INSIGHT

This Issue's Feature:

AFIS Doctoral Symposium: Advancing Systems Analysis and Modeling in French Universities

Design Architecture ROBAFIS AFIS Process
V&V Universities Activities
Research Innovation System DSMLs
Interoperability HMI
System-of-Systems Education
Design Assessment
Industries Forum
Situation Engineering
Requirements Modelling MBSE INCOSE
RobAFIS
Competences Formalizing Specification Operations
Railway
Pedagogy French Manufacturing INSIGHT Risk Workshops

DECEMBER 2015
VOLUME 18 / ISSUE 4

INCOSE

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

DESIGNING THE NEXT FRONTIER

CYBER PHYSICAL SYSTEMS

INTEGRATING THE PHYSICAL, HUMAN AND COMPUTATIONAL WORLD

Cyber Physical Systems are the future. At Stevens, the uniquely designed systems and software engineering division offers academic and research programs that are creating progressive intelligence to lead the new era in systems design – building systems that bridge the cyber-world of computing and communications, with the physical world of humanity and the internet of things.

Learn more at stevens.edu/systems

1870

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

DECEMBER 2015 VOLUME 18 / ISSUE 4

What's inside this issue

About This Publication

INFORMATION ABOUT INCOSE

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

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

OVERVIEW

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

EDITORIAL BOARD AND STAFF

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

Assistant Editor — Lisa Hoverman
lisa.hoverman@incose.org

Theme Editors
Hervé Panetto — herve.panetto@univ-lorraine.fr
Frédérique Mayer — frederique.mayer@univ-lorraine.fr
Eric Bonjour — eric.bonjour@univ-lorraine.fr

Senior Advertising Account Manager — Roland Espinosa
respinosa@wiley.org — 201-748-6919

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

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

2015 INCOSE BOARD OF DIRECTORS

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

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

PERMISSIONS

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

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

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

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

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

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

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

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

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

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

ARTICLE SUBMISSION

INSIGHT@incose.org

Publication Schedule. ***INSIGHT*** is published four times per year. Issue and article submission deadlines are as follows:
- April 2016 issue – 5 February
- July 2016 issue – 5 June
- October 2016 issue – 5 August
- December 2016 issue – 5 October

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

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

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

WPI

SYSTEMS THINKING
ONLINE GRADUATE CERTIFICATE

FOR PROGRAM DETAILS VISIT US ONLINE

cpe.wpi.edu/+thinking

CORPORATE AND PROFESSIONAL EDUCATION

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

ADVERTISE

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

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

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

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

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

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

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

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

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

CONTACT

Questions or comments concerning:

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

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

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

ADVERTISER INDEX December volume 18-4

CORPORATE ADVISORY BOARD — MEMBER COMPANIES

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

FROM THE EDITOR-IN-CHIEF

William Miller, insight@incose.org

This December in the 25th year of INCOSE marks the fourth issue of *INSIGHT* in cooperation with John Wiley & Sons publisher as a magazine for systems engineering practitioners. *INSIGHT*'s mission is to provide informative articles on advancing the state of the practice of systems engineering. The intent is to accelerate the dissemination of knowledge to close the gap between the state of practice and the state of the art as captured in *Systems Engineering*, the Journal of INCOSE, also published by Wiley.

The focus of the December issue of *INSIGHT* is the French Chapter of INCOSE, Association Française d'Ingénierie Système (AFIS) Doctoral Symposium: Advancing Systems Analysis and Modeling in French Universities. The theme papers in the December issue promote research contributions for an interdisciplinary and collaborative engineering, based on models. Articles from theme editors Hervé Panetto, Frédérique Mayer, Eric Bonjour and authors address the following topics:

1. Theme Editorial
2. ROBAFIS Student Competition Actuality: A Continuously Evolving Pedagogy for Systems Engineering
3. Improving Human-Machine Interaction Requirements for Maintenance Enabling Systems Specification
4. Graphical Models for RAMS Assessment and Risk Analysis of Systems of Systems Under Uncertainty
5. Design Process for Complex Systems Engineering Based on Interface Model
6. A Design Methodology And Representation Formalism for Changeable Systems – Application to Manufacturing Systems
7. A Method for Formalizing Requirements Interoperation in Complex Systems Engineering
8. A Tooled Approach for Designing Executable and Verifiable Modeling Languages
9. ScOLA, A Scenario Oriented Modeling Language for Railway Systems.

In addition, we are pleased to include a separate paper "Realizing the Potential of Connected Fitness Technologies: A Case for Systems Engineering Involvement" and a book review "Systems Thinking Made Simple: New Hope for Solving Wicked Problems."

We thank you, our readers, for both your laudatory and constructive feedback this first year transitioning *INSIGHT* from a news/feature magazine to a practitioner's magazine published in cooperation with Wiley. I thank assistant editor Lisa Hoverman, Chuck Eng for layout and design, our theme editors in 2015, assistant director for INCOSE publications Bob Kenley, Holly Witte in the publications office, and the staff at Wiley. We look forward to serving you in 2016 and beyond. ■

The INCOSE Foundation Where You Make the Difference

INCOSE FOUNDATION ™

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

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

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

Lead.
Build Expertise.
Solve Industry Challenges.

Master of Science in Systems Engineering

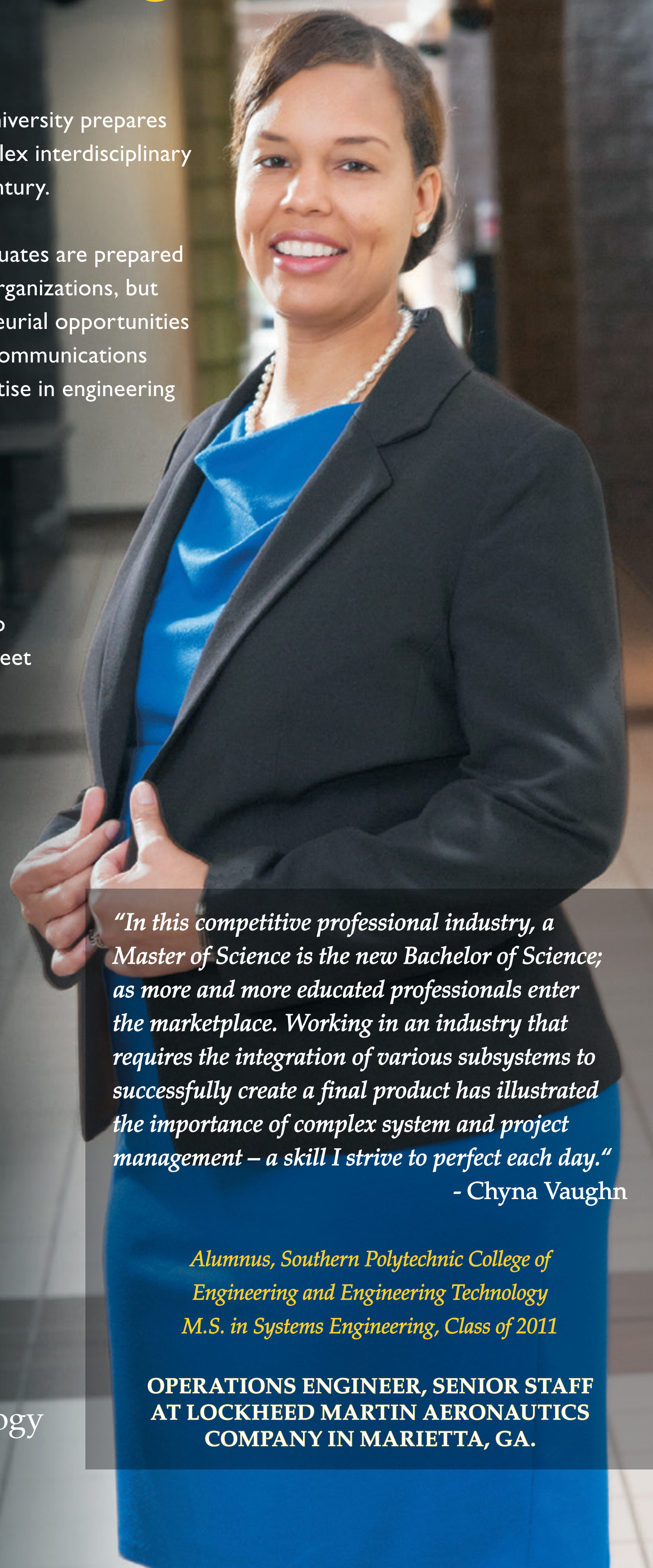
The M.S. in Systems Engineering at Kennesaw State University prepares graduates to lead engineering teams and manage complex interdisciplinary projects, solving the industry challenges of the 21st century.

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

Program highlights:

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

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



"In this competitive professional industry, a Master of Science is the new Bachelor of Science; as more and more educated professionals enter the marketplace. Working in an industry that requires the integration of various subsystems to successfully create a final product has illustrated the importance of complex system and project management – a skill I strive to perfect each day."

- Chyna Vaughn

Alumnus, Southern Polytechnic College of Engineering and Engineering Technology
M.S. in Systems Engineering, Class of 2011

OPERATIONS ENGINEER, SENIOR STAFF AT LOCKHEED MARTIN AERONAUTICS COMPANY IN MARIETTA, GA.

Kennesaw State University

Southern Polytechnic College of Engineering and Engineering Technology

AFIS Doctoral Symposium: Advancing Systems Analysis and Modeling in French Universities

Hervé Panetto, herve.panetto@univ-lorraine.fr; **Frédérique Mayer,** frederique.mayer@univ-lorraine.fr; **and Eric Bonjour,** eric.bonjour@univ-lorraine.fr

This special issue of *INSIGHT* section includes the main contributions presented in workshops held during the Systems Engineering Academia-Industry Forum. The aim of this issue is to provide an overview of the French research in the domain of systems engineering.

The Systems Engineering Academia-Industry Forum is organized by AFIS (Association Française d'Ingénierie Système), the French chapter of INCOSE, and supported by French universities as a regular series, usually every two years. The forum provides the opportunity for both academics and industrials to debate on:

- Education in systems engineering and developing competencies in systems engineering for professional situations,
- Developing and promoting the research in systems engineering.

Thus, workshops and plenary lectures are held during the forum in order to cover the theme of education, research, and practices of systems engineering.

In December 2014, the fifth edition of the forum was held in Cergy-Pontoise with the support of the university of Cergy-Pontoise. This edition focused on the important subject of "Interdisciplinary and Collaborative Engineering Based on Models" broken down in 11 topics:

- Model-based Systems Engineering,
- Research Activities in Systems Engineering,
- From the Systems Engineering Deployed by the Gen X to the Systems Thinking Needed by the Gen Y,
- Model-based Systems Engineering and PLM: Which Challenge,
- Innovation and Systems Engineering: Discrepancy or Complementarity,
- Systems Engineering and SysML for Education in Secondary Schools,
- Architecture and Allocation of Requirements,
- From Design to the Operations,
- Pedagogies and Systems Engineering: Challenges and Issues,
- ROBAFIS Challenge – Organized by AFIS - Promotes Project-based Collaborative Learning in Systems Engineering
- Doctoral Program in Systems Engineering as a Challenge for Research in French Universities.

The last topic gave rise to a specific workshop, the doctoral seminar, offering the opportunity for doctoral students to present and to discuss their doctoral works concerning systems engineering, with academics and industrials.

For this issue of *INSIGHT*, we invited doctoral students and their professors to submit an extended version of their presentations to emphasize the research aspects of systems engineering. We selected seven papers to include in this edition in order to promote research contributions for interdisciplinary and collaborative systems engineering, based on models.

This emerging paradigm is a challenge for the discipline of systems engineering in order to establish the relation of interoperability for coupling the representation of a system-situation (Lawson 2010), that designates and defines a problem or an opportunity, and the interdisciplinary representations co-specified by the respondent, engineering orchestrated by systems engineering, as a coherent collaborative whole (Bouffaron et al. 2014). The interest for this systemic vision in both engineering, education, and research, has been underlined by the international community of systems engineering (BKCASE Editorial Board 2015) in order to avoid systemic failures (Boardman and Sauser 2008) too often seen *a posteriori*.

The first paper, *Improving Human-Machine Interaction Requirements for Maintenance Enabling Systems Specification*, authored by Romain Lieber and Gérard Morel is a relevant example of such systemic vision expected by systems engineering to enable a system as a coherent socio-technical whole. As the performances of such a system depend on the synergies of the different interactions that take place between technical and human systems when operating a common object, the engineering requires that traditional systems engineering frameworks evolve in order to take into account those critical interactions that arise after the specification phase. The authors explore the paradigm of the hypothesis of possible inter-operations between physiological and technical processes for human-machine modeling. More precisely, within a model-based systems engineering (MBSE) approach, the authors focus on the specification of a physical-physiological perception interaction for a human to perceive correctly, the meaning of symbolic properties that technical objects afford.

The concept of Systems of Systems addressed by Siqi Qiu, Mohamed Sallak, Walter Schön and Zohra Cherfi-Boulanger in their paper *Graphical Models for RAMS Assessment and Risk Analysis of Systems of Systems Under Uncertainty* is another example of making systemic systems in uncertain circumstances, such as the European Rail Traffic Management System. By considering the hardware aspect, the network aspect, and the human factors that characterize these systems, the authors show there is a crucial need for a collaborative works between specialized engineering disciplines in order to reinforce the assessment of their requirement of Reliability, Availability, Maintainability, Safety (RAMS) and the analysis of the risk to which they may be subjected. Thus, the main contributions of this work consists in the proposition of a methodology to model and evaluate a European Rail Traffic Management System as a Systems of Systems, from which the RAMS can be evaluated by considering the unavailability of the whole SoS as an emergent property and from which some uncertainties that can be seen can be quantitatively modeled

Beyond the paradigm of Interdisciplinary and collaborative engineering based on models, systems engineering needs to address the notion of complexity of any system to be engineered. Thus, the works of Chen Zheng, Julien Le Duigou, Matthieu Bricogne and Benoît Eynard relates to the *Design Process for Complex Systems Engineering Based on Interface Modelling*. As shown by the authors, the rise of technologies leads to the multiplicity of disciplines when designing a technical system. The need to integrate all these disciplines leads to the sense of a technical system as a complex whole. In their works, the authors propose systems engineering in order to clarify and elicit customers' needs and required functionality of such complex systems in the early stage of a development cycle. Thus, the authors propose a new method for modelling interfaces of complex systems. The goal is to support the design process of complex systems based on the V-model by making use of the proposed interface modeling.

In the same vein, Nadège Benkamoun, Khalid Kouiss, Carrey Dilliott, Philippe Ducreuzot, Jean-Philippe Marcon, Anne-Lise Huyet and Michel Dhome argue that changeability is a characteristic that refers to systems engineering artefacts in order to suggest a new formalism for representing the architecture of a complex system in a rationale way. In the paper *A Design Methodology and Representation Formalism for Changeable Systems – Application To Manufacturing Systems*, the authors show how systems engineering elements (requirements and structural components) trace to each other in order to give a comprehensive view of a complex system and how changeability requirements – deriving from a stereotype profile of requirement – can allocate to any of those elements. They demonstrate that the advantages of tracing the design rationale in changeability design are multiple: Identify the relating elements to a changing need; Analyze change propagation and impacts between system engineering elements; Move toward a semi-automated design process with the development of a knowledge-based system about changeability possibilities.

