure Diagnostics Into Model-based System Engineering

Emile van Gerwen, emile.vangerwen@tno.nl; **Leonardo Barbini**; and **Thomas Nägele**

■ **ABSTRACT**
Ever-increasing system complexity is challenging for development engineers and service personnel troubleshooting system failures in the field. This paper presents a systematic, scalable approach to attain a diagnostic model. Automatic transformation into computational models is used 1) at design time to improve the diagnosability of the system, and 2) during operation for guided root cause analysis by calculating the most probable failures and suggesting diagnostic procedures based on available data and observations. The approach combines nicely with model-based systems engineering, showing the added value of using diagnostic models both during the design of a system and during operation when the system needs to be diagnosed.

INTRODUCTION

Getting a grip on the ever-increasing system complexity is challenging for development engineers and service personnel troubleshooting system failures in the field. Traditionally, supporting the service organization from R&D goes along the lines of a failure mode and effect analysis (FMEA) that delivers a spreadsheet of possible causes and observable effects from which a service manual or, at best, a decision tree is derived. Many factors make this process suboptimal, including human factors, determining system-level effects from subcomponent analysis, and keeping up with new product variations. Consequently, the outcome is often incomplete, inconsistent, and hardly reusable.

This paper presents a systematic, scalable approach to attain a diagnostic model. After automatic transformation into computational models, these are used 1) at design time to improve the diagnosability of the system and 2) during operation for guided root cause analysis by calculating the most probable failures and suggesting diagnostic procedures based on available data and observations.

We emphasize that the approach combines nicely with model-based systems engineering, as it empowers diagnostics support, yet another incentive to push the model-based systems engineering efforts.

High-tech systems manufacturers are currently validating the proposed approach.

DIAGNOSING HIGH-TECH SYSTEMS IS HARD

The service engineer tasked with diagnosing a failing high-tech system has a tough job. The ever-increasing complexity of high-tech industrial systems makes it impossible to know all the intrinsic details of their behavior. However, these are needed for system-level diagnostics.

Equipped with a 10000-page service manual on a laptop, the service engineer must efficiently get to the root cause of the system failure. In many cases, this does not work, and the engineer relies on experience. However, building up this experience is getting more challenging because of several trends in the industry. For one, systems change regularly, with systems deployed in the field being upgraded and getting updates all the time. Experience can get stale in no time. Secondly, the trend to

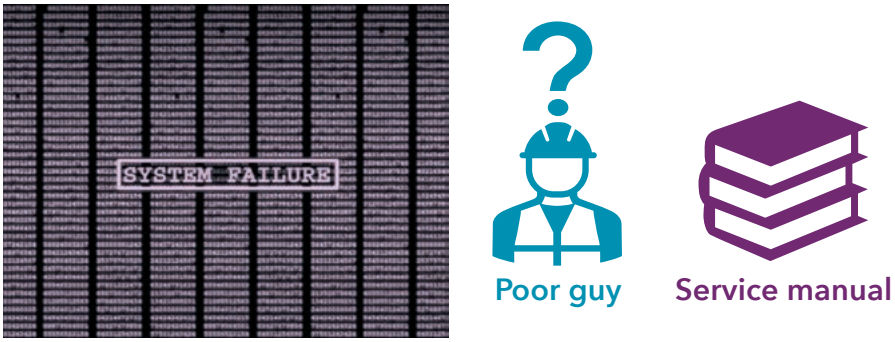


Figure 1. Diagnosing high-tech system failures is hard and the traditional service manuals are often insufficient for the job

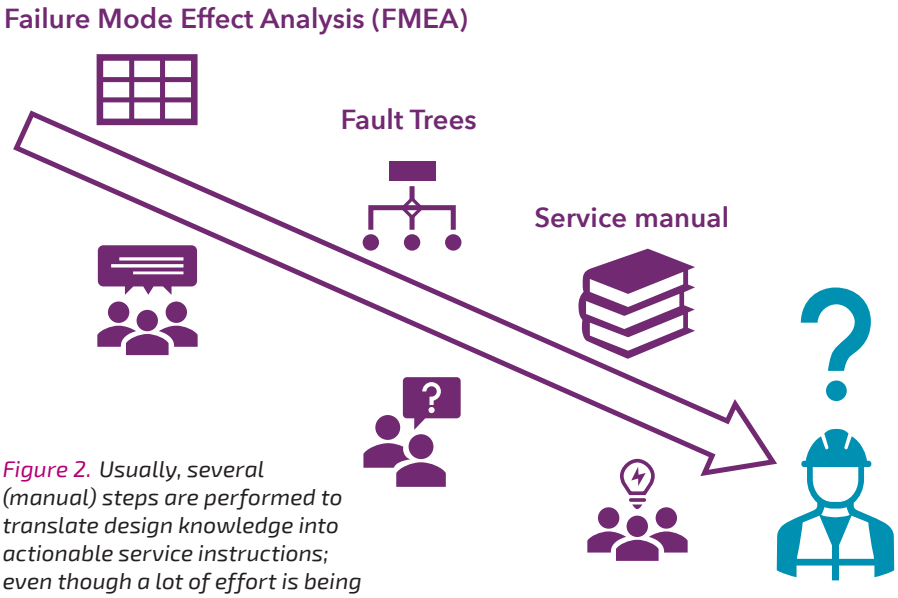


Figure 2. Usually, several (manual) steps are performed to translate design knowledge into actionable service instructions; even though a lot of effort is being put on the creation of diagnostic procedures and manuals, the service engineer is still missing information to diagnose system failures quickly

delegate service to local agents for cost saving and faster response times makes it difficult to build up expertise for a specific brand or model, as they often have a large portfolio to deal with.

WHY CURRENT PRACTICES FAIL

So how does the service engineer get the needed information?

When the design of a system is almost completed, it is common practice to conduct a Failure Mode and Effect Analysis (FMEA) to identify potential failure modes and their consequences. If a potential failure is severe but not critical enough to warrant a redesign, this analysis is passed to the technical writers to incorporate this scenario in the service manual. There are a few things that make this a dreary process.

To begin with, system designers are not at ease thinking about failures. This is the opposite of their training and experience: thinking of how the system should work. Also, they usually have in-depth knowledge of a small part of a system, and it is hard for them to live up to the service engineer's need for system-level diagnostic reasoning. The main challenge, however, is transforming the FMEA outcome to the service manual – the main instrument provided to the service engineer to perform the diagnostic task. An FMEA analysis is a bottom-up approach that takes component failures as starting point and reasons towards system failure symptoms. The service manual contains diagnostics procedures to achieve precisely the reverse: given some system failure symptoms, find the root cause. There is no well-defined way to perform such a transformation. This task is delegated to the same system designers involved in the FMEA analysis, encountering the exact issues described above. Usually, design engineers first transform the outcome of the FMEA into fault trees or Decision Trees (DTs). However, building these trees requires considerable work, and a top-down structure is not trivial, so we see that DTs are incomplete and sometimes even inconsistent. The situation worsens after a system design change: finding all the locations in the DT that need an update is challenging and time-consuming. Technical writers then document these DTs as diagnostic procedures in the service manual. Figure 2 shows the described approach.

EARLIER ATTEMPTS

Fuelled by successes in the medical domain, in the '90s, the graphical probabilistic models appeared (Console and Dressier 1999). This allowed reasoning with uncertainty, which is unavoidably present in a diagnostic setting. Creating these models, however, turned out to be a work of art; there were not enough artists around to apply this on an industrial scale.

Driven by storage capacity and computation power, this century brought us the hopes of using data analysis to circumvent the need to create models by hand. The idea is to gather machine operational data to identify signatures of failures and use those data to learn a diagnostic model. In practice, a lot of data is needed to learn such a model. However, the amount of data on failing systems is limited because if there is a lot of data for a single failure, you can better fix the system design instead of diagnosing it faster.

Additionally, data only becomes available after the systems have been in the field for some time, making this method of limited use shortly after system introduction, when the experience levels of service engineers are low. Lastly, we see that because of the variety of machine execution patterns, black-box approaches such as neural networks are not satisfactory. The resulting models need extensive curation by experts, an activity that data-driven methods try to avoid in the first place.

Our method combines the strengths of the diagnostic approaches discussed above together with the evolving model-based systems engineering methods to reach a structured and scalable way of diagnostic support.

DIAGNOSTICS IN THE CONTEXT OF MODEL-BASED SYSTEMS ENGINEERING

With model-based systems engineering methods gaining traction in the high-tech industry, models are more frequently used as carriers of design information rather than relying on documents. The use of formalized information for the design of a system allows for formal verification and validation of its building blocks and a formal description of how these are interconnected.

Our vision is that the design information formalized in design models according to the model-based systems engineering paradigm could also be used for diagnostic purposes. These models already describe a coherent system and precisely follow the system's decomposition into smaller building blocks until only replaceable parts are listed. The system decomposition that models describe following the model-based

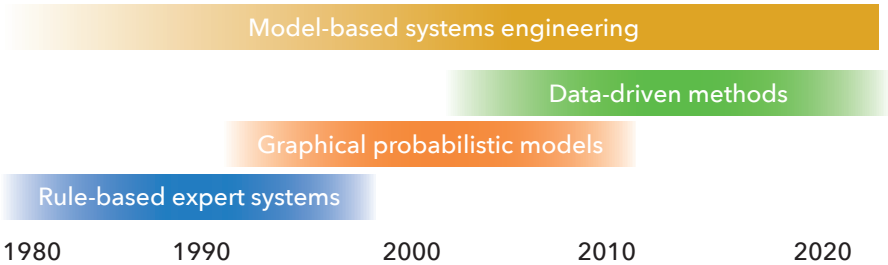


Figure 3. Throughout the years many approaches have been developed

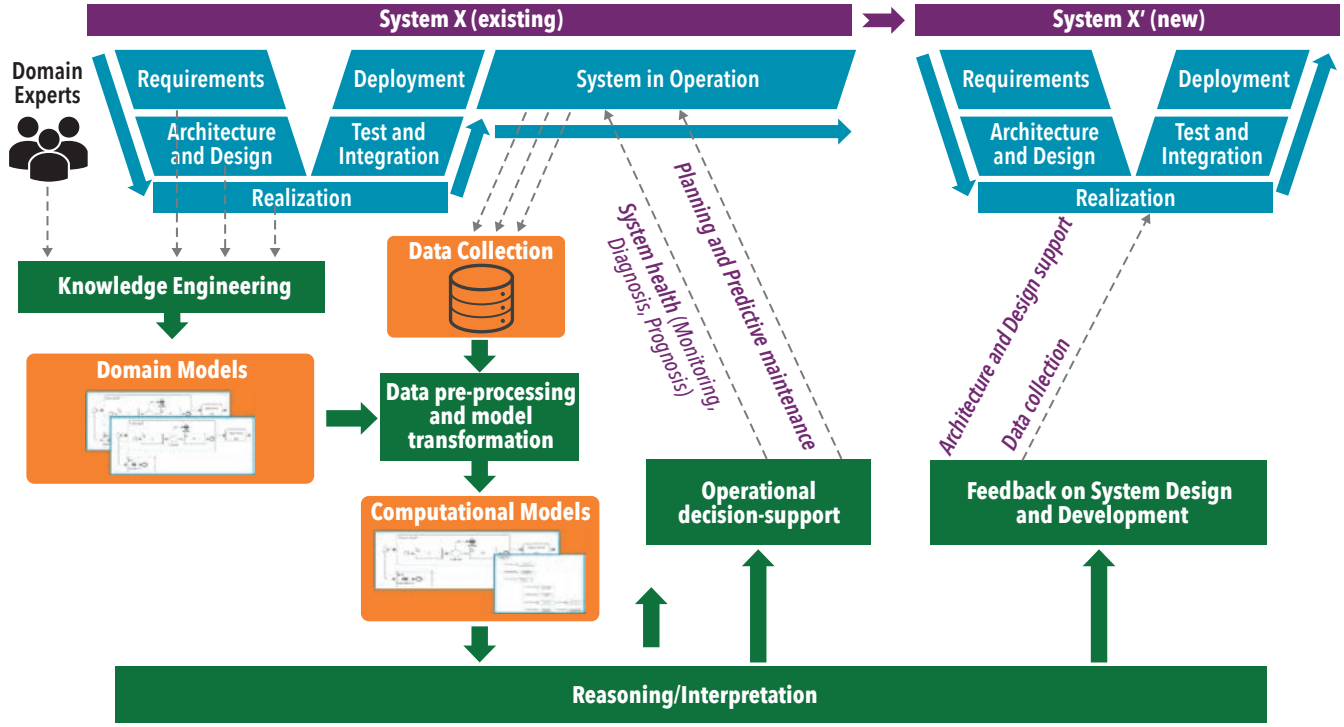


Figure 4. Reasoning and interpreting based on domain models and operational data is important throughout the whole system lifecycle, both for monitoring and diagnostics as well as for improving the next generation system

systems engineering approach primarily specifies the desired behavior of the system: the behavior the system is designed for. For diagnostics, the aspect of what happens if the system is not behaving as expected becomes relevant. This information is often not included in such a design. We aim to integrate diagnostics into the model-based systems engineering process by adding information about the system's failure manifestations. This allows for broader use of the models created during the design process.

METHODOLOGY

We developed a model-based methodology for diagnostics that embeds diagnostics through models throughout the lifecycle of a system, as shown in Figure 4. The diagnostic model – shown as a *Domain Model* in the picture – is based on design information, describes the system's expected behavior, and adds diagnostic-specific information.

This systematic approach improves the completeness of the final diagnostic deliverables for the service department, as its completeness no longer depends on engineers manually building them. Finally, the use of compositional models enables model reuse. This avoids the need for manually recreating all diagnostic deliverables upon every new system generation, as common parts in the machine do not need to be remodeled.

Building a diagnostic model of a system is challenging, as it takes enormous knowledge of the system. This information is not always readily available, as part of it resides in people's heads or is scattered throughout the organization. Once the domain model is created, it is transformed into a computational model and used through the whole system's lifecycle, serving as a single source of truth from a diagnostics perspective. The transformation is described in (Barbini et al. 2020) and (Barbini et al. 2021). Early during the design of the system, the model can help to assist in finding observability limitations to ensure the system remains diagnosable. When the system is deployed in the field, the model guides the service engineer to the best next step in getting the system up and running again. Figure 5 gives an overview of how a computational diagnostic model can be used, described in detail in the following subsections.

MODEL CREATION

During the development of a system, requirements are defined, and significant effort is put into getting the architecture and design correct. This is all covered in documents, drawings, and simulation models. These artifacts precisely how the system should behave and often contain in-

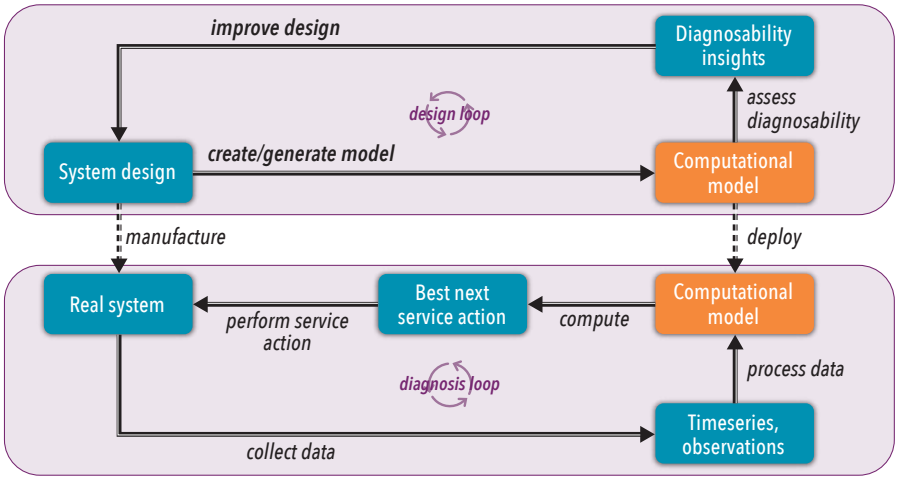


Figure 5. Workflow overview: during system design, the computational diagnostic model is used to assess the system's diagnosability; once the system is deployed, the same model aids the service engineer by suggesting the best action, such as recommending a particular test

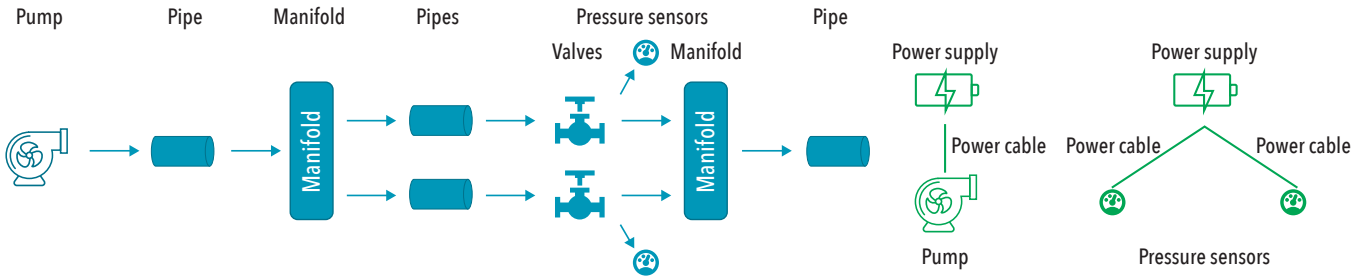


Figure 6. A schematic view on the design of a hydraulic system; the picture on the left shows the hydraulic view of the system and the picture on the right shows the electrical view; note that in this picture it is rather simple to map a component from one picture to another, but this is often not this straight-forward

formation about possible system malfunctions. Our approach formalizes the relevant information of this body of knowledge on the system design by creating a diagnostic model. This model describes the expected behavior of the relevant parts of the system as well as the possible faults that may occur. The behavior is specified per building block of the system – such as a component or a module – and these building blocks are used to compose a bigger system. This compositional approach reduces the complexity of the model creation, as the engineer only has to reason over faults and their effects at a local level. The composition of all those building blocks takes care of the propagation of the local fault effects to different parts of the system.

DOCUMENTATION

During the development of a system, plenty of design documents are delivered explaining how the system should behave, how to achieve this behavior, and why certain decisions are made. This information covers the process, from higher-level requirements and functional descriptions to highly detailed 3D drawings about the system mechanics. Altogether, these documents provide a good view of the system design, often in a semi-structured way.

Through knowledge engineering, the relevant design information for diagnostics is extracted from the documents and used to build a domain model for diagnostics. This is not always an easy task, as there could be many different views on the same system, and these do not always trivially map to each other. Figure 6 shows two different views on the same system. In this example, the pump and pressure sensors appear in both views, meaning that these components play their role in each of the shown domains: hydraulic and electrical. Combining all views into one coherent model is challenging.

Expert knowledge

Experts from the relevant fields of system design typically write the design documentation. These experts often make implicit assumptions based on what they consider trivial or common knowledge within their area of expertise. Consequently, certain pieces of the puzzle that describe the expected or failure behavior of the system are missing in the documentation. These pieces of the puzzle only reside in the heads of the experts who designed the system. In our experience, the strict modeling approach greatly aids knowledge elicitation and often reveals missing or seemingly contradicting information.

Tests

During the system's design, it is typically considered that failures are not fully observable. There are cases in which a service engineer should perform one or more tests on the system to identify the root cause of downtime. To guide the service engineer through the repair process, tests and replacement procedures are being specified. These specifications are included in the diagnostic model. In this way, the model guides the service engineer from start to end of a downtime, starting with root cause analysis through tests and finishing with the necessary repair action.

Data

While design documentation, expert knowledge, and test information are used to build the initial diagnostic model, data generated by the system, once deployed, can be used to increase the accuracy of the diagnostic model. A deployed system generates an enormous amount of data for control and performance monitoring. Additionally, the service engineers collect data when repairing the system during downtime. These combined data sources reveal component and system reliability, failure rates, and possibly unforeseen failure manifestations and their possible repair actions. Statistical analysis of this data improves diagnostic accuracy by fine-tuning its parameters. Feedback from the service engineer helps to find the best test strategy and is used to improve the quality of the model's diagnostic capabilities. Finally, data collected from a system of one generation is valuable to improve the diagnostic model and the system design of the next generation.

MODEL USAGE

The diagnostic model is used both at design time and during system operation.

Table 1. Fragment of an FMEA result for the example hydraulic system.

ID	Service part	Failure mode	Failure signature
FM-01	PSU for pump	Broken	Pressure_1: NoPressure, Pressure_2: NoPressure
FM-02	PSU for sensors	Broken	Pressure_1: NoReading, Pressure_2: NoReading
FM-03	Power cable Pump	Broken or Disconnected	Pressure_1: NoPressure, Pressure_2: NoPressure
FM-04	Power cable Sensor_1	Broken or Disconnected	Pressure_1: NoReading
FM-05	Power cable Sensor_2	Broken or Disconnected	Pressure_2: NoReading

Table 2. An example of a decision matrix that is generated to distinguish between the power cable or the PSU being broken.

Measurement	Power cable Pump	PSU for pump
Power outlet of PSU for pump	On	Off

At design time, the model is used to assess the system's diagnosability and to generate diagnostic deliverables. During operation, the model can assist the service engineer in performing the correct service action to get the system up and running as quickly as possible.

Design for diagnostics

Design for diagnostics aims to introduce diagnostic reasoning early in the design process. The goal is to deliver a diagnosable system to the customer. During the design, most focus is on the system's functionality. Techniques described above, such as FMEA and DT, are used when the design is almost finished, and no major changes to the design can be made to improve the system's diagnosability.

To assess the system's diagnosability early in the design phase, we propose to build the diagnostic model in parallel with the design. The diagnostic model can be used to see how a failure in one of the components affects the other components and, ultimately, the system behavior. The sensors in the system define observability, as these are the only places where data can be obtained without performing additional measurements. Our diagnosability analysis comprises an algorithm (Barbini et al. 2021) that assesses the observability of every failure in the system and expresses it in terms of expected sensor readings: a failure signature. All possible failure modes in the system and their failure signatures are collected in a Failure Mode and Effects Analysis (FMEA) table. A fragment of an FMEA for the small hydraulic system, as depicted in Figure 6, is shown in Table 1.

Note that this algorithm creates the failure signatures for a single failure, as computing signatures for all possible failure combinations is computationally infeasible.

The computed failure signatures are used to verify the visibility of failures and find ambiguities in the observability of failures. Since most systems are not fully observable there are typically multiple different failures that have identical readings on the sensors. In the example above, failures *FM-01* and *FM-03* give a "*No Pressure*" reading on the two available pressure sensors, indicating that an additional measurement is needed to distinguish between these failure modes.

The analysis computes a decision matrix for each of the groups of failures indistinguishable from each other. The decision matrix guides the service engineer to the root cause by suggesting additional points in the system to measure. By manually performing the additional measurements, the service engineer ends up with only one possible failure to repair. The decision matrix can also serve as advise for placing other sensors in the system. Table 2 shows an example of a decision matrix that helps to distinguish between *FM-01* and *FM-03*, as shown in Table 1. When both pressure sensors report an absence of water pressure, the table indicates to measure whether there is power at the outlet of the Power Supply Unit (PSU) for the pump. If there is power, the power cable towards the pump is broken, and if there is none, it must be a PSU failure. Based on this table, the system designer can also choose to put an additional sensor at the power outlet of the PSU to avoid the need for manual testing when the system is deployed. Following this approach brings the potential to improve the diagnosability of the system during the design so that the service engineer can repair the system more efficiently.

Using the model to automatically derive failure observability and ambiguity can help a system design. The approach helps design space exploration for placing sensors in the most effective places and reduces the manual effort in creating diagnostic deliverables. Once the model is changed, the deliverables follow automatically.

Diagnostics in operation

When the system is deployed in the field, the diagnostic model is used to assist the service engineer in determining the correct repair action. For this, the model is instantiated with the information logged by the system. The model computes a first diagnosis as a list of possible next steps. These steps could point to a possibly broken part that needs to be replaced or a suggestion to execute a diagnostic test, for example performing an additional measurement or reading a status light. In the latter case the service engineer performs the suggested measurement, supplementing the gathered data. The model then computes a new diagnosis. This iterative process is repeated until the model advises a repair action, that should resolve the issue. This process is illustrated by Figure 7.

CASE STUDIES

The diagnostic approach presented above has been applied to several industrial case studies. This section describes two of these applications.

The first case study concerns a lithography system's hydraulic cooling module that supplies water at a desired flow and temperature to several other modules with-

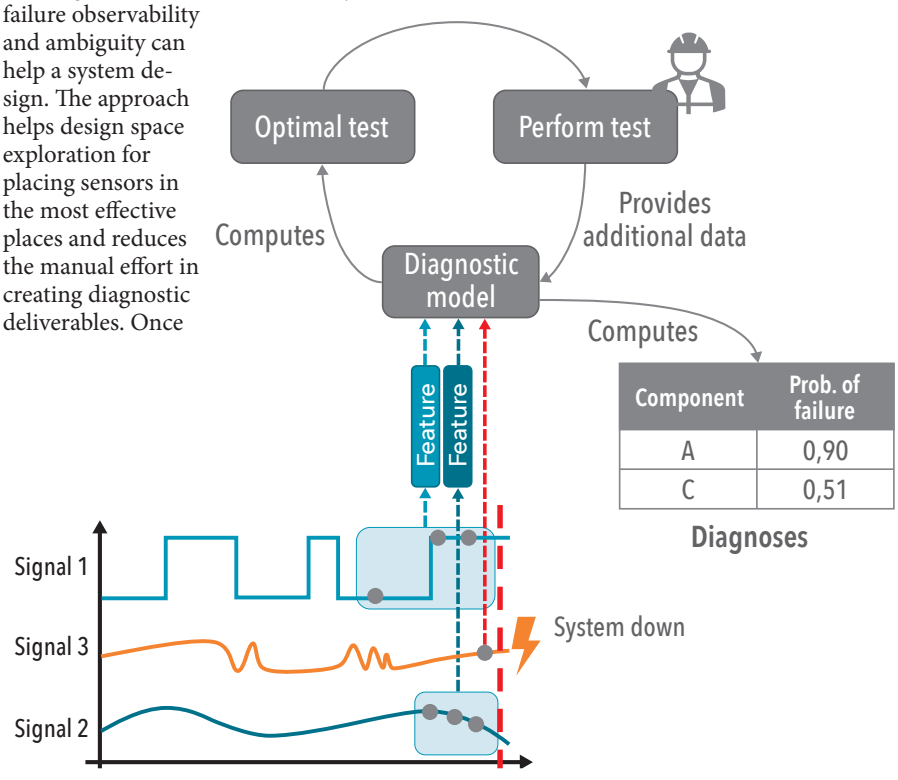


Figure 7. Our iterative approach uses signals and features computed from those signals to guide the service engineer to the right diagnosis through additional measurements

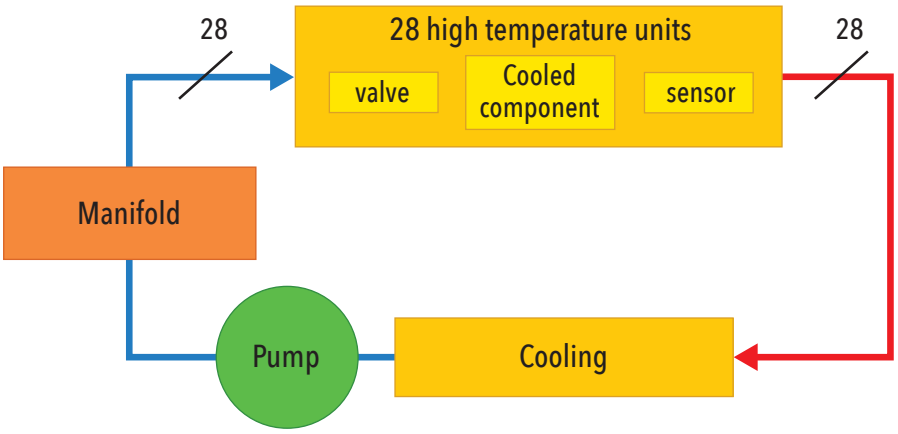


Figure 8. Schematic overview of the hydraulic cooling system

in the system. A schematic overview of the system is shown in Figure 8.

Containing more than 200 interconnected replaceable parts, the hydraulic cooling module is a good case study to test the scalability of the modeling. Despite its many components, the types for which the behavior must be specified are less than 20. Examples of part types in the module include pipe, valve, cartridge heater, differential pressure sensor, and temperature sensor. The small number of part types drastically reduces the time spent on modeling. To specify the behavior of the parts, the information present in documents was often sufficient, and only in a few cases more detailed discussions with experts were needed.

The interconnection of the large number of parts following the system specification presented a challenge: the module was described in several documents, each for a specific engineering domain, such as hydraulic, electrical, and safety logic, while the diagnostic model must be specified across these domains. The modeler manually merged all these descriptions into one model.

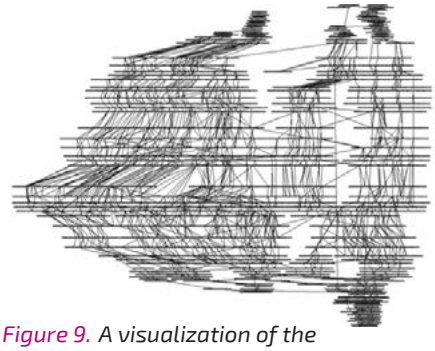


Figure 9. A visualization of the computational model of an industrial hydraulic system containing several thousands of elements; fortunately, no-one has to deal with these models directly

The diagnostic model generated the FMEA during the system's redesign, which experts reviewed. In this way, we could actively assess the value of our approach during design. The experts recognized that the proposed methodology creates an FMEA which is complete as it is derived from the system design and that the overall process offered substantial time savings since the FMEA was automatically generated.

Further, the methodology generated the expected failure signature per failing part. This allows designers to spot observability limitations, such as failing parts with identical signatures and failing parts with a signature equal to the system's expected behavior. These helped designers to reason where to add additional sensors to resolve diagnostics ambiguities.

The computational model generated for the module contains more than 500 elements and is shown in Figure 9. Even with such a large model, no computational scalability issues were identified during the case study, with computations requiring only a few seconds.

The second case study concerns the paper input module of an industrial printer, which transports a sheet of paper from one of the multiple trays to the printing module. The goal of this case study was to test the operational usage of the model. A database of historical service data for the industrial printer was used to add prior knowledge on the failure probability to maximize the diagnostic accuracy of the model.

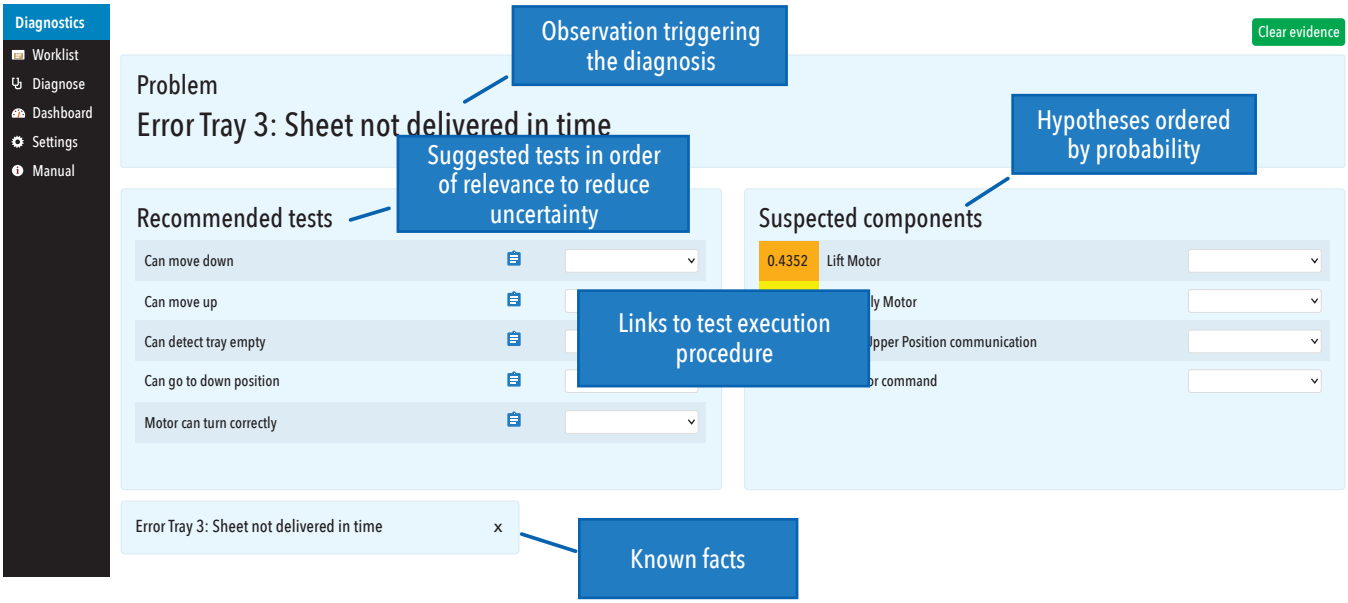


Figure 10. A graphical user interface guides the service engineer through the diagnostic process by suggesting next best tests, iteratively leading to a service action repairing the system

Figure 11. Stepping stones towards proactive performance diagnostics

As it is clear from the figure above, the size of the model quickly grows, making it very difficult for a service engineer to interact with it. In this case study, a graphical user interface was built on top of the model to allow user-friendly interaction. The interface is shown in Figure 10.

It is possible to insert the error triggering the diagnostic procedure on the *Problem* part. Based on the available machine data, the model presents an ordered list of the most likely serviceable parts to have failed, as shown in the *Suspected components* part. The order is given by the computed probability of failure based on the inserted evidence, the prior probabilities of component failure, and the modeled system behavior.

Due to the complexity of industrial systems often, several components have a similar failure probability. As a result, it remains unclear which part to replace in the system. To tackle such a situation, we developed an algorithm that uses the model to compute the most relevant tests to execute to increase diagnostic accuracy.

These results are shown on the *Recommended tests* part of the interface. Execution of one of the suggested tests provides additional observations to add to the model. The model then recomputes a new set of recommended tests, and this procedure is repeated, allowing the service engineer to conclude the diagnosis for the module iteratively.

RESEARCH TOPICS

The diagnostic methodology described in this paper primarily targets the diagnosis of a system that stopped completely. However, there are many other situations for which a system needs to be diagnosed. One example of such a situation is when a system still delivers a product, but the product is of insufficient quality: this is a performance problem. We are researching a general approach to extend our diagnostic models to support performance diagnostics. Modeling system components in more detail and defining performance-related features at the component level are being developed for performance diagnostics.

Another active area of research on performance diagnostics is investigating how to derive indicators at the system level, for example, on-product performance, from the system's components' performance indicators. For the example of a printer, a performance indicator at the system level is print quality. This derivation is challenging since many components, and external environmental and physical quantities influence on-product performance. Research is ongoing on how to augment the diagnostic models with knowledge of the functional decomposition of the system from the model-based systems engineering workflow. This information will then allow modeling of a degrading component's effect via its function on the on-product performance.