In their paper, Anderson Luis Szejka, Alexis Aubry, Hervé Panetto, Osiris Canciglieri Júnior, and Eduardo Rocha Loures propose *A Method for Formalizing Requirements Interoperation in Complex Systems Engineering*, and address the problem of definition, modelling and formalisation of the system requirements and their relationships. They are formally defined in terms of transformation (translation, conversion and sharing) and traceability, based on an ontological approach. The authors demonstrate that there are significant lacks of interoperability between requirements in systems engineering due to heterogeneity of information from multiple domains and different phases of the systems lifecycle. Then they propose a method dealing with requirements interoperability across different phase of the life cycle and domains to formally model requirements interoperation in terms of transformation, traceability and conflicts analysis.

Blazo Nastov, Vincent Chapurlat, Christophe Dony, and François Pfister propose *A Framework for The V&V (Verification & Validation) in Model Based Systems Engineering: Towards Executable And Verifiable DSMLs* (Domain Specific Modeling Languages). They are developing a formal approach for modeling the dynamic semantics of DSMLs to achieve executable, verifiable and interoperable DSMLs (eviDSMLs). It is based on the idea that among DSML concepts, some of them take part in the dynamics of the DSML, called "evolving" concepts where the expected behaviors design uses software engineering languages.

Finally, Melissa Issad, Leila Kloul, Antoine Rauzy, and Karim Berkani developed *A Scenario Oriented Modelling Language for Railway Systems* that allows a textual and graphical representation of a railway system. The language can be extended to any other complex systems easily since it relies on generic system architectures.

To add value to this *INSIGHT* issue, the editors chose to include a paper related to the important issues that the ROBAFIS challenge procures to the education of systems engineering. In the paper *RobAFIS Student Competition Actuality: A Continuously Evolving Pedagogy for Systems Engineering*, Jean-Claude Tucoulou and David Gouyon describe the different methods of this particular educational challenge that encourages students to place themselves in an engineering role so that they practice systems engineering processes by designing, realizing, and validating a technical system in an operational situation.

These contributions provide valuable material for systems engineering research and education as tracks for future works. ■

We are grateful to the authors for their impressive contributions and to the reviewers for their valuable assistance to the scientific relevance of this issue of *INSIGHT*.

LIST OF AUTHORS AND ARTICLES

- Hervé Panetto, Frédérique Mayer, Eric Bonjour - *Editorial*
- Jean-Claude Tucoulou and David Gouyon - *RobAFIS Student Competition Actuality: A Continuously Evolving Pedagogy for Systems Engineering*
- Romain Lieber and Gérard Morel – *Improving Human-Machine Interaction Requirements for Maintenance Enabling Systems Specification*
- Siqi Qiu, Mohamed Sallak, Walter Schön and Zohra Cherfi-Boulanger – *Graphical Models for Reliability, Availability, Maintainability, and Safety Assessment and Risk Analysis of Systems of Systems Under Uncertainty*
- Chen Zheng, Julien Le Duigou, Matthieu Bricogne and Benoît Eynard – *Design Process for Complex Systems Engineering Based on Interface Model*
- Nadège Benkamoun, Khalid Kouiss Carrey Dilliott, Philippe Ducreuzot, Jean-Philippe Marcon, Anne-Lise Huyet and Michel Dhome – *A Design Methodology and Representation Formalism for Changeable Systems – Application to Manufacturing Systems*
- Anderson Luis Szejka, Alexis Aubry, Hervé Panetto, Osiris Canciglieri Júnior, and Eduardo Rocha Loures – *A Method for Formalizing Requirements Interoperation in Complex Systems Engineering*
- Blazo Nastov, Vincent Chapurlat, Christophe Dony, and François Pfister – *A Tooled Approach for Designing*

Executable and Verifiable Modeling Languages

- Melissa Issad, Leila Kloul, Antoine Rauzy, and Karim Berkani – *ScOLA, a Scenario Oriented Modeling Language for Railway Systems*

REFERENCES

- BKCASE Editorial Board. 2015. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*. Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Vol. v. 1.4. R.D. Adcock (EIC). http://www.sebokwiki.org.
- Boardman, J., and B. Sauser. 2008. *Systems thinking: Coping with 21st century problems*. Boca Raton, US-FL: CRC Press.
- Bouffaron, F., J. M. Dupont, F. Mayer, and G. Morel. 2014. Integrative construct for model-based human-system integration: a case study. Paper presented at the 19th International Federation of Automatic Control World Congress, IFAC'14. Cape Town, ZA. 24-29 August. https://hal.archives-ouvertes.fr/hal-01061943.
- Lawson, H. 2010. *A Journey Through the Systems Landscape*. College Publications. London, UK: Kings College.

About the Editors

Dr. Hervé Panetto is a full Professor of Enterprise Information Systems at the Université de Lorraine, TELECOM Nancy. He conducts research at Centre de Recherche en Automatique (CRAN) Joint Research Unit with the Centre National de la Recheche Scientifique (CNRS), where he manages a project on the use of ontology for formalising models related to the interoperability of production information systems. His work includes European projects like the Smart-fm project awarded by the Intelligent Manufacturing Systems (IMS) and the Sixth Framework Programme (FP6) Interoperability Research for Networked Enterprises Applications and Software Network of Excellence (INTEROP NoE) under the Information Society Technologies (IST) Fifth Framework Programme for Research (FP5). He is a member of the Editorial Board of the *Annual Reviews in Control, Computers In Industry, the Journal of Applied Mathematics*, Hindawi, the *International Journal on Universal Computer Science*, the scientific journal *Facta Universitatis*, series Mechanical Engineering, and an Associate Editor of the international *Journal of Intelligent Manufacturing* (JIM), Springer and the *Enterprise Information Systems* (EIS) journal, Taylor & Francis. After being Chair of the IFAC TC5.3 "Enterprise Integration and Networking," he is currently, Chair of the IFAC Coordinating Committee 5 on Manufacturing and Logistics. He received the IFAC France Award in 2013 and the INCOSE 2015 Outstanding Service Award.

Dr. Frédérique Mayer is Associate Professor at Université de Lorraine Équipe de Recherche sur les Processus Innovatifs (ERPI). Her research project is a contribution to the definition and the designation of the systemic interoperability between the interdisciplinary representations of a situation, a system and the engineering required to realize a system as a whole. In particular, her research focuses on the systemic thinking-based process for modelling a complex system in the field of systems engineering and enterprise modelling. She was the chairwoman of the Technical Committee 9.2 on "Social Impact of Automation" of IFAC and she is a member of the International Council on Systems Engineering (INCOSE). The main courses she teaches are on Systems Integration and Engineering, Enterprise Modelling and Systems Thinking-based approaches. In 2015, she received the INCOSE 2015 Outstanding Service Award

Dr. Eric Bonjour is a full professor specialized in systems engineering, at the Université de Lorraine Équipe de Recherche sur les Processus Innovatifs (ERPI) and Graduate School of Innovation and Systems Engineering (ENSGSI). In summary, his main research interests touch upon the fields of innovation, systems engineering, product architecture, product family, knowledge management, competency management, and project organization. He supervised 9 PhD theses related to these topics. He is the Vice-chair of the French chapter of INCOSE (AFIS), responsible for "Research – Training" topics (since 2012) and a co-head of the "Knowledge management and competence management in industrial companies" working group, affiliated to the National Research Council GDR MACS (groupement de recherche en modelisation, analyse et conduite des systemes dynamiques) in CNRS since 2007. He was the academic co-chair of the Program Committee of an international conference dedicated to systems engineering: Complex Systems Design and Management (CSD&M) in Paris, 2015. He has published more than 70 papers for conferences, journals, and books. In 2015, he received the INCOSE Outstanding Service Award 2015. ■

26th annual INCOSE international symposium
Edinburgh, UK
July 18–21, 2016
Edinburgh International Convention Centre

RobAFIS Student Competition Actuality: A Continuously Evolving Pedagogy for Systems Engineering

Jean-Claude Tucoulou, jeanclaude.tucoulou@incose.org **and David Gouyon** david.gouyon@incose.org

This paper presents the AFIS Robotics competition (RobAFIS), an annual event since 2006, by AFIS, the French chapter of INCOSE. This competition, as well as its pedagogical objectives, appeared in previous editions of *INSIGHT* (Tucoulou et al. 2011 and Tucoulou and Gouyon 2013).

RobAFIS enhances AFIS action, offering educational and research institutions an operation to better understand and develop the use of systems engineering best practices, as recommended and formalized by AFIS and more recently by the Graduate Reference Curriculum for Systems Engineering (GRCSE®) (Pyster et al. 2012). The recommended reference document for RobAFIS is the book by Fiorèse and Meinadier (2012).

Since 2007, the RobAFIS competition repositories and development files provided by teams are available as examples in the member area of the AFIS web site ("RobAFIS workspace" http://www.afis.fr), in order to be viewed and analyzed by the students and teachers.

Students and their supervising teachers have the opportunity to exchange with the jury AFIS expert members, working in industry or teaching systems engineering. During development, these experts answer, via a FAQ page on a RobAFIS dedicated collaborative space (http://www.robafis.fr), questions about technical or methodological issues related to stakeholder requirements or to the development document.

The main objective of RobAFIS is to highlight the benefits of basing systems engineering education on a project lifecycle realization: a full lifecycle including the implementation of an operational system, deployed by a client, in a real environment.

2015: "10TH ANNIVERSARY SPECIAL EDITION"

Each year this project benefits from previous edition feedback, in order to continuously improve the educational value of the event. Systems engineering best practices are also those on which AFIS technical committees have worked in last years, to ensure that companies can improve their methods, and more generally to progress on the engineering of the product and services they develop.

This competition is open to bachelor or master degree students, in a systems engineering discipline. Since the beginning of RobAFIS in 2006, about twenty-five different institutions participated at least one time, with an average of twelve registered teams each year.

To celebrate the 10th anniversary of RobAFIS, we invite all of the participating teams since 2006 to join RobAFIS in 2016, while encouraging brand new teams to participate.

RECENT TECHNICAL SUBJECTS:

2014: Product Lines And Remotely Operable Systems Problematics

The specification basis of the 2014 system was the design and assembly of three configurations, able to ensure three different object transport scenarios. Each configuration had to contain a common platform and a modular subsystem adapted to scenario features.

The main objective was first to highlight the importance of functional and physical architectures study phase, of optimization and justification of the selected architecture, and then to apprehend the concepts of integrated or modular (open) solutions specific to product line variants architecture.

With the introduction of two operating modes during mission progress (automatic mode and remote operated mode), students encounterd the problem of human introduction during operation. During the final phase, two ergonomics experts worked with the students during the configuration audit and the operational phase. They assessed the consideration of human factors in engineering and the usability of the Human Machine Interface (HMI), and exchanged with students and teachers on this topic.

2015: Interactions Between Operators and Technical Systems

The objective in 2015 is to highlight the consideration of human factors in systems engineering. This results in the choice of a subject in which the system in operational configuration includes a team of two operators working together. Operators interact with the robot and the objects of the deployment environment. This year, students need only design one configuration, but with a system of interest including the robot, two operators, the burdens (requiring relocation), and the platform on which the mobile robot operates.

Given the major role played by the human factor in the implementation of the system, the ergonomics experts intervene during the final phase to assess the quality of operators HMI and to give immediate feedback to participants.

INTRODUCTION OF A DEVELOPMENT IN TWO PHASES

To enhance the distinction between system architectural choices and technology choices, in 2015 we proposed a model of development in two phases:

- Phase 1: an upstream study phase, with the supply of a preliminary development

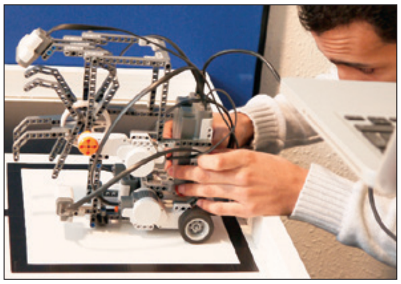


Figure 1. Engineering phase (test)

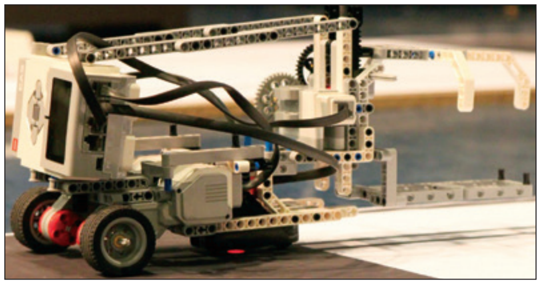


Figure 2: Deployment phase

document, focusing on the identification of possible solutions (at least 3 candidate solutions) and on the justified choice of the selected solution, on the basis of studied solutions drafts;
- Phase 2: a full development phase, with the supply of a detailed development document and an operational prototype, corresponding to the solution selected in the first phase.

In order to differentiate the sequencing of these two phases and the absence of iteration between phases 1 and 2, students receive phase 1 documents two weeks before phase 2 documents. The aim is to highlight the specific nature and contribution of both phases, in terms of progressive definition, the first one corresponding mainly to a system vision, the second one to a more product oriented vision.

We elaborated the scoring scale of development documents and their deliverables in order to highlight the respective issues attached to their good achievement and their contributions to engineering and product overall quality.

PRELIMINARY DEVELOPMENT DOCUMENT ARCHITECTURE

At the end of phase 1, student teams have to supply a preliminary development document, structured into 3 deliverables (RobAFIS 2015):

1. Preliminary version of requirement referential (Deliverable 10)
2. Presentation of possible architectural designs (Deliverable 20)
3. Justification of architecture choice (Deliverable 40)

DEVELOPMENT DOCUMENT ARCHITECTURE

Students achieve the results of the full development phase 2 with guidance in the form of a detailed development document, structured into 8 deliverables (RobAFIS 2015):

1. Final requirement referential (Deliverable 10)
2. Final architectural design (Deliverable 20)
3. Reference configuration (Deliverable 30)
4. Justification of definition (Deliverable 40)
5. Integration, verification, validation plan (Deliverable 50)
6. Maintainability study and maintenance definition (Deliverable 60)
7. Project management (Deliverable 70)
8. Assembly and verification instructions (Deliverable 80)

BEST PRACTICES FOR ENGINEERING QUALITY

With ten years of existence, the RobAFIS competition gives evidence to various best practices, R1 through R10:

- R1: The analysis of the *operational environment* and the related systems, source of requirements and constraints complementary to those included in the initial functional specifications.
- R2: The study of the functional architecture, an *essential step* for the requirement analysis and the physical architecture definition.
- R3: The search for *candidate architectures* and the justified choice of *the selected one*.
- R4: Performance allocation to functions, subsystems, and elementary components, with *values* and *tolerances* including component characteristic dispersions.
- R5: A *comprehensive* requirement repository applicable to the system and its constituents enriched by requirements identified during the design and applicable to the higher and system levels.
- R6: The realization of a *robust* solution incorporating *functioning* margins able to absorb dispersions related to specific constituents or resulting from the integration.
- R7: A justification guaranteeing the ability to achieve the mission, for all *scenarios* and for implementation *boundary conditions*.
- R8: The integration of the *support systems engineering* in the *system-of-interest* engineering.
- R9: The integration of *human factors* in systems engineering and in operating systems.
- R10: The importance of a *preliminary development phase* before the *full development phase*.