Prediction of decreasing performance to avoid unscheduled downtime is the next research challenge and a highly desired outcome by many industries. However, research in this domain is still ongoing at the component level and is only in its infancy on the system level. Prediction in industrial systems is also complicated by the presence of control software, which by design compensates, at the subsystem level, for the behavior of a degrading component. As a result, predictions cannot rely solely on the capability of a module to fulfill its function but must also factor in the amount of control needed.

CONCLUSION

The increasing system complexity and product introduction cycles combined with a shortage of well-trained and experienced service engineers require a change in the service department's way of working. Furthermore, we see that service and service organizations become more critical for businesses. Connecting earlier approaches, such as rule-based systems and graphical probabilistic models, to model-based engineering practices leads to a scalable, maintainable model that drives the diagnostics of system failures.

This paper showed the added value of using diagnostic models both during the design of a system and during operation when the system needs to be diagnosed. The model is used to assess the diagnosability of a system and can be used to compare different configurations of sensors with each other. Additionally, it guides the service engineer to the proper repair actions for a broken system.

With the increased adoption of model-based systems engineering methods in the industry, our methodology aims to improve computational diagnostics by leveraging the design information created by those methods. Our vision is to embed diagnostics in model-based systems engineering methods to deal with the system complexity and reduce the overhead needed to build a diagnostic model. This way, the diagnostic improvements, and guided diagnosis will come at little additional cost. ■

ACKNOWLEDGMENTS

The research is carried out as part of the SD2ACT and Carefree projects under the responsibility of TNO-ESI in cooperation with ASML and Canon Production Printing, supported by the Netherlands Ministry of Economic Affairs, the Netherlands Organisation of Applied Scientific Research TNO, and TKI-HTSM.

REFERENCES

- Barbini, L., C. Bratosin, and E. van Gerwen. 2020. "Model based diagnosis in complex industrial systems: a methodology." In PHM Society European Conference.
- Barbini, L., C. Bratosin, and T. Nägele. 2021. "Embedding Diagnosability of Complex Industrial Systems Into the Design Process Using a Model-Based Methodology." In *PHM Society European Conference*, 2021.
- Console, L., and O. Dressier.1999. "Model-based diagnosis in the real world: lessons learned and challenges remaining." In *IJCAI'99: Proceedings of the 16th international joint conference on Artificial intelligence*.
- Scherer, W. T., and C. C. White III. 1989. "A Survey of Expert Systems for Equipment Maintenance and Diagnostics." In *Knowledge-Based System Diagnosis, Supervision, and Control*, Springer New York, US-NY: 285-300.

DECEMBER 2022
VOLUME 25/ ISSUE 4

Distilling Reference Architectures in the High-tech Equipment Industry

Richard Doornbos, richard.doornbos@tno.nl; **Jelena Marincic,** jelena.marincic@tno.nl; **Alexandr Vasenev,** alexandr.vasenev@tno.nl; and **Jacco Wesselius.** jacco.wesselius@tno.nl

■ **ABSTRACT**

Companies in the high-tech equipment industry are continuously looking for ways to optimize their business. A notoriously difficult part of optimizing is the R&D activities, as risks and uncertainties are inherent. In our experience, creating and using a reference architecture for a product or portfolio to guide future developments is a good way to improve R&D effectiveness and efficiency. But developing a reference architecture by capturing the relevant information and establishing the structure, the models and their interrelations, the tools, and secondly, getting clarity on how to use such reference is not easy. In this article, we describe a method to 'distill' a reference architecture using the knowledge built-up in years of developing products and using the customer and business values to capture the key architectural decisions for future products. We explain the purpose and usage of a reference architecture and how to organize it. The experiences obtained in Thermo Fisher Scientific have proven the importance and practicality of this approach.

■ **KEYWORDS:** reference architecture; platform architecture; architecting; transmission electron microscope

INTRODUCTION

ESI (TNO-ESI 2022) is a Dutch open innovation center for systems design and engineering in the high-tech equipment industry. To learn more about ESI as an organization, we invite the reader to take a look at the introductory article of this section. One of the ESI's research competencies is systems architecting which relates business vision to technical decisions. These relations are not always obvious - business managers and technical experts use different "languages," and typically, within organizations, they are not set up to communicate daily. The system architect's role is to ensure these two worlds are in sync. We build methods to support the systems architect in this role.

The methods focus on building models to describe the essential relations between business and technical aspects, thus making them explicit. In a nutshell, we support R&D departments in increasing their effectiveness by using systems architecting.

Our partner network consists of OEM (Original Equipment Manufacturer) companies that sell their equipment to other businesses. These machines are highly complex high-tech systems designed by different domain experts, including mechanical, electronic, electrical, software, chemical engineers, and physicists. The complexity makes it difficult for one expert to grasp the whole system. The complexity is multiplied even further because these companies are multi-site, multi-supplier (in an eco-system), and multi-country (and therefore multi-culture).

At the partners' customers, these equipment systems are a part of a larger workflow in which other systems and machines achieve the customer's goals. It is, therefore, important for OEMs to understand the needs of their customers and how their equipment participates and contributes to achieving that goal.

These manufacturers are *the* leaders in their respective domains. The market forces them to improve their products and services to stay on top constantly. As a result, new and improved systems are offered regularly.

The systems improve their performance and qualities and come with new features to better accommodate customer needs. These multimillion-dollar products stay in the field for a long time, sometimes decades. The market requires backward compatibility, so the systems' old versions get upgrades and improvements. All the above comes with yet another push, which is a shorter time to market.

For these reasons, it is essential to be flexible and provide the machines that fit the requirements of their partners and simultaneously deliver the product on time. The latter requires the efficiency and effectiveness of R&D. One of the ways to increase R&D efficiency is to reuse technical solutions. For these reasons, platforms are introduced. From a platform, components are chosen to be integrated into different product lines and families. Another way of improving efficiency is to develop and use a *reference architecture* for the platforms and portfolio of products.

In this article, we first describe what reference architecture is and why it is created and used. Next, we describe a

valuable way of structuring the information into layers, allowing straightforward reasoning across the various layers and perspectives. An example of our practical experiences is briefly shown in the section that describes the TEM reference architecture of Thermo Fisher Scientific. The last section puts things into perspective by discussing observations and best practices from our practical work.

SCOPE AND PURPOSE OF REFERENCE ARCHITECTURES

The term reference architecture is widely used for varying concepts. To understand the methodology described in this document, it is important to grasp what we consider a reference architecture.

Reference Architectures (RAs) capture the essence of existing architectures and the vision of future needs and evolution to guide in the development of new architectures (Cloutier, et al. 2010). In the methodology described in this paper, the reference architecture also intends to capture the company's technical *strategy* to relate long-term value for customers to long-term business value. Therefore, the reference architecture must provide a shared vocabulary, a shared architectural vision, and rationales. This approach is consistent with how the US Department of Defense (DoD) defines reference architecture: "an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions" (DoD 2010).

Note that multiple RAs can simultaneously exist within a subject area where each

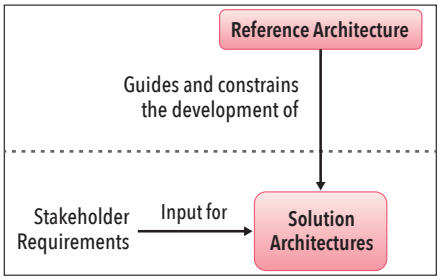


Figure 1. The purpose of a reference architecture according to DoD (from (DoD 2010))

represents a different aspect (DoD 2010). For example, a reference architecture for communication can co-exist with one for systems safety.

A reference architecture should not aim to detail every architectural structure and aspect. It addresses a specific area with a suitable level of detail and abstraction. It should only capture the core aspects to *guide* architectural decision-making. There should be enough freedom to allow specific product requirements to be fulfilled across a product family or for particular customers. Therefore, reference architectures can be very different for different areas and organizations.

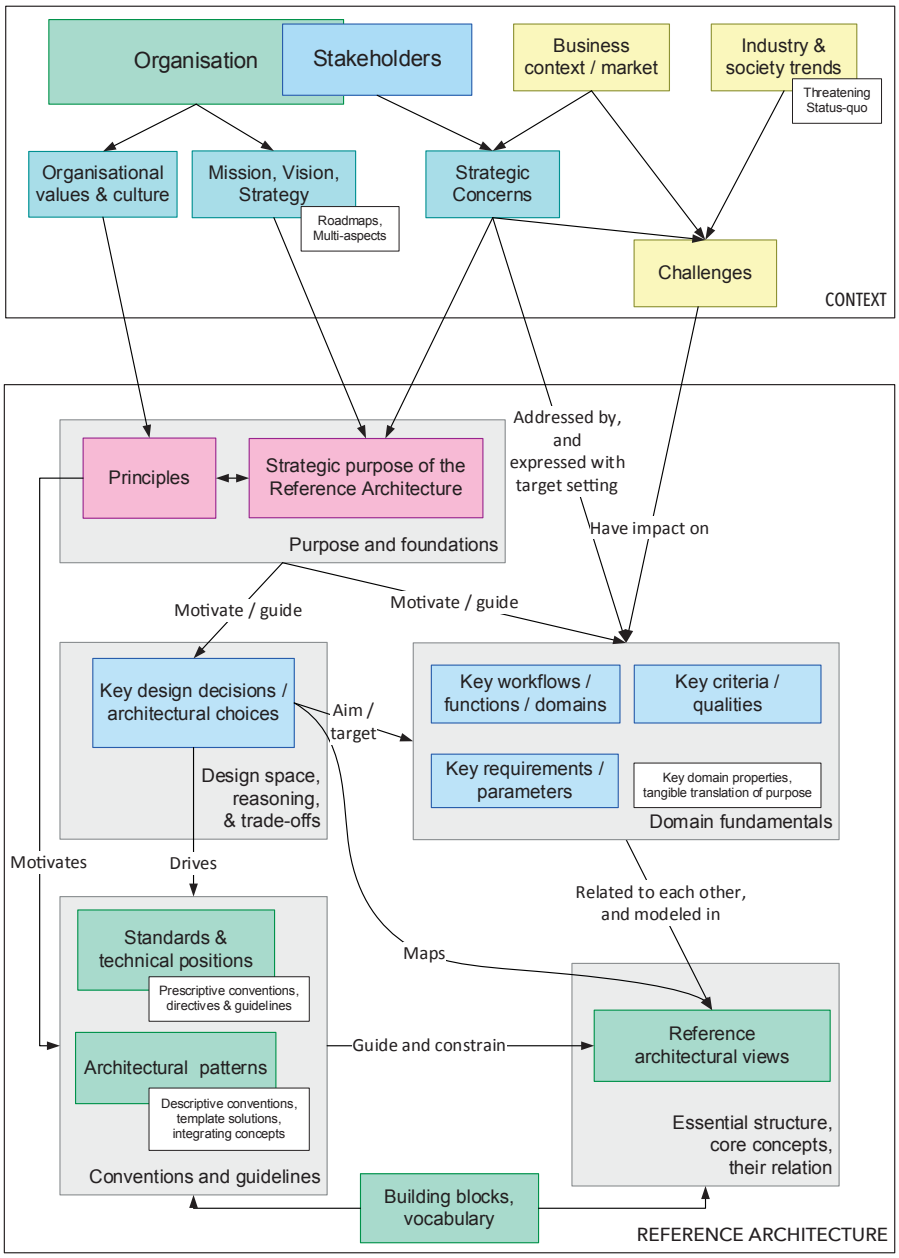


Figure 2. Overview of a generic reference architecture and its context (TNO-ESI 2021)

The *contents* of a reference architecture include aspects relevant to the Technical architecture, Business architecture, and Customer context. (Cloutier, et al. 2010). These may consist of elements of strategic purpose, vocabulary, principles, methods (as technical guidance), patterns, technical architecture (high-level system-wide architecture blueprint), standards, and tools (DoD 2010).

Reference architectures contain the relevant concepts to address the company's needs to help architects in their work. Figure 2 shows the elements of a reference architecture and its drivers (Vasenev and Hendriks 2019, Vasenev 2018). It combines concepts from multiple sources (DoD 2010, Eklund and Bosch 2014, Bach, Otten and Sax 2017, Pelliccione et al. 2017) and can serve as a generic model for reference architectures and their context.

Summarizing, a reference architecture guides platform and product/solution architects; see Table 1. Reference Architectures bring value by considering the key value and business drivers and relating them to key architectural decisions. A common language and validated solution patterns provide consistency in the realized products and enable easier adherence to standards. As a result, product archi-

Table 1. Characteristics of reference architectures

	Reference
Goals	Guides and constraints instantiations of multiple architectures and solutions
Contents	Well-founded guidelines for architecting decisions in technical, business, and customer contexts
Direct user	Platform architects, product and solution architects
Inputs	Business, organizational, industry- and societally-driven aspects
Usage	Architectural decisions for product and platform development, strategy definition

tects can effectively work on multi-site, multi-supplier, and multi-vendor product creation (Muller 2020). Platform architects can focus on the platform strategy for dynamic environments that impose many changes on the products (Muller 2008).

THE STRUCTURE OF A REFERENCE ARCHITECTURE

It is crucial to establish a way to structure the reference architecture elements, as it will help to explain and focus the next steps. It clarifies the building blocks of a reference architecture, how they are related, and how they contribute to its purpose.

Elaborating on the model in Figure 2, we focus on organizing the Reference Architectural Views (bottom right). Inspired by the CAFCR methodology (the acronym denotes Customer, Application, Functional, Conceptual, Realization) (Muller 2020), we created a 6-layer structure to organize the information blocks (Doornbos, Marincic, et al. 2021) that can be considered ISO 42010 *Views* (ISO/IEC/IEEE 2011), see Figure 3.

In the 6-layer structure, the relations between the layers are made explicit. Models positioned at a layer are linked to models at the next higher and lower layers. This structure enables both top-down and bottom-up reasoning (as indicated by the arrows in Figure 3) by creating *chains of reasoning*:

- *Top-down* they explain how customer and business values are realized: via a workflow description, the involved functions are described, which in their turn are mapped on system components and realizations;
- *Bottom-up* they express how properties of components (realizations in the lowest layer) influence values, via their impact on functions, workflows and – ultimately – system qualities.

Each layer contains information that abstracts from more detailed and complete models, descriptions, formulas, and documents. The most important and difficult task here is finding the right level of detail, where a product architecture should be complete and more detailed, and a reference architecture should only describe and constrain (DoD 2010) the elements of strategic concern. Deciding which elements to describe and which constraints to apply are a matter of architectural strategy closely linked to the organization's business strategy.

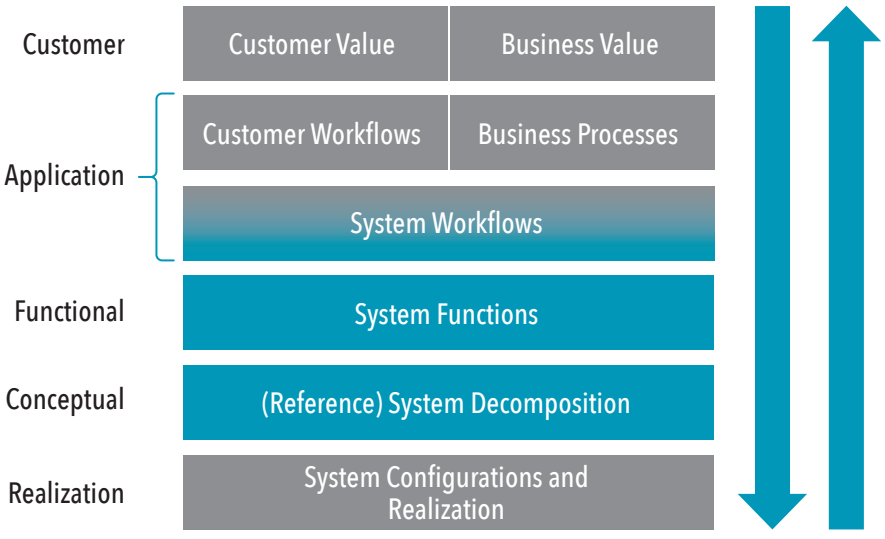


Figure 3. The 6-Layer reference architectural model. On the left, the CAFCR aspects are shown for reference

LAYER 1: CUSTOMER VALUES AND BUSINESS VALUE

Customer and business values are addressed in marketing research and in a branch of modern systems engineering called Value-Driven Design (VDD). Customer value can be defined (Anderson, Jain and Chintagunta 1992) as technical, economic, service, and social benefits that a customer receives in exchange for the price they pay. In this layer of the reference architecture, those values are currently addressed and will be addressed in the future as part of the business strategy.

The customer values can be structured in a causal diagram and related to the other layers: customer and system workflows, system functions, and system decomposition. These relations guide the key architectural decisions on reference workflows, abstract functional and system decompositions, and reference interfaces.

LAYERS 2 AND 3: WORKFLOWS

How customers achieve their goals is addressed in the Customer Workflows layer. Here, the "what" of the customer is translated into the "how." A high-tech system typically performs a task or several tasks for the customer as part of *their (business) process*. This process can be expressed in a workflow model consisting of a sequence of steps, which can also occur in parallel. Many customer values depend on the properties of these workflows. The workflow models can be separated according to ownership:

- customer workflows: the overall workflow of a customer to realize a value;
- system workflows: the part of customer workflows in which the system of interest is being used, describing the system's workflow and clarifying the system's contribution to the customer's value creation.

LAYER 4: FUNCTIONAL ARCHITECTURE

The functional architecture is a view of the system that describes *what* the system does. It is a crucial central perspective that links to many other system views. For instance, a system workflow step (layer 3) involves the *execution* of a system function

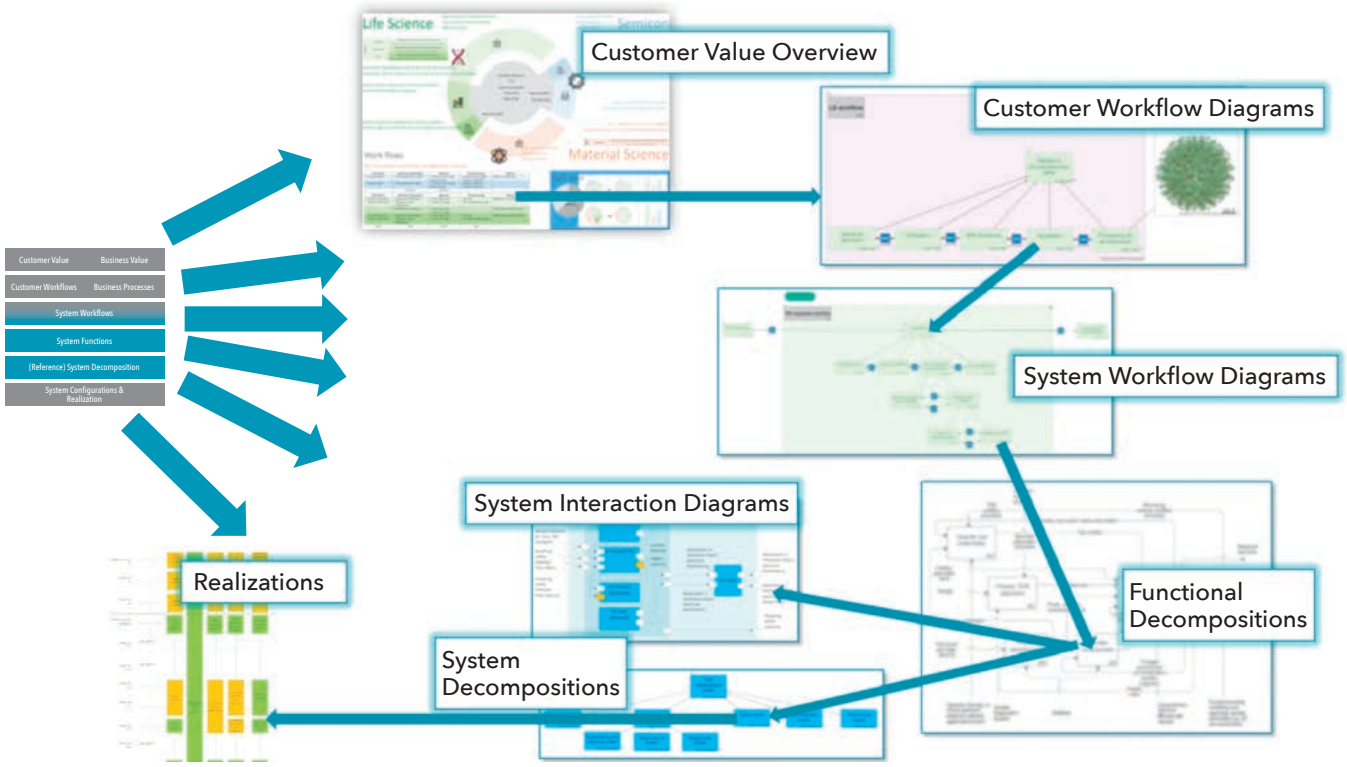


Figure 4. Overview of the TEM reference architecture (Doornbos, Marincic, et al. 2021) (Doornbos 2021). The models in the layers are connected explicitly, as shown by the thin arrows

(or of multiple functions). Furthermore, a system function is allocated to a system component (layer 5).

A *system function* transforms inputs (information, energy, material) into outputs, performed by the system under study. Each function can be decomposed into smaller functions that can be described separately, typically called functional decomposition. The functional decomposition is usually expressed graphically as a tree of functions. Other model types commonly used in this layer are behavioral and timing models.

LAYER 5: SYSTEM DECOMPOSITION: (SUB-) SYSTEMS AND INTERFACES

Decomposing the system into sub-systems and their interfaces delivers the elements of the structural part of the reference architecture. A system decomposition consists of the following:

- a collection of subsystems, where each subsystem has a collection of defined interfaces;
- a collection of connections between the interfaces of these subsystems;
- an allocation of functions to the subsystems;
- an allocation of the flows between the functions to connections between the subsystem interfaces.

A good system decomposition and system structure enable the realization of the system functionality specified by the functional decomposition and realization of key system qualities. Note that the descriptions in the layers are abstract–they generalize from product or platform architectures. As such, they have fewer hierarchy levels than the system tree and only describe some of the interfaces in full detail. Next to decomposition models, also component models, interface models, and interaction models can be used.

LAYER 6: REALIZATIONS

The Realization layer describes concrete implemented components or building blocks used to realize a product. This layer shows which individual components can form groups to work together seamlessly, usually in the form of a variability or system configuration model. These individual components can be further configured during the manufacturing of the product. Typically, realization elements have their lifecycle and roadmaps and are preferably modular and easy to exchange.

Even though the realization layer is, strictly speaking, not part of the reference architecture, it is a valuable source of information. It couples the reference architecture (and its system decomposition) to the reality of systems that have been manufactured in the past (the installed base) and about systems that are (or will be) manufactured. Apart from being a source of inspiration for creating the reference system decomposition, it is used to:

- identify gaps and differences between the reference system decomposition and the reality of current systems; the discrepancies can be used to identify (i) errors or over-simplifications in the abstract system decomposition; (ii) elements that need to be put on the architectural roadmap to improve the design of systems that are currently being manufactured and shipped;
- identify variation points needed in the reference architecture by creating an overview of the full system configuration landscape, providing an overview of system variants in the field and those rolling out of the factory;
- make informed decisions about the product variants that will be commercially supported (as new shipments and in-service) in configuration management processes.

AN EXAMPLE: A REFERENCE ARCHITECTURE FOR TRANSMISSION ELECTRON MICROSCOPES

An example of a complex product is a transmission electron microscope (TEM). In a TEM, an electron beam is used to illuminate a specimen. The electron beam is accelerated and directed through a vacuum, using high voltages and electric and magnetic fields. Once the beam passes

through the specimen, detection systems, and cameras create an image. Using various configurations and technical options, a large variety of biomedical and material science applications is enabled.

The need to create a reference architecture for the portfolio of TEMs in Thermo Fisher Scientific comes from the fact that (1) the systems are getting larger and more complex, (2) system customization is increasing – more variation and late customization options for customers, (3) more market dynamic mean faster time-to-market needed – being early is more important than being perfect, (4) there is a growing legacy, and is related to that configuration management complexity, (5) the software contribution to the system is growing meaning that software is becoming a key element for all the above. And all this without resources and the number of people in the organization increasing very fast (McCormack 2021) (McCormack 2021).

A Reference Architecture's purpose is to guide the development of architectures for new versions of the system or extended systems and product families. It targets maximizing development effectiveness (the customer and business impact) and development efficiency (speed, cost, and effort).

Using the structure as described above, we established a Reference Architecture. Doing so resulted in many benefits (McCormack 2021) (McCormack 2021): concrete, connected models of the systems (see Figure 4) used for many purposes, such as mapping customer values to product configurations, managing modules and their ownership, and explicit linking beyond architecting to design activities, quality management, and operations.

DISCUSSION

A reference architecture can provide multiple benefits while demanding some effort. According to (Martinez-Fernandez et al. 2015) benefits include interoperability, reduced development costs and time-to-market, improved communications, and reduced risks. Some extra effort is necessary to adapt existing processes and learn new terminology. Our experience shows that the adoption of the reference architecture is successful when attention, incentive, and time are given to its introduction (McCormack 2021) (McCormack 2021).

Interestingly, we observed that constructing a reference architecture (RA) also provides benefits, highlights attention areas, and calls for tailored approaches. In this section, we reflect on our experiences.

At the beginning of constructing a reference architecture, direct benefits include building shared understanding between stakeholders and communicating best practices. For instance, stakeholders can respect the space to reason on the level of reusable constructs. The result can directly be documented as simple figures, such as high-level functional and system decomposition, and fed back to the development process. In addition to being directly useful, they spark interest in more formal modeling with MBSE languages and tools.

Maturing a reference architecture as the next step raises multiple questions and dilemmas. For instance, how to avoid capturing too much of earlier and future solutions? Was sufficient depth of design rationale and constraints already identified? When diagrams and models become more accepted, how to restrict architects' drive to capture as much system knowledge and design decisions as possible? Which tools to use (whiteboard, Visio, prototyping tools, MBSE solutions)?

Addressing these questions should respect the project purpose, specifics, and concerns of involved stakeholders. Several guiding principles assisted us in modeling:

- Avoid direct re-use of artifacts. As the purposes of existing models are often specific, their direct re-use is typically impossible. The models can have too much or too few details, outdated or irrelevant info, or not act as representatives for the larger scope. Depending on the availability and quality of documentation, one would have to abstract and generalize design concepts, mechanisms, and architectural intentions.
- Deal with an abundance of information while tolerating incompleteness. Prudence is needed to handle a tremendous amount of information (including undocumented expert knowledge) without details of particular applications/product lines/market segments. It should account for when to start and when to stop modeling. The modeler shall identify what is relevant, capture remarks for future iterations, and clarify assumptions.
- Keep focus via traceability links. The purpose of each RA layer should be clear to all stakeholders. The layer's elements should be traceable to key constructs of higher-layer constructs and business goals. For instance, how a component contributes to a customer's workflow.

To have early, clear, and continuous communication with key stakeholders, we adopted some guidelines:

- Start small and ensure solving relevant and feasible tasks. Senior management should support the core team in maintaining the vision. Domain experts, in turn, shall ensure coverage of sufficient details, correctness, and meaningfulness of architectural elements.
- Choose key representative stakeholders wisely, as many people will interact with the reference architecture.
- Choose a format that is easy to explain to non-technical stakeholders and allows for essential differences to be highlighted. For instance, IDEF0 can support function-focused discussions. A selected subset of SysML constructs can help discuss (sub)components and interfaces.
- Expect and resolve confrontations caused by different departure points. For example, reasoning from business values will lead to results that differ if one departs from Customer values. While such conflicts happen at all levels, they are prominent at the system decomposition level. An answer can be to do activities in parallel (different system decompositions), align regularly, and keep the storylines consistent. Short update cycles shall incorporate prototyping and reviews.

Constructing a company-specific reference architecture goes beyond technical knowledge elicitation and modeling. It also calls for softer skills and a way of working sufficiently embedded into an organization. This implies numerous interactions with key stakeholders, familiarity with the organization's business and culture, and accessibility of key experts. Open and in-depth content discussions are a prerequisite for it. Several aspects empowered the construction of the reference architecture. First, as ESI, we act from the perspective of an independent not-for-profit expertise group on methodologies with long-term relations with the organization. Trust-based relationships with stakeholders, their goodwill, and practical focus facilitated the process. Such aspects are critical, as developing a reference architecture can take substantial time (depending on the size of the company and the number of people involved) to identify, mature, and communicate proper abstraction levels. Second, the chosen bottom-up and technical approach ensured the practicality of outcomes and assisted in addressing on acceptance of the results. Finally, we approached the construction of a reference architecture as an applied research project to address how to apply generic concepts in daily practice in the company's context.

In our project, several factors positively contributed to embedding the reference architecture into the organization:

- Support and demand for changes by

company management, next to drive and leadership-by-example from the in-company champion.
- Direct value identification, for example, to construct a (missing) abstraction level useful for communication and the introduction of new employees.
- Clarity of how reference elements help individuals (such as avoiding unnecessary modeling)
- Using RA for guiding and constraining decisions was aligned with in-organization processes. One example is clear ownership of reference architecture elements and each function.

High-tech companies can similarly organize their reference architecture developments. The process itself will spur many opportunities and questions. We shared some of our observations. Yet, one can reasonably expect that the softer aspects will call for a tailored approach to make the process of constructing a reference architecture and its outcomes a success.

CONCLUSION AND FUTURE WORK

In this article, we presented a framework for distilling reference architectures. We described (1) a high-level overview of the method to create reference architectures of high-tech systems and (2) discussed the practical aspect of introducing and building reference architectures in an organization.

The method we have been designing is our generic deliverable, described in more detail here (Doornbos, Marincic, et al. 2021). We are validating the methodology at another partner company establishing reference and platform architecture. In this process, we have discovered that the generic method has to be further specialized for specific technical and business contexts. We are also extending the method with rationales for the most critical decisions.

Regarding the practical aspect, the reference architecture, built with Thermo Fisher Scientific, created impact and continues to be used successfully (McCormack 2021) (McCormack 2021). It resulted in assigning new ownerships and establishing new or adjusting existing processes to enable effective use of the reference architecture. The reference architecture will be further extended when needed. But for now, the current reference structures have reached a level of maturity; they are stable and do not require regular changes.

A practical challenge for each company remains the selection of the languages, formalisms, and tools to document reference and platform architectures. In the Eindhoven region, companies are starting to adopt an MBSE way of working. Therefore, formalism and tool selection is currently a part of this challenge. ■

ACKNOWLEDGMENTS

The research is carried out as part of the PaloAlto program under the responsibility of ESI (TNO) in cooperation with Thermo Fisher Scientific. The research activities are supported by the Netherlands Ministry of Economic Affairs and TKI-HTSM.

We want to thank Thermo Fisher Scientific architects and management for a fruitful collaboration and excellent and inspiring discussions.

REFERENCES

- Anderson, J.C., D.C. Jain, and P.K. Chintagunta. "Customer Value Assesment in Business Markets: A State-of-Practice-Study." *Journal of Business-to-Business Marketing* (Routledge) 1 (1992): 3–29.
- Bach, J., S. Otten, and E. Sax. "A taxonomy and systematic approach for automotive system architectures-from functional chains to functional networks." *International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2017)*. Porto Portugal, 2017. 90-101.
- Cloutier, R., G. Muller, D. Verma, R. Nilchiani, E. Hole, and M. Bone. "The Concept of Reference Architectures." *Systems Engineering* (Wiley InterScience (www.interscience.wiley.com)) 13, no. 1 (2010): 14-27.
- DoD. "Reference architecture description." June 2010.
- Doornbos, Richard. "Towards a method for creating reference architectures - a journey together with Thermo Fisher Scientific (video)." *Towards a method for creating reference architectures - a journey together with Thermo Fisher Scientific (video)*. 2021.
- Doornbos, R., J. Marincic, A. Vasenev, and J. Wesselius. "Reference architectures for product families." techreport, TNO ESI, 2021.
- Eklund, U., and J. Bosch. "Architecture for embedded open software ecosystems." *Journal of Systems and Software* 92, no. 1 (2014): 128-142.
- ISO/IEC/IEEE. "ISO/IEC/IEEE 42010:2011, Systems and software engineering — Architecture description." Tech. rep., International Organization for Standardization, 2011.
- Martinez-Fernandez, S., P.S.M. Dos Santos, C.P. Ayala, X. Franch, and G.H. Travassos. "Aggregating Empirical Evidence about the Benefits and Drawbacks of Software Reference Architectures." *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 2015. 1-10.
- McCormack, J. "Reference Architecting (video)." *Reference Architecting (video)*. 2021.
- ———. "Reference Architecting Track." *Reference Architecting Track*. April 2021.
- Muller, Gerrit. "A Reference Architecture Primer." September 3, 2020. https://www.gaudisite.nl/ReferenceArchitecturePrimerPaper.pdf (accessed November 4, 2020).
- ———. "Architectural Reasoning; CAFCR." Vers. Version 3.4. *Gaudí Systems Architecting*. October 11, 2020. https://www.gaudisite.nl/ArchitecturalReasoningBook.pdf (accessed November 23, 2020).
- ———. "How Reference Architectures support the evolution of Product Families." January 2008. https://www.gaudisite.nl/CSER2008_Muller_EvolvabilityRA.pdf (accessed November 4, 2020).
- Pelliccione, P. et al. "Automotive Architecture Framework: The Experience of Volvo Cars." *Journal of Systems Architecture* 77 (March 2017).
- TNO-ESI. *ESI website*. 2022. https://www.tno.nl/en/about-tno/organisation/units/information-communication-technology/embedded-systems-innovation/ (accessed December 22, 2022).
- ———. "Reference architectures for product families." 2021. https://esi.nl/research/output/methods/reference-architectures-for-product-families (accessed February 12, 2021).
- Vasenev, A. "Structuring of Methods to Estimate Benefits of Partial Networking." Edited by Oleg Gusikhin and Markus Helfert. *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018)*. Funchal, Madeira, Portugal: SCITEPRESS – Science and Technology Publications, Lda, 2018. 575-581.
- Vasenev, A., and T. Hendriks. "Structured problem exploration approach for the pre-concept stage of system development." *14th Annual Conference System of Systems Engineering (SoSE)*. Anchorage, AK, USA: IEEE, 2019. 370-375.

ABOUT THE AUTHORS

Richard Doornbos is a senior research fellow with Embedded Systems Innovation (ESI), a part of the Netherlands Organization for applied scientific research (TNO). He conducts applied research on system architecting methods for complex high-tech systems. He currently investigates how to apply model-based systems architecting and systems engineering techniques in multidisciplinary architect teams, focused on reference architectures, system modeling, and aspects such as architecting skills, tool support, and team cooperation.

His competencies include systems architecting, systems thinking, road mapping, facilitation and coaching, modeling techniques, and their applications; in a company engineering context.

Jelena Marincic is a senior research fellow with Embedded Systems Innovation (ESI), a part of the Netherlands Organization for Applied Scientific Research (TNO). Her specialisms are model-based software and systems architecting. As part of her career, she advised leading global companies in The Netherlands on introducing model-based methods, techniques, and tools, including ASML, Thermo Fisher Scientific, and Canon CPP.

Alexandr Vasenev is an experienced researcher with a systems background who focuses on developing and applying design methodologies. His interests include eliciting requirements, analyzing systems, and creating user-oriented solutions. His work identifies practical methods to create and apply platforms and reference architecture in the enterprise context.