COOPERATION WITH GFSE: ROBSE 2015.

RobAFIS receives international recognition, during INCOSE IW and IS (Gouyon et al. 2013 and Tucoulou and Auvray 2014), INCOSE EMEA, and in collaborations with other local chapters. As an example, the GfSE, the German chapter of INCOSE, works with the AFIS and has since 2014. The GfSE (Mr. Schulze and Ms. Schlüter) organized the first edition of RobSE, a student competition corresponding to RobAFIS, with the support of INVENSITY GmbH, Corporate member of the GfSE (Mr. Martinez and Mr. Zutter) and the aid of the AFIS and of Jean-Claude Tucoulou RobAFIS Team Leader.

The event organization followed a visit of the GfSE in France for the 2014 RobAFIS event. This visit confirmed the strong interest of the GfSE to try to duplicate the concept in Germany. AFIS kindly provided the subject that forms the basis of the 2015 RobSE event. This reflects the AFIS-GfSE desire to grow this interesting event together to what could be an international INCOSE event in the future; we are looking forward to this cooperation.

As of this writing, 3 teams are registered from 2 universities (Hochschule Pforzheim and Hochschule Esslingen). With 2 registered teams, the Hochschule Pforzheim will host the event the 17th and 18th of December 2015. ■

REFERENCES

- Fiorèse S. and Meinadier J. P. (eds.). 2012. *To Discover and Understand Systems Engineering* (in French). Toulouse, FR: Cépaduès Editions.
- Gouyon, D., A. Faisandier, and J. C. Tucoulou. 2013. "RobAFIS – A Systems Engineering Student Competition." Paper presented at the 23rd Annual International Symposium of INCOSE, Philadelphia, US-PA, 24–27 June.
- Pyster, A., D. H. Olwell, T. L. J. Ferris, N. Hutchison, S. Enck, J. Anthony, D. Henry, and A. Squires (eds.). 2012. Graduate Reference Curriculum for Systems Engineering (GRCSE®). Hoboken, US-NJ: The Trustees of the Stevens Institute of Technology. www.bkcase.org/grcse/.
- Tucoulou, J. C., D. Gouyon, and E. Bonjour. 2011. "A Student Challenge in Systems Engineering: RobAFIS." INSIGHT 14 (4): 9-11.
- Tucoulou, J. C. and D. Gouyon. 2013. "RobAFIS Student Competition: From Systems Engineering to the Engineering of One System." *INSIGHT* 16 (4): 7-10.
- Tucoulou J. C., and G. Auvray. 2014. "Systems Engineering Competition." Paper presented at the 24th Annual International Symposium of INCOSE, Las Vegas, US-NV, 30 June – 3 July.

Improving Human-Machine Interaction Requirements for Maintenance Enabling Systems Specification

Romain Lieber, romain.lieber@gmail.com **and Gérard Morel,** Gerard.morel@univ-lorraine.fr

Maintenance enabling systems link technical objects and human beings within system-targeted operational situations. Their performances depend on the synergies of the different interactions that take place between the constitutive parts when operating a common object. Engineering such systems requires that traditional systems engineering (SE) framework evolve in order to take into account those critical interactions since the specification phase. The paradigm we have explored in our work is the hypothesis of possible interoperations between physiological and technical processes for human-machine modeling. Within a model-based systems engineering (MBSE) approach (Pyster et al. 2012), we focus on the specification of a physical-physiological sensory interaction *for a human to perceive rightly the meaning of symbolic properties technical objects afford.*

1. Our System of Interest (SoI): Visual Control of an Orange Signal

We observed that typical maintenance related-incidents involve equipment doors which remain unlatched but closed after maintenance task completion prior to walk around inspections. In this situation, operators are both power-sources and perception-sinks manipulating various mechanisms. For example operators control access to legacy/non-functioning equipment located behind the door that requires update/replacement. A Latch-Lock procedure (noted P_{LL}) specifies standard manipulation actions and their related perceptions, according to which, the objective is to ensure the reinstatement of the operational requirements linked to the door Latch-Lock state (noted R_{LL}). The operator sastisfies this requirement when controlling latch alignment with the surface of the doors (noted A_{LL}).

The Latch-Lock mechanism (noted M_{LL}) affords a lot of interaction possibilities to control in terms of alignment. More precisely, M_{LL} is a source of potential sensory signals perceived by the operators considering their physiological perceptive thresholds. Each mechanism signal is a physical quantity that propagates through space and time to reach each human operator in order to trigger an action that is stored in the form of learnt knowledge somewhere in the brain. We have investigated an orange visual signal (= conventional visual alert signal triggering attention) placed on the sides of the latch after further operational issues (Figure 1) leading to a new visual control requirement to be satisfied (noted REC_{LL}).

Thus, we questioned the physical-physiological nature of the human-machine perceptive interaction in order to communicate symbolic properties exhibited by M_{LL}. From a technical perspective, orange color represents a property of light characterized by a given wavelength that is a physical scale corresponding to a quantity of photons. To the human, orange color is captured by specific cells called cones (a type of photoreceptors) located on the very small part of the retina called the fovea centralis (a cellular mosaic). This overall consideration focused our attention on the physical-physiological communication upstream, the symbolic communication between artifact and human (Gibson 1975). So, we first investigated on a modeling framework that allows us to understand this perceptive process in order to specify physiological requirements on our SoI.

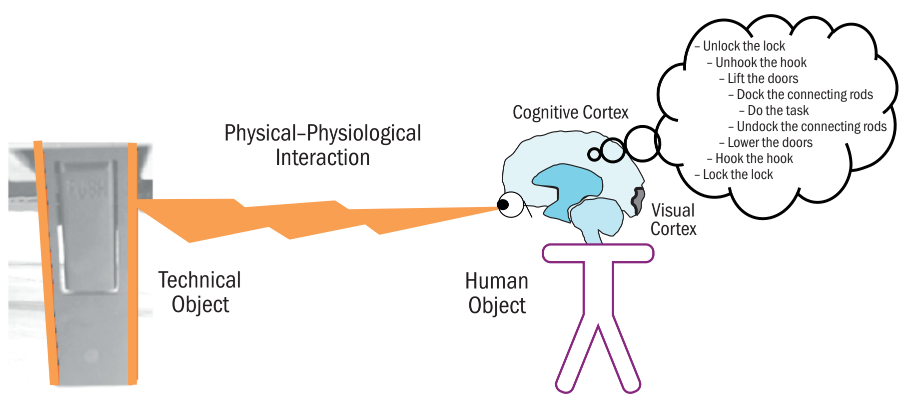


Figure 1. Visual physical-physiological interaction between the technical-object (M_{LL}) and the human-object (visual perceptive system composed of the eye-fovea centralis and the visual cortex)

2. Human-Machine Interaction Modeling Issues

We assume that by focusing on the understanding of the physical-physiological interaction nature we can succeed in the objective of specifying measurable requirements that meet MBSE requirements. The basis for this understanding is from works related to perception and action physiology (Berthoz 2012) and Integrative Physiology also called Mathematical Theory of Integrative Physiology (MTIP) (Chauvet 1993a, 1993b, and 1993c). Through Chauvet's theory, we understand physiological processes are hierarchically organized within space and time scales and stimulated by a set of functional interactions ψ_{LL} that spread over structural discontinuities (Figure 2a). Such discontinuities modify the nature of ψ_{LL},that is important when transmitting a physical flow to a physiological environment.

This interaction ψ_{LL} can trigger many mental processes that lead to the realization of the Latch-Lock actions thanks to the corresponding knowledge (noted K_{LL}) stored in the 'cognitive cortex.' Summarizing this functional organizational understanding led us to consider each physiological process involved in this 'perception-cognition loop' as a kind of thyristor in order to highlight the importance of the right stimuli to propagate ψ_{LL}. Specifically this applies to the first physical-physiological process emitted by the technical object.

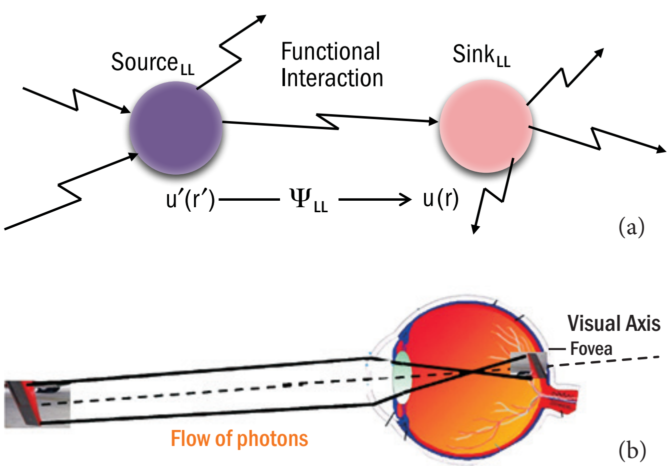


Figure 2. (a) Perceptive Functional Interaction ψ_{LL} between a SourceLatchLock (visual signal) located in r' and a SinkLatchLock (the fovea centralis) located in r.
(b) A technical source located in the physical environment interacts with a Physiological Source (eye or H_{LL}) located in the physiological environment

We focus on the propagation of ψ_{LL} from the latch to the fovea assuming from MTIP the different structural discontinuities of the eye introduce no modifications to the nature of the flow of photons (the receiving eye is a normal-functioning organ). This hypothesis enables us to consider current available physiological data for this specific area in order to specify the law $K_{LL2.2}$ at a scale-factor sufficient for a MBSE specification phase (see Table 1).

The amount of electrical power coming from the transmutation of the 'orange color photon quantity' dictates the spreading of the stored knowledge K_{LL} received by the fovea. This quantity depends directly on the reflected one from latch according to the law M_{LL2} (Figure 2b). The luminance of a light source is a photometric measure of light intensity that is dependent on the human eye sensibility. We evaluate the corresponding physical light power in watts according to the wavelength of the photons. That leads to some qualitative and quantitative physical-physiological requirements addressed in Table 1. Note that the alignment requirement K_{LL2} impacts others human factors such as $K_{LL2.1}$ related to the operator posture and technical factors such as $M_{LL2.3}$ related to the latch position.

An MTIP-based model enables us to functionally understand the physiological behavior of a human being in order to make hypotheses for computational simulation purposes according to available data and the scale-factor related to the MBSE decision-making process. This measurability requirement limited our case-study to the specification of the physical-physiological orange-signal perception.

3. Human Machine Interaction Specification Process and Verification Issues

The physical-physiological interaction is one of the interactions formalized as Human Factors requirements during maintenance system specification process. We formalize the specification of technical-human interactions within an MBSE process thru the *Requirements Analysis Model for Socio-Technical Systems* (Hall et al. 2005). This model highlights the fact that at least three types of interaction specifications (I_{TW}, I_{HW}, I_{TH}) enable us to specify concurrently the targeted socio-technical system, S_{LL}. I_{TW} specifies the target system technical requirements prescribing the Latch-Lock Procedure P_{LL} that implements the Latch-Lock mechanism M_{LL} within the operational context (W_{LL}). I_{HW} specifies the target system human factors requirements prescribing the capabilities K_{LL} that the maintainer H_{LL} has to own to operate within W_{LL}. I_{TH} specifies the target system human-machine interaction requirements prescribing the requested interface properties within W_{LL}. Our studied physical-physiological interaction specification describes one type of the human-machine interaction specification I_{TH} that can impact the design of the considered target system.

Table 1. Collection of Physical-Physiological Interaction Requirements we highlighted during our study

ID	DESCRIPTION
K_{LL2}	«Physiological Requirement»: The Visual-Sink$_{LL}$ must be aligned on the Signal-Source$_{LL}$
$K_{LL2.1}$	«Physiological Requirement»: Anthropometric axes must be aligned according to the visual axis
$K_{LL2.2}$	«Physiological Requirement»: The photons flow received by the fovea must be specified according to the law: nbλ= 2.10-13 / (hν) = 6.10^5 (s-1)
M_{LL2}	«Technical Requirement»: The reflectance coefficient ρ determines the fraction of the reflected power Po versus the received power Pi according to $P_o(x,t) = \rho P_i(x,t)$
$M_{LL2.3}$	The latch must be aligned on the visual axis according to the transformation matrix: $Z_o = -oT_v Z_v$

The specification of the system as a whole S_{LL} leads us to consider it as a set of functions (satisfying R_{LL}) at the MBSE architectural functional phase. Among these functions, we focus on the perception one. The transition to the MBSE architectural organic phase leads us to consider this function allocation on both technical and human components. To do so, the related two domains have to collaborate to specify the perception interactions as a whole. As a consequence in an MBSE context, we highlight the need of two new roles within the systems engineering domain: technical and human factor architects in order to better balance functions allocation to organic components respectively technical or human-based. Generally speaking, the system specification process (Figure 3) can be seen as a series of transformations between iterative problem-space_source and solution-space_sink within different domains. A problem-space describes requirements that solution-spaces have to satisfy by prescribing requirements based on domain skills.

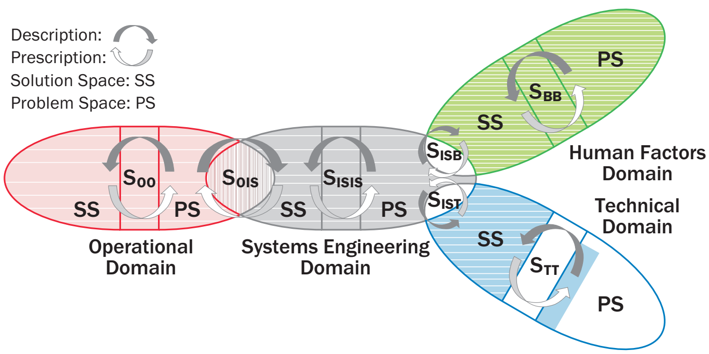


Figure 3. Systems engineering, pivotal domain between two expert domains respectively, technical and human

We pursued the MBSE approach (Lieber 2013) by simulating human-machine SysML models enabling us to verify numerically R_{LL}. More precisely, we described maintenance plausible contexts scenarios through use case diagrams enabling us to specify physical-physiological interaction requirements of the alignment that meet the operational recommendation REC_{LL}. This specification is not sufficient to properly formalize the requirements of the interfaces considered as the starting point of the different contextualized interactions. Based on an MTIP framework, the human factors domain-of-interest (physiology) prescribes to systems engineering domain a set of measurable requirements $\{K_{LL2}, K_{LL2.1}, K_{LL2.2}\}$ to be satisfied in order to meet REC_{LL}. We formalize these requirements thru SysML parametric diagrams in the form of mathematical equations constraints. Thus, we simulate scenarios of light rendering in order to determine the physiological lighting efficiency to perceive or not perceive an orange visual signal (Figure 4).

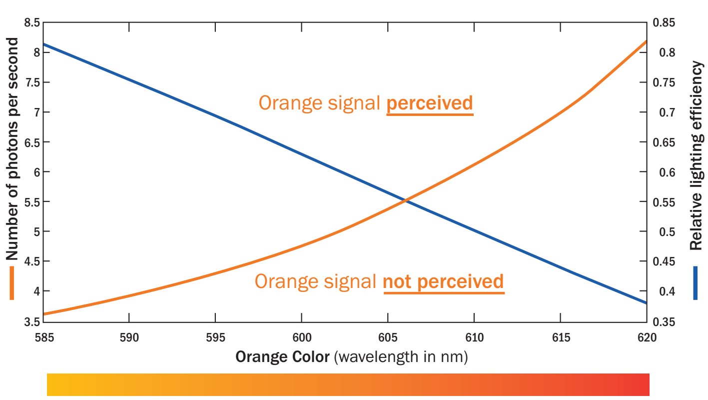


Figure 4. Result of a physical-physiological interaction simulation using the computer algebra system Matlab®

4. Conclusion

We find that functionally defining a system as a whole prior to architecting it with two kinds of organic parts {human and technical} is rare in current systems engineering organizations. Indeed, we find many human considerations as an extra burden within the teams that don't know how to deal with too often non-measurable requirements. By proposing such a rationalized specification process that focuses on the physical-physiological nature of the interaction we aim to improve the current situation and insist on the necessary collaborative work (under the systems architect responsibility) between human and technical domains to obtain *measurable human-machine requirements* that satisfy stakeholders' operational requirements. ■

REFERENCES

- Berthoz, A. 2012. "Neuronal basis to decision-making. An approach of cognitive neuroscience." *Annales Médicopsychologiques*, 170(2) :115-119.
- Bouffaron, F., D. Gouyon, D. Dobre, and G. Morel. 2012. "Revisiting the interoperation relationships between Systems Engineering collaborative processes." Paper presented at the 14th IFAC Symposium. Bucarest, RO, 23-25 May.
- Chauvet, G. 1993a. "Hierarchical Functional Organization of Formal Biological Systems: A Dynamical Approach. I. The Increase of Complexity by Self-Association Increases the Domain of Stability." *Philosophical Transaction of the Royal Society London*, 339(1290): 425-444.
- Chauvet, G. 1993b. "Hierarchical Functional Organization of Formal Biological Systems: A Dynamical Approach. II. The Concept of Non-Symmetry Leads to a Criterion of Evolution Deduced from an Optimum Principle of the (O-FBS) Sub-System." *Philosophical Transaction of the Royal Society London*, 339(1290): 445-461.
- Chauvet, G. 1993c. "Hierarchical Functional Organization of Formal Biological Systems: A Dynamical Approach. II. The Concept of Non-Locality Leads to a Field Theory Describing the Dynamics at Each Level of Organization of the (D-FBS) Sub-System." *Philosophical Transaction of the Royal Society London*, 339(1290): 463-481.
- Gibson, J. J. 1977. "The Theory of Affordances." In *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, edited by R. Shaw & J. Bransford, 127-143 Hoboken, US-NJ: John Wiley & Sons Inc.
- Hall, J. and L. Rapanotti. 2005. "Problem Frames for Socio-Technical Systems." In *Requirements Engineering for Socio-Technical Systems*, edited by J. L. Maté & A. Silva, 318-339. Hershey, US-PA: IGI-Global.
- Lieber, R. 2013. "Spécification d'exigences physico-physiologique en ingénierie d'un système support de maintenance aéronautique." PhD diss., Lorraine University (Lorraine, FR).
- Pyster, A., D. Olwell, N. Hutchison, S. Enck, J. Anthony, D. Henry, and A. Squires. 2012. "Guide to the Systems Engineering Body of Knowledge (SEBoK) version 1.0.1." Hoboken, US-NJ: The Trustees of the Stevens Institute of Technology.

Graphical Models for Reliability, Availability, Maintainability, and Safety Assessment and Risk Analysis of Systems of Systems Under Uncertainty

Siqi Qiu, siqiqiu@sjtu.edu.cn; **Mohamed Sallak**, sallakmo@utc.fr; **Walter Schön**, wschon@utc.fr; **and Zohra Cherfi-Boulanger**, zohra.cherfi-boulanger@utc.fr

The objective of this paper is to present our original work on proposing graphical models for RAMS (Reliability, Availability, Maintainability, Safety) assessment and risk analysis of Systems of Systems (SoSs) under uncertainty. Firstly, we proposed dysfunctional models of a railway signalling system, European Rail Traffic Management System (ERTMS), which is considered an SoS. Then, we evaluated some RAMS attributes of the whole SoS. At last, different kinds of uncertainties were taken into account quantitatively.

SoSs are large systems whose components are themselves systems which interact to realize a common goal, and for which the malfunction of a single system can have some serious consequences on the performance of the whole SoS. So far, SoSs do not have a universally accepted definition. Jamshidi (2008) considered that SoSs are large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal. INCOSE (Haskins 2006) considered that the term SoS should be applied to a system-of-interest whose system elements are themselves systems; typically these entail large-scale inter-disciplinary problems with multiple, heterogeneous and distributed systems. Maier (1998) proposed five traits to distinguish very large and complex but monolithic systems from SoSs: Operational Independence of the Elements, Managerial Independence of the Elements, Evolutionary Development, Emergent Behavior, and Geographic Distribution. Among different definitions, Maier's criteria on SoS encompass all the characteristics of our research object: European Rail Traffic Management System (ERTMS).

The main international standard for train control and command systems is the ERTMS. Originally designed for European Union countries, ERTMS has now become a global standard used by several other non-European countries, such as China, Australia, Brazil, and Mexico. ERTMS has two parts, the first part being ETCS (European Train Control System), which is a standard for train control systems, and the second part being the GSM-R (Global System for Mobile communications-Railways), which is an international wireless communications standard for railway communication and applications. ETCS has three levels. ERTMS/ETCS Level 1 is the classic standard of European railway signalling system. The driver controls the train according to the lineside signals. In ERTMS/ETCS Level 2, radio communication is implemented to realize the communication between ground and train. ERTMS/ETCS Level 3 introduces a moving block section technology. Accurate and continuous position data is directly supplied to the control centre by the train's positioning system. ERTMS/ETCS Level 3 is currently in testing phase. As ERTMS/ETCS Level 2 is now widely implemented, our main modeling subject is the ERTMS/ETCS Level 2.

Figure 1 describes precisely the ERTMS/ETCS Level 2 inspired by the work of Flammini (2009). It consists of three parts:

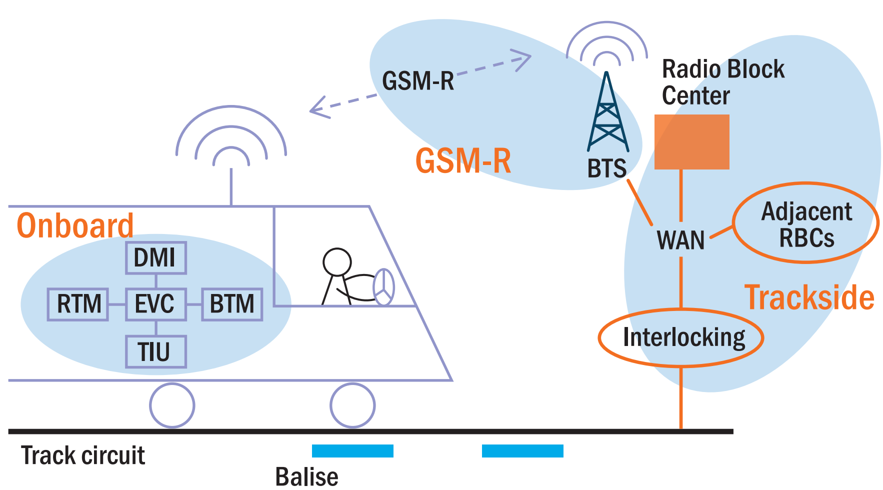


Figure 1. Railway signalling system equipped ERTMS/ETCS Level 2

The onboard system, the trackside system and the GSM-R system. The onboard system is installed in the train to control train movements. It uses the information received from the trackside system to create a "braking curve." The train driver should respect this speed profile in order to slow down or brake before stop signals or emergencies. It also receives telegrams from balises and sends position reports to the trackside system via GSM-R. The trackside system performs train routing, acquires the track circuit occupation status, detects train position, and sends correct movement authorities and static speed profiles that the trains and their operators should comply with.

Emergence is a disputable topic in the SoS domain. There is no precise and generally accepted definition of emergence. Given the focus of our research on railway systems, we use our own particular definition of an emergent behavior for an SoS: despite redundancy, taking over from human operators and other automatic procedures, the occurrence of a failure at the SoS level is considered as emergent behavior. A failure at SoS level refers to a failure which has not been considered on any of the systems of the SoS and that can happen only when those systems are cooperating.

Before proposing the models of the studied system, we need to choose appropriate modeling languages. Modeling languages are usually developed to deal with certain issues. There is not a language which can perfectly model a real system and deal with all issues. A modeling language is chosen according to the characteristics of the studied systems and the studied problems. The dynamic behavior, the composition of the SoS, and the integration of uncertainty are the important issues in our work. The chosen modeling languages should solve these problems.

Uncertainty is also an important part of our work. There are always deviations between the real world and its representation in models. Probability distribution is usually used to quantify the natural variability of random phenomena. However, when the uncertainty arises from incompleteness or imprecision of knowledge and data, probability is no longer appropriate to quantify it (Aven and Nøkland 2010, Aven 2011). This kind of uncertainty is called epistemic uncertainty. Researchers developed several uncertainty theories for epistemic uncertainty quantification, including Bayesian theory, imprecise probability theory, possibility theory, belief functions theory, and more. The studied uncertainty in this work is epistemic. We propose the use of belief functions theory because it is well adapted to model the uncertainty of systems by quantifying the belief masses of the uncertainty provided by experts. The belief functions theory also called Dempster-Shafer theory, is a generalization of the Bayesian theory of probabilities. Whereas the Bayesian theory requires probabilities for each question of interest, belief functions allow us to base degrees of belief for one question from probabilities for another. The belief mass is used to express one's degree of belief.

For systems that involve humankind such as railway and aeronautical systems, some safety standards introduce Reliability, Availability, Maintainability, and Safety (RAMS) requirements that stakeholders require to ensure the safety design of such systems. 'RAMS allocation' deals with the setting of RAMS goals for individual systems such that an SOS meets a specified RAMS goal. Engineers decompose SoS-level allocations successively using dependability models (Reliability Block Diagrams, Fault Trees, and more.) until the engineers apportion an appropriate set of RAMS measures to all the systems comprising the SoS. In railway systems, a common method for managing complexity in RAMS allocation is to divide the SoS into systems according to function. However, decomposing RAMS requirements is far from being straightforward. Quantifiable goals such as cost or performance may be decomposed by allocating a fixed limit on each system, but the RAMS requirements of the SoS cannot be expressed simply as the sum or the product of the RAMS requirements of its constituent systems. Consider a railway availability goal: "The operational availability of the ERTMS/ETCS, due to all the causes of failure, shall be not less than 0.99973" (EEIG ERTMS Users Group 1998). The concept of "operational availability" at the SoS level does not have the same meaning for individual systems and is not the sum or the product of operational availabilities of individual systems. It's important that the design of these SoSs takes into account the RAMS requirements of safety standards not only at the system level but also at the SoS level.

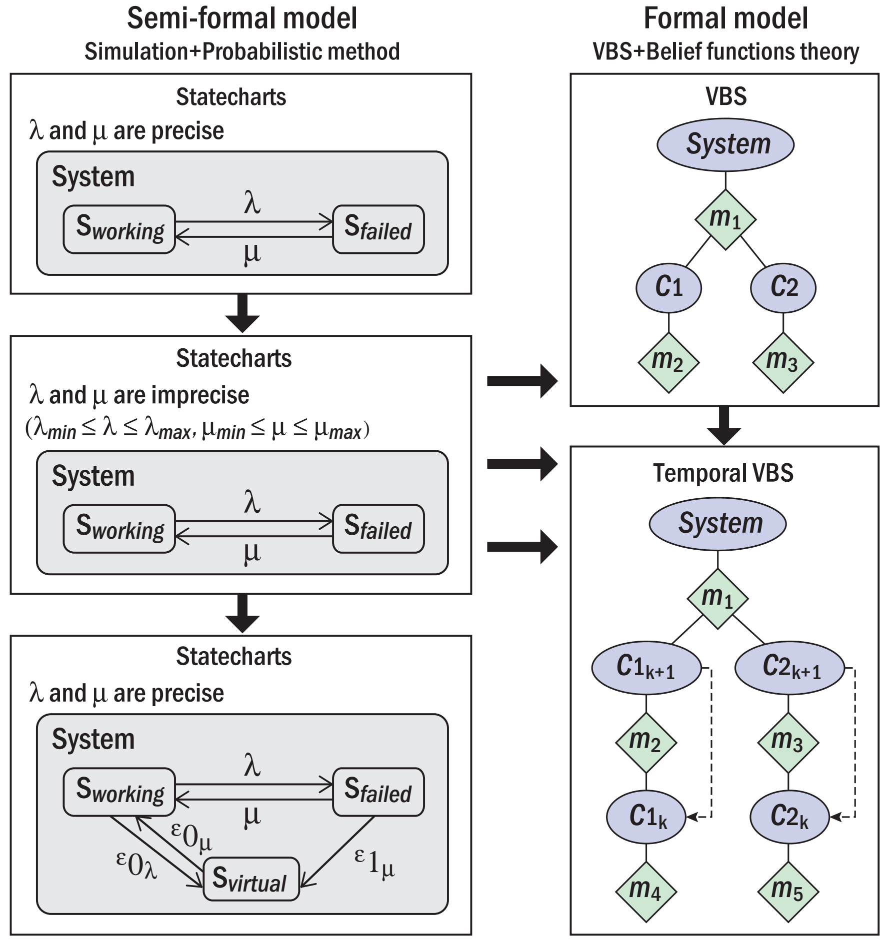


Figure 2. Two proposed models

Figure 2 shows our two proposed models. Models of ERTMS/ETCS Level 2 are too huge that only generic diagrams are given here. Engineers first considered ERTMS/ETCS Level 2 as an SoS and modeled it using statecharts. Engineers obtain RAMS parameters of ERTMS/ETCS Level 2 by analyzing the results of simulations. Statecharts are suitable for modeling the dynamic behavior of the SoS, and they support the hierarchy of states and orthogonal regions. When modeling systems in statecharts, two kinds of epistemic uncertainties may exist in the model: epistemic parametric uncertainty and epistemic state uncertainty. Though statecharts are dynamic and suitable to model dynamic behaviors of the SoS, they cannot handle the uncertainties represented by belief masses. This forces us to find another language that can deal with all kinds of uncertainties.

Engineers introduced valuation-based system (VBS) as a general language for incorporating uncertainty in expert systems. It encodes knowledge using functions called valuations. Uncertainties are represented by valuations. There are two operators called combination and marginalization that operate on valuations. Combination corresponds to aggregation of knowledge. Marginalization corresponds to coarsening of knowledge. We describe the process of reasoning in VBS simply as finding the marginal of the joint valuation for each variable in the system. We obtain RAMS parameters of ERTMS/ETCS Level 2 by valuation of the variable which represents the SoS. VBS has the following advantages:

- VBS provides a compact representation of systems components and their dependencies;
- VBS is well adapted to represent and propagate all types of uncertainties in models;
- VBS can model and evaluate performances of multi-state systems.