Jacco Wesselius is currently the business director at ESI (TNO). In his previous role as Senior Program Manager at ESI, he guided the MBSE study, including the collaboration with the industry sector. Before joining ESI (TNO) in 2018, he held various positions as a Program Manager and System Architect in the High-Tech Equipment industry. Jacco received a Ph. D. degree in Computer Science/Software Engineering from Delft University of Technology (1992).

Neurohr and Möhhlmann continued from page 50

- Zhang, X., J. Tao, K. Tan, M. Torngren, S. Gaspar, M. Jose, M. Ramli, X. Tao, M. Gyllenhammar, F. Wotawa, N. Mohan, M. Nica, and H. Felbinger. 2022. "Finding Critical Scenarios for Automated Driving Systems: A Systematic Mapping Study." in *IEEE Transactions on Software Engineering*. doi: 10.1109/TSE.2022.3170122

ABOUT THE AUTHORS

Birte Neurohr (née Kramer) received her BSc and MSc degrees in mathematics from the Carl von Ossietzky University Oldenburg in 2015 and 2017, respectively. She is currently pursuing a PhD degree with the Institute of Systems Engineering for Future Mobility, German Aerospace Center. Since 2017 she has been a researcher in scenario-based verification and validation of automated vehicles. From 2017-2021 she was with OFFIS e.V, and since 2022 she has been a researcher with the newly founded Institute of Systems Engineering for Future Mobility, German Aerospace Center. Her research interest includes simulation validity and quality criteria for simulations.

DR.-ING. Eike Möhlmann is leading the group for "Evidence for Trustworthiness" at the Institute of Systems Engineering for Future Mobility, German Aerospace Center. His main interests are methods for analyzing and optimizing cyber-physical systems with a strong focus on ensuring and assuring trustworthiness, especially safety and security. He obtained a PhD in control and automated stability analysis from the University of Oldenburg. His professional experience includes formal modeling, analysis, simulation, and testing.

Pairing Bayesian Methods and Systems Theory to Enable Test and Evaluation of Learning-Based Systems

Paul Wach, paulw86@vt.edu; **Justin Krometis,** jkrometis@vt.edu; **Atharva Sonanis,** asonanis@purdue.edu; **Dinesh Verma,** dinesh.verma@stevens.edu; **Jitesh Panchal,** panchal@purdue.edu; **Laura Freeman,** laura.freeman@vt.edu; and **Peter Beling,** beling@vt.edu

ABSTRACT
Modern engineered systems, and learning-based systems, in particular, provide unprecedented complexity that requires advancement in our methods to achieve confidence in mission success through test and evaluation (T&E). We define learning-based systems as engineered systems that incorporate a learning algorithm (artificial intelligence) component of the overall system. A part of the unparalleled complexity is the rate at which learning-based systems change over traditional engineered systems. Where traditional systems are expected to steadily decline (change) in performance due to time (aging), learning-based systems undergo a constant change which must be better understood to achieve high confidence in mission success. To this end, we propose pairing Bayesian methods with systems theory to quantify changes in operational conditions, changes in adversarial actions, resultant changes in the learning-based system structure, and resultant confidence measures in mission success. We provide insights, in this article, into our overall goal and progress toward developing a framework for evaluation through an understanding of equivalence of testing.

KEYWORDS: Test and Evaluation; systems theory; Bayesian; Learning; artificial intelligence

INTRODUCTION

Test and evaluation (T&E) frameworks for learning-based systems (LBS) are currently in their nascent stage, with existing frameworks lacking specificity and needing to be piloted against actual LBS. By the term LBS, we refer to an array of systems, based on artificial intelligence (AI), with adaptive learning behavior stemming from training data, such as machine learning (ML) computer vision algorithms. A particular challenge arises when considering the impacts of changes in operational conditions and adversarial actions, which may notably vary over the life-cycle of an LBS and cause deviation of the LBS from design limits (Lanus 2021). Traditional systems employ a black-box T&E method of providing sampled inputs, from which outputs are measured against expectations. LBS's complexity and dynamics suggest challenges in applying traditional methods (Freeman 2020).

This paper reports on the status of a Systems Engineering Research Center (SERC) project that aims to establish theory and methods for how T&E requirements can and should change as a function of the test team's knowledge of LBS technical specifications. An overarching objective of this research is to characterize the balance between the design of T&E activities and the cost of data/model rights acquisition for LBS. This informs government decision-makers on the emerging necessity for a new policy. We focus this research article on building from past research on a notional networked munition system of systems for ground denial, referred to as the Silverfish Testbed (Carter 2019), which we leverage to provide insights to our initial T&E framework for LBS.

We develop a framework consisting of Bayesian methods and a system theoretic basis for the mathematical characterization of equivalence between pairs, referred to as a morphism. The project experimented with two pilot scenarios to demonstrate how multiple testing phases contribute to evaluating an LBS, using morphisms as guiding principles. The pilot scenarios center on an unmanned aerial vehicle (UAV), providing vehicle and human detection functions in the Silverfish notional weapons system. These detection functions use the You Only Look Once (YOLO) image recognition agent (Redmon 2016) trained on the Common Object in Context (COCO) data set of images (Lin 2014) and paired with simulations and real drones. From knowledge of morphic equivalence, we frame the correlation between scenarios and resulting confidence in mission success through Bayesian methods.

We share insights from our initial framework, practical development, and expected future activities in the following sections.

GOALS AND OBJECTIVES

The complexity of T&E for LBS is unparalleled when compared to traditional systems. LBS have a rate of evolution based on behavior changes due to the data ingestion rate, which generally has a high frequency, such as in the measure of fractions of a second. Traditional systems, alternatively, are expected to have a low frequency of behavior change, even with changes in input. Furthermore, traditional systems may typically be viewed as deterministic, whereas LBS are viewed from a probabilistic context. Such distinctions between traditional systems and LBS suggest that new T&E methods are necessary to cope with the magnitude of complexity.

Further complexity arises from the necessity to rely on surrogate analogies to achieve confidence in mission success during developmental testing (DT) of LBS. First, the environments and operational conditions of the mission are often analogies to the full scope of the mission set. For example, a system developed for a mission to Mars would leverage a surrogate analogy to the Mars environments on Earth (such as desert climate) to gain confidence in mission success before deployment to the actual Mars environment. Second, the real system may not be available during DT; surrogate analogous systems are used instead. For example, in our case, we use simulation and a low-cost drone as surrogate analogies for the UAV "real" (fielded) system.

This research is driven toward developing a T&E framework for LBS through the necessity to understand the equivalence between and confidence from using the surrogate analogies versus the fielded system and actual mission. An overarching goal of this research is to reach the characterization of the tradespace between the design of T&E activities and the cost of changes in policy to acquire increased access to data/model rights for LBS. To understand this tradespace, subsequent objectives are defined as follows:

- Characterize the change in operational conditions and adversarial actions;
- Characterize the impact of change in operational conditions and adversarial actions on changes to the system implementation and behavior; and
- Create a T&E framework for LBS that characterizes the balance between T&E activities and data/model rights acquisition costs.

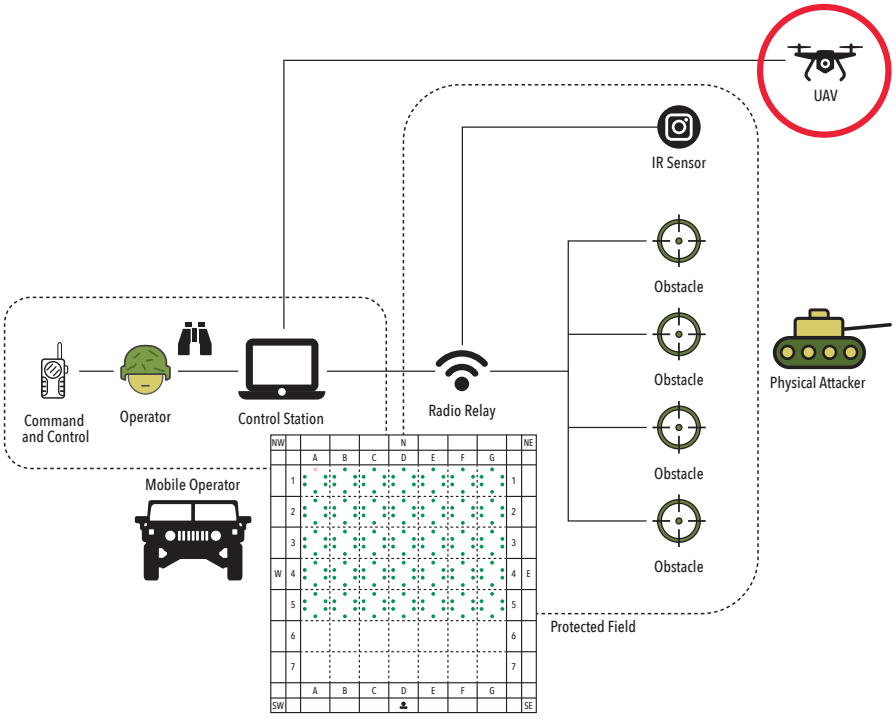


Figure 1. The UAV within Silverfish's notional system of systems context is considered to be the system of interest for this research article

This article provides insights into the creation of the T&E framework. We discuss the framework (1) in terms of notional use for the characterization of changes in operational conditions and adversarial actions, which we refer to as a systems theoretic morphism between the mission and mission surrogates used for T&E; (2) in terms of notional use for the characterization of changes in system implementation and behavior, which we refer to as a systems theoretic morphism between the fielded system and surrogate systems used for T&E; and (3) in terms of notional decision context. The characterization of the balance between T&E activities and data/model rights acquisition cost is left for future research. However, we provide insights into the Bayesian methods that are in development and, when paired with systems theory, will be used to reach the overarching goal.

TESTBED ENVIRONMENTS

The primary testbed for this research is a notional weapons system of systems named Silverfish. Silverfish is used to deny ground to adversaries through a networked munition system with integrated surveillance and situational awareness technology. The system of systems includes the protected area, a UAV that performs surveillance functions, tripwire and infrared ground sensors, and a human operator in charge of command and control. Data from the UAV cameras and the ground sensors are fused to provide situational awareness of the protected area, emphasizing the detection of humans or vehicles. In the event of a detection, the operator is provided with a likelihood that the entity traversing the protected area is a combatant versus a non-combatant. The human is responsible for final decisions, including engaging a target with the networked munitions. We provide the Silverfish notional system in Figure 1 to illustrate the system of systems.

The Silverfish testbed continues to expand from its conception. In the original implementation, Silverfish included a network of connected Raspberry Pi ® to emulate the protected area and ordinance. In line with digital engineering (DE), a model-based systems engineering (MBSE) implementation of Silverfish was defined in the GENESYS tool (Long 2019). More recent progress by our research group has included some initial transition of the MBSE implementation to the Cameo MBSE tool (NoMagic), simulation, and physical testing through the pairing of the YOLO algorithm with UAV/drone hardware.

In this article, our current focus is on the UAV element of Silverfish and T&E for its LBS nature. We refer to the LBS element of the UAV as Agent YOLO, for the name of the computer vision algorithm leveraged therein. The YOLO algorithm provides an open-sourced algorithm to fulfill the intent of a cascade of analogies with respect to the development sequence. The cascade includes T&E surrogates of the Silverfish

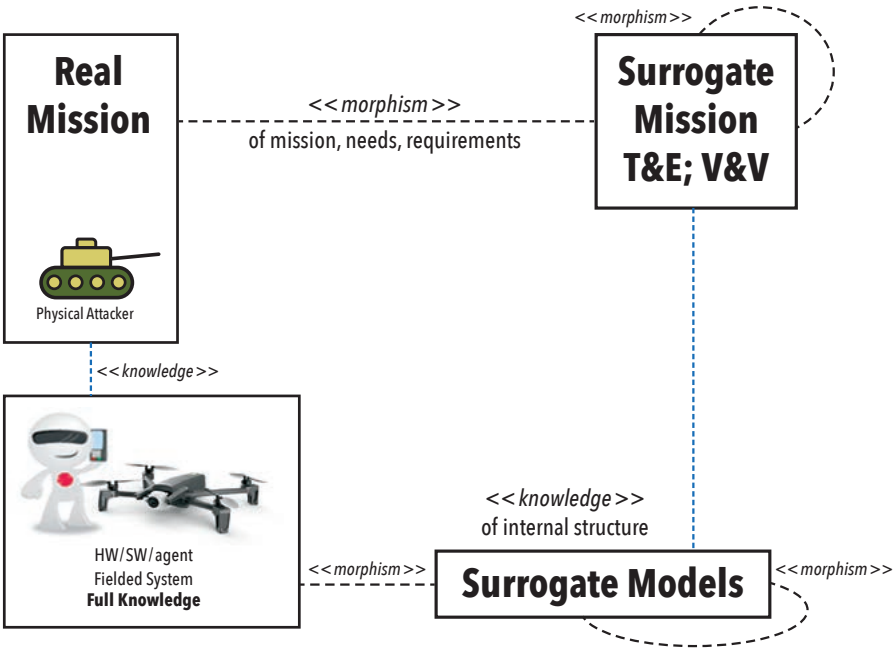


Figure 2. Proposed systems theoretic test and evaluation framework

UAV and surrogates of its mission context within the Silverfish system of systems, which is to surveil a protected area, identify potential attackers, and report the surveillance activities to the human command and control element.

OVERVIEW OF FRAMEWORK

Our framework consists of two parts: (1) systems theoretic characterization of stratification as well as characterization of equivalence referred to as system morphisms and (2) Bayesian method characterization of correlation in confidence in mission success.

We provide a visualization of the systems theoretic aspects of the framework in Figure 2, which builds on the research found in Wach 2021; Wach 2022a; Wach 2022b). The horizontal lines reflect morphic equivalence between surrogate analogies with the real mission and the fielded system; the vertical lines reflect knowledge of the interior structure of the LBS system implementation. Each surrogate may have morphisms relative to other surrogates (mission-mission and model-model). There is a corresponding cost associated with acquiring the data for systems. To account for the many levels of data-driven knowledge, we use systems theory to mathematically characterize the iterative and recursive stratification.

We provide a visualization of the Bayesian aspect of the framework in Figure 3. We use a Bayesian network to characterize the probability of outcomes across the testing phases; the network's edges represent conditional probabilities that can be used to compute the probability of—or the operational cost associated with—outcomes at each layer. In this simple example, we use three layers to represent three different system types that might be evaluated, including in the Silverfish context, System 1 might be a pairing of Agent Yolo with prototype hardware for a developmental test activity, System 2 might be a pairing of Agent Yolo with low-rate initial production hardware (LRIP) in an initial operational T&E (IOT&E) activity, and System 3 might be the real mission and fielded system. We then categorize the outcomes from those systems into two cases, Case A and Case ¬A ("not A"), which might, for example, correspond to "detect" and "no detect" in the context of Silverfish. We elaborate further in the next section; see Figure 5 in particular.

The Bayesian network is paired with the cascade of knowledge of the results of T&E activities, which builds on the research found in Salado (2018). This knowledge includes the systems theoretic characterization of morphic equivalence and internal structure. The combined and framed knowledge impacts overall confidence in mission success from the deployment of the LBS, which can be paired with utility metrics such as cost/schedule for predictive capabilities. In doing so, the framework enables the characterization of the relationship between the design of the evaluation activities and the characterization of equivalence. When we pair the systems theoretic morphisms with Bayesian methods, we have a fabric for connecting information and determining T&E priority. For example, we may select a cheap drone for a T&E activity as a surrogate or a more expensive drone because we believe the drone to have a low probability of mission degradation when considering the overall LBS. Thus, an impact of the framework is the ability to narrow down cases that are most likely to fail or cause problems. By connecting levels of knowledge of the surrogate analogies to confidence, we can weigh the cost of a T&E activities in light of their importance to mission success.

INITIAL RESULTS

In this section, we provide insights into the results. We focus here on a T&E activity consisting of detecting automobiles and using physical drones paired with Agent YOLO, which have various morphic equivalence to the real mission and fielded system. We have a cheap prototype drone paired with Agent YOLO in the first case. In the second case, we have the higher-cost LRIP drone paired with Agent YOLO. Both drone/agent pairs were simultaneously tested and evaluated for detecting automobiles, which is a surrogate mission scenario for detecting a potential attacker. A visualization is shown in Figure 4.

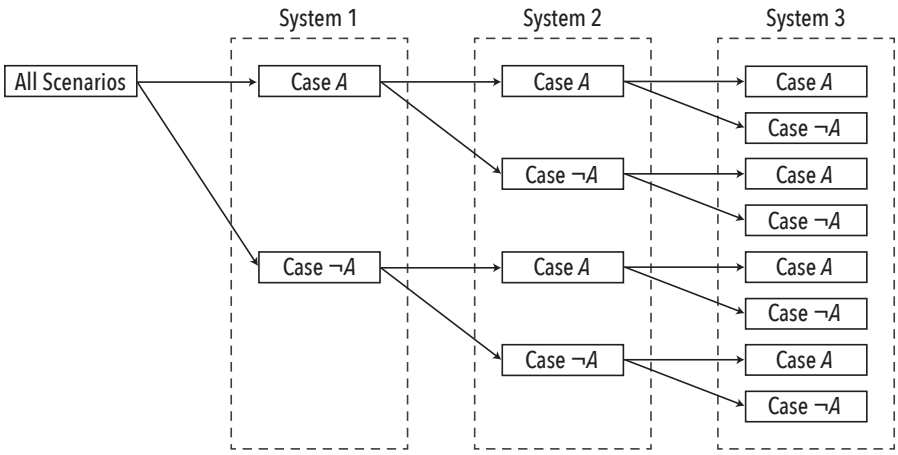


Figure 3. A visualization of the Bayesian aspect of the framework

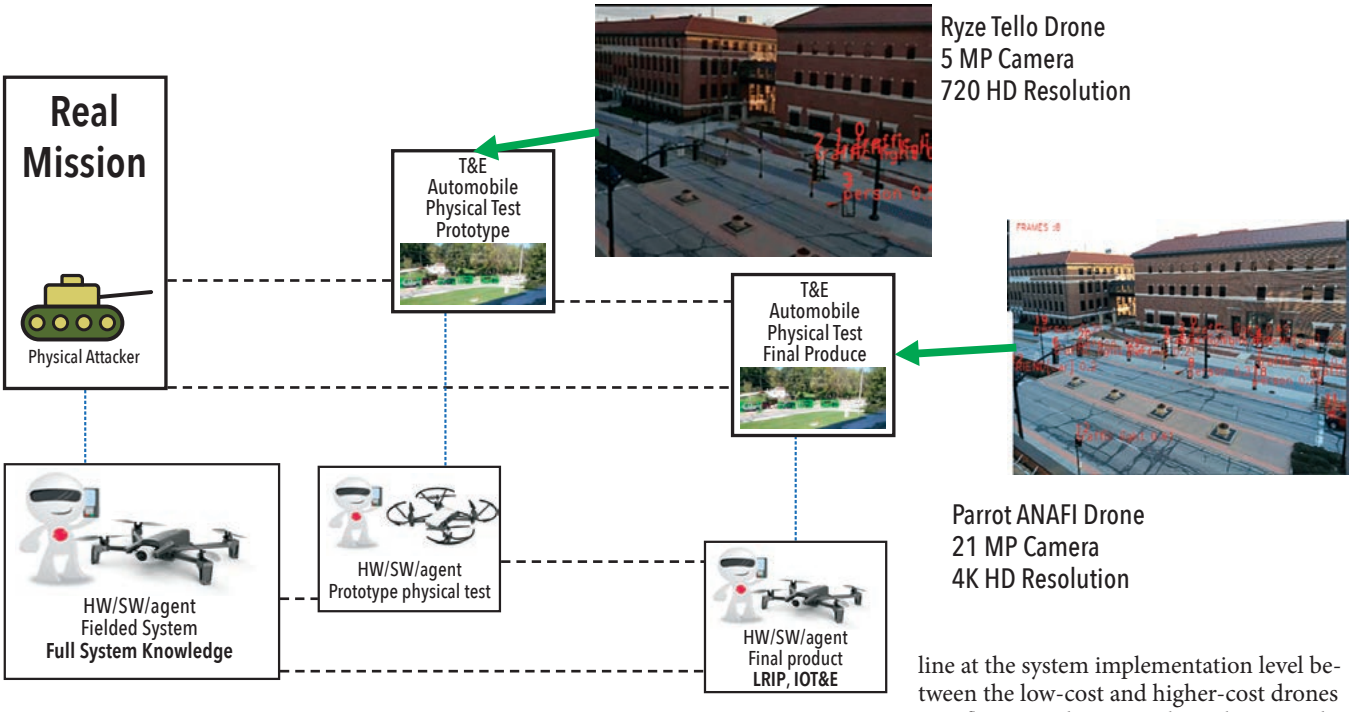


Figure 4. A visualization of the systems theoretic framing of the test context

To further elaborate on the two drones used for this study: The first drone used is a lower-cost drone (Ryze Tello), which has cost-corresponding attributes such as camera megapixels (5 MP) and resolution (720 HD). The second drone used is a higher-cost drone (Parrot ANAFI), which has cost-corresponding attributes such as camera megapixels (21 MP) and resolution (4k HD).

Each drone served as a representation of a phase of system development with the corresponding testing. We treat the low-cost drone as a prototype that may be used in the early development of a system for a developmental test. We treat the higher-cost drone as resembling what may be produced during LRIP for IOT&E.

We used simultaneous testing of the drones, although one would typically expect time to elapse between tests following phased system development. Each drone was positioned side-by-side at the same time of day and in view of the same street. During the test activity, Agent YOLO, paired with each drone, characterized the vehicles as they passed on the street.

The vertical lines in Figure 4 reflect morphic equivalence at each system specification level, similar to Figure 2. In this case, we add a vertical line at the mission level of system specification between the test conducted on the low-cost drone and the test conducted on the higher-cost drone to reflect morphic equivalence between the tests. Also, in this case, we add a vertical line at the system implementation level between the low-cost and higher-cost drones to reflect morphic equivalence between the drones.

The knowledge of morphic equivalence may be complemented by a confidence factor defined by Bayesian methods, as shown in Figure 5.

The images of the street and vehicles passing by are shown in Figure 4 at the top middle for the lower-cost drone and to the right side for the higher-cost drone, which is unaltered and can be observed to have visual differences. Although there is nearly an exact morphic equivalence at the mission level, there is a lower degree of morphic equivalence at the drone system implementation level. The morphisms provide knowledge to frame the overall equivalence, which feeds into confidence in mission success. Using Bayesian methods, the success (or lack thereof) detection and

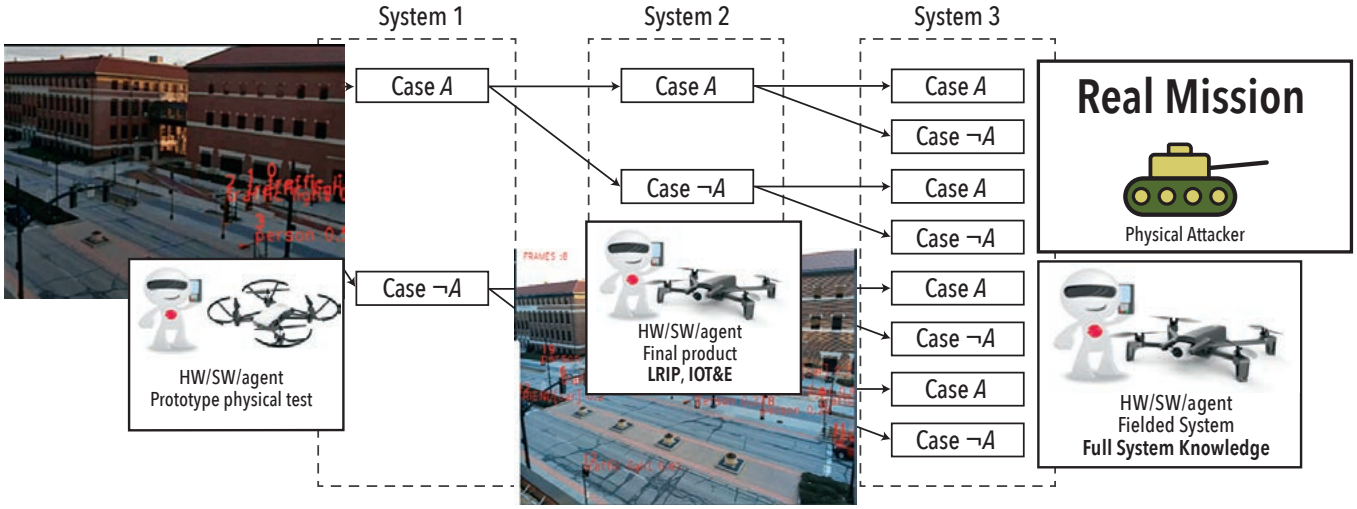


Figure 5. A visualization of the Bayesian propagation of confidence

categorization of the automobiles with the lower-cost drone indicate success for the higher-cost drone. The complementary pairing of system morphisms with Bayesian methods provides the basis for our framework for the T&E of LBS.

FUTURE WORK

Our future efforts are threefold: (1) link the LBS lifecycle, (2) advance the digital engineering aspects, and (3) prove the value to the government.

As discussed in this article, we have focused our initial efforts on the DT aspects of T&E. Our future efforts will continue from DT to later aspects of the LBS lifecycle. We plan to show the propagation of knowledge and confidence in mission success from the DT to the operational, surveillance, and maintenance phases of the LBS lifecycle. Furthermore, knowledge of retirement and legacy systems propagates perceived confidence in new systems, which we will explore in future work.

We are exploring several paths to advance the digital engineering aspects of the framework. One, we are exploring creating plugins for the Cameo MBSE tool and constructs based on the Systems Modeling Language (SysML). We plan to enhance the framework through digital twin and physical twin pairing. We are also exploring creating an expert system to advise the human decision-maker (s) during acquisition and deployment based on the T&E framework. Lastly, we anticipate linking the framework to a "born-digital" Test and Evaluation Master Plan (d-TEMP). These are some of the digitally enhanced efforts either in planning or in progress.

To reach the main goal of this effort, we desire to prove the value to the government and use the framework to assess the tradespace between confidence in mission success and resources necessary for acquiring increased data/model rights to LBS. First, we plan to add utility metrics to the Bayesian methods and simulate policy changes to accomplish this. Second, our data set is currently small, and we would like to expand it with more control. As an example, we are proposing using a controlled group of students traversing a field to emulate the red/blue scenario. Furthermore, we are leveraging commercial-off-the-shelf drones with limited insights and control over their hardware and software, increasing our urgency to create our controlled hardware/software. Last, we plan to up-scale the framework from the controlled development environments to real LBS acquisition, deployment, and policy decision-making.

CONCLUSION

We present a novel framework for the T&E of LBS. The framework consists of a systems theoretic basis for determining equivalence from surrogate analogies used for T&E relative to the real mission and system implementation. The framework uses Bayesian methods to characterize confidence in mission success. We initially framed LBS through simulation and physical testing, which has shown promise. This article is focused on exposure to the framework rather than the data and specifics of the mathematical basis. Finally, we discuss aspirations for the T&E framework for LBS. ■

DISCLAIMER

The Acquisition Innovation Research Center is a multi-university partnership led and managed by the Stevens Institute of Technology and sponsored by the U.S. Department of Defense (DoD) through the Systems Engineering Research Center (SERC)—a DoD University-Affiliated Research Center (UARC).

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) through SERC and AIRC under Contract HQ0034-19-D-0003, TO#0309.

The views, findings, conclusions, and recommendations expressed in this material are solely those of the authors and do not necessarily reflect the views or positions of the United States Government (including the DoD and any government personnel) the Virginia Tech National Security Institute, the Stevens Institute of Technology, and the Purdue University.

No Warranty.

This Material is furnished on an "as-is" basis. The Virginia Tech National Security Institute, the Stevens Institute of Technology, and the Purdue University make no warranties of any kind—either expressed or implied—as to any matter, including (but not limited to) warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material.

The Virginia Tech National Security Institute, the Stevens Institute of Technology, and the Purdue University do not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

This material has been approved for public release and unlimited distribution.

REFERENCES

- Beling, P., B. Horowitz, C. Fleming, S. Adams, G. Bakirtzis, B. Carter, T. Sherburne, C. Elks, A. Collins, and B. Simon. 2019. "Model-based engineering for functional risk assessment and design of cyber resilient systems." University of Virginia Charlottesville United States.
- ———. 2018. *Model-based systems engineering*. CRC press.
- 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.
- Cody, T., S. Adams, and P.A. Beling. 2019. "A systems theoretic perspective on transfer learning." *2019 IEEE International Systems Conference (SysCon)*, 1–7. IEEE.
- Consortium, Coco. 2022. 'COCO Common Objects in Context.'
- ———. 2020. "Test and Evaluation for Artificial Intelligence." *INSIGHT* 23 (1): 27–30.
- Fleming, C.H., C. Elks, G. Bakirtzis, S. Adams, B. Carter, P. Beling, and B. Horowitz. 2021. "Cyberphysical Security Through Resiliency: A Systems-Centric Approach." *Computer* 54, (6): 36–45.
- Freeman, L.. 2020. 'Test and Evaluation for Artificial Intelligence.' *INSIGHT* 23 (1): 27–30. https://doi.org/10.1002/inst.12281.
- Lanus, E., I. Hernandez, A. Dachowicz, L. Freeman, M. Grande, A. Lang, J.H. Panchal, A. Patrick, and S. Welch. 2021. "Test and Evaluation Framework for Multi-Agent Systems of Autonomous Intelligent Agents." *arXiv preprint arXiv:2101.* 10430. http://arxiv.org/abs/2101.10430.
- Lin, T.Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. 2014. "Microsoft coco: Common objects in context." *European conference on computer vision*, 740–55. Springer.
- ———. 2021. "Test and Evaluation Framework for Multi-Agent Systems of Autonomous Intelligent Agents." *arXiv [cs, eess]*, Ιανουάριος. arXiv. https://doi.org/10.1109/SOSE52739.2021.9497472.
- Long, D. 2019. "MBSE 2.0: The Future of MBSE."
- McDermott, T.A., M.R. Blackburn, and P.A. Beling. 2021. "Artificial Intelligence and Future of Systems Engineering." Systems Engineering and Artificial Intelligence, 47–59. Springer.
- NoMagic. Accessed 2022 Oct 10 "Cameo Systems Modeler." https://www.nomagic.com/products/cameo-systems-modeler
- ———. 2016. "You Only Look Once: Unified, Real-Time Object Detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779–88.

- Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–88.
- Salado, A., and H. Kannan. 2018. "A mathematical model of verification strategies." *Systems Engineering* 21 (6): 593–608.
- Wach, P., B.P. Zeigler, and A. Salado. 2021. "Conjoining Wymore's Systems Theoretic Framework and the DEVS Modeling Formalism: Toward Scientific Foundations for MBSE." *Applied Sciences* 11 (11): 4936.
- Wach, P., P. Beling, and A. Salado. 2022a. "Formalizing the Representativeness of Verification Models Using Morphisms." INCOSE-IS, Detroit, MI, USA.
- ———. 2022b. "Initial Systems Theoretic Metamodel of Verification Artifacts." CSER, Norwegian University of Science and Technology, Norway.

ABOUT THE AUTHORS

Paul Wach has research interests include the intersection of theoretical foundations of systems engineering, digital transformation, and artificial intelligence. Dr. Wach is a member of the Intelligent Systems Division at the Virginia Tech National Security Institute. He was the President and Founder of the Virginia Tech student division of INCOSE. Dr. Wach also works for The Aerospace Corporation, leading enterprise digital engineering transformation. His prior work experience is with the Department of Energy, two National Laboratories, and the medical industry. Dr. Wach received a B.S. in Biomedical Engineering from Georgia Tech, an M.S. in Mechanical Engineering from the University of South Carolina, and a Ph.D. in Industrial and Systems Engineering from Virginia Tech.

Justin Krometis is a research assistant professor in the Intelligent Systems Division of the Virginia Tech National Security Institute. Before joining NSI, Dr. Krometis worked in Virginia Tech's Advanced Research Computing department for ten years as a Computational Scientist supporting research computing. Before that, he worked in the public and private sectors doing transportation modeling for planning and evacuation applications; hurricane, pandemic, and other emergency preparedness; and project management. His research is in the development of theoretical and computational frameworks to address analytics problems, such as how to incorporate and balance data and expert opinion into decision-making, how to fuse data from multiple sources, and how to estimate model parameters, including high- or infinite-dimensional quantities, from noisy data. Areas of interest include Bayesian inference, parameter estimation, machine learning, data science, and experimental design. Dr. Krometis holds a Ph.D. in mathematics, an M.S. in mathematics, a B.S. in mathematics, and a BS in physics, all from Virginia Tech.

Atharva Sonanis is an M.S. in Mechanical Engineering student working with Professor Jitesh Panchal at the Design Engineering Laboratory at Purdue University (DELP). Atharva's research interests include robotics, controls, computer vision, machine learning, and systems. He received his B.E. in Mechanical Engineering from M.I.T. College of Engineering, Pune. While pursuing his bachelor's degree, he was selected as a Cummins Scholar and received the opportunity to work at Cummins Technical Centre India in the R&D department. He also holds a Mechanical Engineering diploma from Government Polytechnic, Miraj.

Dinesh Verma is a 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 the 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 a Ph.D. in industrial and systems engineering.

Jitesh Panchal is a Professor of Mechanical Engineering at Purdue University. He received his BTech (2000) from the Indian Institute of Technology (IIT) Guwahati, and MS (2003) and Ph.D. (2005) in Mechanical Engineering from Georgia Institute of Technology. Dr. Panchal's research interests are in (1) design at the interface of social and physical phenomena, (2) computational methods and tools for digital engineering, and (3) secure design and manufacturing. He is a recipient of the CAREER award from the National Science Foundation (NSF); Young Engineer Award, Guest Associate Editor Award, and three best paper awards from ASME; and was recognized by the B.F.S. Schaefer Outstanding Young Faculty Scholar Award, the Ruth and Joel Spira Award, and is one of the Most Impactful Faculty Inventors at Purdue University. He is a co-author of two books and has co-edited one book on engineering systems design. He has served on the editorial board of international journals, including the ASME Journal of Mechanical Design and the ASME Journal of Computing and Information Science in Engineering. He is a program chair of the ASME IDETC/CIE conference and the past chair of the ASME Computers and Information in Engineering (CIE) division.

Laura Freeman is a Research Associate Professor of Statistics and dual-hatted as the Deputy Director of the Virginia Tech National Security Institute and Assistant Dean for Research for the College of Science. Her research leverages experimental methods for conducting research that combines cyber-physical systems, data science, artificial intelligence (AI), and machine learning to address critical challenges in national security. She develops new methods for test and evaluation focusing on emerging system technology. Previously, Dr. Freeman was the assistant director of the Operational Evaluation Division at the Institute for Defense Analyses (IDA). Dr. Freeman also served as the acting senior technical advisor for the Director of Operational Test and Evaluation (DOT&E). Dr. Freeman has a BS in aerospace engineering, an M.S. in statistics, and a Ph.D. in statistics, all from Virginia Tech.

Peter Beling is a professor in the Grado Department of Industrial and Systems Engineering and Director of the Intelligent Systems Division at the Virginia Tech National Security Institute. Dr. Beling's research interests lie at the intersections of systems engineering and artificial intelligence (AI), including AI adoption, reinforcement learning, transfer learning, and digital engineering. His research has found applications in various domains, including mission engineering, cyber resilience of cyber-physical systems, prognostics and health management, and smart manufacturing. He received his Ph.D. in operations research from the University of California at Berkeley.

DECEMBER 2022
VOLUME 25/ ISSUE 4

Human Models for Future Mobility

Andreas Lüdtke, andreas.luedtke@dlr.de; **Jan-Patrick Osterloh,** jan-patrick.osterloh@dlr.de; **Jakob Suchan,** jakob.suchan@dlr.de; and **Alexander Trende,** alexander.trende@dlr.de

■ **ABSTRACT**
The new DLR Institute of Systems Engineering for Future Mobility (DLR SE) opened its doors at the beginning of 2022. As the new DLR institute emerged from the former OFFIS Division Transportation, it can draw on more than 30 years of experience in the research field on safety critical systems. With the transition to the German Aerospace Center (DLR), the institute has developed a new research roadmap focusing on technical trustworthiness for highly automated and autonomous systems, as described in the article "DLR Institute of Systems Engineering for Future Mobility – Technical Trustworthiness as a Basis for Highly Automated and Autonomous Systems" in this journal. In this paper, we describe how the Group *Human Centered Engineering* (HCE) contributes to this roadmap with our methods of "virtual test drivers" and "virtual co-drivers."
■ **KEYWORDS:** highly automated systems; autonomous systems; human factors; test driver; artificial intelligence; cognitive architectures; co-driver; trustworthiness

INTRODUCTION

The Human Centered Engineering Group of the new DLR Institute of Systems Engineering for Future Mobility (DLR SE) researches human models that can be used as *virtual test drivers* or as *virtual co-drivers*. These models can recognize and predict human behavior. The main objective of both models is to improve the safety and trustworthiness of human-machine interaction. Each model focuses on different aspects of the human-machine interaction and can be used independently. Our *virtual test drivers* are used to analyze and test the design of a system, for instance, by analyzing design variants of human-machine interaction for safety critical systems or variants of assistant systems. Before testing with real humans, such virtual tests can be done very early in the system development process.

On the other hand, our virtual co-drivers are used to recognize the driver's state and to predict their actions to initiate interventions in hazardous situations. Both models use different techniques and methods. Thus, we investigate these techniques and formalisms to model how humans interact with machines in complex traffic situations. We research not only driver models but also models of seafarers and aircraft pilots. In the following, we describe both use cases and our modeling techniques.

HUMAN MODELS AS VIRTUAL TEST DRIVERS

The long history of automation research in the aircraft industry has shown that human factors play a crucial role whenever automation is increased in a transportation system because automation often leads to characteristic human errors (Parasuraman 1997; Sarter et al. 1997), like mode confusion, situation awareness problems, unexpected mode reversions, and inappropriate use of automation. Human factors expertise is needed to mitigate those human errors, especially during the transition phase between manual driving and fully autonomous driving, when the tasks of the human driver shift from driving to monitoring and control. Unfortunately, the needed human factors expertise is often absent or, if present, is applied in very late design phases when fixes are expensive to implement. Furthermore, the human factors tooling is often not, or only loosely connected (via requirements management) to the tooling of the engineers developing the system. The idea behind the development of the "virtual test driver" is to bridge this gap by providing systems engineers a tool suite that a) can be integrated into their tooling, allowing b) simulation of human behavior in early design phases and c) allows formal modeling of the user behavior. Especially the last point provides several benefits since the formal model of the user provides a semantically defined language for discussions between the different stakeholders, like system engineers, human factor experts, and management. We have developed such a virtual test driver in form of a cognitive architecture, which is introduced in the following section.

Human Modelling

To implement the virtual test driver, the Human-Centered Engineering Group of DLR SE implemented the cognitive architecture CASCaS (*c*ognitive *a*rchitecture for safety *c*ritical *ta*sk simulation). Cognitive architectures provide computational models of the theories about the structures of the human mind and how these theories work together to manage intelligent behavior in complex environments. A good overview and comparison of cognitive architectures can be found in Wickens et al., 2013. By combining psychological and physiological models of human behavior, the architecture can be used to predict human behavior via simulation.

The virtual test driver consists of two parts. First, the cognitive architecture CAS-

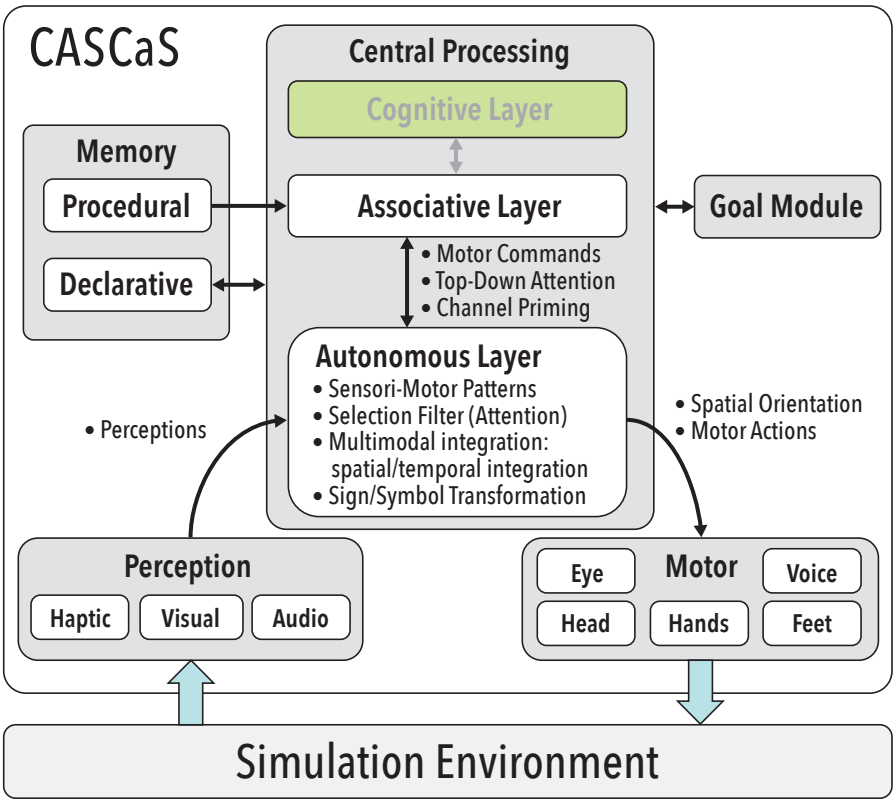


Figure 1. The architecture of the virtual test driver CASCaS (cognitive architecture for safety critical task simulation); cognitive layer not implemented in CASCaS

CaS which is used as a simulation tool and secondly, the formalized knowledge of the user. Figure 1 shows the components that are implemented in CASCaS.

The perception module is connected via a dedicated interface in the simulation environment to the simulator and is responsible for the symbolic perception of the environment. The visual component, for example, implements aspects of human vision like the foveal and peripheral view to determine which information can be perceived in the current situation. As a counterpart, the eye and head components in the motor module are responsible for calculating and performing the shifts of the eye and head toward a new position. The other components in the Motor module are for interacting with the system, for instance pressing buttons and speaking. The perceived information is written into the Memory module, which not only stores the declarative (for example, the current lane, the current speed limit, or the next navigation target) and procedural knowledge (including the information on how to drive; see below on formalization of knowledge) but has also implemented processes for retrieval and forgetting. The main component that controls everything is the central processing module, which implements three different layers of cognition (Anderson 2000):

- cognitive layer: decision-making in unfamiliar situations (not implemented in CASCaS)
- associative layer: rule-based behavior and decision making
- autonomous layer: processing without thinking in daily operations, such as sensory-motor programs like steering, braking

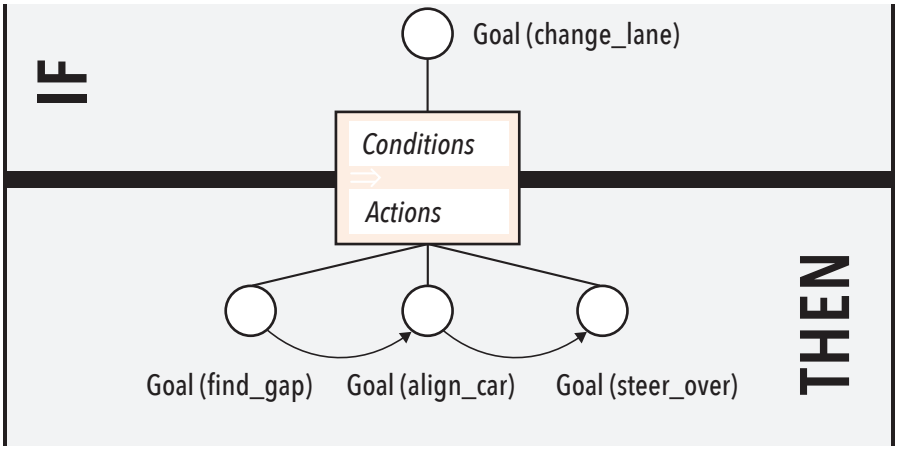


Figure 2. Example GSM rule for changing the lane: IF your goal is to change the lane, THEN you must first find a gap, afterward you must align the car with the gap, and finally, you must steer over into the gap

The goal module is a standard module for the central processing layers and manages the goals of the virtual driver (follow car, lane change left/right) concerning multitasking and the order of goals to be executed. The goals themselves are part of the formalized user knowledge. A major part of preparing the simulation is the definition and formalization of the user's knowledge, in this case, the driver. The knowledge is modeled in the form of GSM rules, which are then loaded into the memory module at runtime into CASCaS. GSM means "goal-state-mean" and are if-then rules that belong to a goal, as depicted in Figure 2. The "then" part is executed when the goal is active and the "if" conditions evaluate to true. The "then" consists of motor actions, memory operations, or sub-goal management.

CASCaS has been extensively validated by comparing the output of the architecture and a domain-specific set of rules with experimental human data in the aviation domain (Frische, Osterloh and Luedtke 2010a, Frische, Osterloh, and Luedtke 2010b), as well as in the automotive domain (Wortelen, Baumann and Luedtke 2013).

HUMAN MODELS AS VIRTUAL CO-DRIVERS

The virtual co-driver is designed to monitor the human driver and help carry out specific tasks, including driving a car, navigating a ship, or flying an aircraft. This results in a shift of control from the driver to the automation and poses the need for close cooperation between the driver and the co-driver to establish safe and efficient human-machine cooperation. Therefore, it is essential that both interaction partners can understand each other's intentions and capabilities.

The work of the Human Centered Engineering (HCE) group encompasses both directions of understanding involved in this process. On the one hand, this research

is concerned with designing machines that act and communicate in a way such that humans can understand the rationale behind their behavior, which includes the need for the systems to adapt their performed actions to the expectations of the human driver and communicate their decisions and relevant information coherently, such as using HMIs or other visual-auditory communication devices (Harre and Feuerstack 2018). On the other hand, and more directly related to the development of a virtual co-driver, as discussed in this section, is the ability of autonomous systems to understand and support the human driver. The system needs a human model that can recognize or even predict human behavior or intention and anticipate possibly safety-critical situations to provide the right type and amount of assistance in a particular situation. Such human models need to maintain a representation of the driver's mental state to monitor and interpret the driver's body functions and relate this to sensed interactions and dynamics within the environment. In this sense, key abilities of the virtual co-driver are to a) capture the state of the driver, including using technologies such as eye-tracking, physiological and neural measurements b) model and interpret the mental state, intentions, needs, and preferences of the driver, to predict the driver's conception of the situation, and d) sense the environment and simulate possible dynamics within it, including the effects of the driver's decisions on the current situation. Research and development conducted in this direction involve the development of general AI-based methods (building on probabilistic models, neural networks, and logic-based reasoning) for modeling and predicting human factors and environmental interactions with empirical human studies to inform and validate these methods.

Modeling with Probabilistic Techniques

Probabilistic models are a suitable framework to account for individual differences between humans and the broadness of human behavior. Probabilistic graphical models like Bayesian networks are suitable model types for the inference of human behavior and intention due to their ability to provide predictions even with uncertainty (Pearl 1998). For instance, a model that recognizes whether the driver wants to turn does not have direct access to the human decision-making process. However, the model can use behavioral driving variables and user-specific information to estimate the driver's intent to turn at an intersection (Trende et al. 2021). Such intention recognition systems can be used as building blocks of advanced driver assistance systems and warn the driver in safety-critical situations (Trende et al. 2022).

CONCLUSION

We described two use cases in the automotive domain for human models being developed at the Human-Centered Engineering Group of the new DLR Institute of Systems Engineering for Future Mobility. First, human models and especially cognitive architectures can be used to serve as virtual test drivers to investigate, for example, human errors during automation in a safe, simulated environment. The simulated behavior of the virtual test driver can be easily analyzed due to the deterministic nature of the cognitive architecture. This helps to derive design guidelines for improving the investigated automation system. Second, we presented probabilistic models as virtual co-drivers. These models can be used to infer and monitor the state or intention of a human driver in real time. These models can activate intervention strategies, like human-machine interactions or adaptive automation, if an unfavorable user state or user intention is detected. Such intervention strategies can potentially reduce the number and severity of safety-critical situations during driving. ■

REFERENCES

- Anderson, J. R. 2000. *Learning and memory: An Integrated Approach*. 2nd ed. New York; Chichester: Wiley.
- Frische, F., J.-P. Osterloh, A. Lüdtke. 2010a. "Simulating Visual Attention Allocation of Pilots in an Advanced Cockpit Environment," in *Proceedings of the MODSIM World Conference and Expo 2010*.
- Frische, F., J.-P. Osterloh, A. Lüdtke. 2010b. "Modelling and Validating Pilots Visual Attention Allocation during the Interaction with an Advanced Flight Management System." In *Proceedings of Human Modelling in Assisted Transportation* (HMAT), Springer.
- Harre, M.C., and S. Feuerstack. 2018. "The Konect value—a quantitative method for estimating perception time and accuracy for HMI designs." *Behaviour & Information Technology* 37(9): 894-903.
- Lüdtke, A., L. Weber, J.-P. Osterloh, and B. Wortelen. 2009. *Modeling Pilot and Driver Behavior for Human Error Simulation*. 403-412. doi: 10.1007/978-3-642-02809-0_43.
- Osterloh, J.-P., J. Rieger, and A. Lüdtke. 2017. *Modelling Workload of a Virtual Driver*. Proceedings of the 15th International Conference on Cognitive Modeling (ICCM).
- Parasuraman, R., and V. Riley. 1997. "Humans and automation: Use, misuse, disuse, abuse." In *Human factors* 39(2): 230-253.
- Pearl, J. 1998. Graphical models for probabilistic and causal reasoning. Quantified representation of uncertainty and imprecision, 367–389.
- Sarter, N. B., D.D. Woods, and C.E. Billings. 1997. "Automation surprises." In *Handbook of human factors and ergonomics 2nd edition*, edited by G. Salvendy, Wiley
- Trende, A., A. Unni, M. Jablonski, B. Biebl, A. Lüdtke, M. Fränzle, and J.W. Rieger. 2022. *Driver's turning intent recognition model based on brain activation and contextual information*. Frontiers in Neuroergonomics, 23.
- Trende, A., A. Unni, J. Rieger, and M. Fraenzle. 2021. "Modelling Turning Intention in Unsignalized Intersections with Bayesian Networks." In *International Conference on Human-Computer Interaction*, 289-296. Springer, Cham.
- Wickens, C., A. Sebok, J. Keller, S. Peters, R. Small, S. Hutchins, L. Algarín, B.F. Gore, B.L. Hooey, and D.C. Foyle. 2013. Modeling and Evaluating Pilot Performance in NextGen: Review of and Recommendations Regarding Pilot Modelling Efforts, Architectures and Validation Studies. NASA/TM-2013-216504, Ames Research Center, Moffett Field, California. https://hdl.handle.net/2060/20140002223.
- Wortelen, B., M. Baumann, A. Lüdtke. 2013. "Dynamic simulation and prediction of drivers' attention distribution." In *Transportation Research Part F: Traffic Psychology and Behaviour*. Volume 21, 278-294, ISSN 1369-8478, https://doi.org/10.1016/ j.trf.2013.09.019.

ABOUT THE AUTHORS

Andreas Lüdtke received his diploma degree in computer science in 1997 and achieved a PhD summa cum laude degree in 2004. From 2004 until 2021, he worked at OFFIS as senior principal scientist, where he established the group Human-Centered Design. In 2022 he transitioned with his group to the newly founded DLR SE. He has published more than 150 publications on human modeling

> continued on page 79

NeuroRAN Rethinking Virtualization for AI-native Radio Access Networks in 6G

Paris Carbone; **Gyorgy Dán**; **James Gross,** jamesgr@kth.se; **Bo Göransson,** bo.goransson@ericsson.com; and **Marina Petrova,** petrovam@kth.se

■ **ABSTRACT**
Network softwarization has revolutionized the architecture of cellular wireless networks. State-of-the-art container based virtual radio access networks (vRAN) provide enormous flexibility and reduced life-cycle management costs, but they also come with prohibitive energy consumption. We argue that for future AI-native wireless networks to be flexible and energy efficient, there is a need for a new abstraction in network softwarization that caters for neural network type of workloads and allows a large degree of service composability. In this paper we present the NeuroRAN architecture, which leverages stateful function as a user facing execution model, and is complemented with virtualized resources and decentralized resource management. We show that neural network based implementations of common transceiver functional blocks fit the proposed architecture, and we discuss key research challenges related to compilation and code generation, resource management, reliability and security.
■ **KEYWORDS:** function as a service, serverless computing, network softwarization, neural networks, energy efficiency, radio access network, AI-native wireless

INTRODUCTION

Over the last years, we are witnessing significant efforts in designing a software-based virtualized radio access network (vRAN) architecture running on commercial off the shelf (COTS) hardware, in an attempt to reduce development and maintenance costs, and to replace static and monolithic architectures with programmable and flexible ones. Software-based vRAN on COTS hardware indeed offers enormous flexibility. Flexibility is essential to support emerging applications that will tightly integrate with physical processes, for example, augmented reality and cognitive assistants, real-time cyber-physical control systems, as well as situational awareness systems. At the same time it is a precondition for further antenna densification, which is the primary approach to providing increased capacity, lowering latency and increasing reliability. Yet, vRAN also comes with prohibitive energy consumption and would require a tenfold increase in energy efficiency to allow wide scale adoption. Moreover, as of today, a fast and flexible programmable control framework, which can jointly meet real-time requirements of lower layers of the protocol stack, and can autonomously adapt to the time-varying network dynamics does not exist.

As 5G networks are rolled out all over the world, the requirements for next generation (6G) systems are starting to be discussed in academia and in industry. The main application-level drivers for a future radio access network include increased trustworthiness and the ability to cope with emerging applications, such as XR/VR, gaming, smart sensors, internet-of-sense applications, and digital twins. Many of these applications require or generate massive amounts of data and have tight delay requirements. At the same time, over the last few years we have been witnessing an unprecedented push in communications research towards data-driven approaches.

While the degree of functional involvement of machine learning varies so far with respect to the discussed approaches, many presented studies demonstrate on-par performance of data-driven approaches in comparison to legacy model-based ones. Thus, it appears quite likely that next generation RANs will include widespread use of AI within transceivers. At the same time, it goes without saying that they should be an enabler for a sustainable society, and that all this has to come at an affordable cost.

In this paper, we argue that the promises of machine learning (ML) in next-generation systems require suitable software architectures to actually deliver. These go well beyond standard container-based approaches leveraged today with respect to flexible RAN solutions such as O-RAN (O-RAN 2022). The main components of such future architecture relate to efficient virtualization with respect to more adapt hardware, for instance, for neural network types of workloads, the provisioning of virtualized resources towards functional software blocks allowing fine-grained composition and function splits, as well as on-demand deployment of corresponding

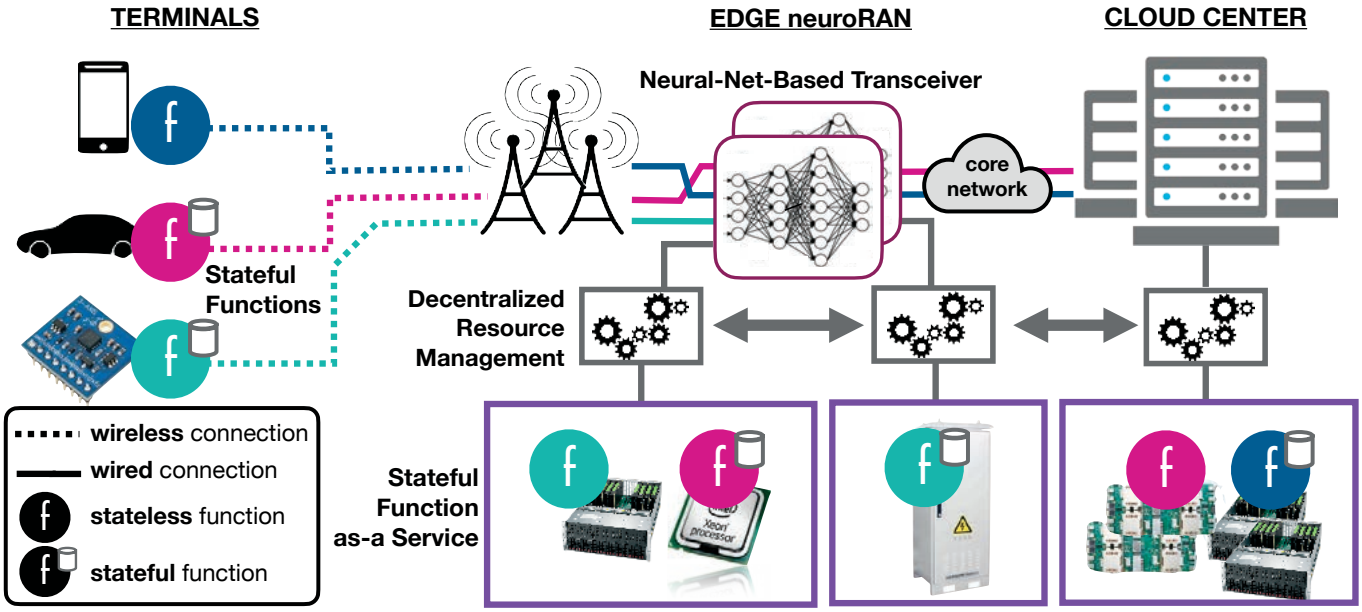


Figure 1. AI-native RAN architecture overview

RAN transceiver code. To this end, we present NeuroRAN, illustrated in Figure 1, a matching software architecture for next-generation mobile networks, while also discussing newly arising challenges in the intersection between software architectures and ML-based transceiver software for future RANs.

The rest of the paper is organized as follows. In Section II we briefly discuss the evolution towards softwarized and virtualized mobile networks architectures and present current trends in deploying neural networks for design and adaptation of wireless communication in Section III. In Section IV we propose NeuroRAN, our flexible AI-native RAN architecture and discuss the main research challenges to be addressed in Section V. Finally in Section VI we conclude the paper.

SECTION II – SOFTWARIZATION OF MOBILE NETWORK ARCHITECTURES

Early product-grade LTE implementations were leveraging custom hardware and software in the RAN as well as in the EPC (evolved packet core). They were optimized for minimal energy consumption, but left little room for functional enhancements and required high coordination bandwidth among eNBs (base stations) with the advent of sophisticated interference coordination schemes. RAN softwarization was arguably triggered by the emergence of software-defined networking (SDN) and network function virtualization (NFV) as new implementation paradigms for wide-area networks, allowing enhanced flexibility and separation of control and data plane functionality for enhanced control. Vendors of LTE networks introduced the virtualized EPC, which too allowed a higher degree of flexibility and efficiency. Moreover, softwarization triggered the move away from the traditional non-split RAN architecture in LTE, where within the same cabinet all baseband processing and analog-to-digital processing is performed, to the split architecture in cloud RAN (C-RAN), where baseband units of different eNBs can be centrally pooled while A/D conversion is performed locally by radio units. An important benefit of pooling baseband units was that it facilitates interference coordination in LTE Advanced by running associated and virtualized baseband units on the same resource pool. The introduction of C-RAN allowed to lift the hardware dependency in the pool assignment, allowing baseband units to be moved almost freely over a compute infrastructure. Nevertheless, due to the legacy of the basic function definition of LTE systems, the potential of network function virtualization could not be fully leveraged.

The architecture of 5G systems has been defined from the start with softwarization and virtualization in mind. Loosely speaking, the approach taken is to define smaller functional units that can be operated independently and thus be placed more flexibly over virtualized resources. For the RAN, this has led to the definition of radio units (RUs), distributed units (DUs) and central units (CUs) being defined as the building blocks of gNBs, the equivalent entities of LTE eNBs in 5G. The resulting architecture is more flexible, as only the RUs are dependent on specific hardware for A/D conversion and for analog handling of the signal. In contrast, DUs and CUs can be executed on COTS hardware for realizing the lower and upper parts of the RAN network stack. Frameworks like O-RAN or SD-RAN realize these vRAN implementations of 5G systems, and represent the state-of-the-art in fully softwarized mobile radio access networks.

The downside to the increase in flexibility is increased power consumption. While power consumption of a RAN results from the intimate relationship between architecture, implementation, run-time optimization and other aspects like cooling, focusing on the power consumption of a virtualized transceiver alone running on a general purpose processor steeply increases the consumption in comparison with processors custom-made for this purpose. This is not specific to any transceiver algorithm, but applies in general to almost all virtualized functions. In existing wireless transceivers this has led to the use of customized hardware environments, mostly utilizing accelerators for certain functionality, at the price of losing flexibility. With the advent of massive MIMO and its exponentially increasing processing demands, the necessity to reduce the power consumption in future virtualized RANs becomes paramount.

SECTION III – NEURAL NETWORK ABSTRACTION AND APPLICATIONS IN MOBILE AND WIRELESS NETWORKS

The significant breakthroughs in neural networks for classification and pattern recognition associated with the advances in training acceleration through the use of GPUs about ten years ago, have led to enormous interest in using machine learning for various problems in communication systems. In particular, deep learning (DL) has recently shown

great potential to become a powerful tool to design, optimize, adapt, and secure wireless communications. Deep learning makes use of deep neural networks (DNNs) which are cascades of parallel processing layers with individual connectivity degrees. In general, neural networks provide flexible abstractions to any functional input/output relationship at hand, for which sufficient training data is available. Through training, DL is capable of approximating functions and complex inter-relationships of variables that are hard to accurately describe using mathematical models. By doing so, DNNs enable novel approaches to the design of wireless communication systems without the knowledge of accurate mathematical models, for example, unknown channel models.

While the potential performance improvements of a DNN-based wireless communication system design are currently receiving significant research attention, it is evident that flexibility of the implemented processing structure, as well as implementation costs and adaptability with the evolution of communication systems are strong additional arguments for the application of DNNs in communication systems. In the following we will discuss few representative applications of DNNs to communications transceiver and protocol design.

DNNs for Low-Layer Transceiver Architectures

A plethora of research works have recently emerged studying machine learning applications to communication systems design. With respect to implemented architectures, works either focus on substituting individual functions in the transceiver chains, or more progressively substituting larger blocks, primarily in the physical layer. Examples of the first category comprise for instance works on signal detection (Samuel 2019), channel estimation (Neuman 2018), or signal demapping in broadband wireless communication systems (Shental 2019). In all cases, it is shown that deep learning, given sufficient training data, is on par with legacy (model-based) approaches, typically assuming though synthetic benchmarks instead of real-world implementations. For those benchmarks, the resulting approaches can have a lower complexity though.

Works focusing on larger functional blocks typically propose to substitute several processing steps of the physical layer by a suitable aggregate DNN. A good example is for instance the recent seminal work (Honkala 2021). It introduces a DNN-based OFDM receiver implementation, converting frequency-domain signal samples into uncoded bits with soft information. Thus, equalization, channel estimation and signal demapping are substituted by one trained neural network. The work demonstrates the applicability of this approach to 5G new radio compliant signals, showing on par performance with legacy receiver structures. Complexity-wise, the proposed neural network performs best with roughly 1 million parameters in a ResNet structure (a special type of connectivity structure of the neura network), while scaling it to larger bandwidths asymptotically becomes equivalent to linear minimum mean square error (LMMSE) receivers.

Another fundamental approach to substituting entire blocks of transceivers is given by end-to-end approaches (Dorner 2018). Here, in contrast to (Honkala 2021)., the entire transmitter and receiver are substituted through DNNs, which allows for channel- specific signaling schemes. In detail, variational autoencoders are utilized for the joint training of transmitter and receiver. Training such special deep neural data structures encompasses an end-to-end consideration, giving the approach also its name. In other words, end-to-end approaches are most consequent in moving away from model-based transceivers, potentially jeopardizing traditional system standardization. The authors showed that for narrowband, single-carrier systems the approach is in principle viable, achieving results that are on par with model- based implementations. In terms of real-time performance, the implementation is not yet up to speed, though.

DNNs for Higher-Layer Functionality

While the bulk of information processing of any wireless transceiver is related to the physical layer, deep learning has been also considered higher up in the (wireless) network stack. Focusing again first on works that consider to substitute individual functions of the network stack, efforts have been made for instance with respect to performing resource allocation for 5G networks by DNNs (Imtiaz 2021). Further works consider channel allocation for dynamic spectrum access (Naparstek 2019) or focus on improving sensing/classifying accuracy for dynamic spectrum access systems (Davaslioglu 2018).

In contrast to substituting individual functions, a different category of works focuses on automating the parameter tuning of communication protocols. When it comes to learning-based medium access protocol approaches, the research is in its infancy and mainly addresses learning optimal channel access policies. Dynamic protocol composition from a set of atomic components by means of deep neural networks has been presented in (Pasandi 2020). More work has been done in the transport layer, where deep neural networks have been proposed to design congestion control algorithms and learn an optimal TCP congestion control policy from rich parameter observations of the network environment (Queuing delay, inter-arrival times, round trip time (RTT), lost packets) (Zhang 2020). In contrast, conventional congestion control only considers several measurements such as packet loss and RTT as indicators of congestion, and cannot easily adapt to new networks or leverage experiences from the past.

The encouraging initial results on deploying DNNs in the design of communication system components and functions at different layers suggest an upcoming paradigm shift in the way how wireless and mobile networks will be architected in the 6G era and beyond. Such a paradigm shift would, however, require a software architecture that supports the flexible composition of DNN processing chains in an efficient way. In the following we propose such an architecture.

SECTION IV – AI-NATIVE SOFTWARIZED RAN: THE NEURORAN ARCHITECTURE

Neural networks show great promise for various baseband and PHY processing tasks, and could become building blocks of a future, flexible RAN architecture. Yet, adopting neural processing on top of existing software abstractions will unlikely result in efficient operation. Existing software and resource abstractions, virtual machines and containers, were optimized for data center environments with abundant, homogeneous hardware and for slowly changing environments. Consequently, they would result in significant memory and computational overhead in the implementation of an AI-native RAN. In lack of an abstraction for neural processing they would also fall short on flexibility due to the reliance on custom hardware accelerators, for example, FPGAs, and cannot efficiently support adaptive composition of processing chains on short time scales.

For a softwarized AI-native RAN to be efficient and flexible, there is a need for a software architecture where AI-native functions are ”first class citizens”, as opposed to virtual machines (VMs) and containers. The architecture should match the data-centric abstraction of computing that neural processing provides, agnostic to instructions sets, memory hierarchy, and more, which facilitates code reuse, portability and hardware maintenance. Furthermore, it should support fine-grained provisioning of resources, based on fine-grained compute actions without the need to allocate compute and network resources for long time periods (hours, days).

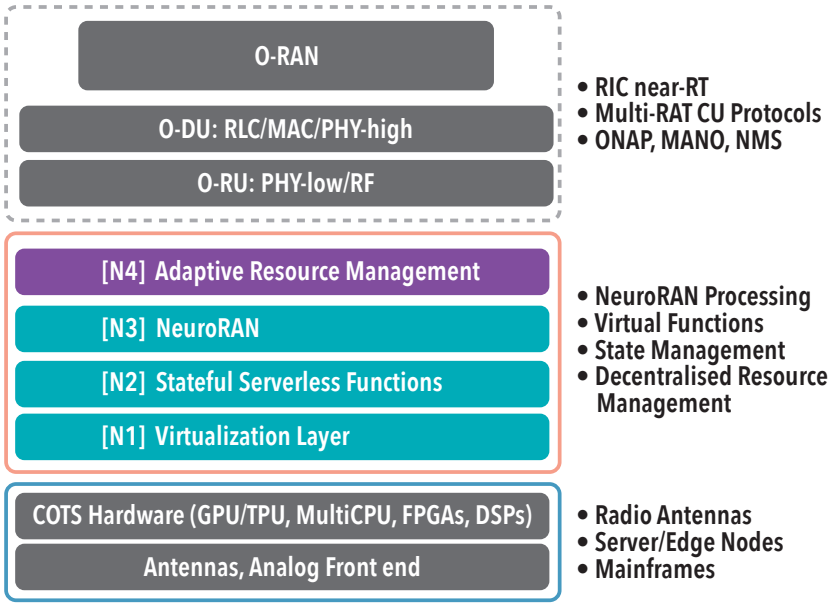


Figure 2. NeuroRAN Software Component Stack in the O-RAN Architecture

Table 1. Feature comparison of container-based and stateful FAAS abstraction for AI-native RAN

Software Feature	Container-based	sFaaS
Memory footprint	Moderate	Low
CPU overhead	Significant	Low
Scalability	Custom	On-Demand
Maintenance	Custom	Automatic
Compute/Latency-Intensity	Custom	On-Demand
Hardware/Energy Consumption	Static	Dynamic
Composability	Coarse	Fine
Platform independence	No	Yes
Resource management	Coarse	Fine
Data path	Memory	Message passing
Reliability	Custom	Built-in

To address this gap, we propose the NeuroRAN architecture, which complements the O-RAN design through four main components, illustrated in Figure 2.

First, we adopt the emerging paradigm of stateful function as a service (sFaaS) (Sreekanti 2020), which is becoming dominant in the domain of cloud virtualization for scalable applications (Gonzalez 2019) Functions are ideal building blocks to compose decentralized dataflow applications (for example, stateful streaming (Carbone 2015)) and services, incorporating data subscription and session management and on-demand scalability (group communication, multiplayer gaming, collaborative editing apps, remote and VR control systems). sFaaS could essentially replace traditional virtualization technologies through an abstraction focusing on functionality that allows resources to be allocated to actual demand, as opposed to dimensioning for peak demand. Doing so brings savings in terms of execution overhead and hence energy consumption. Second, NeuroRAN, a novel abstraction of neural-network on demand that can be supported and offered on top of stateful FaaS. NeuroRAN uses neural processing as a ubiquitous abstraction for common tensor-centric workloads extended with native support for tensor-centric hardware that is energy-efficient for tensor computations (FPGAs, TPUs) as a service. Transceiver functional blocks and other compute workloads can be made available as functions, available in different implementations that can be used interchangeably, depending on instantaneous compute, memory and storage capacity availability. Third, adoption of virtualization of compute, storage and communication resources to enable seamless migration of functions over heterogeneous hardware, alongside their corresponding state and data dependencies. Fourth, decentralized resource management, which is itself a decentralized tensor-centric data flow application in the proposed framework, which can be migrated and executed on-demand. Table I summarizes the main advantages of the proposed architecture compared to a state-of-the-art container-based architecture, as utilized for instance in O-RAN.