Thus, we modeled the ERTMS/ETCS Level 2 in VBS and analyzed uncertainties using belief functions theory. However, VBS is a static modeling language. It only supports the evaluation of the performance at a given instant. As we were interested in the evolution of the performance of the ERTMS Level 2, we proposed a temporal VBS approach which allows evaluating the performance as a function of time. We suppose that the state of a component only depends on its previous state (Markov model), so that the state probability of the component can be calculated by a recurrence relation.

As shown by statistics, human factors cause a large amount of accidents in railway transport. Although engineers do not take into account human factors in railway standards, the necessity of taking human errors into account in railway accidents analysis is an idea widely accepted by all the experts. In the future, we will propose a formal quantitative model of human factors in order to integrate it into a global model of the accident risk analysis. ■

REFERENCES

- Aven T. and T. Nøkland. 2010. “On the Use of Uncertainty Importance Measures in Reliability and Risk Analysis.” *Reliability Engineering & System Safety* 95(2): 127–133.
- Aven T. 2011. “Interpretations of Alternative Uncertainty Representations in a Reliability and Risk Analysis Context.” *Reliability Engineering & System Safety* 96(3): 353–360.
- EEIG ERTMS Users Group. 1998. ERTMS/ETCS RAMS Requirements Specification. Brussels, BE: International Union of Railways (UIC).
- Flammini, F. 2009. “Model-based Dependability Evaluation of Complex Critical Control Systems.” PhD diss., Università Degli Studi di Napoli Federico II (Napoli, IT).
- Haskins, C., ed. 2006. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 3. San Diego, US-CA: INCOSE.
- Jamshidi, M. 2008. *Systems of Systems Engineering: Principles and Applications*. Boca Raton, US-FL: CRC Press.
- Maier, M. W. 1998. “Architecting Principles for Systems-of-Systems.” *Systems Engineering* 1 (4): 267–284.

A LEGACY OF SOLVING
COMPLEX PROBLEMS

At Engility, systems engineering is at the core of what we do — supporting the government’s most critical missions and programs. That’s why defense, intelligence and civilian agencies turn to us. Our proven systems engineering and integration tools, techniques, technology and training helps to assure their systems are efficiently designed, accurately built and engineered for mission success.

ENGILITY
Your Mission. Our Commitment.
engilitycorp.com

Requirements Analysis and Specification Writing

Requirements Analysis and Specification Writing are sciences practiced by many, mastered by surprisingly few. And yet, the payoff from achieving excellence in these areas is large. The two aspects, Requirements Analysis and Specification Writing, are treated as separate but related topics, in modules of three and two days duration respectively.

The three-day Requirements Analysis course addresses the techniques used to capture, validate and gain a complete understanding of requirements communicated at all stages of the system life cycle. The two-day Specification Writing course addresses in detail the conversion of individual requirements into effective requirements specifications. The course focuses on the structure and language of requirements specification.

The two courses are complementary, with little overlap. They may therefore be taken together, or taken individually. The two courses comprise Project Performance International's popular 5-Day public course in Requirements Analysis and Specification Writing. The course is delivered worldwide.

Upcoming Course Deliveries

January 25 - January 29, 2016	**Seattle, WA**
March 7 - March 11, 2016	**London, United Kingdom**
April 4 - April 8, 2016	**East Orlando, FL**
May 2 - May 6, 2016	**Washington, DC**
May 30 - June 3, 2016	**Las Vegas, NV**
June 13 - June 17, 2016	**Munich, Germany**
July 25 - July 29, 2016	**Birmingham, United Kingdom**
August 29 - September 2, 2016	**Amsterdam, the Netherlands**

"The best thing about the course was demonstrating how poor requirements lead to waste, and how, by being methodical we can 'hopefully' solve a lot of angst between the user and the contractor."

- delegate

"The best thing about the course was the amount of detail presented, along with supporting courseware material to bring back for study/reference."

- delegate, United Kingdom

"The course added some very important fundamental knowledge to my understanding of requirements analysis and how to write a requirements specification."

- delegate, Washington DC, USA

PROJECT PERFORMANCE INTERNATIONAL

Improve the performance of your organization and enhance your career by taking PPI's training. Visit our website to find out more information:

www.ppi-int.com

Stand out from the crowd
Become SEP certified

This **5-Day CSEP Exam Preparation Course** combines a mixture of presentations, workshops and practice examinations. In particular, the Q&A sessions ensure that any unclear topics are answered in a format that provides adequate time for the content to be mastered. All our courses are facilitated by world-class leaders, who are highly experienced and knowledgeable in dealing with all aspects of CSEP training.

CTI's 5-Day CSEP Exam Preparation Course provides you the perfect grounding to take the INCOSE CSEP examination with confidence.

CONSIDER AN ON-SITE COURSE FOR YOUR TEAM

Prepare your team to sit the INCOSE CSEP exam with an on-site CSEP Preparation 5-day course. On-site training is an effective and cost effective way to train a team. CSEP public training courses are available on-site worldwide. These courses are based on version 4 of the INCOSE Systems Engineering Handbook. Take advantage of:

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

"Feel like I have the tools necessary to work towards taking the exam. The course material laid a good foundation/understanding of systems engineering fundamentals." - delegate, USA

"A very valuable learning experience; well-taught and facilitated. The instructor was very knowledgeable well-prepared, pleasant and accommodating." - delegate, USA

"Thanks again for the best preparation course. Thanks to you, I felt really comfortable in studying the handbook and finally taking the exam." - delegate, Sweden

Gain international recognition of your systems engineering knowledge and a competitive edge in your career. Find out more by visiting our website:

www.certificationtraining-int.com

CERTIFICATION TRAINING INTERNATIONAL
A PPI Company

DECEMBER 2015
VOLUME 18 / ISSUE 4

Design Process for Complex Systems Engineering Based on Interface Model

Chen Zheng, chen.zheng@utc.fr; **Julien Le Duigou,** julien.le-duigou@utc.fr; **Matthieu Bricogne,** matthieu.bricogne@utc.fr; **and Benoît Eynard,** benoit.eynard@utc.fr

1. Introduction

Current trends indicate that systems become increasingly complex, leading to constantly changing requirements and the introduction of new technologies (Blanchard 2004). Design of complex systems requires contribution from numerous technical disciplines and expertise. Mechatronic systems, for example result from integration of electrical/electronic systems, mechanical parts, and information processing (Carryer 2011). With technology development, more disciplines, such as optical, hydraulic, pneumatic disciplines, and more, integrate into complex systems. Systems engineering is a multidisciplinary approach to enable the realisation of successful complex systems (INCOSE 2015). During systems engineering activities, designers select concepts to employ in solving a given design problem and decide how to interconnect these concepts into an appropriate system architecture. The complex system is one engineers decompose into sub-systems (or components) designed by different disciplines and the interfaces between them. However, most of current studies on design only focus on the decomposition and integration of sub-systems (or components), and neglect the significance of the interface (Zheng 2014). The interface in a complex system refers to any logical or physical relationship required to integrate the boundaries of components or of systems with their environment (Liang and Paredis 2004). In order to initiate detailed design activities with the multidisciplinary collaboration issued from systems engineering activities, this paper introduces an interface model and a design method based on this interface model to help the designers to achieve an integrated multidisciplinary design of complex systems during the systems engineering activities.

2. V-model for Design of Complex Systems

The V-model is a way of representing the systems engineering and development process. It presents a general flow for the product development process. The V-model starts with clarification of users' requirements and ends with a user-validated system (Forsberg and Mooz 1998). According to the V-model, the left side of the "V" depicts the "system definition and decomposition." Considering the customers' requirements and the required functionalities of the complex system, the designers should define and decompose the complex system into sub-systems (or components) designed by different disciplines and the interfaces between them. Design teams from different disciplines design the individual sub-systems (or components) in the detail design phase. After the detail design phase, the systems engineer integrates and tests the sub-systems (or components), which corresponds to the right side of the "V" (United States Department of Transportation 2007).

This introduction shows that multidisciplinary collaboration of the discipline-specific design phase plays a key role during the whole design process of complex systems. However, the V-model is too generic to support the multidisciplinary collaboration of detail design phase.

In order to overcome the limitations of multidisciplinary collaboration of the V-model during the detail design phase, we introduce an interface model and a variance of V-model based on the interface model. In the next section we present the details of the interface model.

3. Interface Model

The topic of interface is at the heart of the multidisciplinary nature of systems engineering. As discussed before, decomposing the complex system into sub-systems (or components) and the interfaces between them allows engineers to achieve the desired system architecture during the architecture design process. However, engineers neglect the interfaces in current studies on system design process. In this paper, an interface model is proposed to provide a common representation for the interfaces defined by different disciplines, such as geometrical interfaces, control data flow, software interfaces, electrical standard ports, and more. Figure 1 shows the interface model represented thanks to the UML class diagram.

First, the proposed interface model provides required details describing the interface. It contains classes to define the attributes of one interface and its ports. The

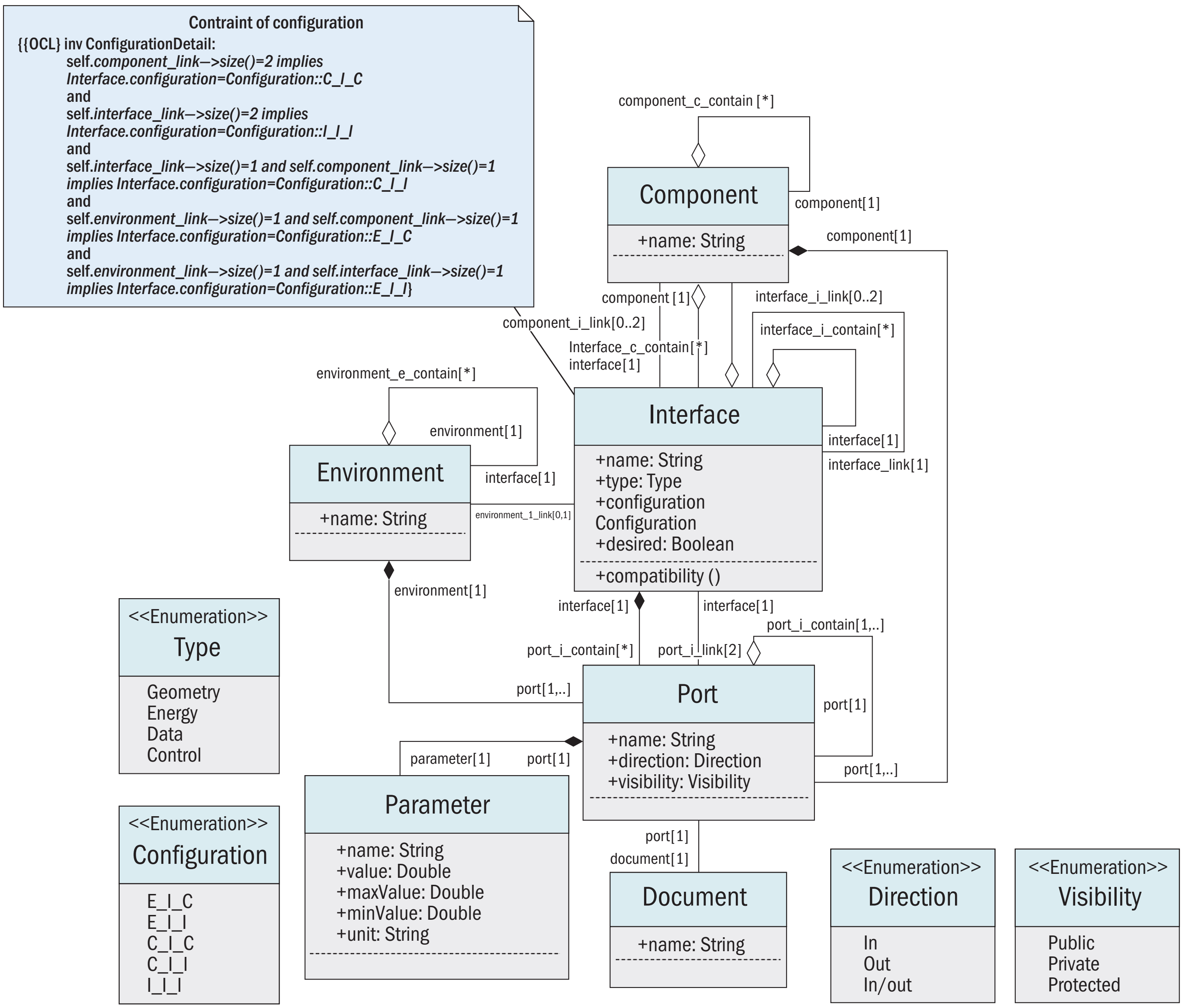


Figure 1. UML class diagram of interface

term "port" is the primary location through which one element of a system interacts with other elements. We define the interface attributes taking into consideration three different features: **type**, **configuration** and **desired**. **Type** attribute focuses on which types of transfer (geometric, energy, control, or data) occur through one interface. **Configuration** attribute describes which elements the interface links. **Desired** attribute (a Boolean) expresses whether the interface creates positive effects (data or energy transmission) or unintended side-effects (heat, magnetic fields, vibration and other side effects). Moreover, the interface model contains one method, **compatibility**, included to check the compatibility of the interface. On the one hand, the attributes contained by the interface model provide a common representation for the interfaces defined by design teams of different disciplines. On the other hand, the **compatibility** method helps the designers to guarantee the different components integrate correctly and ensure the multidisciplinary integrated design among design teams and an early testing and verification.

Second, the interface model represents the relationship between the interface and component of complex systems. We decompose a component into several sub-components connected by sub-interfaces, so the class **component** is an aggregation of interface and itself. In order to refine the architecture of the complex system during the design process, an interface decomposes into a group of sub-components and sub-interfaces, so the class interface can be an aggregation of **component** and itself.

With the support of the proposed interface model, we propose a variant of the V-model in the next section to overcome the limitations of V-model during the system architecture design process.

4. Variant of V-model Based on the Proposed Interface Model

The design of complex systems requires the multidisciplinary collaboration of different design teams during the detail design phase. The concurrent design process is where the design tasks for different sub-systems (or components) occur via different design teams in parallel. All the sub-systems (or components) designed in the detail design phase should integrate correctly with each other. In the proposed interface model-based design method, the designers of different disciplines can use a common representation to define the interfaces existing in complex systems. Once the data model of an interface is instantiated, the interface compatibility should be checked by this **compatibility** method. If the components prove to be incompatible with each other, an iterative process can ensue. With the support of the interface model, the integration of sub-systems (or components) designed by the design teams from different disciplines can be guaranteed and the multidisciplinary collaboration among design teams can be ensured during the detail design phase. We propose a case study by means of a three dimensional (3D) measurement system to demonstrate the design method based on the interface model in the following section.