Starting from the bottom of the proposed NeuroRAN middleware stack (N1-N4 in Figure 2) this leads to the following software components.

- **RAN Virtualization Layer:** We propose the creation of a lightweight virtualization service that offers resource as a service capability to edge nodes with heterogeneous hardware in mind. This includes compute and memory resources such as GPUs, TPUs, flash memory, NVMe, RAM, and network resources. The virtualization layer should guarantee strong isolation and security of the resources, as well as the flexibility to compose and utilize custom configurations by combining memory, storage, network and compute components to facilitate the implementation of sFaaS on top. The virtualization layer should natively support the orchestration of the resources to meet performance metrics required by the utilized services (energy consumption, time allocation, local IO, network IO, vCPU instructions).
- **Stateful Function as a Service:** As a user-facing execution model we propose the adoption of sFaaS. We envision the use of this paradigm both for lower layer network protocol functions (RU, DU and CU) that exist in the O- RAN architecture as well as for the deployment of applications and services that can be executed on demand on top of edge nodes. The main differentiation to serverless functions known in cloud computing (e.g., Amazon Lambda) is the addition of explicit state and support for building end-to-end decentralized dataflow services composed out of interconnected functions that can communicate through remote invocations or via message passing.
- **NeuroRAN:** As discussed in the previous section, most RAN functions that

Table 2. Overview of stateful and stateless transceiver functional blocks

Function	Stateful?	Parameters	State
A/D Conversion	No	Baseband sampling frequency	
STO, CFO correction	No	Signal type	
PHY PDU extraction	Yes		Scheduling grant
Channel estimation	No	Signal type, pilot position	
Equalization	No	Signal type, channel estimate	
Signal (de)mapping	Yes		Modulation scheme
(De-)Interleaving Encoding,	No	Interleaving scheme	Coding scheme,
Decoding	Yes		
CRC insertion,check	No	Block length, CRC scheme Key,	coding block length
En-/De-cryption	Yes	encryption scheme	Cipher state

today are model-driven can be substituted by data-driven implementations. NeuroRAN allows for automating the deployment of data-driven implementations of RAN functions and other edge micro-services, including network functions (NFs) in the core network and emerging network data analytics function (NWDAF) services. NeuroRAN instances can be created, instantiated, trained, and used for inference via a standard programming interface.

- **Decentralized Resource Management:** A crucial component of the proposed middleware architecture is decentralized resource management. Supporting a FaaS deployment model requires the ability of the middleware to migrate functions and their corresponding state across nodes as well as scaling out resources elastically when needed. A resource management middleware can support and optimise these functions towards fair allocation of resources across decentralized edge nodes. Furthermore, this type of service must be adaptive to cope with the dynamic nature of edge networks (failure detection, service discovery, reconfiguration, and more). In this setting service level agreements (SLAs) can specify requirements in terms of latency, throughput, and availability, and the objective of resource management is to minimize energy consumption subject to meeting all SLAs.

While many candidate application workloads for edge computing are inherently tensor-centric, such as visual analytics, autonomous industrial systems, cars and drones, a fundamental question for the feasibility of the proposed architecture is whether typical transceiver functional blocks can be implemented in the FaaS model. Table II shows a list of radio transceiver functional blocks, their typical parameters and whether they require state to be maintained. The table indicates that out of ten functional blocks four require a stateful FaaS implementation, and for three of those the state is due to their dependence on scheduling decisions, which are performed at the same time scale as the functions would be invoked.

SECTION V – RESEARCH CHALLENGES

The proposed architecture introduces a set of core challenges that we foresee are necessary to be addressed. We categorize them in a bottom-up fashion from the virtualization layer up to resource and service management.

- **Virtualization over Heterogeneous Hardware:** At the lowest level we identify a set of problems that need to be addressed in order to enable seamless support for sFaaS, ranging from compilation and code generation to deployment and provisioning. Heterogeneous hardware should be supported out of the box without the need for reconfiguring and recompiling libraries in edge nodes. To that end, there is a need for on-demand code generation techniques that can translate high-level sFaaS program specifications to low-level instructions supported by the underlying hardware (GPUs, multicore x86, TPUs and DSPs, for compatibility with legacy RAN hardware). Recent works in compiler research adopt intermediate code representations and build on widely adopted LLVM libraries, such as multi-level intermediate representation (MILR (Lattner 2020)), which already support a number of existing representations and compilation tools for current and upcoming hardware architectures. Finally, we foresee it to be necessary to employ decentralized provisioning of services using sFaaS with respect to 3 core usage metrics: aggregated number of invocations, IO (state size) and data transferred across functions. The management of all the provisioning of these metrics needs to be made consistently and efficiently, which is a challenging task itself in a heterogeneous environment.
- Resource Management in Dynamic and Heterogeneous Environments: A key enabler of the proposed architecture is decentralized scheduling enabling on-demand resource allocation, instantiation, migration and invocation of stateful FaaS. Existing approaches to resource management do not apply well to neural network abstractions and cannot provide throughput and latency guarantees in highly distributed environments. Recent efforts on applying deep learning to resource management would need to be extended to the multi-agent setting, but the convergence and stability of the resulting systems is not well understood. A promising direction could be the use of graph convolutional network (GCN) embeddings, possibly starting from an initial model through iterative refinement and including performance SLAs as multi-objective optimisation variables in the training process of the network. Doing so could provide predictable performance in the spirit of safe machine learning.
- **Reliability and Security:** Among the biggest challenges of adapting cloud computing technologies to decentralized edge networks are reliability and security. Reliability not only involves provisioning of resources in response to failures, but also processing guarantees (number of times a function is executed) and consistency guarantees (for service state) despite failures. This is especially challenging in edge networks under high churn. In addition, FaaS requires strong isolation guarantees (process memory, non-volatile storage, virtual CPUs) offered by the underlying virtualization layer, which is important both for accurate provisioning as well as for the secure execution of FaaS. Finally, state management for FaaS over COTS

opens many new challenges such as the need for automated state partitioning, migration, encryption-support as well as consistency guarantees at a dataflow- or FaaS-level in the presence of partial failures.

SECTION VI – CONCLUSION

Motivated by the increasing importance of flexibility and energy-efficiency in RAN operations, in this article we discussed the requirements that the migration to an AI-native RAN would impose on software abstractions in beyond 5G networks. We presented arguments that show that combining flexibility with energy efficiency would require going beyond existing abstractions, and could be possible by extending the emerging serverless computing paradigm to a stateful FaaS model combined with a neural abstraction of computing functions. The proposed architecture has the potential to meet the performance and reliability requirements of beyond 5G wireless networks, but it remains to understand what tradeoffs it involves in terms of security. ■

REFERENCES

- O-RAN Alliance. 2022. https://www.o-ran.org/.
- Carbone, P., S. Ewen, and S. Haridi. 2017. "State management in Apache Flink: consistent stateful distributed stream processing." in Proc. of the VLDB.
- Davaslioglu, K., and Y. Sagduyu. 2018. "Generative Adversarial Learning for Spectrum Sensing." *Proceedings of the IEEE International Conference on Communications (ICC)*.
- Dorner, S., S. Cammerer, J. Hoydis, and S. ten Brink. 2018. "Deep learning based communication over the air." *IEEE Journal of Selected Topics in Signal Processing*, 12 (1): 132–143.
- Gonzalez, J., R. Popa, I. Stoica, and D. Patterson. 2019. "Cloud Programming Simplified: A Berkeley View on Serverless Computing." *Berkeley Technical Report*.
- Honkala, M., D. Korpi, and J. Huttunen. 2021. "DeepRx: Fully Convolutional Deep Learning Receiver." *IEEE Transactions on Wireless Communications*.
- Imtiaz, S., G. Koudouridis, and J. Gross. 2021. "Coordinates-Based Resource Allocation through Supervised Machine Learning." IEEE *Transactions on Cognitive Communications and Networks*.
- Lattner, C., M. Amini, U. Bondhugula, A. Cohen, A. Davis, J. Pienaar, R. Riddle, T. Shpeisman, N. Vasilache, and O. Zinenko. 2020. "MLIR: A compiler infrastructure for the end of Moore's law." arXiv:2002.11054.
- Naparstek, O., and K. Cohen. 2019. "Deep Multi-User Reinforcement Learning for Distributed Dynamic Spectrum Access." *IEEE Transactions on Wireless Communications*, 18 (1): 310-323.
- Neumann, D., T. Wiese, and W. Utschick. 2018. "Learning the MMSE Channel Estimator." *IEEE Transactions on Signal Processing*, 66 (11): 2905-2917.
- Pasandi, H., and T. Nadeem. 2020. "MAC Protocol Design Optimization Using Deep Learning." *Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*.
- Samuel, N., T. Diskin, and A. Wiesel. 2019. "Learning to detect." *IEEE Transactions on Signal Processing*, 67 (10): 2554-2564.
- Shental, O., and J. Hoydis. 2019. "Machine LLRning: Learning to softly demodulate." *Proceedings of IEEE Globecom Workshops*.
- Sreekanti, V., V. Wu., C. Xiayue, J. Schleier-Smith, J. Gonzalez, J. Hellerstein, and T. Alexey. 2020. "Cloudburst: Stateful Functions-as-a-Service." *Proceedings of the VLDB Endowment*.
- Zhang, T., and S. Mao. 2020. "Machine Learning for End-to-End Congestion Control." *IEEE Communications Magazine*, 58 (6): 52-57.

ABOUT THE AUTHORS

P. Carbone, **György Dán**, and **James Gross**, are with KTH Royal Institute of Technology, Stockholm, Sweden.

Bo Göransson is with Ericsson Standards & Technology and KTH Royal Institute of Technology, Stockholm, Sweden.

M. Petrova is with RWTH Aachen University, Germany and KTH Royal Institute of Technology, Stockholm, Sweden.

Lüdtke et al. continued from page 73

and model-based design. He was responsible for human modeling in several national and international projects and has coordinated five European projects. In 2014 he founded the spin-off Humatects GmbH to commercialize research results on human modeling.

Jan-Patrick Osterloh received his diploma in computer science from the Carl von Ossietzky University Oldenburg in 2005. He started as a researcher at OFFIS e.V. in the Transportation Division in 2005. He became a senior research engineer at OFFIS in 2009 and transitioned into the Institute of Systems Engineering for Future Mobility at the German Aerospace Center in 2022. His research interests are cognitive modeling, human factors, simulation, and training, mainly in the aeronautics domain but also in automotive.

Jakob Suchan is a researcher at the Institute of Systems Engineering for Future Mobility at the German Aerospace Center (DLR), where he is part of the Human-Centered Engineering group. Previously, he was based as a doctoral researcher at the Human-Centered Cognitive Assistance Lab at the faculty of mathematics and informatics, University of Bremen, Germany. His research interests are on semantically grounded human-centered abstraction and reasoning, focusing on interpretation, explanation, and learning of visuospatial dynamics and (human) interactions.

Alexander Trende received his BSc and MSc in physics from Ludwig Maximilian University of Munich and Carl-von-Ossietzky University Oldenburg, respectively. He is currently researching at the Institute of Systems Engineering for Future Mobility at the German Aerospace Center. He is also pursuing his PhD in computer science at Carl-von-Ossietzky University. His research interests include applied machine learning and human factors.

AI4SE and SE4AI: Setting the Roadmap toward Human-Machine Co-Learning

Kara Pepe, kpepe@stevens.edu; and **Nicole Hutchison,** nicole.hutchison@stevens.edu

ABSTRACT
Artificial intelligence (AI) and machine learning (ML) technology are becoming increasingly critical in systems: both to provide new capabilities and in the practice of systems engineering itself, especially as digital transformation improves the automation of many routine engineering tasks. The application of AI, ML, and autonomy to complex and critical systems encourage the development of new systems engineering methods, processes, and tools. This article highlights a series of workshops conducted jointly by the US Army Combat Capabilities Development Command Armaments Center (CCDC AC) Systems Engineering Directorate and the Systems Engineering Research Center (SERC). These workshops focus on the relationships between AI and systems engineering and elicit input from hundreds of stakeholders across government, industry, and academia. They also provide critical direction to the SERC's research roadmap on AI/autonomy as it looks towards the long-term outcome of "human-machine co-learning." Though the workshops are US-centric, the lessons and insights gained are applicable globally.

INTRODUCTION

Systems engineering is undergoing a digital transformation that will lead to the further use of Artificial intelligence (AI) and machine learning (ML) technology to automate many routine engineering tasks. AI seeks to provide intellectual processes like those of humans, including the "ability to reason, discover meaning, generalize, or learn from past experience (Copeland 2021)." ML is a branch of AI and computer science which focuses on the use of data and algorithms to imitate the way that humans learn and gradually improve accuracy (https://www.ibm.com/topics/machine-learning). ML-based approaches can include logistic regression, neural networks, deep learning, decision trees, and so forth. Automation is defined as a technology that performs tasks independently, without continuous input from an operator (Groover 2020). Tasks can be fully automated (autonomous) or semi-automated, requiring human oversight. Automation is also defined as "the execution by a machine agent of a function that was previously carried out by a human. What is considered automation will, therefore, change with time (Parasuraman and Riley 1997)." For this article, "autonomy" is well-designed and highly capable automation. Thus, the application of AI, ML, and autonomy to complex and critical systems *encourage* the development of new systems engineering methods, processes, and tools.

To address this evolving reality, the US Army Combat Capabilities Development Command Armaments Center (CCDC AC) Systems Engineering Directorate and the Systems Engineering Research Center (SERC), a university affiliated research center (UARC) for the US Department of Defense (DoD), co-founded the artificial intelligence for systems engineering and systems engineering for artificial intelligence research and application workshops (AI4SE and SE4AI). So far, three workshops have been held: the first on October 28-29, 2020 (virtual), the second on October 20-21, 2021 (virtual), and the third on September 21-22, 2022 (hybrid). These events gathered government, academic, and industry communities to learn from leaders using AI in systems engineering and to share ideas focused on the workshops' main objectives: how to define relevant systems engineering and AI challenges, explore methodologies to address them, and identify ways to collaborate and conduct research that addresses these challenges in the coming years.

MOTIVATION

The annual AI4SE and SE4AI workshops aim to prepare future systems engineering processes for a world where humans and machines co-adapt and team to evolve complex missions in response to dynamic operational conditions. As the volume of

information needed to develop and test increasingly complex systems grows, AI will be critical to support digital engineering (DE), yielding time and testing efficiencies; building up assurance models; and advancing the move toward cognitive technologies. This will result in an increasing need for organizations and systems to be agile enough to keep up with the dynamic nature of AI, driving toward the goal of delivering the most relevant and effective tools to the warfighter in the field.

CCDC AC is working to understand and manage the innovations AI brings to systems engineering processes, which can lead to benefits for both areas: systems engineering can support AI/ML weapon system projects and initiatives, and using AI/ML in systems engineering best practices can improve CCDC AC's lifecycle support. In 2020, CCDC AC partnered with the SERC to establish the AI4SE and SE4AI workshops, which provide an annual opportunity for exchange among and contributions by representatives of government, academia, and industry toward continued refinement and rich application of AI and ML. Discussions in the 2020 and 2021 workshops focused on identifying how AI can support and create efficiencies in systems engineering; developing ways to apply effective systems engineering to AI-intensive systems; and exploring DE and its relationship with AI.

CRITICAL CHALLENGES

Roadblocks to progress have been evident since the inception of the annual workshops, and both events to date sought to address identified challenges such as:

1. capturing and developing systems engineering artifacts;
2. determining, in a dynamic fashion, AI value on the battlefield;
3. analyzing AI solutions;
4. structuring data for consumption by AI;
5. building trust in AI through transparency and reliability; and
6. providing workforce training and development for effective integration of AI.

To address these challenges, CCDC AC and the SERC jointly hosted two-day virtual workshops in 2020 and 2021. The remote nature of these workshops allowed for wider participation and sharing of ideas and insights among participants. Keynote and introductory speakers provided their relevant perspectives for the daily sessions. The results from the third workshop held in September 2022 are still being integrated into the roadmaps and the published report forthcoming.

WORKSHOP DESCRIPTIONS

AI4SE & SE4AI 2020

The 2020 inaugural workshop recognized that AI and ML are actively shaping the development of weapon systems and have the potential to transform the battlefield "from the back office to the front lines (Esper 2020)." Realizing the promise of AI and ML prompts the development of multidisciplinary activities in which systems engineering methodology can be applied to the development of AI technologies that will, in turn, streamline and improve the development of systems. It was acknowledged that the span of such activities should include workforce development; requirements and architecture; systems engineering technical management; systems analysis; and advancing systems engineering methodology. Thus, it makes sense that data and people were reoccurring themes throughout the workshop's various tracks and presentations.

Over 200 workshop attendees represented more than 70 organizations, including other government agencies (OGA) as well as industry and academia affiliates. The agenda was structured into four sessions:

1. Machine learning/artificial intelligence (ML/AI);
2. Artificial intelligence for systems engineering (AI4SE);
3. Systems engineering for artificial intelligence (SE4AI); and
4. Digital engineering (DE).

The workshop included three to four presentations on each of these relevant topics. Audience members collaborated and asked questions throughout the sessions.

A theme of the workshop was that of data and its efficient and reliable acquisition, analysis, verification, maintenance, and distribution. This is a key component across all initiatives, spanning the research and development lifecycle with the goal of delivering relevant and trustworthy products to the warfighter on the field faster. Tasks related to data emphasize the importance of collaboration among all stakeholders to leverage existing expertise, address identified gaps, produce efficiencies, drive other efforts, and achieve wins, as well as to change culture and evolve ways of operating toward a common infrastructure.

People are a key component of transformation initiatives, as well as a challenge, highlighting the need to change the "narrative" — that is, the established thinking and approach toward outcomes and the recruitment and retention of capable people — to enable effective AI workforce development. Continuous training is needed to prepare individuals, keep knowledge up-to-date, and compensate for skills lost as more tasks become automated.

The 15 presentations (https://sercuarc.org/event/ai4se-and-se4ai-workshop/) delivered over October 28-29, 2020 focused on relevant topics within the four areas of:

1. **Machine Learning/Artificial Intelligence (ML/AI)**
   Presentations highlighted research and initiatives focused on AI-enabled tools that can: extract semantic insights based on predefined points of view; determine when a particular engagement is balanced and how to optimally achieve imbalance; and defend against adversarial attacks by gaining insight into their behavior and similarities.
2. **Artificial Intelligence for Systems Engineering (AI4SE)**
   Presentations highlighted how AI can: support the (human) engineer in tasks such as identifying gaps as requirements are formulated and detecting patterns-of-interest within model- based systems engineering (MBSE) models; and automate time-consuming and costly processes subject to error, such as data analysis, and the development of embedded operational control systems.
3. **Systems Engineering for Artificial Intelligence (SE4AI)**
   Presentations highlighted the benefits of applying the rigors of systems engineering to the development of complex AI and ML-enabled technologies and capabilities that can accomplish tasks such as: anticipating and detecting drops in model performance in new environments; facilitating knowledge transfer for rapid and effective mission integration; generating human data sets for ML analyses; and identifying the human limitations of using AI aids.
4. **Digital Engineering (DE)**
   Presentations highlighted research focused on how to design intelligent systems that can complement, support, and improve the efficiency and reliability of manual processes. Examples and case studies included creating ML methods to: reduce system development time, cost, and design errors; design digital twin architectures that support AI and ML formalisms working side-by-side as a team; support on-terrain decision making; and reduce the number of human test subjects required in developing assessment methods.

AI4SE & SE4AI 2021

The 2021 workshop enabled exchange among and contributions by representatives of government, academia, and industry toward continued refinement and rich application of AI and ML. Presentations and panels focused on opportunities for AI to augment systems engineering processes to maximize outcomes and improve workflows, particularly system verification, validation, and resilience – critical aspects of today's complex and evolving systems – as well as the issues raised by bringing together AI and the engineering workforce, including recruitment and training.

In 2021, over 250 attendees again spanned various OGAs as well as industry and academia affiliates. The agenda was structured into three tracks:

Verification and validation (V&V) of AI systems;

1. Lifecycle adaptation and resilience; and
2. Augmented engineering.

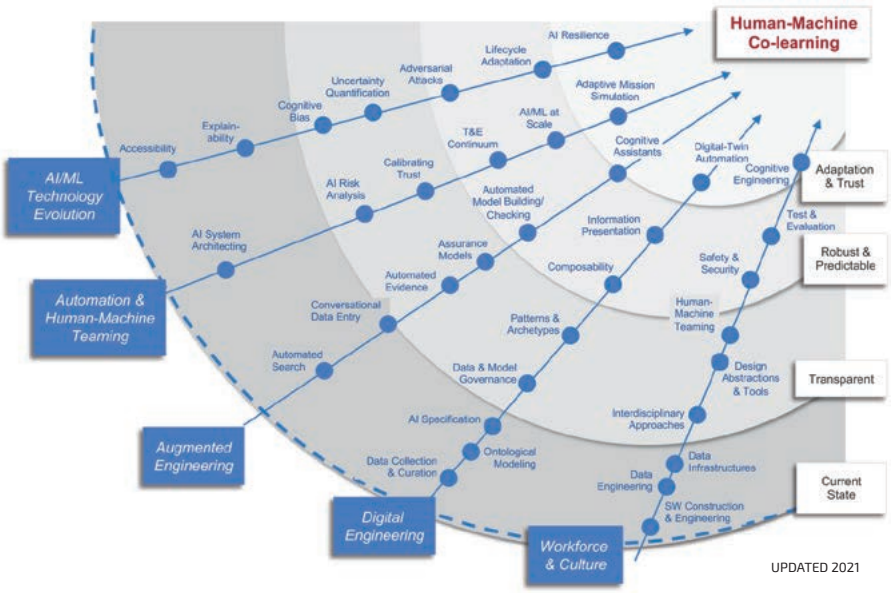


Figure 1. SERC AI/autonomy roadmap (https://sercuarc.org/wp-content/uploads/2021/08/ROADMAPS_3.5.pdf)

The SERC Research Roadmap (2021), a high-level, aspirational research vision linked to the goals of the DoD *Digital Engineering Strategy* (2021), was highlighted at the 2021 workshop as the basis for much of the work the SERC is undertaking (DoD 2018). It focuses on how AI can support DE as the volume of information needed to develop and test increasingly complex systems grows, yielding time and testing efficiencies, building up assurance models and advancing the move toward cognitive technologies.

Conclusions drawn from the 2021 workshop included that AI's promise is linked to its effective use of data; however, systems can only adapt to the data on which they are trained. There is a need to prioritize the design of systems that collect quality data for the most useful application and with attention to biases and suitability. Human-AI teaming will be critical to successfully implementing such systems, as people provide greater agility than AI in novel, unforeseen situations inherent in warfare. Transparency is critical to AI reliability, which in turn ensures human trust in machines.

At the workforce level, education and continuous training are critical to managing new disruptive technology, enabling the workforce to build the appropriate infrastructure, and making the infrastructure easier to use. Education and training also minimize resistance to change.

The 18 presentations (https://sercuarc.org/event/ai4se-and-se4ai-workshop-2021/) delivered over October 20-21, 2021 focused on relevant topics within the three areas of:

1. **Verification and Validation of AI Systems**
   Presentations highlighted research focused on novel testing tools to support the development of intelligent systems and capabilities for use in dynamic environments. Approaches included reframing ML from a problem-solving endeavor to a system and using AI-assisted tools during development and testing to identify hazardous outcomes that have not yet occurred in operations.
2. **Lifecycle Adaptation and Resilience**
   Novel design alternatives were presented that can be used throughout a system's lifecycle, including case studies demonstrating the applications of cognitive assistants to support systems engineers during the problem formulation phase of trade space exploration. Digital twins of live and/or hypothetical insider threat detection enterprises, created for the purpose of performing testing and evaluation on continuous monitoring systems sensitive to disruptions, were also presented.
3. **Augmented Engineering**
   Presentations highlighted how applying AI/ML-enabled capabilities and tools to engineering can expand the possibilities and time- and resource-saving benefits of the resulting optimized technologies. Case studies included the use of natural language processing (NLP) to reduce the resource and time cost of translating free text; effectively manage knowledge for intelligence purposes; and extract usable representations from text to generate actionable models in near real-time that require minimal to no human intervention for updates/upgrades.

WORKSHOP OUTCOMES AND LESSONS LEARNED

The inaugural workshop concluded that current SE practices do not support the long-term outcome of "human-machine co-learning", which will necessitate an upcoming evolutionary phase in the systems engineering community in three "waves":

- technologies and approaches that increase the transparency of decisions produced by AI systems;
- the production of systems that learn and are robust and predictable in the type of key applications normal to systems engineering; and
- systems that adapt and learn dynamically from their environments, which will develop trust in machine-to-machine and human-to-machine (and maybe machine-to-human) interactions.

AI AND AUTONOMY ROADMAP

These conclusions were used to refine the SERC AI and autonomy research roadmap that was presented in the 2021 workshop and that now converges into human-machine co-learning. The research roadmap spans five categories as illustrated in Figure 1:

1. AI/ML technology evolution;
2. Automation and human-machine teaming;
3. Augmented engineering;

4. Digital engineering; and
5. Workforce and culture.

The range of issues explored in 2020 demonstrated the complexity of the topic. The three tracks of the 2021 workshop focused on ways to view and work through the complexity of human-machine teaming and the associated endeavors being undertaken by systems engineering. Presentations and discussions considered what systems engineering can do to address the limitations of AI, and what challenges AI presents to the systems engineering discipline. V&V and test and evaluation (T&E) of new intelligent systems must be conducted in ways that account for both current and future uncertainties of the tests being used, as well as for confidence in the enhanced systems.

The resulting outcomes and insights from 2020 and 2021 include:

- *View AI as a system, not a technology*. Designers, requirement setters, and systems engineers should consider what can be done to mitigate the uncertainty of AI and provide guardrails against unexpected events from the start of system design. Testing of AI should be done in the context of a system, and systems can be shaped to recognize and compensate for the limitations of the AI. Powerful tools are currently available, but these do not yet address the needs of the systems engineering community. The systems engineering community needs to consider whether there is a way to stimulate or organize an expression of its demand for the specific, appropriate products it needs.
- *Focus on how AI4SE and SE4AI come together in the lifecycle*. Digital twins become indispensable tools for managing the lifecycle of AI systems and are also indispensable to the systems engineers when thinking about the design of those systems.
- *Expand thinking around training at all levels*. Training must be an investment focus, both for the users and the developers of systems with AI. New AI capabilities can train warfighters in use of the next generation of systems as well as train those who create the tools to deal with systems that rely on AI. The US needs to catch up to other global players who are already considering and working on this.
- *Reconsider the research infrastructure.* There is a need to envision and create an infrastructure that enables the study—at every stage of development and use—of human-machine teaming issues and outcomes. A re-envisioned infrastructure could allow for the study of better integration of live and virtual testbeds to accelerate learning around new technologies.
- *Navigate cultural and policy roadblocks*. AI-based applications will permeate future systems engineeing and acquisition practices. AI applications are fundamentally digital (data and software) and will evolve with the digital transformation of government policy and practice. Services will have to share data, integrate with each other, and reconsider existing siloed cultures and business processes. The Acquisition Innovation Research Center (AIRC) is undertaking research on policies related to new technologies and whether they drive or inhibit progress.

Overall, it is important to remember that people, from systems engineers to end users, want tools that deliver new capabilities. The tools need to increase productivity and efficiency, helping these capabilities reach users at the speed of relevance. AI and automation will enable that. Individuals also seek richer, more meaningful jobs; AI can assist this, too, by automating mundane tasks. Systems engineers need to identify the problems to be solved and the opportunities to be harnessed.

Unprecedented technology generates uncertainty. Any loss of confidence and trust in new technologies and systems will set back progress. Safety is a primary concern, but there is a need to be practical about the level of confidence and trust achievable. Attention needs to be paid to the risks posed by not moving forward technologically or falling behind the competition. The greatest risk is ultimately passed on to the end user. The mix of industry, academia, and government represented at the 2021 and 2020 workshops is where the answers and solutions lie and will continue to come.

The 2022 Workshop

The 2022 AI4SE & SE4AI Workshop took place September 21-22 as a hybrid event at Stevens Institute of Technology, in Hoboken, NJ, again hosted jointly by CCDC AC and the SERC (https://sercuarc.org/event/ai4se-and-se4ai-workshop/). The objective of the 2022 workshop was to generate relevant research ideas aligned with US priorities intended to produce actionable applications of AI4SE and SE4AI. Topics of relevance included: applications of AI on systems engineering and systems engineering on AI projects; digital engineering and methods in support of AI; automation and human-machine teaming; advances in explainable AI; cognitive assistants and decision-aiding tools; and workforce development. The results of the workshop are still being integrated into the SERC AI research roadmap and will be available along with the public report later.

CONCLUSIONS

The goal of SERC research in AI and autonomy is to lead the development of systems engineering dynamic processes that leverage the speed and rigor of rapidly evolving modeling, simulation, and analysis enabled by computational intelligence. The technical domains of AI, ML, and autonomy encompass a broad range of methods, processes, tools, and technologies that are still emerging. The research roadmap in Figure 1 illustrates this evolutionary framework and includes abstraction and high-level design methods; approaches to design for "X" (where "X" is a particular characteristic); and design for test and certification. This leads to the ability to specify technology requirements; develop tools to accelerate and scale design, modeling, and simulation at the mission level; and incorporate AI into operational testing.

The vectors of this notional roadmap span five categories:

- *AI/ML technology evolution* recognizes that the technological implementation of AI systems will evolve and will need to evolve in directions relevant to systems engineering. Most of these can be related to the development of transparency and trust in technology.
- *Automation and human-machine teaming* recognizes that the purpose of AI in systems is generally to provide automation of human tasks and decisions.
- *Augmented engineering* recognizes that AI technologies will gradually be used more and more to augment the work of engineering.
- *Digital engineering* recognizes that the current digital engineering transformation will be an enabler for augmented engineering.
- *Workforce and culture* recognizes a significant transformation will need to be accomplished in the systems engineering workforce, with significantly greater integration of software and human behavioral sciences at the forefront.

Continued engagement of a broad community of researchers and practitioners is critical to evolving a research portfolio that enables the co-evolution of systems engineering and AI. CCDC AC and the SERC will continue to conduct annual workshops to foster this coordination. ■

REFERENCES

- Copeland, B. J. 2021. Artificial Intelligence. https://www.britannica.com/technology/artificial-intelligence.
- DoD. 2018. Digital Engineering Strategy. Arlington, US-VA: US Department of Defense (DoD). June 2018. https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf
- Esper, M. T. 2020. "Secretary of Defense Remarks for DoD Artificial Intelligence Symposium and Exposition." (speech). AI2020: Department of Defense (DOD) Artificial Intelligence Symposium and Exposition hosted by the Joint Artificial Intelligence Center (JAIC). 9 Sept. https://www.defense.gov/News/Speeches/Speech/Article/2341130/secretary-of-defense-remarks-for-dod-artificial-intelligence-symposium-and-expo/
- Groover, M. 2020. *Automation, Production Systems, and Computer-Integrated Manufacturing*, 5th ed. New York, US-NY: Pearson.
- IBM. https://www.ibm.com/topics/machine-learning
- Parasuraman, R., and V. Riley. 1997. "Humans and automation: Use, misuse, disuse and abuse." *Human Factors*, 39 (2): 230–253.
- SERC. 2020. AI4SE & SE4AI Workshop 2020. Hoboken, US-NJ: Systems Engineering Research Center (SERC). 28-29 October 2020. https://sercuarc.org/event/ai4se-and-se4ai-workshop/
- SERC. 2021a. Systems Engineering Research Roadmaps. Hoboken, US-NJ: Systems Engineering Research Center, Stevens Institute of Technology. https://sercuarc.org/wp-content/uploads/2021/08/ROADMAPS_3.5.pdf
- SERC. 2021b. AI4SE & SE4AI Workshop 2021. Hoboken, US-NJ: Systems Engineering Research Center (SERC). 20-21 October 2021. https://sercuarc.org/event/ai4se-and-se4ai-workshop-2021/
- SERC. 2022. AI4SE & SE4AI Workshop 2022. Hoboken, US-NJ: Systems Engineering Research Center (SERC). 21-22 September 2022. https://sercuarc.org/event/ai4se-and-se4ai-workshop-2022/ _**report pending public release*

ABOUT THE AUTHORS

Kara M. Pepe is the director of operations at the Systems Engineering Research Center (SERC), a university affiliated research center for the Department of Defense at Stevens Institute of Technology. Prior to joining SERC, Kara was the director of industry and government relations at the Center for Complex Systems and Enterprises, working with the various government, private sector, and non-profit organizations that engaged and funded research initiatives for the center. She received her ME in systems engineering and BE in engineering management from Stevens and is currently pursuing a PhD. Her research focuses on digital engineering in general with an emphasis on workforce development. Kara is a member of INCOSE, NDIA, SWE, and ASEM.

Dr. Nicole Hutchison is a principal investigator (PI) and research engineer at the Systems Engineering Research Center (SERC). Her primary work through the SERC has been in human capital development research. This has included the development of competency frameworks for systems engineering the Helix project (http://helix-se.org/), digital engineering (https://sercuarc.org/serc-programs-projects/project/86), and mission engineering (https://sercuarc.org/serc-programs-projects/project/58). Currently, Dr. Hutchison is the PI for the simulation training environment for digital engineering (STEDE), a project that is developing realistic models to be used in training the DoD acquisition workforce in a way that builds hands-on digital engineering skills. She previously served on the BKCASE research team, which resulted in the development of the *Guide to the Systems Engineering Body of Knowledge* (SEBoK). She is currently the managing editor for the SEBoK and the lead editor for the "Enabling Systems Engineering" section of the SEBoK. Before joining the SERC, she spent 5 years working for Analytic Services, Inc., supporting the US Departments of Defense, Homeland Security, Health and Human Services, and Justice. She holds a PhD in systems engineering from Stevens and is a certified systems engineering professional (CSEP) through INCOSE.

DECEMBER 2022
VOLUME 25/ ISSUE 4

Modular Over-the-air Software Updates for Safety-critical Real-time Systems

Domenik Helms, domenik.helms@dlr.de; **Patrick Uven,** patrick.uven@dlr.de; and **Kim Grüttner,** kim.gruettner@dlr.de

■ **ABSTRACT**
Automotive software is undergoing a rapid change toward artificial intelligence and towards more and more connectedness with other systems. For both, an incremental design paradigm is desired, where the car's software is frequently updated after production but still can guarantee the highest automotive safety standards. We present a design flow and tool framework enabling a DevOps paradigm for automotive software development. DevOps means that software is developed in a continuous loop of development, deployment, usage in the field, collection of runtime data and feedback to the developers for the next design iteration. The software developers get support in defining, developing, and verifying new software functions based on the data gathered in the field by the previous software generation. The software developers can define contracts describing the time and resource assumptions on the integration environment and guarantees for other dependent software components in the system. These contracts allow a composition of software components and proof obligations to be discharged at design time through virtual integration testing and runtime through continuous monitoring of assumptions and guarantees on the software component's interfaces. An update package, consisting of the software component and its contracts, is then automatically created, transferred over the air, and deployed in the car. Monitors derived from the contracts allow for supervising the system's behavior, detecting failures at runtime, and annotating the situation to be included in a data collection, fueling the next design iteration.

■ **KEYWORDS:** DevOps; safety-critical; over the air updates; contracts; monitoring

WHY DOES THE AUTOMOTIVE INDUSTRY NEED UPDATES?

Updates for software have been around for almost as long as software exists. No matter how extensively software is tested before delivery, there can always be situations during operation that should have been considered during testing. Moreover, the environmental conditions for the software can always change over time, for example, through changes in the environment, new hardware, or new requirements for the program.

The procurement and installation of updates used to be a very time-consuming, partly manual process at the time of the first Windows programs. It has long since been solved, with the latest app marketplaces such as Google Play or the Microsoft Store allowing developers to post updates in these stores. The availability and compatibility of an update are then automatically identified. That update is automatically installed–depending on the software and hardware configuration–without user interaction. The separation of all high-level functionality into individual and separately updateable program parts (apps) is also known. It has been common practice for a long time on PCs and cell phones.

For non-safety-critical systems, updates are already commonplace today only because there are no life-threatening consequences in the event of an update error. The update software, of course, can have errors like any software. In addition, installing the new software or its interaction with existing software components can also lead to errors or complete system failure. There is currently no safe solution for safety-critical systems such as driving cars that rule out a system failure with potentially fatal consequences due to a faulty update or triggered by the update process itself, even for rare edge cases. Cars are susceptible because they run programs with complex real-time requirements, so certain programs must run in fixed time bounds; for example, to retrieve each new output of a sensor exactly once and then respond correctly and with a constrained delay to the sensor data.

A common practice in the example of the car is currently to apply updates and test the correctness of the updating process in the workshop during car maintenance in safety

to avoid these real-time problems. However, this already takes hours today, mainly because the entire vehicle software is updated at once to exclude accidental, malicious interactions with other program parts.

As vehicle software becomes more complex, the transfer and installation of a new complete update will take longer and longer, and more and more individual vehicle configurations (with/without lane assist, with/without parking assist) will also become more complex. On the other hand, new usage concepts such as car sharing will lead to ever higher utilization times, for example, fewer and fewer downtimes during which updates can be performed. Thirdly, with more and more driving functions, more and more updates will also become necessary. For AI-based systems, car manufacturers want the systems to detect rare and critical situations during operation and learn from them. The improved artificial intelligence is then to be distributed to all vehicles so that the entire fleet can benefit from the experience of individual vehicles. For systems that communicate with each other or with the cloud, it should also be possible to update them after delivery, as an update of one system may render updates to other communicating systems necessary.

All in all, updates will become more frequent, take longer, and vehicles will be out of service less time and less often to receive and implement such updates in a non-critical driving state.

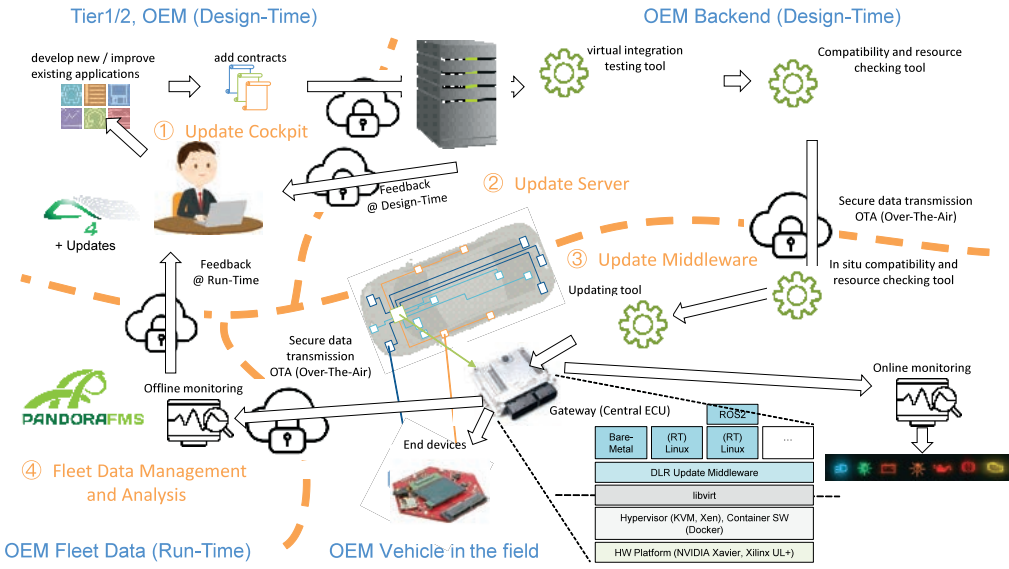


Figure 1. Concept for an automotive continuous development / continuous integration loop: ① The designer uses fleet data to develop an update for an automotive function and defines contracts to describe the system's time and resource requirements; ② The update is virtually tested for compatibility, timing, and resource demand. Monitors are generated from the contracts, which will subsequently safeguard the running function; ③ The update package is transferred via a wireless connection to the car, where it is again checked for compatibility and resource demand; Then it is installed without influencing other running functions using modularization and partitioning; Monitors constantly supervise the functions in accordance with the time and resource specifications; In the car, monitoring events can be used to circumvent critical situations outside the designer's initial assumptions; ④ Rare events in terms of timing, resource constraints, and rare functional behavior are transferred back to the manufacturer and collected for the next design iteration

SAFE UPDATES BY MODULARIZATION AND PARTITIONING

On the technical side, vehicles have long had a need and tendency to consolidate various software functions into a few central hardware units. In the past, almost every function executed in software, from the transmission of a switch position to the control of engine functions, was performed by a separate, independent hardware component. Still, for decades now, more and more functions have been consolidated into fewer and fewer individual hardware components. Consequently, this leads to mixed critical systems: hardware components that can simultaneously execute applications with burdensome real-time requirements, such as a brake assistant, and non-critical comfort functions, such as the navigation system software or air conditioning control.

To prevent mutual interference between the safety-critical components or between the safety-critical and non-critical components, it was proposed that all individual functions are separated by a basic operating software, a hypervisor, in such a way that the function of all other components is guaranteed even during an update of individual components (AGL 2018). To this end, the hypervisor creates an environment that appears to each function running on it as a separate hardware platform. The functions are executed in isolation from the remaining system and can only react with functions in other hypervisor partitions via defined interfaces. A set of functions can thus be modularized and partitioned into safety and non-safety relevant sets, implying different requirements on the hypervisor partition's capabilities in terms of timing and resource guarantees.

Modularization gives each component a clear and consistent outer limit in terms of functionality and interfaces to other functions. Partitioning through a hypervisor further adds controllability and guarantees access to shared system resources such as memory or communication components. Also, in a system of modularized and partitioned functions, it is possible to predict the timing behavior of the overall system, even for different variants and system configurations before and after an update. Such tests are possible during design with a virtual prototype (virtual integration testing) and later before deployment in the existing system. At both points, it is possible to check whether the update can be integrated safely before it is implemented. Furthermore, partitions allow the independent update of functions residing in different partitions, even at run-time, which can become important for future systems.

In summary, automotive manufacturers and suppliers need to be able to let the car's hardware identify, download, and install updates on the fly and without user interaction, just as they do on a PC or cell phone. Because over-the-air uplink is limited in bandwidth, updates will be frequent in upcoming vehicle generations, modularized functions should be supported inside a partitioned HW/SW environment. Instead of all software having to be updated at once, individual software components and partitions can then be updated while others can remain as they are.

Depending on the manufacturer's requirements and the customer's acceptance, modularized and partitioned software components can keep running undisturbed while other components are updated. Therefore, neither the update process nor the updated software component itself must not pose a risk to the vehicle's safety at any time.

AN AUTOMOTIVE CONTINUOUS DEVELOPMENT/CONTINUOUS INTEGRATION LOOP.

Our design flow and software tool infrastructure that implements the above concepts of modularity and partition in

combination with time and resource contracts and run-time monitors is visualized in Figure 1 and consists of four stages.

① *Update Cockpit*

The starting point to developing an update of an automotive function is an *Update Cockpit*, which is based on the App4MC software (Höttger et al. 2017). The update cockpit allows the definition of the application as an updatable unit plus contract-based metadata for the system integration. These metadata contain the interface definitions, the specification of timing behavior, split up in assumptions and guarantees (timing contracts), as well as the specification of resource requirements, split up in assumptions and guarantees (resource contracts). Furthermore, the update cockpit allows users to represent and modify the overall system configuration, for example, vehicle software for a specific vehicle platform. Modifications such as removing, modifying, or adding of software components trigger design time virtual integration and compatibility checks to confirm the validity of the new configuration (see ②). If the virtual integration and compatibility checks fail at design time, the new configuration is rejected so that it is not deployable in the field, and diagnostic trace information is delivered and visualized through the updated cockpit.

② *Update Server*

The update server provides simulation-based or analytical integration testing for timing contracts and a compatibility check for resource contracts (virtual integration testing). We use the MULTIC (Design Approach for Multi-Layer Time Coherency in ADAS and Automated Driving) software (Damm et al. 2019) to quickly and virtually test for various hardware and software configurations, which might be out in the field and which, for practical reasons, cannot be all available as actual hardware instances at the manufacturer's site. If hardware components are available, this can be optionally supported by the hardware-in-the-loop (HiL) tests of the target HW platform or the entire vehicle's electronics, network, wiring, and software architecture, for example, E/E architecture. If the integration or compatibility checks fail, the developer or system integrator delivers diagnostic traces as feedback. If the integration and compatibility checks are passed, an update packed for the vehicles in the field is prepared and submitted *Over-The-Air* (OTA) through a secured transmission into the vehicle.

③ *Update Middleware*

The updated middleware allows the update server (see ②) to securely transfer a new update package over a wireless connection to one central gateway ECU. The update may concern the ECU itself, or it may be targeted to one other subcomponent (end device) in the car, not directly connected to the outside world but connected to the gateway ECU. After the update is complete, it is validated for data integrity. Then, the compatibility and resource checking is repeated in the field against the hardware specifications of the end device's hardware to be updated to avoid catastrophic configuration failures. This includes verifying the proper hardware and software configuration, the system's health status, including the presence and severity of hardware aging, the configuration of the software container, and the associated monitor updates. Updates and monitoring of end devices are optionally supported and fully controlled through the gateway. The updated middleware uses libvirt (Ashley 2019) as a common abstraction layer above different state-of-the-art hypervisors (Kivity et al. 2007) (Barham et al. 2003) and containerization software (Rad et al. 2017) and HW platforms.

To implement the update, there are different options: For an **offline update**, the entire system is brought to a safe state and paused, the function to be updated is removed, and a function with the new software version is set up, started, and connected to the system. Such an update is feasible if the vehicle is at rest or parked. For an **online update**, only the function to be updated is stopped, while the other system functions remain up and running. As soon as the updated function is set up, it is reconnected to the remaining system. Such an update is feasible for uncritical system components. For a **runtime update** (zero delay update), the function to be updated remains operational, while a second instance with the update is set up and connected to its inputs while its outputs are ignored at first. Such a component then runs in a *shadow mode*, where it runs together with the entire system, and its functional behavior can be compared to the old version, which is still in charge of producing the required outputs. Between two executions, the control is handed over to the updated function by an atomic instruction. Afterward, the old function can be removed. Such an update is, in principle, able to perform updates even while driving, depending on the user's acceptance.

Once the update is implemented to the end device of the ECU gateway itself and the updated function is running again, the gateway ECU starts the monitors derived from the resource and timing contracts (see ①) included in the update package from the update server. The monitors supervise the function's timing behavior and resource demand and constantly compare it against the designer's assumptions described in the contracts. This can be accompanied by functional monitors, supervising rare events occurring and rare functional behavior of the system.

Functional monitoring may either be enabled by adding self-supervision to the function (such as novelty detection for neural networks) or by applying learning-based techniques to the system's behavior. A learning-based functional monitor does observe relevant system parameters, such as the occurrence and timing of function calls. In an early learning phase (in the lab), it can be trained to predict the near future's parameters from the parameter's history. Such a system can be trained without human supervision just by observing the system acting in a typical environment. In the field, the monitor will constantly observe the system, predict the present behavior only by knowing the past behavior and then compare the prediction with the actual system behavior.

④ *Fleet Data Management and Analysis*

On the one hand, the timing and resource contracts were defined by the designer under certain assumptions concerning the operating domains where the system is designed to operate, for example, its operational design domain. An operational design domain occurs with a pattern of external events, reaction timing of other systems, and performance of the own hardware, which might, in practice, undergo some degradations due to aging effects. On the other hand, automotive functions, especially AI-based ones, can detect their own uncertainty or the novelty of an occurring situation. In all cases of a violation of the operational design domain assumptions or functional misbehavior as detected by monitors (see ③), the past stimuli entering the system and its own internal state (history) together with long-term data of the hardware health status can be collected, compressed and submitted to the fleet data management server. We currently support the integration of the Pandora Flexible Monitoring System (PandoraFMS 2022). For example, Pandora FMS can detect when a network system is unresponsive, an application is defaced, or a memory leak has occurred. Pandora also monitors hardware components and operating systems. It can generate reports and send notifications to the developer

team when problems occur. This way, the update cockpit is fed in information from the field to support the next update, reacting to the collected data by either redefining the operational design domain and updating specification of the timing and resource distribution through contracts or use collected stimuli to reimplement or retrain the function in order to address unexpected, rare events better.

CONCLUSION

We envision a seamless development and operation flow for safety-critical field devices, bringing together existing tools and bridging the gaps with our own proof-of-concept developments. Update server and update middleware were specified and developed in a series of industry-led research projects, existing as running hardware/software prototypes. The security of the update transmission and implementation was regarded together with safety. Nonetheless, there are still open issues coming from the very fact that the update process can initiate new software, potentially having unpredictable cross-influences with the existing software parts. The fleet data management and analysis are under intense research and development right now and are still in the conceptual phase. For the Update cockpit, all relevant information, tools, and methodology are already available, though it still must be integrated into a comfortable develop environment. ■

ACKNOWLEDGEMENTS

This work received support by the European Commission's Horizon 2020 program under the UP2DATE project (grant agreement 871465).

REFERENCES

- AGL 2018. "The AGL Software Defined Connected Car Architecture." *The Linux Foundation Automotive Grade Linux (AGL) Virtualization Expert Group (EG-VIRT)*
- Ashley, D. 2019. *Foundations of Libvirt Development*. Apress Berkeley, CA.
- Barham, P. 2003, "Xen and the art of virtualization." *ACM SIGOPS operating systems review* 37 (5): 164-177.
- Damm, W. 2019. "Multi-layer time coherency in the development of ADAS/AD systems: design approach and tooling." Paper presented at the Workshop on Design Automation for CPS and IoT (DESTION'19), Montreal, CA, April.
- Höttger, R. 2017. "APP4MC: Application platform project for multi- and many-core systems." *it – Information Technology*, 59 (5): 243-251.
- Kivity, A 2007, "kvm: the Linux virtual machine monitor." *Proceedings of the Linux symposium*, 1 (8): 225-230.
- PandoraFMS 2022. "Make smarter decisions with the data of your business." https://pandorafms.com/en/
- Rad, B.B. 2017, "An Introduction to Docker and Analysis of its Performance." *Intl J of Computer Science and Network Security*, 17 (3): 228-235.

ABOUT THE AUTHORS

Domenik Helms is the manager of the research group Deployments and Updates at the DLR Institute of Systems Engineering for Future Mobility. His background is in electronic design automation and embedded systems.

Patrick Uven is a researcher at the DLR Institute of Systems Engineering for Future Mobility. His background is in technical computer science and embedded systems.

Kim Grüttner is head of the department of System Evolution and Operation within the DLR Institute of Systems Engineering for Future Mobility. His background is in technical computer science and embedded systems.

DECEMBER 2022
VOLUME 25/ ISSUE 4

Getting a Grip on the Ever-Changing Software in Cyber-Physical Systems

Wytse Oortwijn, wytse.oortwijn@tno.nl; **Dennis Hendriks,** dennis.hendriks@tno.nl; **Arjan van der Meer,** arjan.van.der.meer-arzz@asml.com; and **Bas Huijbrechts,** bas.huijbrechts@tno.nl

ABSTRACT

As industrial cyber-physical systems grow ever more complex, their software grows naturally and changes continuously. In order to make risk-free changes to their software, it is crucial to understand how the system behaves, and how software changes have an impact on system behavior. We propose a generic two-fold approach to infer state machine models capturing system behavior, and to compare these models to determine and visualize the impact of software changes on system behavior, in a way to make them easily understandable for engineers. Our approach has been applied in the industry at ASML to help prevent software regression problems during critical software redesigns. In that, our approach has been shown to reduce risk and to be valuable.

INTRODUCTION

Industrial cyber-physical systems are growing more and more complex. Such systems typically consist of many mechanic and mechatronic components (the physical part) controlled by software (the cyber part), making software a crucial part of these systems. As cyber-physical systems grow ever more sophisticated, their software grows naturally and changes continuously with additional features, support for next-generation hardware, and bug fixing.

Managing the complexity of changing software is notoriously challenging. In order to change software without risk, it is crucial to (1) understand how the system behaves in every possible scenario with respect to the parts of the software to be changed and (2) understand the impact of the software changes on system behavior. This article addresses these two challenges.

Elaborating on (1), modern industrial cyber-physical systems may consist of millions of lines of specialized control code. It is impossible for any engineering team to comprehend all the possible ways such a software system might behave (Gulzar, Zhu, and Han 2019). Furthermore, often the original engineers of (parts of) the software have long since left the company, for example, due to retirement, and thereby have taken much knowledge with them on how the software works. How can software be changed without risk if there is no full understanding of how it works?

Despite these difficulties, engineers are tasked with maintaining and changing the software such that current behavior is unaffected, and no bugs are introduced or exposed. Elaborating on challenge (2), overseeing how software changes impact the system behavior is critical since a single overlooked software bug could break the system, which may have a significant economic and societal impact (Schuts, Hooman, and Vaandrager 2016). The typical lack of such an overview makes it particularly risky to change the software, which may lead to a "don't touch it" culture in which engineers become afraid to make changes. How can software correctly be changed without risk if there is no understanding of the full impact of their changes on system behavior?

Both these challenges should be addressed to enable cyber-physical systems' rapid evolution and prevent software complexity from hampering their reliability and growth.

We address these two challenges by researching methods and tools for getting a grip on the ever-changing software in cyber-physical systems. We developed a generic two-fold approach that (A) provides insight into how the system behaves, and (B), after changing the software, indicates what the impact is on the system behavior. More specifically, our approach can (A) soundly *infer* the operational behaviors of software-intensive systems and capture these in concise, human-understandable models; and (B) *compare* these behavior models and visualize any differences between them to be able to understand the impact of software changes on system behavior. Both these parts are fully automatic, come with mathematical guarantees, and do not require expert knowledge — they can be used by the same engineers that maintain the systems' software.

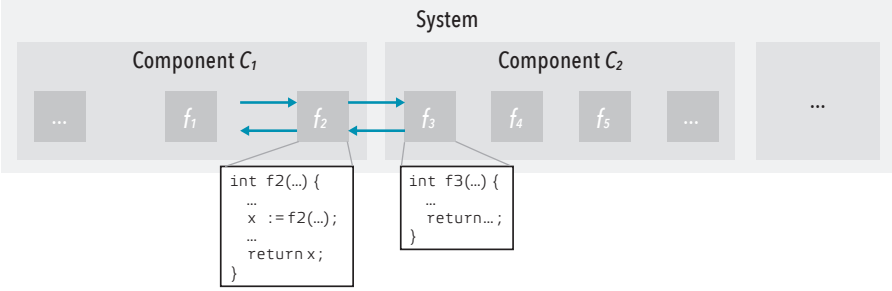


Figure 1. A software system composed of components, which in turn are composed of functions.

Our inference approach (A) infers concise multi-level behavioral models of large software-intensive systems based on observations in the form of execution traces of these systems. These traces are interpreted using domain knowledge, for example, knowledge of the software architecture and deployment, rather than state-of-the-art algorithms that typically use heuristics to ensure the models fit how engineers and architects look at the system. The inferred models can (B) be compared with each other to get insight into any changes in system behavior. Our comparison methodology provides a multi-level overview of behavioral differences, designed to guide users step by step through only the relevant differences by gradually zooming into more and more detail.

We applied our approach in the industry to help prevent software regression problems during critical software redesigns. In that, our approach has been shown to reduce risk and to be valuable. Potential regression problems were identified in hours instead of days during the implementation and validation phase and were fixed before delivery. Afterward, our approach increased the confidence of engineers that the redesigns were correctly performed and did not introduce other regression bugs.

The remainder of this article gives the necessary background on software-intensive systems and their behavior, presents our approach in more detail, and elaborates on how our approach was applied in the industry to help prevent software regression problems.

BACKGROUND

Structure of software-intensive systems

To manage the complexity of software-intensive systems, a common industrial practice is to use a *component-based architecture* (McIlroy, Buxton, Naur, and Randell 1968). Figure 1 illustrates such an architecture. The idea is that the overall software system is divided into components and further into sub-components in a divide-and-conquer fashion, ending up in software components (C_1, C_2, …) that are as independent as possible so that they each have their distinct responsibility and can be worked on by a single engineering team (Szyperski, Gruntz and Murer 2002, Vitharana 2003).

Components encapsulate functionality (for example, the functions f1, f2, … implemented in some programming language) and have interfaces by which components can communicate with each other, illustrated by colored lines between component functions. The interface of a component describes how other components are allowed to communicate with it by exposing functions that can be called externally. In the figure, C_1 communicates with C_2 by calling f3 inside the body of f2, meaning that the interface of C_2 exposes f3 to be called externally.

Software behavior

The advantage of a component-based architecture is that engineering teams do not need to be experts of the whole system, which is infeasible, but only of the single component that they work on. However, to correctly change a software component, it is still essential for engineers to understand how it (directly or indirectly) communicates with the other components in the system. Without such understanding, any changes made to a component may unintendedly affect the executions of other components, which may lead to (sometimes hidden) software bugs that ripple through the system via communication. Any such bugs could then ultimately break the system.

It is essential that engineers understand the *communication behavior* of the system (at least concerning the component under change), meaning the communications induced by all possible executions of the system. We refer to the communications induced by a single system execution as a *trace*, an ordered sequence of communication events (function calls, handles, returns, and more) in the order they happened in the system execution. Then the (communication) behavior of the whole software system is defined as the set of all possible system traces.

…
C_1. f2 ( ) handle
…
C_1. f3 ( ) call
…
C_2. f3 ( ) handle
…
C_2. f3 ( ) handle return
…
C_1. f3 ( ) call return
…
C_1. f2 ( ) handle return
…

Figure 2. An example system trace

Figure 2 shows an example trace that may be observed during system execution. At some point during execution, it may be that f2 is called on C_1 by another function (or another component). C_1 will handle this call when it is ready to do so, indicated by the C_1.f2( ) handle event, and execute this function. During this function execution, a call is made to f3 on C_2, indicated by C_1.f3( ) call. C_2 will handle this call (C_2.f3( ) handle), execute its body, and return the computed result (C_2.f3( ) handle return). The returned value is, in turn, received by C_1 (indicated by C_1.f3( ) call return), and eventually, f2 itself returns (C_1.f2( ) handle return), after which the system continues its execution.

Representing software behavior as state machines

Learning the system behavior by manually inspecting execution traces is generally infeasible. Execution traces are typically massive since much communication activity may occur while running the system—possibly many thousands of communications per second. Moreover, traces may contain significant duplication since component functions may be executed many times, especially when components perform repetitive tasks. Furthermore, components often execute concurrently and communicate asynchronously, meaning that communication events of different components may interleave. Consequently, multiple similar system executions may yield different traces, further complicating manual inspection.

Instead of manually considering such traces, we concisely and intuitively represent the communication behavior of individual functions as *state machines*. Figure 3 shows (simplified) example state machine representations of the

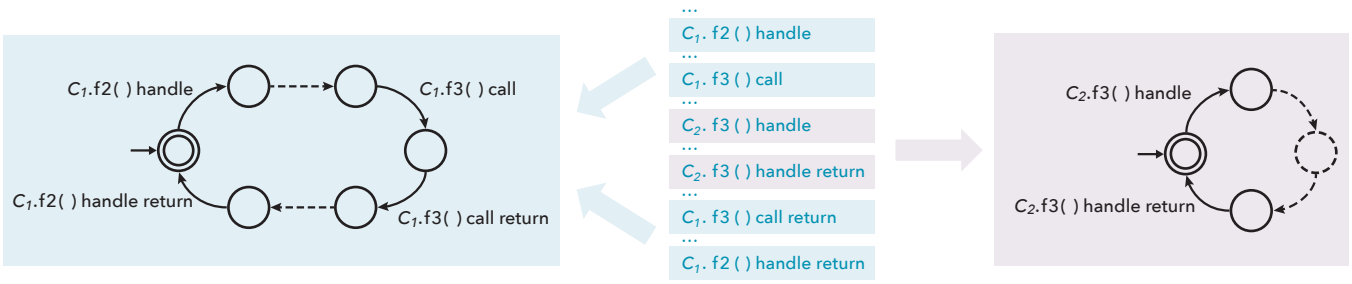


Figure 3. State machine representations of the communication behaviors of f2 *and* f3

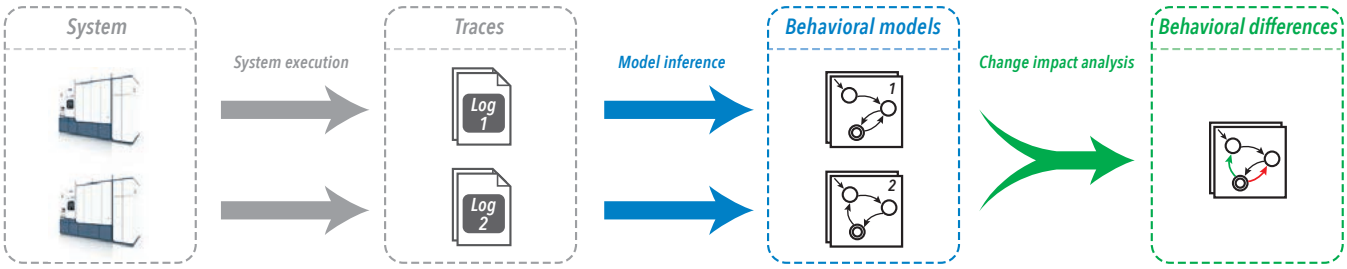


Figure 4. Our Constructive Model Inference and Change Impact Analysis approach

communication behavior of f2 and f3 with respect to the trace in Figure 2. State machines consist of *states* and *transitions* labeled with a communication event. Starting from the *initial state*, the state with an otherwise-unconnected incoming arrow, one could follow the transitions of a state machine to see how the corresponding function behaves, which is consistent with the communication events in the trace. Note that the state machine models for realistic software systems are significantly more complex than the ones shown in the figure. The Industrial Application section shows more realistic models.

Out of a single execution trace, we can construct concise state machine representations of the observed behaviors of all component functions. Moreover, we can *generalize* beyond this observed behavior (Hooimeijer, Geilen, Groote, Hendriks, and Schiffelers 2022), for example, by looping the state machines to indicate that the corresponding functions can be executed any number of times. This is illustrated in Figure 3, where the handle return transitions lead back to the initial states. Such generalizations lead to compact representations of behavior as they merge duplication and are justified by the component-oriented nature of the systems.

APPROACH

To correctly make changes to a software component without risk, it is crucial to (1) oversee how exactly their functions are ever used and (2) oversee how these changes impact system behavior. These two challenges are not immediately solved by investigating (state machine representations of) observed system behaviors as described above. For example, such state machines do not directly show the behavior of an entire component or (sub-)system or how different functions interact using communication.

Our automated approach alleviates these two challenges by (A) inferring concise multi-level state machine models of system behavior understandable by engineers and (B) comparing these models to understand the differences in the system behaviors.

Figure 4 gives a high-level overview of our approach. Our inference approach (displayed in blue) infers state machine models from system observations in the form of execution traces (displayed in grey). If multiple versions of the software system are available, for example, before and after having made a software change, then for both these situations, models can be inferred. If multiple such models are available, then our Change Impact Analysis approach (displayed in green) can compare these models and highlight any differences in the behaviors they represent.

Since our method operates on communication traces and state machines rather than source code, it is independent of implementation technologies like programming languages. This makes our approach generic and widely applicable to component-based software systems. Additionally, users only need elementary knowledge about state machines to interpret the results.

Constructive Model Inference

Our inference approach, named *Constructive Model Inference* (Hooimeijer, Geilen, Groote, Hendriks, and Schiffelers 2022), infers state machine models of the behavior of large software-intensive systems based on observations of these systems in the form of execution traces. These traces can be obtained by running the system and recording all observed calls to interface functions or by sniffing the network. Traces can be obtained from production runs and test scenarios to obtain good coverage of the system's normal and exceptional behavior. The traces are then interpreted using domain knowledge, for example, knowledge of the component-based software architecture rather than heuristics, to ensure the models fit how engineers and architects look at the system. Rather than inferring one single big monolithic model of the system behavior, we infer small models for all the individual functions of all the components for which behavior is observed. These smaller models then capture how the functions interact with each other. By inferring a small behavioral model for every function, the models become concise enough to be understood by engineers, making it manageable to understand and trust the behavior of (parts of) the system.

The state machine models of a component's functions together model the component's behavior. All these component models together model the behavior of the system. As a result, we can distinguish between three levels of models: the *system level*, the *component level*, and the *function level*. The models inferred by our approach are presented to the end user in these three levels. Starting from the system level, this multi-level presentation allows engineers to zoom gradually into the relevant parts of the system behavior, for example, for

changing a particular component or understanding how a legacy component interacts with the rest of the system.

Constructive Model Inference comes with mathematical guarantees that the inferred system models are correct representations of (generalizations of) the observed system behavior. An example of a multi-layered model is shown later in the Industrial Application section.

Change Impact Analysis

Once the behavior of a software component is understood, one can modify it. However, such insight alone does not yet solve the challenge of assessing the impact of software changes on system behavior, which is crucial for determining whether the software change is done correctly. Even using behavioral models of the before and after versions of the changed software obtained by our inference approach described above, it remains non-trivial to manually compare these models to determine whether the software change negatively affected the system behavior.

The second part of our approach, named *Change Impact Analysis* (Hendriks, van der Meer, and Oortwijn 2022), provides automation for performing such comparisons. More specifically, it automatically compares state machine models with each other and outputs an intuitive report of the differences between the behaviors they represent, to be assessed by engineers. Change Impact Analysis takes into account that models can be considered at the system, component, and function levels, allowing for the choice of level to analyze for behavioral differences. After automatically analyzing the chosen level of detail, our comparison methodology guides users step by step through relevant differences, allowing them to ignore irrelevant parts.

Change Impact Analysis highlights any behavioral differences between the input state machines and visualizes these on the state machines' structures to make them tangible for engineers. So, in addition to comparing the behaviors represented by state machines (using standard automata-theoretic notions of language equivalence and inclusion), our approach also compares their structures (Walkinshaw and Bogdanov 2013).

INDUSTRIAL APPLICATION

We have applied our approach in the industry at ASML to demonstrate its applicability to large industrial-scale cyber-physical systems. ASML designs and develops lithography machines which are essential for manufacturing computer chips. Our approach has shown to be able to provide insight into the behavior of such systems (through the lens of software) and any changes in behavior resulting from software changes in a way that is understood by engineers and architects, allowing them to find unintended regressions during software redesigns that are hard to find otherwise. Our approach has been shown to perform well and typically gives results within an order of minutes.

To illustrate our model inference and comparison approach, we highlight four case studies performed at ASML.

Case 1: Constructive Model Inference

For the first case study, we demonstrate that our approach can infer concise models from a large trace of system observations containing approximately 100 million communication events. This trace has been obtained by running the system in a testing environment and observing all communication. Our inference approach constructs concise models for all component functions for which behavior has been observed and presents these in a multi-layered manner.

We highlight two components, anonymously referred to as **A** and **B**, for which models have been inferred to illustrate how they look.

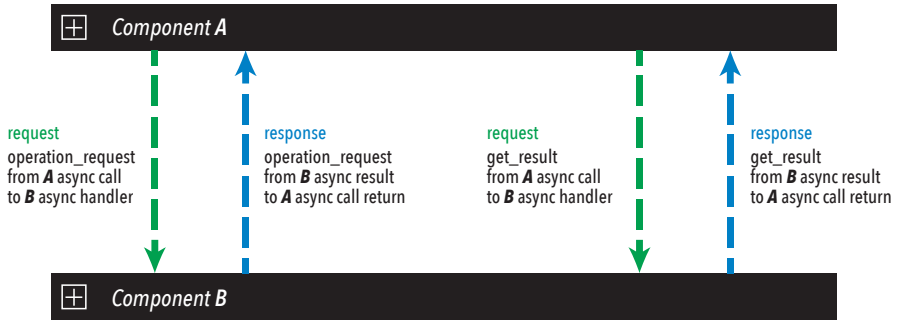


Figure 5. A system model showing two components that communicate by calling each other's functions

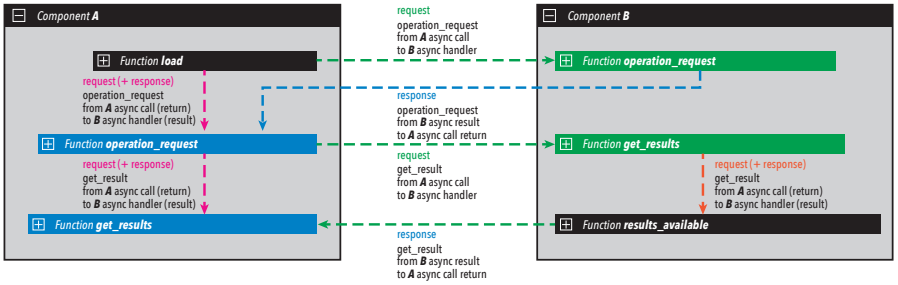


*Figure 6. The models of components **A** and **B**, showing the functions they provide and how communication flows between them*

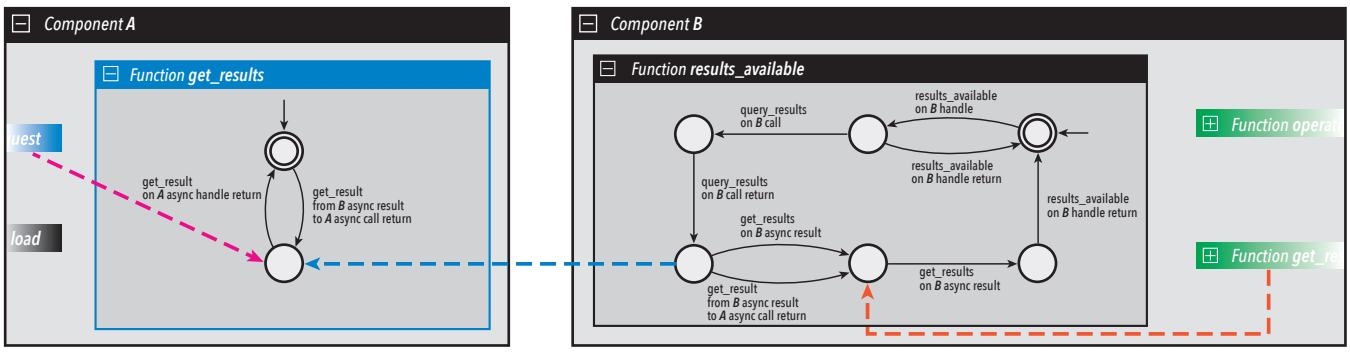


*Figure 7. The models of two functions of **A** and **B** showing their detailed communication behavior as state machines*

Figure 5 shows these models from the system-level perspective. This level provides high-level insights into the communications between these components, indicated by the colored transitions. Green transitions represent requests, while blue transitions represent responses to requests. In the figure, component **A** requests component **B** to perform some operation and later requests the result of that operation. Component **B** gives responses to both of these requests.