5. Case study

The case study chosen to demonstrate the contribution of the proposed design process draws origin from a three dimensional measurement system (Dupont et al. 2011). This measurement system is for reconstruction of an object surface based on optical measurement. It is a complex system integrating synergistically the electrical/electronic system, mechanical parts, information processing, optical technology, and medical knowledge. The envisioned application for this system is for colonoscopy purposes. So one of the major leading issues is that the image guide must remain flexible and compact. To illustrate our proposition around an encountered design issue, we chose a scenario focusing on the image guide. At the early stage of the design process, we defined an interface between the DMD (Digital Micro-mirror Device) and the image guide. On one hand, during the design process, the designers from the optical team choose the image guide while the requirement concerning the acceptable bending diameter constrains the image guide. On the other hand, in order to achieve a better quality of image reconstruction, the engineers should consider the geometric shapes of the DMD and the guide image.

First, the interface model provides a common representation for the interfaces defined by design teams of different disciplines. For example, the optical team designers choose the image guide design while the bending diameter of the image guide is constrained by the intestinal structure decided by the members of medical team. The common representation of the interface proposed by the interface model can solve the conflict problem of the two teams of different disciplines. Second, once the data model of an interface is instantiated, engineers should check the interface compatibility by this method (**compatibility**). If the components prove to be incompatible with each other, an iterative process can follow. For instance, the interface between the DMD and the image guide proves to be incompatible when the designer checks the interface compatibility by using the **compatibility** method, because the image diameter of the image guide and those of the DMD are different. Such incompatible interfaces indicate that the image guide and the DMD cannot connect with each other directly and it further decomposes into a lenses system and two sub-interfaces so that the system architecture is improved. The **compatibility** method ensures early testing and avoids numerous design iterations and changes. The case study demonstrates that the common representation of interface proposed by the interface model can well support the multidisciplinary collaboration during the detail design phase.

6. Conclusion

Systems engineering can help the designers to transform approved customers' needs, and requirements, into a system design solution, and the V-model is a way to describe the systems engineering and development process. However, it does not propose an effective approach to support the multidisciplinary collaboration during the detail design phase. The paper proposed an interface model to provide a common representation for the interfaces defined by different disciplines. And a variant of V-model based on this interface model is then introduced, which can be used as an effective guide to help the different design teams to ensure an early testing and avoid numerous design iterations and changes so as to achieve the multidisciplinary collaboration during the detail design phase. ■

REFERENCES

- Blanchard, B. S. 2004. *Systems Engineering Management*. Upper Saddle River, US-NJ: Prentice Hall.
- Carryer, J. E., M. Ohline, and T. W. Kenny 2011. *Introduction to Mechatronic Design*. Upper Saddle River, US-NJ: Prentice Hall.
- Dupont, E., F. Lamarque and C. Prelle. 2011. "3D triangulation system based on out-of-axis aperture configuration for micro-scaled objects shape measurement." Paper presented at the International Society for Optical Engineering. San Francisco, US-CA, 22 January.
- Forsberg, K. and H. Mooz. 1998. "Systems Engineering For Faster, Cheaper, Better." Paper presented at the 9th Annual International Symposium of INCOSE, Vancouver, CA, 26–30 July.
- INCOSE. 2015. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Hoboken, US-NJ: John Wiley & Sons.
- Liang, V. C. and C. J. J. Paredis. 2004. "A Port Ontology for Conceptual Design of Systems." *Computing and Information Science in Engineering* 4(3): 206–217.
- United States Department of Transportation (DOT). 2007. Systems Engineering for Intelligent Transportation Systems. Washington, US-DC:DOT
- Zheng, C., M. Bricogne, J. Le Duigou, and B. Eynard. 2014. "Survey on Mechatronic Engineering: A Focus on Design Methods and Product Models." *Advanced Engineering Informatics* 28(3): 241-257.

A Design Methodology and Representation Formalism for Changeable Systems – Application to Manufacturing Systems

Nadège Benkamoun, nbenkamoun@ifma.fr; **Khalid Kouiss,** Khalid.Kouiss@ifma.fr; **Carey Dilliott,** Carey.Dilliott@faurecia.fr; **Philippe Ducreuzot,** Philippe.Ducreuzot@faurecia.fr; **Jean-Philippe Marcon,** Jean-Philippe.Marcon@faurecia.fr; **Anne-Lise Huyet,** anne-lise.huyet@ifma.fr; **and Michel Dhome,** Michel.Dhome@lasmea.univ-bpclermont.fr

Highly changing environments of systems raise the challenge for increasing their lifecycle. The systems need to be changeable in order to respond to new and changing needs. As an example from the industrial engineering field, manufacturing systems are more and more required to adapt to highly changing requirements coming from new product variants, fluctuation of production volumes, product family evolution, new process plans, new technology, or merely new strategic decision. Changeability can be simply defined as the degree to which a system is able to adapt to changing circumstance. Changeability is however dependent on the design process capability to formalize, propagate, and resolve changes within systems engineering elements. According to a systemic view, changeability shall then not only be tackled in terms of physical possibilities, but also applied to any artifact delivered by the systems engineering process from conceptual design to detailed design.

This work suggests a comprehensive formalism that aims to rationally trace dependencies between systems engineering elements. This rationale-based architecture is a new representation for studying change propagation. In this context, we propose a changeability requirements profile to enable the elicitation of changing sources in the architecture. The illustration aspects are from a real manufacturing systems design project performed in the automotive supplier company Faurecia.

FRAMEWORK FOR PROGRESSIVE AND ITERATIVE DESIGN PROCESS

Framework Principles

We first introduce the design process framework on which the presented rationale relationship formalism will rely. Based on systems engineering and design theory principles we formalized a progressive and iterative design framework, Figure 1. It is iterative because design is a constant "zigzagging" flow between problem domain and solution domain. At the same time, it is progressive from general systems-level artifacts to detailed components. The design framework is composed of three main activities: requirement analysis, identification of system elements, and architecture design (in the sense of a structural organization of those elements). The presented activities continuously exchange information from generic ones to detailed ones according the described process. We define the initial needs during the problem definition activity, formalize them as requirements, analyze and derive them into new requirements within the requirement analysis activity, and then document them as system requirements. We allocate system requirements to structural elements during the design system elements synthesis activity. Finally, the core of the synthesis process takes place at the architecture design phase where the system architect organizes and arranges structural elements in order to form a consistent architecture, also driven by requirements.

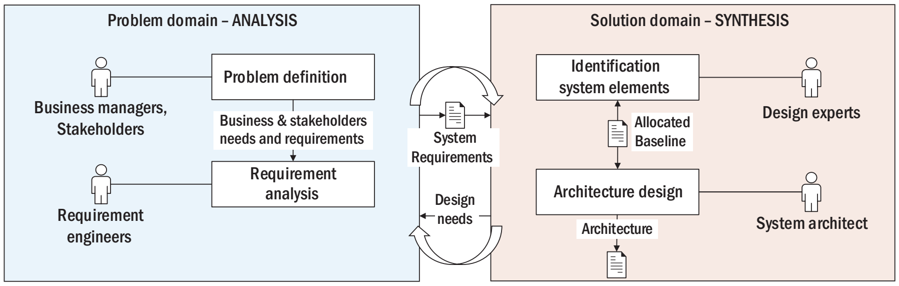


Figure 1. Design framework for progressive and iterative design process

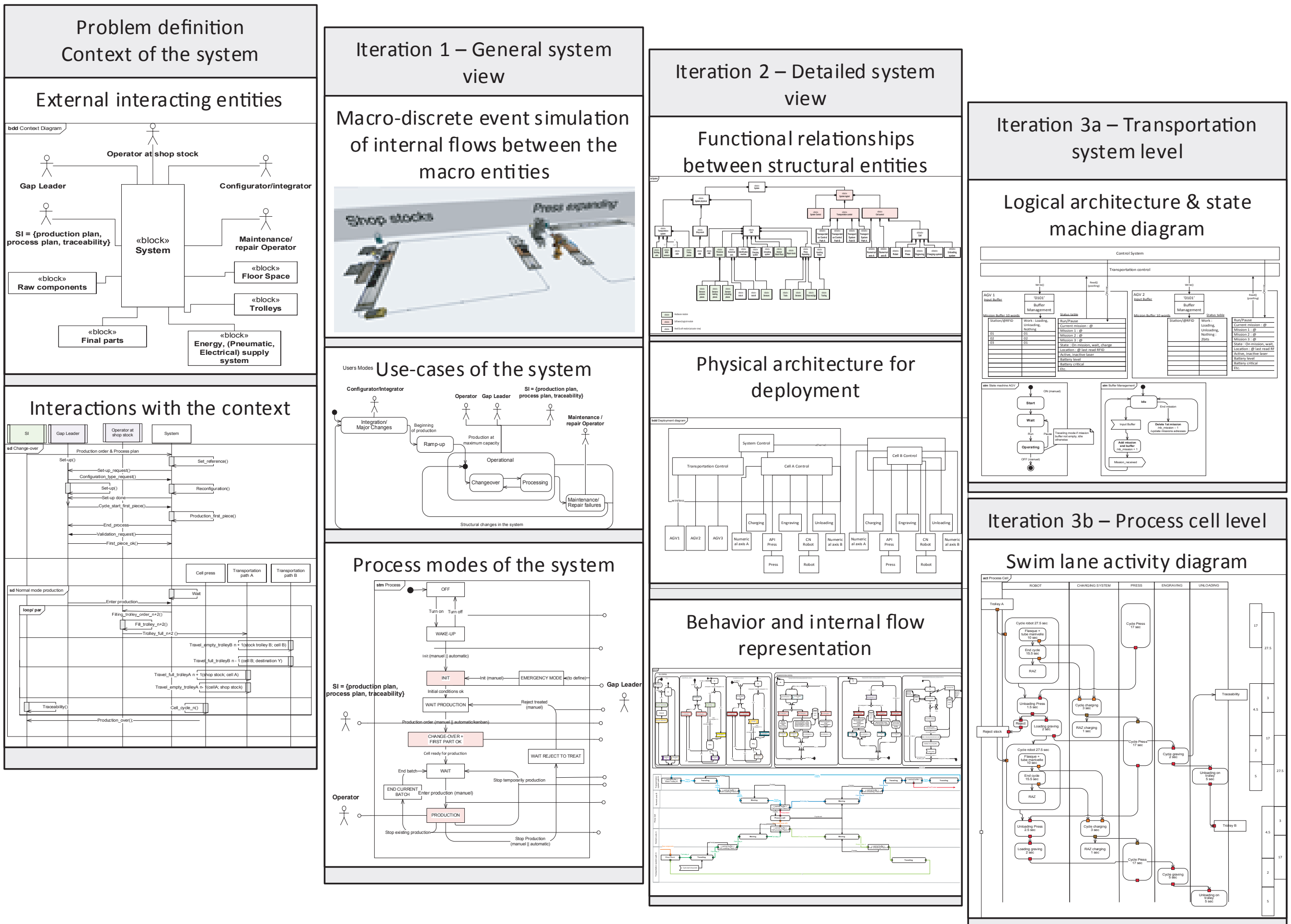


Figure 2. Application of the design framework to manufacturing system design case study

Illustration From a Manufacturing Cell Design Project for an Automotive Supplier

Figure 2 illustrates the successive process iterations on an industrial project. In short, the systems we were to (re) design were a shop stock for raw parts, a manufacturing cell composed of a robot, a press and an identification machine, and a transportation system, AGV (Automated Guided Vehicles), between them. The first iterations of the design process through increasingly detailed viewpoints: general systems view (iteration 1), detailed systems view with the generic entities of the systems (iteration 2) and design at the transportation systems level (iteration 3a) and the process cell level (iteration 3b). The process encompasses MBSE (model-based systems engineering) approaches through SysML and domain-dependent models (discrete event simulation software, robotic cell simulation). It helped to guarantee the consistency of the design process among the different stakeholders and actors from different disciplines (manufacturing process experts, robotic engineers, automation engineers, integrators, suppliers, logistic experts, automated guided vehicles suppliers, traceability experts, and ergonomic experts).

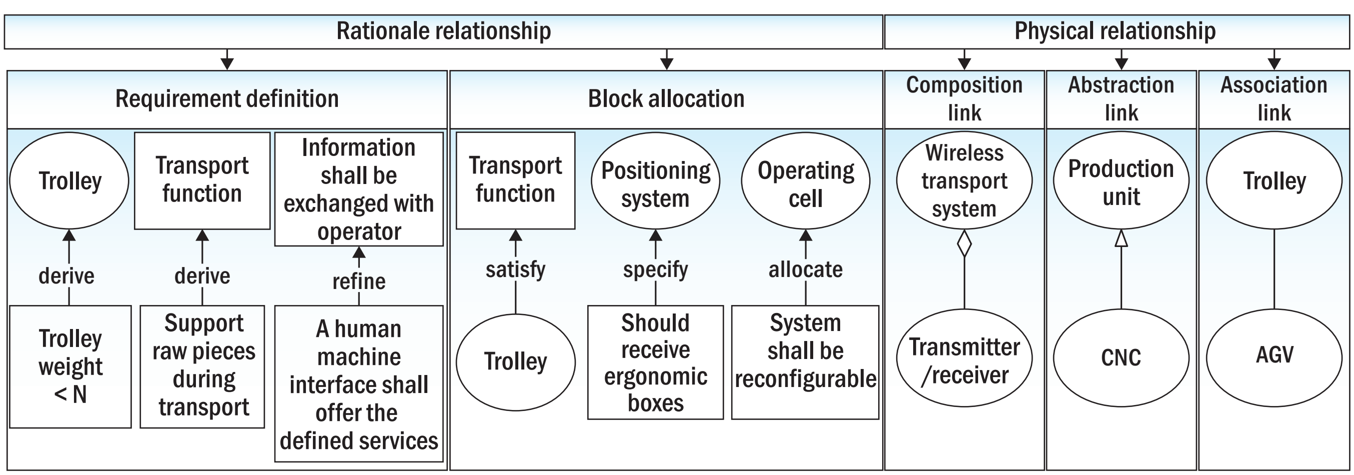


Figure 3. Rationale-based architectural elements relationship formalism

RATIONALE-BASED ARCHITECTURE FOR CHANGEABILITY REPRESENTATION

We defined two architectural views through the earlier defined framework: a solution domain architecture and a rationale-based architecture that traced dependency links between the involved systems engineering elements. The first one originates from the architecture design activity where structural elements together model what the system shall be for the implementation phase. The second is the overall result of the decision process where engineers design and trace requirements and structural elements successively. Several works, from rationale design (Regli et al. 2014) to requirements engineering, aim at tracing the designer intention in order to understand it, reuse it, and track change propagation in the complex project system. With the same motivations, the ontology in Figure 3 enables traceability of the successive design elements (requirements of the problem domain and structural blocks of the solution domain). It deepens the SysML formalism (Friendenthal, Steiner, and Moore 2011) of relationships, especially between structural blocks and requirements (requirement <derives> from block, a block <satisfies> a requirement, a requirement <allocates> to a block). It also integrates block-to-block relationships in the same systems architecture representation. Continually with the design process, engineers formalize rationale-based architectures. We present some relationships in Figure 3.