Figure 6 zooms into the component models and shows the functions of **A** and **B**. This component-level perspective provides insight into the flow of communication between the components. The colored transitions can be followed from top to bottom to see the control flow of how the different functions of **A** and **B** handle these communications.

Figure 7 zooms deeper into the detailed computation and communication behavior of two selected functions, represented as state machines. The behavior of these two functions can be understood by following the transitions of their state machines, starting from their initial states.

Even though the observed system behavior consists of millions of communication events, the inferred state machine models are manageable (typically in the order of tens to hundreds of states) and make it feasible to obtain insight into the complex system behavior. Domain experts confirmed that the models accurately describe system behavior significantly better than models previously inferred using heuristic-based methods that do not exploit domain knowledge.

Case 2: Identifying a known software regression

For the second case study, we look at a real-world software regression in which part of the code base was changed in a way the engineers believed was trivial. However, these code changes later unexpectedly turned out to cause a software defect, leading to a drop in system performance.

Our approach was applied retrospectively to demonstrate that it can find regressions early. We considered the software versions before and after the code change and inferred models for both these situations, again from execution traces containing about 100 million observed communication events each. The inferred models are then compared with each other using Change Impact Analysis.

Figure 8 shows one of the overviews from the comparison report, highlighting which component functions behave differently after the software change. The few highlighted functions can be inspected further, allowing all others to be ignored since their behaviors did not change.

Figure 9 shows the first function with behavioral differences, named ‘operation_queued’. These differences are indicated in the structure of the state machine, using red and green colors. States and transitions colored green indicate new behavior only present in the after situation, while red indicates old behavior that only occurred in the before situation. Black states and transitions indicate unchanged behavior that is present in both situations. In this case, the call to ‘determine_parameters’ is no longer performed by ‘operation_queued’ after the code change.

The second function with behavioral differences, ‘start_operation,’ is displayed in Figure 10. Upon inspection, the call to ‘determine_parameters’ now appears here. Domain experts identified and classified this as a delay to a later callback that causes a regression in system performance.

Function		Before vs. after defect
prepare_queuing	*(function 1 of 113)*	Unchanged
operation_queued	*(function 2 of 113)*	*Changed*
calibrate_level	*(function 3 of 113)*	Unchanged
check_memory_state	*(function 4 of 113)*	Unchanged
prepare_operation	*(function 5 of 113)*	Unchanged
start_operation	*(function 6 of 113)*	*Changed*
operation_finished	*(function 7 of 113)*	Unchanged
…	*(function … of 113)*	…

Figure 8. Comparison results on the level of served functions; all functions with behavioral differences are highlighted in orange

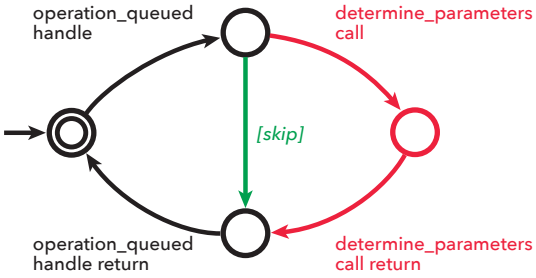


Figure 9. The first function with behavioral differences, ‘operation_queued’

With our approach, this regression was retrospectively diagnosed in hours instead of days, primarily due to the ability to highlight any behavioral changes quickly and accurately. An upfront application of our approach could have prevented the regression from being delivered.

Case 3: Technology migration

The third case study considers a technology migration involving a single component that used end-of-life technology. The migration involved replacing this old technology with new technology and was expected not to change the (observable) behavior of the component. Our approach was applied to increase confidence that the migration was indeed behavior-preserving. The expectation was that there would be no differences in behavior between the legacy and new software versions.

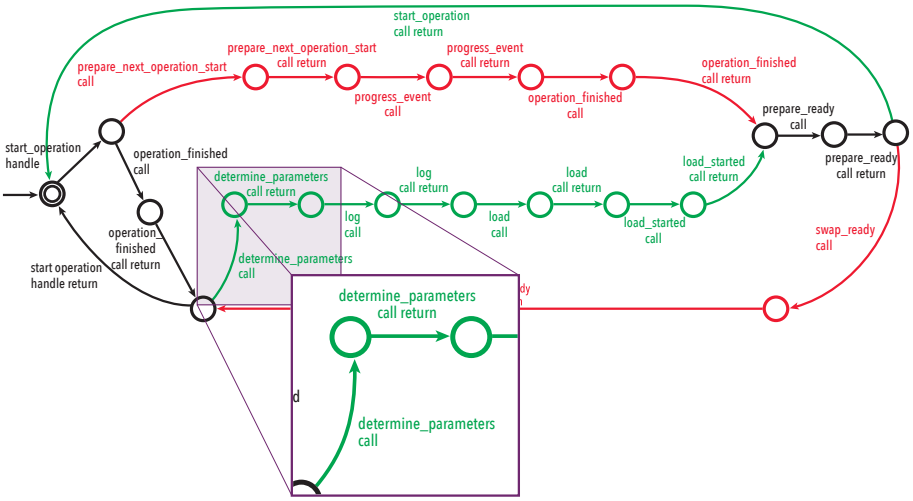


Figure 10. The second function with behavioral differences ‘start_operation’

Function		Legacy vs new
apply	*(function 1 of 11)*	*Changed*
finalize	*(function 2 of 11)*	*Changed*
get_status1	*(function 3 of 11)*	Unchanged
get_status2	*(function 4 of 11)*	Unchanged
initialize1	*(function 5 of 11)*	*Changed*
initialize2	*(function 6 of 11)*	Unchanged
model	*(function 7 of 11)*	*Changed*
prepare	*(function 8 of 11)*	*Changed*
set_context	*(function 9 of 11)*	Unchanged
terminate1	*(function 10 of 11)*	*Changed*
terminate2	*(function 11 of 11)*	Unchanged

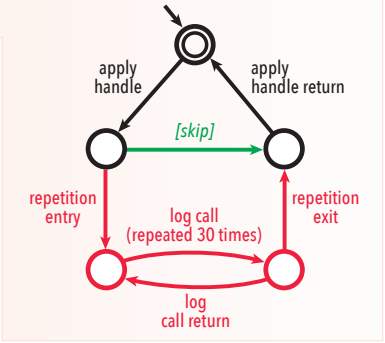


Figure 11. Left side: *Overview of functions with behavioral differences of case 3;* Right side: *behavioral differences of the 'apply' function.*

Figure 12. Top: *First unexpected behavioral difference found in case 4;*

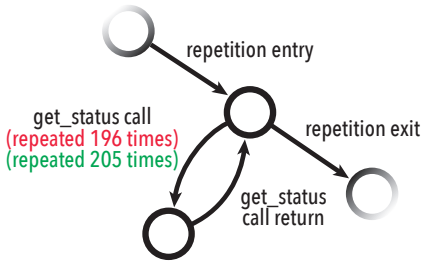
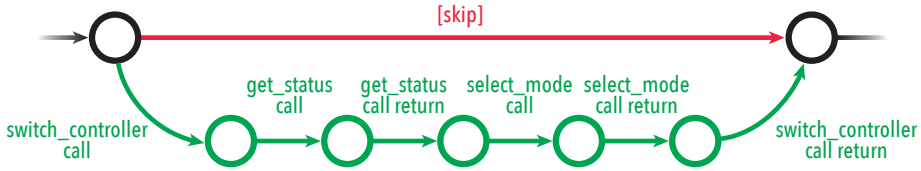


Figure 12. Bottom: *Second unexpected behavioral difference found in case 4* ↓



We inferred behavioral models for the before and after versions of the technology migration and compared the inferred models to search for any differences. Figure 11 (next page) shows the overview in the resulting comparison report with behavioral differences per function. This overview highlights six functions to have behavioral differences which is against expectations. Further inspection of these functions showed that most differences are related to logging. Figure 11 (right side) zooms into the first highlighted function, clearly showing that all logging functionality has disappeared after the migration since it is displayed in red. The migration engineers realized that the connection to the logger had not yet been established and could recognize these differences.

There were no other differences in behavior. The ability to inspect and classify any differences in behavior increased the confidence that the migration was done correctly.

Case 4: Critical redesign

For the fourth case study, we look at an application of our approach during a critical redesign. In this redesign, the software of a single component was refactored. This was risky since there was time pressure on the delivery, the code touched upon a lot of legacy of which knowledge was lost, and the correctness of the refactoring was crucial since the component is on the critical machine path.

Our approach was applied to find behavioral differences in the before and after versions of the redesign, increase confidence that the redesign was successful, and find any regressions early. While doing so, the engineers found two instances of unexpected behavior and fixed these before the software was delivered. Figure 12 (top) shows the first unexpected difference that was identified. While not technically related to the redesign, it shows that a particular function was repeatedly executed 196 times in the before situation and 205 times afterward. The engineers were unaware that their component exhibited such polling behavior and planned to optimize this. Our approach can also filter out such differences should they be considered irrelevant.

Figure 12 (bottom) shows the second instance of unexpected behavior, namely that a particular function that the refactoring should have disabled still appeared to be used. This behavior got classified as a refactoring bug and was fixed before delivery. This bug would have been hard to find otherwise, as it would cause a regression in system performance, which the available tests would not have found. The engineers identified the regression using our approach in less than an hour.

No other behavioral differences were classified as regressions. Our approach provided high confidence and confirmation to the engineers that the redesign was done as intended.

CONCLUSION

To correctly change a software component without risk, it is crucial first to understand how that component behaves and interacts with the system and then to understand the impact of software changes on this behavior. Our approach has been shown to address both challenges. It provides insight into the behavior of complex cyber-physical systems with a component-based architecture. Even though the system can be observed to behave in many ways, we exploit knowledge of the domain and architecture to automatically infer concise intuitive models for the functions of all components that are individually manageable. Then our multi-layered presentation of these models allows engineers to zoom into precisely the parts that are relevant to them. Moreover, Change Impact Analysis allows comparing the inferred models with each other to compare the behaviors of the functions, components, and systems they represent. Any differences in behavior are highlighted and visualized intuitively, allowing engineers to assess whether these changes are expected. Our comparison methodology guides users gradually through relevant differences.

As a result, our approach allows finding potential regression bugs early, before delivery, reducing the risk of introducing regressions and increasing confidence that software changes are done correctly. Our approach builds on principles and assumptions that broadly apply to software-

intensive systems. So, any engineer or architect working on such cyber-physical systems could potentially use our techniques to get a grip on their ever-changing software. ■

ACKNOWLEDGEMENTS

This research is carried out as part of the Transposition project under the responsibility of TNO-ESI in cooperation with ASML. The research activities are supported by the Netherlands Ministry of Economic Affairs and TKI-HTSM.

REFERENCES

- Gulzar, M., Y. Zhu, and X. Han. 2019. Perception and Practices of Differential Testing. *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice:* 71-80. Montreal, CA: IEEE.
- Hendriks, D., A. van der Meer, and W. Oortwijn. 2022. A Multi-level Methodology for Behavioral Comparison of Software-Intensive Systems. *International Conference on Formal Methods for Industrial Critical Systems*: 226-243. Warsaw, PL: Springer, Cham.
- Hooimeijer, B., M. Geilen, J. F., Groote, D. Hendriks, and R. Schiffelers. 2022. Constructive Model Inference: Model Learning for Component-Based Software Architectures. *17th International Conference on Software Technologies* (ICSOFT): 146-158. Lisbon, PT.
- McIlroy, M.D., J. Buxton, P. Naur, and B. Randell. 1968. Mass Produced Software Components. *1st International Conference on Software Engineering*: 88-98. Garmisch Pattenkirchen, DE.
- Schuts, M., J. Hooman, and F. Vaandrager. (2016). Refactoring of Legacy Software using Model Learning and Equivalence Checking: an Industrial Experience Report. *International Conference on Integrated Formal Methods*: 311-325. Springer.
- Szyperski, C., D. Gruntz, and S. Murer. 2002. *Component Software: Beyond Object-Oriented Programming.* Pearson Education.
- Vitharana, P. 2003. Risks and Challenges of Component-based Software Development. *Communications of the ACM 46(8)*, 67-72.
- Walkinshaw, N., and K. Bogdanov. 2013. Automated Comparison of State-Based Software Models in Terms of Their Language and Structure. *ACM Transactions on Software Engineering and Methodology*, 1-37.

ABOUT THE AUTHORS

Wytse Oortwijn is a Research Fellow at TNO-ESI, a Dutch applied research center. He received his PhD and MSc degrees in Computer Science from the University of Twente, at the Formal Methods and Tools research group, in 2015 and 2019, respectively. Moreover, he has worked as a postdoctoral researcher at ETH Zurich (2019-2020) at the Programming Methodology Group. In both these research groups he worked on formal methods and tooling to analyze designs and code implementations of software systems, to ensure their quality. His current work at TNO-ESI aims to make such formal methods and tooling ready to be used in industry.

Dennis Hendriks is a Senior Research Fellow at TNO-ESI, a Dutch applied research center. He also has a part-time position at the department of Software Science at the Radboud University in the Netherlands. He works with both industry and academia, bringing them together to address the complexity challenges of the high-tech industry. In his applied research, he makes academic formal methods ready for industrial use. His current work focusses on methodologies for model inference and change impact analysis of software behavior, and Synthesis-Based Engineering of supervisory controllers.

Arjan van der Meer is a Software Engineer at ASML, a Dutch technology company. He received his PhD and MSc degrees in Computer Science and Engineering from the Eindhoven University of Technology at the Model Driven Software Engineering group. He previously worked with TNO-ESI on implementing formal methods in tooling that can be used in industry. Currently he works at ASML on high-level machine control.

Bas Huijbrechts is a Senior Project Manager at TNO, a Dutch applied research center. He received his MSc degrees in Computer Science at the Eindhoven University of Technology. Having worked as system and solution architect in the telecom industry he was always seeking for the true sense of architecting complex software-centric systems from a technical conscience while striving for true quality. Currently he focuses on driving innovative research initiatives exploiting data-centric methodologies to provide emerging insights in cyber-physical and socio-technical eco systems within the high-tech equipment industry and the energy transition domain.

Merging Agile/DevSecOps into the US DoD Space Acquisition Environment — A Multiple Case Study

Michael Orosz, mdorosz@isi.edu; **Grant Spear**; **Brian Duffy,** bduffy@isi.edu; and **Craig Charlton,** ccharltn@isi.edu

ABSTRACT
Over the past five years, with funding from the US Air Force and the US Space Force, SERC researchers at the University of Southern California's Information Sciences Institute have undertaken a series of case studies that have focused on the introduction of agile and DevSecOps practices into a space-based software-only acquisition environment. These studies have identified best practices and revealed useful lessons learned. While the initial baseline DoDI 5000.02 project was entirely waterfall-based, subsequent projects have introduced agile/DevSecOps methods in progressively increasing levels, with the second project consisting of a roughly a 50/50 hybrid agile/waterfall mix and with the current project consisting of an approximately 70/30 hybrid agile/waterfall mix effort. All projects exhibit similar code complexity and size.

INTRODUCTION

During the past several decades, the commercial sector has gradually transitioned from a traditional waterfall approach to an agile system development process. In this context, the waterfall method refers to a serial approach of systems development where the requirements are first defined, followed by the system design, then system development, integration and testing, and finally deployment into the operational environment. In this approach, each effort or "step" starts once the previous step has been completed. The waterfall method assumes that all system requirements, including end-user needs, are known upfront prior to the start of development and remain relatively unchanged throughout the development process.

The problem with the waterfall approach is that for many systems, including many large enterprise systems such as those found in the US Department of Defense (DoD), system requirements either are not fully understood at the start of the project or they change (evolve) as the project is developed. The agile approach recognizes this challenge and takes a more evolutionary approach to systems development. In agile, frequent system releases are made available to the end-user community. Each release contains functionality that meets some subset of the end-user needs. Subsequent releases incrementally build on the previous release and incorporate new or updated systems requirements and end-user feedback. This approach allows for the project to reflect evolving systems requirements and end-user needs.

The continuous development, security, and operations (DevSecOps) lifecycle loop is closely linked to agile. DevSecOps focuses on integrating and testing as frequently as possible to detect and mitigate system discrepancies early in the loop before they cascade into a bow-wave of problems later in the program. In DevSecOps, testing of the complete system focuses on functional testing (for example, testing new functionality added to the system), regression testing (for example, testing older functionality to ensure it did not "break"), and security testing (identifying cyber or other security vulnerabilities). Ideally, integration and testing are undertaken continuously or near-continuously to detect discrepancies often.

The DoD has mandated that the DoD acquisition environment become more agile (US Congress 2018) to produce system releases more frequently (for example, 6-12 months). As reported in (GAO 2022), the DoD continues to meet challenges in implementing agile and DevSecOps and lags commercial standards. Over the past five years, with funding from the US Air Force and the US Space Force, SERC researchers at the University of Southern California's Information Science Institute (USC/ISI) have undertaken a series of case studies that have focused on the introduction of agile and DevSecOps practices into a space-based software-only acquisition environment. These studies have identified best practices and revealed valuable lessons learned. Figure 1 summarizes the research questions underlying the efforts of the USC/ISI team. Although the DoD is focused on hardware and software-based systems development and deployment, the USC/ISI team focuses

- Are there lessons/best practices that can be applied across all acquisition programs (space-based or non-space-based)?
- What tailoring practices are required when applying agile/DevSecOps to a DoD software acquisition program?
- Can we develop a template that suggests which approach to take when developing software (waterfall, agile or a hybrid of both)?

Figure 1. Research questions underlying the three projects

on software-based systems within the US Space Force domain.

As described in (Orosz et al. 2021), the USC/ISI team initially studied a baseline DoDI 5000.02 waterfall project (Project A), which serves as the baseline of comparison when studying the performance impacts which result from the introduction of varying implementation levels of agile and DevSecOps to the subsequent projects. The second project (Project B) undertaken by the USC/ISI team consisted of a roughly 50/50 hybrid agile/waterfall mix and was roughly the same size and exhibited similar system complexity to the baseline project. As reported (Orosz et al. 2021), the hybrid 50/50 waterfall and agile mix project (Project B) produced approximately 85.4% fewer open problem reports (PRs) than the traditional waterfall project (Project A). Further, an analysis of the performance of the waterfall portion of the 50/50 hybrid project as compared to the agile portion revealed that the agile effort produced approximately 95.7% fewer open problem reports than the waterfall portion of the effort.

Currently, the USC/ISI team is embedded in a 70/30 hybrid agile/waterfall project (Project C) that is estimated to be equivalent in size (software lines of code) and system complexity as the baseline and the 50/50 hybrid projects. Before the start of the 52-month Project C, there was an initial 15-month study to undertake technical discovery, establish an initial Scaled Agile Framework (SAFe®) **implementation, provide workforce training** and establish an initial DevSecOps design. No software was developed during this study. This article summarizes several lessons learned from all three projects – including the 15-month pre-project study.

APPROACH

For each case study, the project team embeds into the system acquisition environment, becoming part of the government's team. Along with participating in the project's day-to-day operations, the project team focuses on developing and refining performance measuring tools, collecting performance metrics, developing training/education modules, and providing subject matter expertise on agile, DevSecOps and systems engineering as related to the project domain.

LESSONS LEARNED AND RECOMMENDATIONS

As recently reported at the Naval Postgraduate School's 19th Annual Acquisition Research Symposium (Orosz et al. 2022, 405) and summarized in Figure 2, an initial set of lessons learned and recommendations were presented and discussed. The focus of this article is to expand on two of the NPS paper recommendations (highlighted in Figure 2) and discuss an additional recommendation based on new observations collected since the NPS paper was published. Specifically, this article focuses on the following three key recommendations when implementing agile practices in a DoD acquisition environment.

- Need for good upfront engineering
- Need to focus on MVP/MMP (Minimum Viable Product/Minimum Marketable Product) when planning/assigning features/stories to an upcoming PI
- Need for frequent Government engagement with the development contractor (need to attend scrums, ceremonies)

- Need for upfront engineering to avoid surprises later
- Access to performance tracking tools – may need to develop custom tools
- Program increment (PI) lengths may change as the project ramps up
- Assign tentative stories to sprints upfront during PI planning to help with resource management and story performance management
- Avoid assigning too many story points to a PI to avoid feature slips and manpower burnout
- Stay focused on the MVP/MMPs when planning
- Need upfront and continuous training
- Access to a near operations environnment may be necessary for projects where access to operations may be delayed

Figure 2. Recommendations from NPS paper; highlighted items are addressed in this article

NEED FOR SOUND UPFRONT ENGINEERING

As noted in the Agile Manifesto (Manifesto 2001), a key tenet of the agile development process is flexibility – the ability to react to changing conditions. This implies that the system design is constantly undergoing change, that developing a detailed design of the system upfront is impossible and that traditional DODI 5000.02 waterfall milestones such as preliminary design reviews (PDRs) and detailed design reviews (DDRs) are no longer relevant, and in many cases, costly as the design will have to be modified. In large part, this is true. However, there is still a need for good upfront systems engineering to ensure a mature project backlog is produced that has identified all required internal and external dependencies and priorities.

Many DoD projects exist within a larger enterprise of interconnected systems. For example, in the US space domain, many programs consist of three interconnected segments (Wikipedia 2022): space, ground (or control), and user (Figure 3). Within these segments are multiple sub-systems, all with interconnections and dependencies. These enterprise systems – and in many cases, even the individual segments within the enterprise – are often developed on separate timelines by different vendors. Understanding these interfaces (internal and external) and dependencies upfront will ensure that the necessary features (i.e., the work items) are defined, placed on the project backlog, assigned priorities, and allocated (initially) to a timeboxed program increment (PI) based on when they are needed within the enterprise. This understanding is necessary for decision-making when moving features, say due to a change in priorities, within the project backlog during system development. When there is a need to move features forward (to the "left" into an earlier PI or to the "right" into a later PI), the impact of that move needs to be considered in the context of the enterprise. This relationship between system interface definition and feature

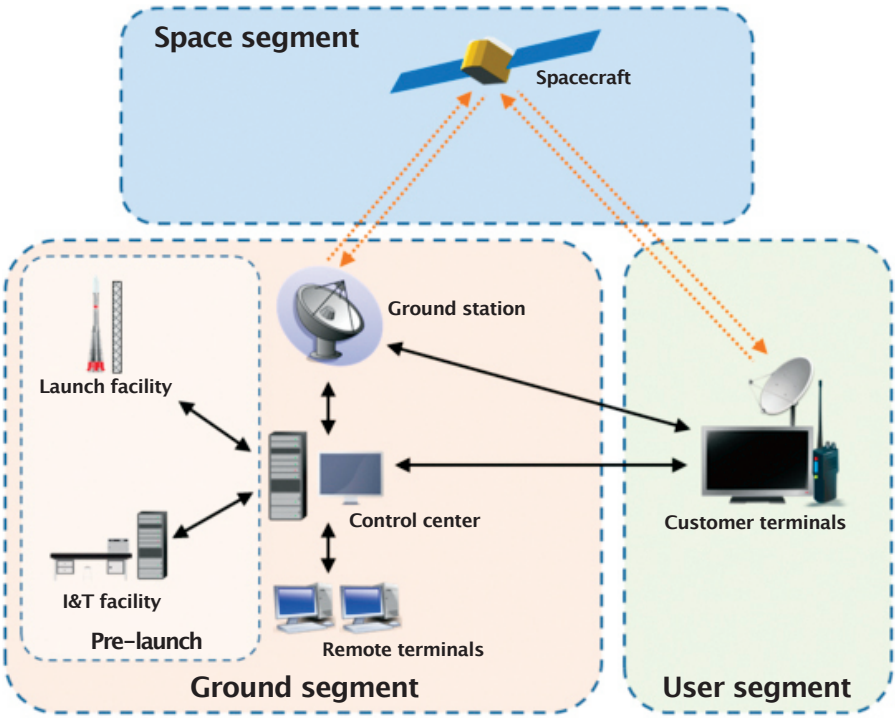


Figure 3. Three segments in the space domain; from (Wikipedia 2022)

implementation differs from the traditional waterfall approach. Identifying both strong and weak dependencies (internal and external) early in the project is essential for ensuring later program success. The difference, however, is that an agile approach means that important features can be made available before the entire system is fully implemented.

It is also important that the system requirements (often listed in the DoD's Capability Development Document (CDD)) be decomposed entirely into the necessary features for placement on the project backlog. This also involves ensuring that the acceptance criteria used by the Government to sign off on a feature are sufficiently defined so that all features linked to a requirement – when completed – fully satisfy the requirement. The intent is not to fully define each feature down to its detailed design, testing approach, acceptance criteria, and other details. Instead, the intent is to ensure sufficient placeholders are inserted into the project backlog with just enough detail to ensure that the requirements are fully decomposed, understood, and agreed to by the stakeholders. The actual details of the design, test procedures, and other supporting details will come later as these features are refined as part of the sprint and PI development/testing process cycle. The critical issue is to avoid missing key components of a requirement (for example, the need to avoid breaking existing system functionality by performing regression testing during the DevSecOps process).

The early and complete decomposition of requirements into features aids in preventing the late discovery of key functionality or capabilities and reduces the potential for any rework before release. In theory, such events should be captured when the product is released to the operators. However, in many cases, especially in the Department of Defense space domain, the end user may not have access to the product for years due to the delayed availability of all the enterprise's segments (which is why a Near Operations Environment–listed in Figure 2 and discussed in (Orosz et al. 2022, 405) is needed).

As for acceptance criteria (AC), there is often a difference between satisfying the acceptance criteria of stories decomposed from a feature and the satisfaction of the feature itself. Paraphrasing an often-used phrase – the sum of the parts does not necessarily equate to the whole, also known as system integration. In the agile environment, it is important to ensure that either the ACs for decomposed stories are adequately defined to satisfy the AC of the parent feature or define an AC for the feature itself that indicates that the feature can be closed when satisfied. ACs – at least the initial description – need to be part of the upfront engineering so that stories can be integrated into functional features. Refinement of the ACs can occur later during grooming and refinement of the feature.

Finally, systems engineering-based planning upfront can help with program management workforce/resource allocations. Having an initial project backlog with assigned priorities which indicates where a particular issue (feature) will be worked, also helps with staffing estimations. If a particular issue requires a unique skill set, knowing when that issue will be addressed can help influence hiring/workforce placement decisions.

FOCUS ON MVP/MMP WHEN PLANNING AND ASSIGNING FEATURES/ STORIES TO AN UPCOMING PI

In many projects, performance tracking is focused on velocities (the number of stories or story points completed) rather than on progress toward a minimum viable product (MVP) or minimum marketable product (MMP). For example, earned value management (EVM) and award fee structures often focus on production metrics such as story points completed during a given performance period. This focus is likely a legacy of performance-tracking systems used in a waterfall environment. A waterfall typically tracks the number of software lines of code (SLOC) written during a given performance period.

When the focus is strictly on velocity, or if project teams are not aware of the priority of completing upcoming MVP/MMP milestones, there is a tendency during PI planning for sprint/development teams to select features and their related stories that will maximize velocity in the upcoming PI. Optimizing for overall velocity may come at the expense of features needed in a near-term milestone. This is not suggesting that tracking program performance via velocity is incorrect; instead, the focus should be prioritized on stories and story points that accomplish the most value in completing an upcoming MVP/MMP event.

It is important during PI planning that planning objectives for the upcoming PI are well defined to ensure that the focus is on the upcoming MVP/MMPs and that the focus is maintained throughout the execution of the targeted PI (to account for feature movements between tentatively assigned PIs due to changing project priorities). For SAFe® implemented projects, it is the responsibility of the Agile Release Train (ART) team to set these objectives and monitor project teams' performance during PI execution.

FREQUENT GOVERNMENT ENGAGEMENT WITH THE AGILE AND DEVSECOPS PROCESS

There is a need for the Government team to attend as many scrums, planning events, grooming activities, ceremonies, and other activities as possible to understand the day-to-day cadence of the entire project. Decisions are made daily regarding story performance, blockage issues, changing

priorities, and other programmatic issues. Often these decisions can impact current and future PI performance. Early detection of issues can jumpstart mitigation steps that prevent such problems from cascading from PI to PI, forming a bow wave of issues in future PIs.

Frequent attendance of these events can also provide insight into higher levels of management, such as cost and schedule performance tracking. For example, in EVM cost/schedule performance tracking, the metrics may often show that the planned work was not accomplished. This can be explained within the context of adaptive priorities on the project. If a feature is blocked and will slip into the following PI, the project team will often advance another feature (from a future PI) to take advantage of available time and resources. This decision would not appear in EVM metrics but can be easily explained by Government observers.

Government involvement as the product customer is also necessary during the feature and story refinement for upcoming PIs. Often, these events are defined as a feature (and decomposed stories) and are worked as part of the current PI. In many cases, features cannot be approved for acceptance (be placed in the upcoming PI's backlog) without Government approval. For example, a feature's acceptance criteria may include a requirement that says that Government approval is needed for the feature to be approved and closed. To ensure that the acceptance criteria are met, the Government should participate in the refining process used to define the feature's AC.

ON-GOING RESEARCH

In addition to continuing to collect and report on lessons learned, the USC/ISI team is also focused on addressing a number of specific challenges in implementing agile/DevSecOps in the DoD acquisition environment. Specifically, the USC/ISI team is focused on the following three research questions:

1. Is there a need for a less rigid requirement specification (can some requirements be partially implemented)?
2. Is there a better approach to cost/schedule performance tracking?
3. Are there easily identified markers that signal whether a waterfall, agile or hybrid agile/waterfall approach should be used when starting a program?

Less rigid requirements specification: There can be little deviation from the defined specifications in the DoD acquisition environment. Requirements changes often require approval up several chains of command, which is a lengthy process and, in many cases, will result in schedule delays as the request for change (RFC) contracting process plays out. One solution that the USC/ISI team is exploring is the concept of dividing requirements into three buckets: those that are needed ASAP, those that are needed but can be delivered at any time during the life of the current project, and a final bucket composed of requirements that are also needed but could be slipped to a later project if needed. An award incentive program could be established for those requirements in the last bucket to motivate hitting as many of the last bucket requirements within the current project as possible. Modular contracting (Gerhardt and Headd 2019) is another possible approach.

Cost and schedule performance tracking: In the hybrid agile/waterfall projects observed, there is often a mismatch between the feature/story tracking system (for example, Jira (Atlassian 2022)) and the integrated master schedule (IMS), which drives earned value management (EVM) project cost and schedule performance calculations. Jira is updated daily to reflect project performance and events, including changing task end dates due to blockages and other program events. Often these changes are not reflected in the IMS until weeks later. Updating the Jira feature information in the IMS is often time-consuming, especially if it is not automated. This may result in the IMS having outdated information as it lags behind the up-to-date status shown within Jira. Unfortunately, this mismatch will often produce an inaccurate picture of project performance. The USC/ISI team is currently exploring options to address this challenge.

Selecting between waterfall, agile, or hybrid agile/waterfall: Although the USC/ISI team is focused on lessons learned in implementing agile in a DoD space-based acquisition program, the team recognizes that a fully implemented agile approach may not be the appropriate method for a given program (a maritime project to replace an obsolete vessel propeller with a new design whose requirements are well defined). A research question of interest to the team is whether there are easily identifiable program markers or characteristics that can suggest a preferred development method to use on a project (agile only, agile/water hybrid, or waterfall-only).

SUMMARY

Over the past five years, researchers at the University of Southern California's Information Sciences Institute have undertaken case studies focused on developing lessons learned and identifying best practices when introducing agile and DevSecOps into the space-based software-only acquisition environment. Based on the research to date, the USC/ISI project team recommends that when implementing agile and DevSecOps processes, it is crucial to focus on sound upfront engineering; to focus on the MVP/MMPs when planning/assigning features/stories for an upcoming PI; and to plan on frequent Government engagement with the development contractor (attend scrums, ceremonies) ■

REFERENCES

- Atlassian. 2022. Jira Software. https://www.atlassian.com/software/jira/guides/getting-started/overview.
- Gerhardt, L. and M. Headd. 2019. "Why We Love Modular Contracting," 18F, Government Services Administration. https://18f.gsa.gov/2019/04/09/why-we-love-modular-contracting/
- Orosz, M., J. Evans, B. Duffy, C. Charlton, and R. Mitchell. 2021. "WRT 1012: Global Positioning Systems–Mission Engineering and Integration of Emerging Technologies." SERC Report, Stevens Institute of Technology Systems Engineering Research Center. https://sercuarc.org/publication/?id=231&pub-type=Technical-Report&publication=WRT%201012:%20Global%20Positioning%20Systems%20-%20Mission%20Engineering%20and%20Integration%20of%20Emerging%20Technologies
- Orosz, M., G., Spear, B. Duffy, and C. Charlton. 2022. "Introducing Agile/DevSecOps into the Space Acquisition Environment," Proceedings, Naval Postgraduate School 19th Annual Acquisition Research Symposium Proceedings, Vol 1: 405–416, Naval Postgraduate School. https://dair.nps.edu/handle/123456789/4541.
- The Manifesto Authors. 2001. "Manifesto for Agile Software Development." https://agilemanifesto.org/.
- US Congress. 2018. "2018 National Defense Authorization Act (NDAA) (H.R. 2810)." https://www.congress.gov/bill/115th-congress/house-bill/2810.
- US Government Accounting Office (GAO). 2022. "Weapon Systems Annual assessment: Challenges to Fielding Capabilities Faster Persist. "https://www.gao.gov/products/gao-22-105230.
- Wikipedia, 2022. "Ground Segment." https://en.wikipedia.org/wiki/Ground_segment.