NEW REQUIREMENT STEREOTYPE ON THE CHANGEABILITY LEVEL OF DESIGNED ARTIFACTS

In order to initially capture requirements for changeability, and thus to encourage designers to think in a future-oriented way, a profile for changeability requirements is defined. This profile is manufacturing-domain dependent, as it lists the potential change drivers that may apply to manufacturing systems. Requirements stereotype follows the classification of change drivers and examples from the earlier presented industrial context.

- <Product family change stereotype> The robot gripper shall be changed
- <Product variant change stereotype> The AGV shall transport different type of parts for different product variant
- <Product volumes change stereotype> The transportation control shall be scalable from one to several AGVs
- <Layout change stereotype> The AGV shall be able to travel in no-planned path
- <Manufacturing process change> The press has changeable tools and fixtures
- <Logistic process change stereotype> The system shall be independent to the input flow: the parts could be brought by type or kits
- <Standard change stereotype> The identification control shall be independent from the chosen technology
- <External strategic motivation stereotype> The internal stock place shall be optional

PERSPECTIVES

Designing changeability according to the functionalities, the requirements or largely, the intentions a system answers, is more meaningful than an operational or physical description. That is why tracing design rationale is necessary prior work to offer a new formalism, based on suggested design framework for iterative and progressive design process. Using requirement stereotypes for changeability, engineers can formalize and apply sources for changeability to any element of the initial rationale architecture. The future work perspectives are now to guide designers into design for changeable solutions and for changeability. In response to changeability requirements, the rationale-based architecture shall embody structural characteristics like modularity and the ability to interface. ■

REFERENCES

- Friendenthal, S., R. Steiner, and A. Moore. 2011. *A Practical Guide to SysML: The Systems Modeling Language*. Philadelphia, US-PA: Elsevier.
- Friorèse, P., and J.-P. Meinadier. 2012. *Découvrir et Comprendre l'Ingénierie Système*. Toulouse, FR: Cépaduès Editions.
- Koren, Y., and M. Shpitalni. 2010. "Design of Reconfigurable Manufacturing Systems." *Journal of Manufacturing Systems* 29 (4): 130-141.
- Regli, W. C., X. Hu, M. Atwood, and W. Sun. 2014. "A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval." *Engineering with Computers* 16 (3-4): 209–35.

This work has been sponsored by the French government research program Investissements d'avenir through the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01), by the European Union through the program Regional competitiveness and employment 2007-2013 (ERDF–Auvergne region) and by the Auvergne region.

Tenure-Track Faculty Positions | Pennsylvania State University

The Harold and Inge Marcus Department of Industrial and Manufacturing Engineering at The Pennsylvania State University invites applications for two full-time, tenuretrack positions at the Assistant Professor level. Exceptional candidates will also be considered for higher ranks. We seek outstanding individuals working in traditional and/or emerging areas of Industrial Engineering. The department is especially interested in the areas of advanced manufacturing processes, manufacturing systems, simulation and stochastic processes. Candidates are expected to teach undergraduate and graduate courses, establish and lead an innovative research program, supervise graduate students, pursue external funding opportunities, and provide service to the University. The first industrial engineering department founded in the world, our department is consistently highly ranked by US News & World Report for both its graduate and undergraduate programs. Interested candidates are encouraged to visit the department (http://www.ie.psu.edu) and college (http://www.engr.psu.edu) websites to learn more.

The start date is Fall 2016. Applicants must have a Ph.D. in Industrial Engineering, Operations Research or a closely related field and have excellent verbal and written communication skills. Interested applicants should submit: (1) a cover letter, (2) a Curriculum Vitae, (3) a one-page research statement, (4) a onepage teaching statement, and (5) a list of 3-5 references. Applicant screening will begin on January 11, 2016 and continue until the position is filled. Apply at http://apptrkr.com/698944 Questions can be directed to the faculty search chair by email to IE-FacultySearch@engr.psu.edu

CAMPUS SECURITY CRIME STATISTICS: For more about safety at Penn State, and to review the Annual Security Report which contains information about crime statistics and other safety and security matters, please go to http://www.police.psu.edu/ clery/ , which will also provide you with detail on how to request a hard copy of the Annual Security Report.

Penn State is an equal opportunity, affirmative action employer, and is committed to providing employment opportunities to all qualified applicants without regard to race, color, religion, age, sex, sexual orientation, gender identity, national origin, disability or protected veteran status.

A Method for Formalizing Requirements Interoperation in Complex Systems Engineering

Anderson Luis Szejka, Anderson.Szejka@pucpr.br; **Alexis Aubry,** Alexis.Aubry@incose.org; **Hervé Panetto,** Herve.Panetto@incose.org; **Osiris Canciglieri Júnior,** Osiris.Canciglieri@pucpr.br; **and Eduardo Rocha Loures,** Eduardo.Loures@pucpr.br

Systems engineering is "*an interdisciplinary approach and means to enable the realization of successful systems*" that must meet and satisfy the needs of its stakeholders, based on concepts from systematic paradigms, methods and standardized processes (BKCASE 2015, ISO/IEC 15288 2015, ISO/IEC 29148 2011). The continuing evolution of systems engineering comprises the consolidation, identification, and formalization of new methods and modelling techniques, which engineers apply in Product Development Engineering (PDE), Process Development (PD), and Software Development (SD). A significant amount of researchers work to formalize and standardize engineering design and to manage systems lifecycle when, in matter of fact, the attention should be on the information comprehension. Moreover, there is currently a debate regarding the semantic problems across different lifecycle phases, which occur because of information misinterpretation and misunderstanding.

Systems engineering project starts with requirements elicitation and negotiation phases in which the main objective is to translate initial customers' needs into functional and non-functional stakeholder requirements. Requirements are statements from the customers' needs to identify and constrain a product, system, or process (BKCASE 2015). They must be unambiguous, clear, unique, consistent, stand-alone, measurable, verifiable, and traceable, that is, SMART (Specific, Measurable, Attainable, Realizable, Time bounded), requirements (Mannion and Keepence 1995). Requirements are the entry point for the process production of specification analysis to a solution system with verified and validated solutions that meet initial needs. Technical information enriches functional and non-functional requirements, transforming into technical functional or non-functional requirements across different phases of systems lifecycle. However, this transformation involves, throughout its systems lifecycle, a set of heterogeneous knowledge, information, and expertise. This results in a semantic gap between the initial requirements and the requirements taken into account during the lifecycle, compromising the results of solution systems. This semantic gap comes from the non-interoperability of information within each requirement.

IEEE (1990) defines interoperability as "*the ability of two or more systems or components to exchange information and to use information that has been exchanged without special effort*." In terms of categorization, interoperability has three levels (Panetto and Molina, 2008; EIF, 2004) as follows:

(i) Technical Interoperability (TI) – it concerns technical properties, enabling machine-to-machine communication to take place. TI is usually associated with hardware and software components, systems and platforms (data and protocols format, physical characteristics, material resistance);
(ii) Semantic Interoperability (SI) – it concerns the real meaning of content that is shared and understandable by any other application;
(iii) Organizational Interoperability (OI) – it is the ability of organizations to effectively communicate and transfer (meaningful) data (information) even though they may use a variety of information systems over widely different infrastructures in or out of enterprises boundaries.

SI is feasible when the meaning associated with the captured information and knowledge flows effectively across different workgroups without any loss of meaning and knowledge (Chungoora et al. 2013). This occurs through the construction of formal domain ontologies (Gruber 1995; Provine et al. 2004; Jovanovic and Gasevic 2005; Noy and Rubin 2008), implemented to different fields such as engineering, medicine, business, and more. Applied to systems engineering, the research questions thus concern "*How heterogeneous information related to requirements can be formalised regarding multiple knowledge domains to provide support during different phases of systems development lifecycle?*" and "*What are the formalised relationships between systems requirements related to multiples domains and impacting in different phases of lifecycle?*"

This article presents an ongoing research project that aims to define a conceptual method to formally model the systems requirements and their relationships in terms of transformation (translation, conversion, and sharing) and traceability, based on an ontological approach. In this way, we are considering two hypothesis: *(H1) Multiple phases of systems lifecycle can be supported by systems requirements in a semantically interoperable manner; and (H2) Systems requirements formalization can ensure the comprehensibility and verifiability, reducing inconsistencies between different domains across the phases of systems lifecycle*.

DECEMBER 2015
VOLUME 18 / ISSUE 4

The approach needs to discover and identify the dependence relationship during any systems engineering. Thus, according to ISO/IEC 15288, 2008, ISO/IEC 29148, 2011 and discussion provided in Szejka et al. (2014), three perspectives are necessary to establish the systems-project: (i) the domain of application; (ii) the systems lifecycle phase; and (iii) the requirements constraining the studied systems. The first perspective concerns the set of heterogeneous knowledge and expertise involved during the systems engineering, including the mechanical expertise, electrical expertise, IT expertise, and so on. The second perspective refers to different phases of systems lifecycle, where each phase has its proper constraints and information. The last perspective considers the consistency of the relationships between requirements since a specific requirement is dependent on one or multiple domains and one or multiple phases of lifecycle.

Based on this context, we can directly identify three interoperation issues. The first concerns the heterogeneity of information coming from multiple domains. It imposes some knowledge representation and analysis for managing requirements and their semantic relationships then, this is associated to hypothesis H2. The second interoperation issue concerns the systems lifecycle phases and the possible impacts between some requirements associated to different lifecycle phases. The requirement definition can impact on other requirements and then, it is necessary to manage and to ensure the consistency of those requirements, so this relates to hypothesis H2. The last interoperation issue concerns the relationships between requirements and their properties of completeness, coherency, uniqueness, univocity, and traceability that it correlates to the hypothesis H1.

Currently, there are a significant number of researchers that are working in this area (Ratchev et al. 2003; Baudry et al. 2007; Canciglieri Jr. and Young 2010; Chungoora et al. 2013). However, these teams focus their research on specific points to transform information between domains and/or single phases of systems lifecycle and/or requirements. In particular, requirements interoperation is the core concern of industry when they are engineering a system (Micouin 2008; Bernard 2012). The main issue is to pragmatically formalize the requirements interoperation, based on ontological models, considering the tacit knowledge related to the processes involved in systems engineering. Thus, specialists typically define requirements for project-systems using some informal document written in natural language (NL), because of the expression richness provided by it (Bryant 2000). They focus on establishing the references for the needs of systems engineering, in order to ensure the requirements completeness, consistency, and coherency.

The main result of this research is to transform functional and non-functional requirements, written in natural language (informal requirements), into formal requirements (Figure 1). It is based on formal Common Logic (CL) (Pan and Liu 2010; Jarrar 2007) and ontology application for transforming requirements written in

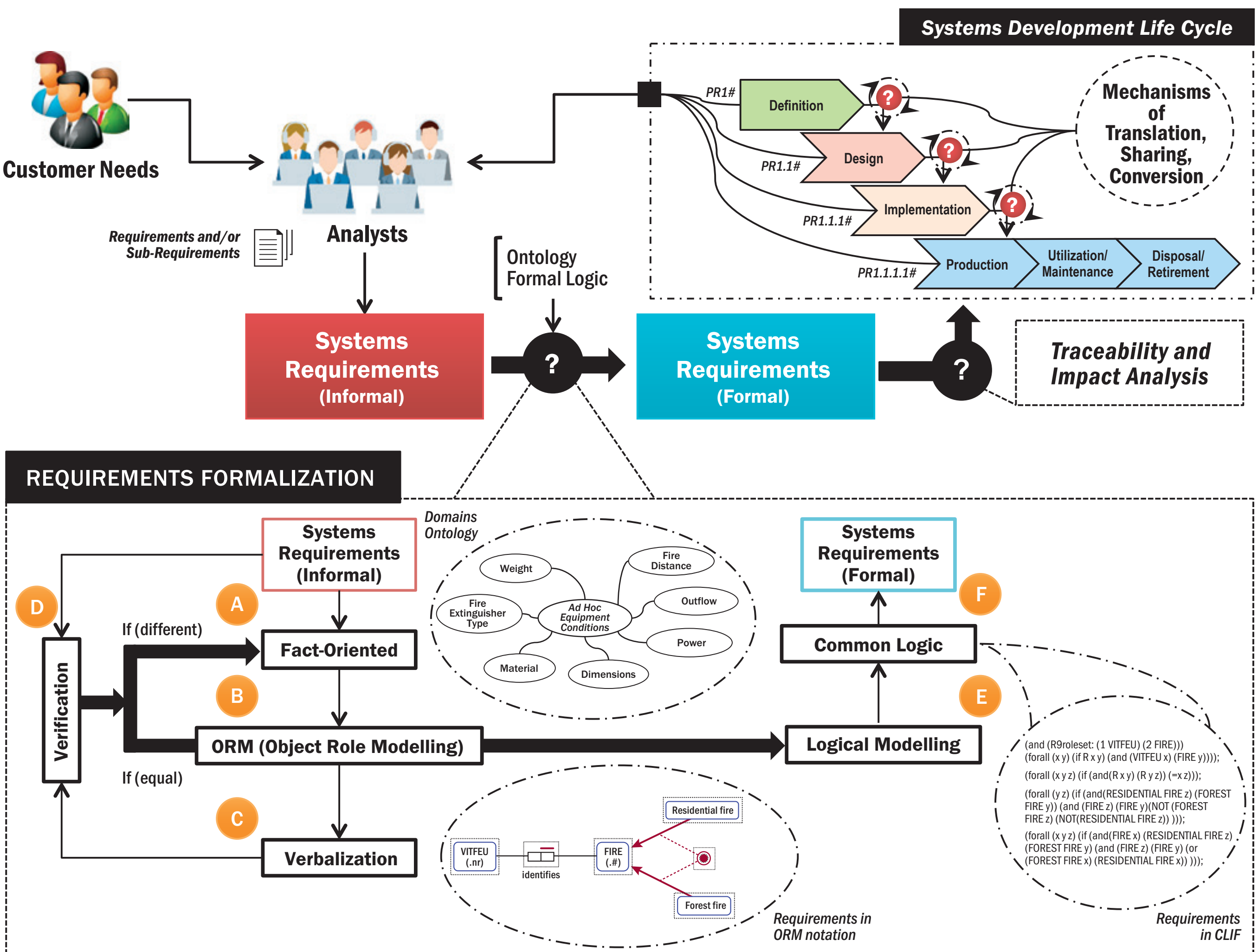


Figure 1. Requirements formalization approach

natural language into requirements written in formal language.

The proposed method is comprised of sub-methods and procedures that they perform in a semi-automatic manner. First, requirements written in natural language are split in 'subject + verb + complement' for building a fact-oriented model (FOM) (Halpin 2006). The requirements are analysed extracting the facts of interest or concepts ('subject' / 'complement') and relationships ('verb') between these facts (Detail A – Figure 1). These relationships can be unary, binary, or ternary (Halpin 2006). Sequentially, engineers must model these facts in a formal logic in order to structure these requirements. This fragmentation puts in a simpler manner the information within each requirement.