Systematic Identification and Analysis of Hazards for Automated Systems

Lina Putze, lina.putze@dlr.de; and **Eckard Böde,** eckard.boede@dlr.de

■ **ABSTRACT**
The introduction of automation into technical systems promises many benefits, including performance increase, improved resource economy, and fewer harmful accidents. In particular, in the automotive sector, automated driving is seen as one key element in Vision Zero by eliminating common accident causes such as driving under the influence, reckless behavior, or distracted drivers. However, this is contrasted by new failure modes and hazards from the latest technologies. In this article, we address the problems of finding common sources of criticality for specific application classes and identifying and quantitatively assessing new sources of harm within particular automated driving systems.
■ **KEYWORDS:** automated driving; hazard analysis; risk assessment; criticality; SOTIF; scenario identification; open context

INTRODUCTION – THE PROBLEMS OF IDENTIFYING RISKS FOR AUTOMATED DRIVING

Accidents due to speeding, distraction, or driving under the influence of alcohol – human misbehavior, intended or unintended, is an important factor in accident statistics. Self-driving vehicles are supposed to increase road safety by reducing the "human" risk factor. Although hazards associated with humans, like a collision due to a distracted driver, might be mitigated, the new technologies come with unknown risks and failure modes. The research topic, *Automation Risks*, focuses on identifying and assessing hazards and scenarios likely to trigger critical situations in the interaction of automated driving systems with their environment. In this article, we will focus on investigating automated driving systems since the methods presented have been developed in close collaboration with partners from the automotive industry. Nonetheless, we are actively adapting to other domains, like the maritime industry.

The safety of road vehicles is a well-known issue in the automotive industry. Due to the rising complexity of interacting safety-critical components, even conventional driving systems need to undergo a systematic safety process corresponding to ISO26262:2018 (ISO2018]). To keep development costs and efforts to a minimum, it is essential to include safety considerations from the beginning of the concept phase and throughout the entire development process because integrating changes in the system during early design phases is significantly easier. Knowledge about the common sources of criticality, for example, from accident databases, is an essential prerequisite for these first safety considerations. Moreover, a comprehensive safety concept requires a systematic identification and analysis of system-specific sources of harm. In the automotive domain, several methods exist for a so-called hazard and risk analysis (HARA), which is well-established in developing road vehicles.

Common hazard and risk analysis methods emphasize functional safety, which focuses on identifying and mitigating possible hazards caused by malfunctioning behavior of safety-related electrical and electronic systems. Assistance systems currently on the market, like adaptive cruise control, lane-keeping assistance, and combinations thereof, still require a human driver to monitor the vehicle and the environment and intervene when necessary. Nonetheless, many of those systems already take over parts of the driving tasks by providing braking, acceleration, and steering support while relying on sensor data that captures the internal and external environment. This comes with new potential sources of harm that take root in the system's specification. Let us consider an automatic emergency braking function (AEB). Despite the absence of faults and malfunctions, such hazards might occur due to incorrect interpretation of sensor input. For example, a poster on the roadside with a picture of a pedestrian crossing the road could be perceived as a natural person resulting in a breaking maneuver that could trigger a collision. This demonstrates that additional examination beyond the functional safety of the system is needed. We need to ensure that the system is robust concerning incorrect or unexpected sensor input, can comprehend situations correctly, and plans and acts responsibly based on these perceptions. These issues concerning the safety of the intended functionality (SOTIF) are addressed by ISO 21448:2022 (ISO 2022).

As assistance systems still have the driver as a redundant and immediately available fallback, such systems only

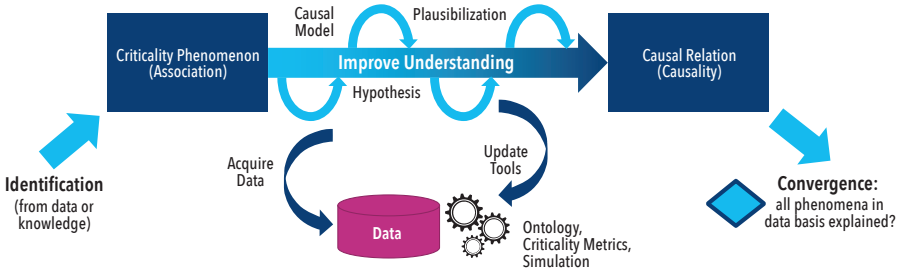


Figure 1. Basic concept of the criticality analysis

require evidence that the safety concept is fail-safe because the system does not provoke any additional risks, for example, by unintended interventions. In contrast to well-established systems, conditionally or highly automated driving functions like a traffic jam chauffeur temporarily release the driver from monitoring the environment for a certain time. This important step in the Levels of Driving Automation comes with additional safety difficulties as the abandonment of the driver as supervising instance involves the loss of a comprehensive and immediately available fallback. Therefore, it is necessary to prove that the system takes all the actions required to mitigate critical situations and that these actions are always carried out correctly and with the right timing: an operational-safe concept is required.

This is particularly problematic since automated driving systems driving on public roads face the challenge of operating safely in an open context. This arbitrarily complex, infinitely dimensional environment includes myriad factors that might lead to harm. Thus, it is infeasible to describe all relevant scenarios explicitly and specify the intended behavior. Moreover, hazards cannot be sufficiently reconstructed from existing real-world data. While there is extensive data for conventional driving systems, the challenges for automated driving systems differ from those for the human driver. For example, falling leaves in autumn are not generally a problem for the human eye, but if they hit the lens of a camera, object detection is not feasible anymore. Therefore, we cannot solely rely on data considering conventional systems and need extensive data that reflects the impact of automated systems on criticality.

To address these outlined issues, our research into *Automation Risks* is based on two main pillars: First, there is the criticality analysis which aims at finding common factors associated with criticality. Its focus is not on a specific system but on abstract application classes, such as the function of a highway chauffeur. Hence, the scope is in a pre-development phase where working groups comprising representatives from regulation authorities, standardization bodies, and industry define standard guidelines that every manufacturer of such a system must meet. In this setting, the criticality analysis will be a systematic approach to identify potential sources of criticality and specify a complete, well-defined set of criticality phenomena to be used as the basis for a homologation concept. Second, we work on a methodology that can be employed to perform a comprehensive hazard and risk analysis for specific highly automated systems that accompany the development process. This automation risks method aims to identify specific scenarios for further verification and validation and define safety goals as a basis for a fail-operational safety concept. The method intends to integrate functional safety (ISO 26262:2018 (ISO 2018)) and SOTIF (ISO 21448:2022 (ISO2022)) concerns.

STRUCTURING THE OPEN CONTEXT – CRITICALITY ANALYSIS

The first method we present is the criticality analysis. Its purpose is to investigate and structure the open context that constitutes the environment of automated vehicles. This includes not only the problem of identifying factors, parameters, and scenarios that have an essential impact on criticality but also abstracting these artifacts and mapping them on a finite set of criticality phenomena. This abstraction structures the criticality-inducing factors into comprehensive but manageable lists that can serve as a foundation for systematic verification and validation processes that enable a homologation for classes of automated systems. Furthermore, it helps to understand the underlying causalities to derive generic safety principles and mechanisms that avoid or mitigate the effects of critical situations.

Therefore, criticality analysis relies on a combined approach of expert-based and data-driven methods that precedes the design phase of specific systems. For example, it can be applied to urban traffic to set up a foundation for developing automated systems in this domain. In addition, it can support the operation and subsequent updates of corresponding systems in a DevOps process by continuously assessing changes in their domain. That might involve specific effects of amendments or enactments of laws and guidelines – a recent example would be the approval of e-scooters for German streets in 2019 – or even effects of climatic or societal changes. One of the fundamental principles of criticality analysis is that it does not only focus on the view of a single vehicle but also looks at the criticality of traffic. In this way, criticality analysis makes it possible to create generally accepted catalogs of criticality phenomena managed by regulation bodies and used by all manufacturers.

The basic approach of the criticality analysis is shown in Figure 1 and consists of three steps which we will present individually in the following.

1. *Identification and selection of criticality-triggering elements*: In the first step of the criticality analysis, candidates for criticality phenomena are selected for which a high correlation with a criticality increase is assumed. Expert knowledge, which is stored, for example, in the form of domain ontologies, test catalogs for vehicle approval, or accident databases, serves as a basis for the selection. Another source is data-driven approaches that systematically evaluate data from driving tests on test fields or in real traffic and data from specific computer simulations.
2. *Plausibilization and elaboration of interactions between criticality phenomena*: In the next step, the individual selected candidates for criticality phenomena are further analyzed. To make their influence on criticality, measurable criticality metrics are employed that quantify specific aspects of criticality. A typical example of such a metric is the time to collision (TTC), indicating the minimal time until a collision occurs, provided no action is taken. To achieve a comprehensive causal understanding of how the different phenomena affect certain aspects of criticality, we model the underlying causal assumptions based on causal theory by Judea Pearl (Pearl 2009). This theory allows the qualitative and quantitative investigation of causal queries based on constructing a so-called causal graph that represents the causal relationships of the different factors on a certain abstraction level. *Figure 2* illustrates such a causal graph for the criticality phenomenon stationary occlusion of traffic participants.
3. *Consolidation and abstraction of criticality phenomena/convergence*: The last step of the criticality analysis maps the identified and relevant criticality phenomena to a manageable and finite set of classes of criticality phenomena.

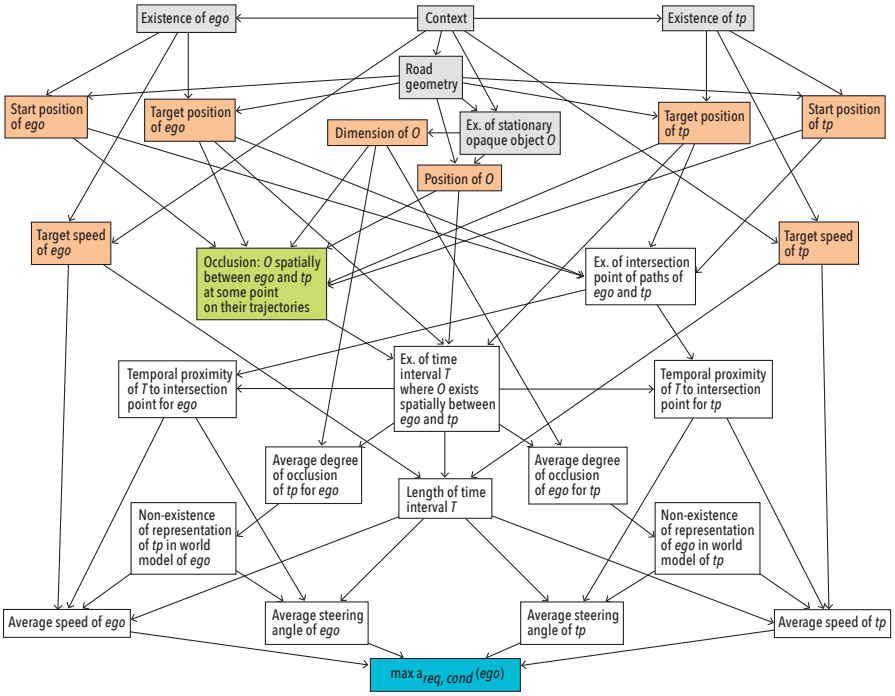


Figure 2. Causal graph for the criticality phenomenon "occlusion of a participant (tp)

This assumes that such a manageable set must exist and that the number of criticality phenomena cannot be unlimited. If this were the case, the amount of data relevant to safe driving would surely exceed the processing capacity of human drivers. However, since we know that humans can drive a vehicle safely even in very complex situations, we can assume that there is a compact representation of the criticality phenomena. The procedure for generating the criticality classes takes each new criticality phenomenon from step 2 and compares it to the already identified classes of criticality phenomena. If these are similar, they are merged into a standard class. Otherwise, a new class is created. The process is continued until it is determined with sufficient statistical certainty that all new phenomena found in step 1 are only ever mapped to already known classes.

During the execution of the method, individual parts, particularly in Step 2, are iterated repeatedly. This is done until the underlying mechanisms are sufficiently understood. A manageable finite set of abstracted criticality phenomena remains, covering all criticality-triggering causes for the investigated system class in a given environment. However, let us note that the method can be presented here only in a highly simplified form, and the figure notably omits details on where and how the feedback loops tie in with the process. For a comprehensive description of the methodology, please refer to Neurohr et al. 2021.

HAZARD AND RISK ANALYSIS FOR AUTOMATED SYSTEMS

The second method we elaborate on is the automation risks method (Kramer et al. 2020) which defines a comprehensive approach to the hazard and risk analysis of automated driving functions. It addresses both functional safety and SOTIF (safety of the intended functionality) by sustaining existing safety processes of the standards ISO 26262:2018 and ISO 21448:2022 and complementing them where necessary (ISO2018, ISO2022). The focus is on hazards that are inherent in the system but are triggered by external influences of the automated function, such as situations where the automated driving function does not react appropriately to its current environment. This includes non-detection or misclassification of objects, such as a bicyclist not detected or misclassified as a pedestrian, erroneous recognition of non-existing objects, and wrong predictions of future events, for example, due to wrong dynamic models. Therefore, the method builds on established analytical techniques for hazard analysis and risk assessment while it adds significant enhancements to enable the applicability to automated systems.

The proposed method is designed to accompany the entire development process. It is beneficial to initiate its application early during the concept phase so that safety considerations can be integrated into the system as early as possible. As shown in Figure 3, the method contains several feedback loops between the concept phase and development that enable the consideration of adjustments in the system, especially the integration and analysis of risk mitigation measures based on the previously gained knowledge, such as the implementation of redundancies or the definition of a higher safety distance.

The approach involves two main parts: the identification of hazardous scenarios (Steps (1) – (5) in Figure 3. Overview of the automation risks method) and the quantification of corresponding risks (steps (6) and (7)).

The first part aims to identify hazards, understand the underlying causal relationships, and deduce scenarios that might

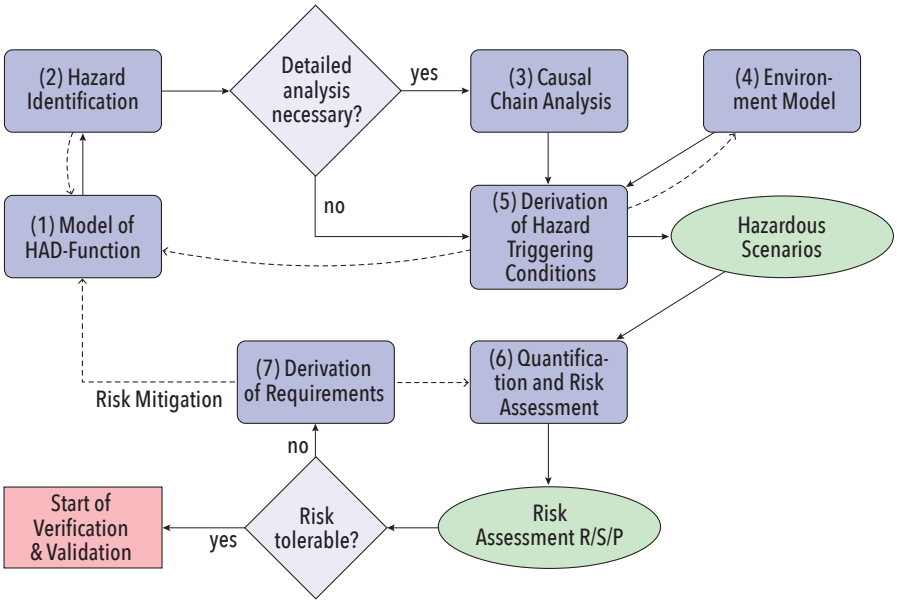


Figure 3. Overview of the automation risks method (Kramer et al. 2019)

ID	Basic Scenario	Basic Maneuver	Correct if (context)	Keyword	Incorrect Vehicle Behavior	Observable Effect(s) in Scenario	Additional Scenario Conditions (necessary for Top Level Event)	Potential Top Level Event
1	slower turn into path challenger	decelerate/ braking	front distance < safety distance	no	necessary breaking maneuver not initialed	ego continues with constant speed	challenger with significantly lower speed or critical Time-To-Collision	front/side collision with challenger
2				less	breaking maneuver not strong enough	Ego does not decelerate to prevent collision	challenger with significantly lower speed or critical Time-To-Collision	front/side collision with challenger

Figure 4. Table for identification of hazards on vehicle level (Kramer et al. 2019)

Functional Unit	Function			Key-word	Local Failure/ Functional Insufficiency	Basic Scenario	System Effect(s) in Scenario	Incorrect Vehicle Behavior	ID(s) of IVB	Possible System Cause(s)	Environ-mental Condition	Relevant for human driver?
	Input	Compu-tation	Output									
Sensors > Front camera > object recognition	camera image	segmen-tation	seg-mented camera image	no	segmented camera image not generated	slower turn into path challenger	challenger not detected by front camera > maneuver planning without information about the challenger	necessary braking maneuver not initiated	1	HW-failure, degradation or design fault	none	no statement
					no segments in camera image recognized	s/a	s/a	s/a	s/a	no night vision lacking sensibility at dark	darkness	likely (human vision also impaired by darkness)

Figure 5. Table for identification of hazards on component level (Kramer et al. 2019)

trigger hazardous events. These hazardous scenarios serve as inputs to the following quantification part. They can also serve as a basis for comprehensive scenario-based testing within the verification and validation process and define a starting point for improvements in the system.

The investigation is based on an initial system description that involves at least an item definition and a functional architecture that describes an architectural model representing system functions, like sensor fusion or trajectory planning and their interactions. To identify hazards caused by incorrect behavior of the automated function, we employ a keyword-based brainstorming approach inspired by the hazard and operability study (HAZOP) (Ericson 2005, 365-381), a technique originated from the chemical industry. The main idea is to combine a set of basic scenarios with a set of basic maneuvers that the automated function could perform with a list of keywords to derive possible incorrect behavior of the automated system that might lead to harm. An example of such a table applied to a highway-chauffeur function is provided in Figure 4.

In the next step, we employ a second HAZOP-inspired approach to examine local failures and functional insufficiencies and their effects on the system and vehicle level by applying keywords to the individual functional units.

Based on the identified hazards, we aim to derive scenario properties that might provoke them. Therefore, we use a modified fault tree analysis (Ericson 2005, 183-222) which analyzes the causal chains starting from the top-level event of a hazard during a basic scenario.

A unique feature is that we denote environmental conditions in the tree wherever necessary for the propagation of a fault. We can derive the triggering scenario properties by reducing the fault tree to these environmental conditions and identifying so-called minimal cut sets. An exemplary dependency graph is shown in Figure 6.

The quantification aims to derive a risk assessment that can be used to determine safety goals based on the afore-identified scenario properties. Therefore, it mainly

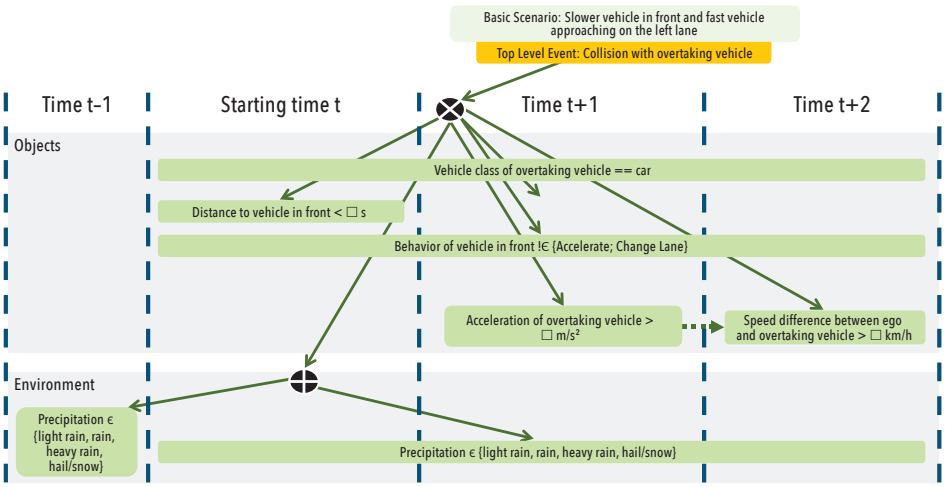


Figure 6. Exemplary part of an environmental fault tree reduced to the environmental conditions and chronologically ordered into discrete time steps

builds on probability estimation. Relying on the probabilities of occurrence of the single environmental conditions and the conditional probabilities that an error propagates in the fault tree, we estimate the probability of a hazard occurring with the help of the single minimal cut sets representing the triggering scenario properties. This serves as a basis for the risk assessment according to the automotive safety integrity level (ASIL) of the ISO 26262:2018 (ISO 2018).

SUMMARY AND OUTLOOK

In this paper, we presented two methods that enable systematic investigation of criticality causes and their effects in the context of automated systems.

Criticality analysis aims at identifying a comprehensive list of all potential sources of criticality in a given application field which serves as input for certification authorities and test organizations to develop detailed homologation guidelines. The method is being developed in the VVMethoden project in close cooperation with representatives from the automotive industry.

The second approach describes an extension of a hazard and risk analysis in which functional safety is combined with SOTIF (safety of the intended functionality). This approach was developed in the PEGASUS project, where it was extensively tested using the example of a highway chauffeur function. A comprehensive description of the approach and the evaluation can be found in (Böde et al. 2019). Furthermore, we have investigated to what extent the approach can be adopted in other application domains. Vander Maelen describes the application of this method to a collision warning system in the maritime domain (Vander Maelen et al. 2019).

Currently, we are working on elaborating the methods, simplifying their application, and investigating other use cases. In two internal projects, we are investigating the suitability of these approaches for hazard detection in automated road traffic (https://verkehrsforschung.dlr.de/de/projekte/kokovi) and for automated ship navigation in port areas (https://verkehrsforschung.dlr.de/de/projekte/das-projekt-futureports-fuer-hochautomatisierte-digitalisierte-und-intermodal-vernetzte). ■

ACKNOWLEDGMENTS

This paper reports on collaborative work by many colleagues from DLR and partners in the projects PEGASUS (https://www.pegasusprojekt.de/en/home) and VVMethoden (https://www.vvm-projekt.de/en/). These projects received funding from the German Federal Ministry for Economic Affairs and Climate Action.

In particular, we want to thank our DLR (formerly OFFIS) colleagues Matthias Büker, Birte Neurohr, Christian Neurohr, Sebastian Vander Maelen, and Lukas Westhofen, the scientific leaders Werner Damm and Martin Fränzle, as well as our industrial cooperation partners Martin Butz (Bosch), Martin Bollmann (ZF), Ulrich Eberle (Stellantis) and Roland Galbas (Bosch) for their substantial contributions.

REFERENCES

- Böde, E., M. Büker, W. Damm, M. Fränzle, B. Kramer, C. Neurohr, and S. Vander Maelen. 2019. "Identifikation und Quantifizierung von Automationsrisiken für hochautomatisierte Fahrfunktionen." Technical report, OFFIS e.V.
- Ericson, C. A. 2005. *Hazard Analysis Techniques for System Safety*. John Wiley & Sons, Inc.
- Kramer, B., C. Neurohr, M. Büker, E. Böde, M. Fränzle, and W. Damm. 2020. "Identification and quantification of hazardous scenarios for automated driving." *International Symposium on Model-Based Safety and Assessment*: 163–178.
- Neurohr, C., L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle, and R. Galbas. 2021. "Criticality analysis for the verification and validation of automated vehicles." *IEEE Access 9*: 18016-18041. doi:10.1109/ACCESS. 2021.3053159.
- Pearl, J. 2009. *Causality* (2nd ed.). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511803161
- Vander Maelen, S., M. Büker, B. Kramer, E. Böde, S. Gerwinn, G. Hake, and A. Hahn. 2019. "An Approach for Safety Assessment of Highly Automated Systems Applied to a Maritime Traffic Alert and Collision Avoidance System." *2019 4th International Conference on System Reliability and Safety (ICSRS)*: 494-503, doi:10.1109/ICSRS48664.2019.8987712.
- ISO (International Organization for Standardization). 2022. ISO 21448:2022. Road vehicles — Safety of the intended functionality. Geneva, CH: ISO.
- ———. 2018. ISO 26262:2018. Road vehicles – Functional safety. Geneva, CH: ISO.
- SAE (Society of Automotive Engineers). 2021. SAE J3016:2021. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Geneva, CH: SAE/ISO.

ABBREVIATIONS OF STANDARDS, CORRELATED WITH REFERENCE-LIST CITATIONS

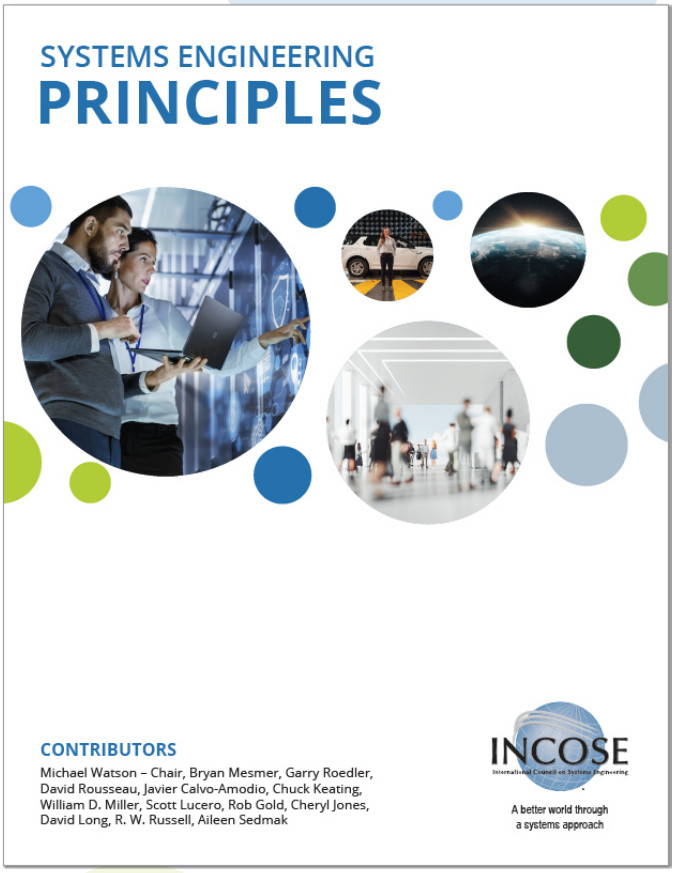
ISO 21448:2022	(ISO 2022)
ISO 26262:2018	(ISO 2018)
SAE J3016:2021	(SAE 2021)

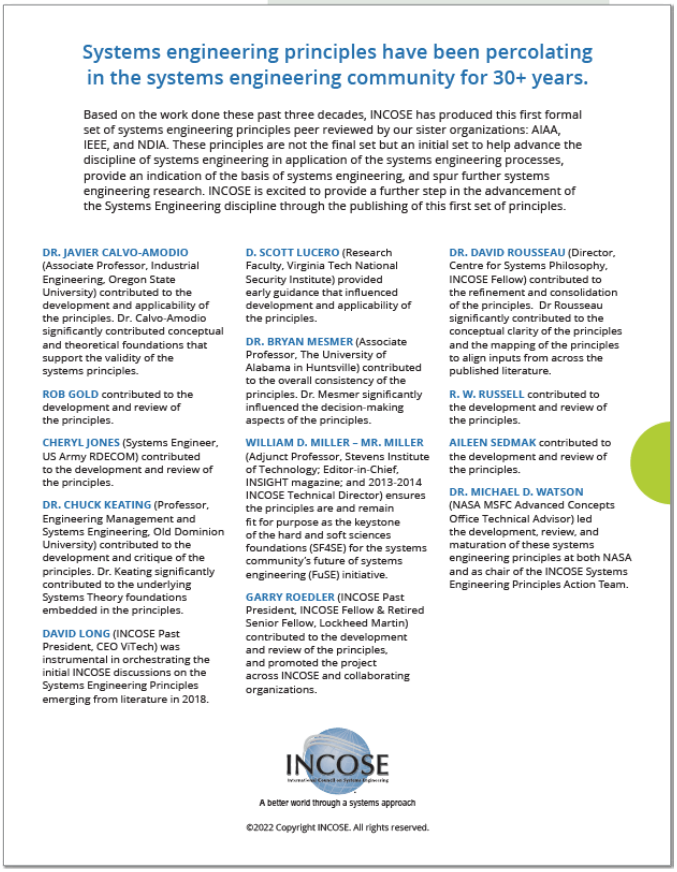
ABOUT THE AUTHORS

Lina Putze is a researcher at the DLR Institute on Systems Engineering for Future Mobility. Her background is in mathematics.

Eckard Böde is a manager of the research group System Concepts and Design Methods at the DLR Institute on Systems Engineering for Future Mobility. His background is in computer science and model-based safety assessment of cyber-physical systems.

How to make complex systems a little less complex.



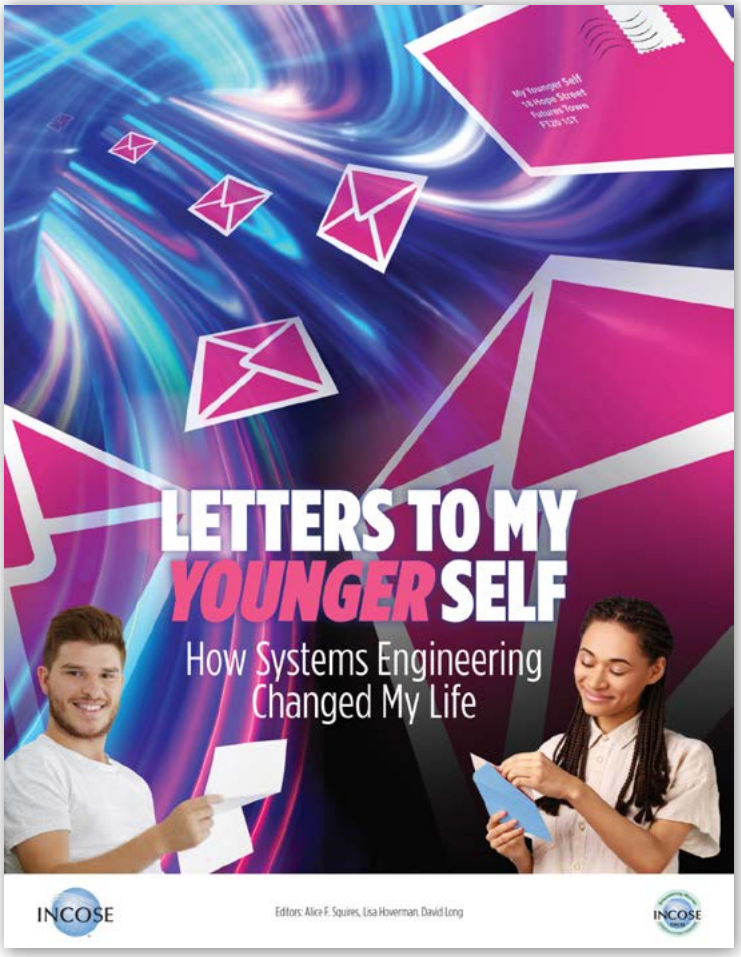


incose.org/seprinciples

INCOSE *A better world through a systems approach / www.incose.org*

Letters to My Younger Self:

How Systems Engineering Changed My Life



Download your copy at incose.org/ltmys

Sponsored by SPEC INNOVATIONS

INCOSE *A better world through a systems approach / www.incose.org*

EMEA WSEC 2023

Europe, Middle East, Africa

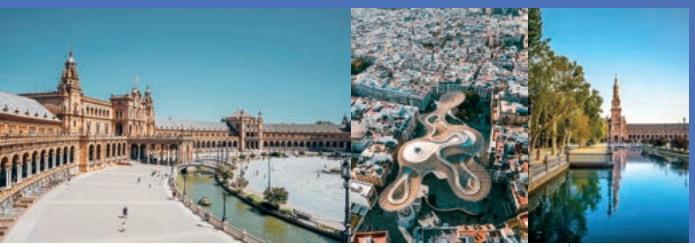
HYBRID EVENT

Workshop and Conference
Sevilla, Spain - 24-26 April, 2023

What is the EMEA WSEC ?

The INCOSE EMEA WSEC 2023 is an event for Systems Engineers from the EMEA region to contribute to the state of the art in Systems Engineering. In 2023, it will be a strategic and a new model event having the conference and the workshop together in one event. We will have full papers, panels, paperless presentations, workshops, and tutorial presentations. Attendees will spend three days working alongside fellow Systems Engineers who are there to make a difference.

This INCOSE EMEA WSEC 2023 on Systems Engineering is organized by the Chapters of the EMEA Sector of INCOSE and hosted by AEIS (The Spanish Chapter of INCOSE).



The Venue
Barceló Sevilla Renacimiento
Av. Álvaro Alonso Barba
41092 Sevilla - Spain

Dates
Monday 24 –
Wednesday 26 April, 2023

Call for content – Deadline extension

Full Papers	**January 10, 2023**
Paperless Presentations	**January 10, 2023**
Acceptance Notification (Paper/Panel/Tutorial/Paperless Presentation)	**End of January, 2023**
Acceptance Notification (Workshop)	**End of March, 2023**

Sponsorship Opportunities

You want to put a spotlight on your company? Our sponsorship opportunities are available on incose.org/emeawsec2023/sponsorship

Future events 2023

JAN 28-31
International Workshop 2023
Torrance, CA, USA
www.incose.org/iw2023

MAR 16-17
Conference on systems engineering research (CSER2023)
Hoboken, NJ, USA
cser.info/cser2023

APR 24-26
EMEA SEC and EMEA Workshop 2023
Sevilla, Spain
www.incose.org/emeawsec2023

MAY 15-19
NAFEMS World Congress NWC23
Tampa, FL, USA

JUL 15-20
INCOSE 33rd Annual International Symposium 2023
Honolulu, HI, USA
www.incose.org/symp2023

OCT 11
AOSEC 2023

33rd Annual INCOSE international symposium

hybrid event

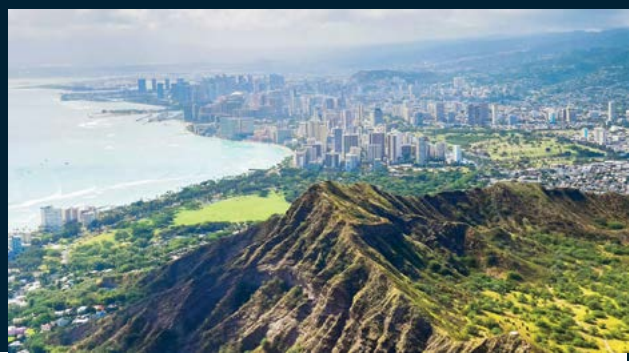
Honolulu HI USA

The Venue

Hawaii Convention Center
1801 Kalākaua Ave
Honolulu, HI 96815 - USA

Dates

Saturday 15 July, 2023
Thursday 20 July, 2023



- From now: Make your hotel reservation
- December 2022: Sponsorship registration open
- March 2023: Registration fees available
- April 2023: Registration open Final program on-line
- June 2023: Virtual platform open
- July 2023: Event (15 - 20 July 2023)

Sponsorship Opportunities

Why become a sponsor?

VISIBILITY

Unique brand of recognition and visibility for your organization

PRACTICE

Access to the latest thinking relevant to the practice of Systems Engineering

SPOTLIGHT

Put a spotlight on your organization's competency in Systems Engineering

ASSOCIATION

Be associated with the highest culture of professionalism and innovation

SUPPORT

Demonstrate organizational support to INCOSE's mission

CONNECTIONS

Put a spotlight on your organization's competency in Systems Engineering

INCOSE's Impact

- 18000+ Members
- 65+ Chapters
- 77+ Countries
- 120+ Corporate Advisory Board Members

Social media

1,950 members	3,750 members	470 members	21,500 members	3,220 members

https://www.incose.org/symp2023