You can use different approaches to structure this fact-oriented model such as: Object Role Modelling (ORM) (Detail B – Figure 1) and Cognition enhanced Natural language Information Analysis Method (CogNIAM) (Bollen 2014). However, the method chosen must be able to model and verbalize the requirements, since it is important to compare the requirements model structured and the requirements originally defined (Detail C – Figure 1). Therefore, it allows the identification of any discrepancy between the model and real desires (Detail D – Figure 1). Although these approaches rely on structured language, they are informal languages. Pan and Liu (2010) mapped some facts from Object Role Modelling into First-Order Logic (FOL). The latter has powerful expression to represent complex rules in a formal way. Some analysed research (Jarrar 2007) provide methods to model those approaches in Description Logic or Common Logic that are based on First-Order Logic, and both methods have an inference machine. Inference machine is important to classify information and query answering. The presented method is based on Common Logic through mapping constraints enriched based on Common Logic Interchange Format (CLIF) (Detail E – Figure 1). This formal structure links with other requirements creating a knowledge model and possibilities of applying traceability methods and conflicts identification methods between requirements (Detail F – Figure 1).

As a conclusion, the presented approach demonstrates that there are significant lacks of interoperability between requirements in systems engineering due to heterogeneity of information from multiple domains and different phases of systems lifecycle. We identified three interoperation issues cross-domains, cross-systems lifecycle, and cross-requirements. These interoperation issues have a direct impact in design success, because they can cause mistakes and misinterpretation with requirements. Moreover, the need exists for a method dealing with requirements interoperability across different phases of the lifecycle and domains. This method aims to formally model requirements interoperation in terms of transformation, traceability, and conflicts analysis. The goal is now to enrich this method and evaluate its performance with different case studies. ■

REFERENCES

- Baudry, B., C. Nebut, and Y. Le Traon. 2007. "Model-Driven Engineering for Requirements Analysis." In *Enterprise Distributed Object Computing Conference, 2007.* 11th IEEE International: 459-467.
- Bernard, Y. 2012. "Requirements Management Within a Full Model-Based Engineering Approach." *Systems Engineering* 15 (2) 119-139.
- BKCASE, Editorial Board. 2015. "The Guide to the Systems Engineering Body of Knowledge (SEBoK)", Version 1.3.2, R.D. Adcock (EIC), Hoboken, US-NJ: The Trustees of the Stevens Institute of Technology.
- Bollen, P. 2014. "Fact-Based Semantic Modelling in the Information and Behavioural Perspectives." In *OTM 2014 Workshops* LNCS 8842: 663-666.
- Bryant, B. R. 2000. "Object-Oriented Natural Language Requirements Specification." In *Proceedings of Computer Science Conference*. Canberra, ACT: 24–30.
- Canciglieri Jr., O. and R. I. M. Young. 2010. "Information Mapping Across Injection Moulding Design and Manufacture Domains." *International Journal of Production Research* 48 (15): 1565–1586.
- Chungoora, N., R. I. M. Young, G. Gunendram, C. Palmer, Z. Usman, A. Cutting-Decelle, J. Harding, and K. Case. 2013. "A Model-Driven Ontology Approach for Manufacturing System Interoperability and Knowledge Sharing." *Computers & Industry* 64 (4): 392–401.
- EIF, European Interoperability Framework for pan-European eGovernment Services. 2004. "Interoperable Delivery of European eGovernment Services to public Administrations". Business and Citizens (IDABC). Brussels, BE: 1-25.
- Gruber, T. R. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." *International Journal of Human-Computer Studies* 43 (5–6): 907–928.
- Halpin, T. 2006. "Object-Role Modelling (ORM/NIAM)." In *International Handbooks on Information Systems*, edited by P. Bernus, K. Mertins, and G. Schmidt, 81-103. New York, US-NY: Springer.
- IEEE (Institute of Electrical and Electronics Engineers). 1990. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries.
- ISO (International Organization for Standardization) and IEC (International Eletrotechnical Commission). 2011. ISO/IEC 29148. "System and software engineering – Life cycle processes – Requirements Engineering". Geneva, CH: ISO/IEC.
- ——. 2015. ISO/IEC 15288. "Systems and software engineering – System life-cycle processes." Geneva, CH: ISO/IEC.
- Jarrar, M. 2007. "Towards Automated Reasoning on ORM Schemes Mapping ORM Into the DLRidf Description Logic." In *Conceptual Modeling* – ER2007, Proceedings 26th International Conference on Conceptual Modeling, November 5-9, edited by C. Parent, K.-D. Schewe, V. C. Storey, and B. Thalheim, 4801: 181-197. Auckland, NZ: Springer.
- Jovanovic, J., and D. Gasˇevic. 2005. "Achieving Knowledge Interoperability: an XML/XSLT Approach". *Expert Systems with Applications* 29 (3): 535–553.
- Mannion M. and Keepence, B. 1995. "SMART Requirements," *Software Engineering Notes* 20 (2): 42–47.
- Micouin P. 2008. "Toward a Property Based Requirements Theory: System Requirements Structured as a Semilattice," *Systems Engineering* 11 (3): 235–245.
- Noy, N.F. and Rubin, D.L. 2008. "Translating the Foundational Model of Anatomy Into OWL." *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (2):133-136.
- Pan, W. and D. Liu. 2010. "Mapping Object Role Modelling into Common Logic Interchange Format." In *3rd International Conference on Advanced Computer Theory and Engineering* (ICACTE) 2: 104–109.
- Panetto, H. and A. Molina. 2008. "Enterprise Integration and Interoperability in Manufacturing Systems: Trends and Issues." *Computers in Industry* 59 (7): 641-646.
- Provine, R., C. Schlenoff, S. Balakirsky, S. Smith, and M. Uschold,. 2004. Ontology-based Methods for Enhancing Autonomous Vehicle Path Planning. *Robotics and Autonomous Systems* 49 (1/2): 123–133.
- Ratchev, S., E Urwin, D. Muller, K. S. Pawar, and I. Moulek. 2013. "Knowledge Based Requirement Engineering for One-of-a-Kind Complex Systems." *Knowledge-Based Systems* 16 (1): 1–5.
- Szejka, A. L., A. Aubry, H. Panetto, O. Canciglieri Jr., and E. R. Loures. 2014. "Towards a Conceptual Framework for Requirements Interoperability in Complex Systems Engineering." In *OTM 2014 Workshops* LNCS 8842: 229-240.

A Tooled Approach for Designing Executable and Verifiable Modeling Languages

DECEMBER 2015
VOLUME 18 / ISSUE 4

Blazo Nastov, blazo.nastov@mines-ales.fr; **Vincent Chapurlat,** vincent.chapurlat@mines-ales.fr; **Christophe Dony,** dony@lirmm.fr; **and François Pfister,** francois.pfister@mines-ales.fr

INTRODUCTION

Model-based systems engineering (MBSE) is "*the formalized application of modeling* [model-centric approach opposed to document centric approach] *to support systems requirements, design, analysis, and verification and validation activities*" (Estefan 2008). It steers experts creating, checking, and analyzing various models of the System of Interest (SoI), each one specifically designed to reach a given modeling objective (analysis of system behavior and performances, safety or other non-functional SOI's properties grouped under the term '–ilities'). A model focuses on a given aspect (functional, logical, physical, or behavioral) detailing a given point of view (user, designer, or other) of the SoI. Engineers create the model using general modeling languages, such as UML or SysML, or by using Domain Specific Modeling Languages (DSML) dedicated to an aspect of the SoI.

In this context, it is crucial, prior to any analysis, to conduct verification (demonstrating the consistency and conformity of a model to its modeling language, rules and patterns), and if possible, validation activities (demonstrating the relevance of a model and its level of respect to stakeholders' requirements), taking into consideration, first each model separately, and then pieced together with the other models of the same SOI (demonstrating the mutual coherence of SOI models, as well as their conformity and adequacy to the modeler's objectives). For this purpose, engineers transform the models into "verifiable" third party models for which model checking (UPPAAL, STEP), theorem proving, or simulation techniques and tools are available, despite the several well-known drawbacks, highlighted hereafter. Models transformation techniques require first expertise and knowledge in the target semantic domain, in transformation languages and tools, for example, ATL (Atlas Transformation Language). Second, demonstrating the "equivalence" between the original, to-be-checked-model, and the transformed model remain limited, often impossible. Finally, engineers should interpret results obtained in the transformed model back according to the used transformation rules and concepts of the original model.

Alternatively, a model transformation can be avoided if the used DSML is sufficiently formal, that is, if the semantics of concepts and relations on which the DSML is based have been non-ambiguously defined. Indeed, DSML definition is an abstract and a concrete syntax promoting concepts and relations between concepts more or less constrained, and generally lacking the specification of semantics. This becomes a major obstacle and limitation for formal verification of models.

Semantics can either be *static* describing only the meaning, independently from the behavior or *dynamic* describing the behavior (Nastov 2014). The abstract and concrete syntaxes of the DSML give static semantics implicitly and partially. Dynamic semantics require additional explaination by formally defining how each DSML concept and relation can evolve taking into consideration temporal evolution rules, events, and data configurations (Nastov et al. 2014). There are several widely accepted tools and approaches for defining static semantics such as EMF/Ecore (for abstract syntax) or Sirius (for graphical concrete syntax), and dynamic semantics such as Object-Oriented languages, for example: Java, Aspect-Oriented languages, like Kermeta (Muller et al. 2005) or executable constraint languages, such as xOCL. Nevertheless, they are related to software programming languages (imperative, object-oriented or aspect-oriented languages) and are still unfamiliar to both modelers and DSML designers in the MBSE context. Indeed, dynamic semantics is to be modeled (not programmed) and formalized with minimal efforts by DSML designers and then used by modelers, by assisting them and automating as much as possible the design of DSMLs and models, and verification and validation (V&V) activities. For this purpose, formal behavioral languages such as Statecharts, Petri Nets, or Finite Automata, can be used. Expected behavior of DSML concepts is described through behavioral models that are furthermore used to simulate models created by using this DSML (each behavioral model executes instances of its corresponding concept). Behavioral models must be checked for construct, coherence, and conformity to DSML metamodel or simulated in order to assist DSML design and V&V activities. However, several questions remain an open issue. For instance, various dynamic semantics of

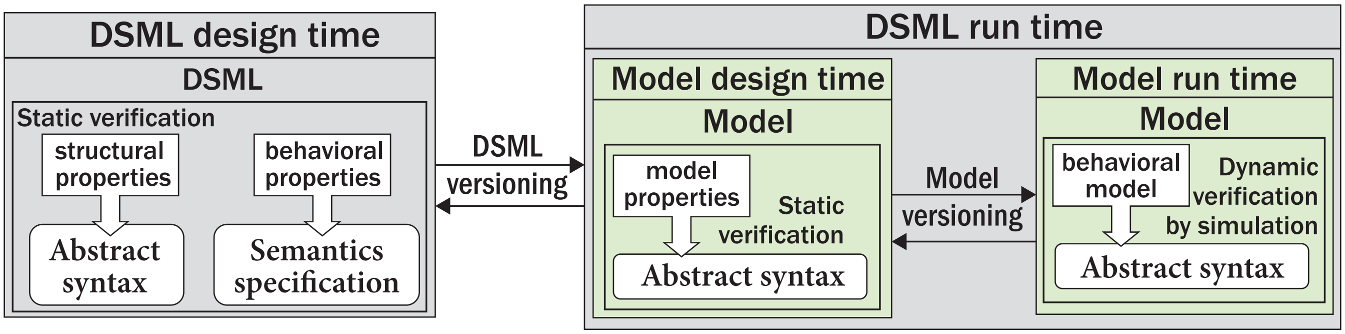


Figure 1. Design time and run time steps

proposed formal behavioral languages exist which unfortunately, may be interpreted differently. Dynamic semantics description remains often limited due to required effort to use such formal languages.

In this paper, we propose a formal approach for modeling dynamic semantics of DSMLs concepts and relations making the concept of executable, verifiable, and interoperable DSML (xviDSML). We aim at reducing the time spent by DSML designers or modelers with third party approaches and to improve without additional effort the model's coherence and relevance. Our approach provides a standardized and a model-based way for specifying the behavior of each concept and relation that can be simulated. In addition we propose a property modeling language that provides the means for formally specifying concept's or relations properties, structural or behavioral, allowing formal verification of the DSML or the models.

EXECUTABLE, VERIFIABLE AND INTEROPERABLE DSMLS: XVIDSML

Among concepts requested in a DSML, some can be chosen to take part in the description of the behavior of the DSML, called "evolving concepts" (Combemale et al. 2012). The expected behavior of each concept, that is the *concept behavioral model* (discussed above) is classically designed using software engineering languages (Kermeta). On the contrary, our approach aims at modeling and not programming dynamic semantics using a formal behavioral language described hereafter. A property modeling language compatible with the proposed behavioral language is also included in the approach, improving simulation and verification capabilities of DSMLs.

Behavioral language overview

The behavioral modeling language used to design *concept behavioral model* is an extension of the Finite Sequential Machine called Interpreted Sequential Machine (ISM) (Larnac et al. 1997). The choice of the ISM is motivated by several advantages it proposes, in comparison to other languages. First, it operates with typed input/output data (primitive type, for example, Boolean, Integer, Real, Character or compound type) and complex expressions built using internal typed data. All concepts from the DSML can be naturally used as a source of data. Second, it separates classical state/transition model here called Control Part (CP) from data specification here called Data Part (DP). This allows it to replace the specification of some states (to be added normally to the CP) as "symbolic" variables in the DP, limiting the combinatorial explosion of CP. Third, ISM has formal underlying structure, formalized in Temporal Logic (TL), allowing formal verification by the use of temporal valid formula. *Concept behavioral model* properties can then be described using TL and checked by using model checking techniques and tools (STEP, MEC, TINA or UPPAAL). In addition, the Temporal Boolean Difference (TBD) can be used to highlight the sensitivity of the present on the future evolution of ISM as proposed in Vandermeulen et al. (1995). Fourth, inputs set denoted I and outputs set denoted O, enable data exchange between various ISMs via a central mechanism called Black Board (BB). This allows a certain level of interoperability between DSMLs, that is, to centralize data exchanges and assume temporal synchronization rules by managing an Execution Scheduler (ES) module that take into account a logical "environmental" time scale. A proposed evolution algorithm allows us to interpret formally, to provide a set of execution and simulation mechanisms of an ISM without ambiguity. Finally, this evolution algorithm is managed by a Controller associated to each ISM independently authorizing parallel execution of *concept behavioral models* and taking into account multiscale time and stability management hypothesis.

Property modeling language overview

A property modeling language meta-model is currently designed and tooled as a concretization of the property modeling language CREI (Causes Relation Effects Indicators) (Chapurlat 2013). This language allows us to design and verify different types of formal properties depending on the current phase of the DSML design or of model lifecycle, as illustrated in Figure 1, allowing a formal V&V strategy based on properties proof.

First, during DSML design time, engineers design DSMLs by the means of an abstract syntax, a concrete syntax, and a behavioral specification. Two types of properties get specified and checked: *structural properties*, concerning the abstract syntax, or *behavioral properties* concerning the behavioral specification. DSML design time supports only "static" verification, ensuring the well-constructed-ness of a DSML, the well-constructed-ness of its abstract syntax and of its behavioral specification. During the model design time, engineers design models by the means of their abstract syntaxes, allowing the specification of *model properties*. We distinguish two types of model properties, one called *model properties*, specified for the current model, and the other named *system properties*, specified for a grid of models that represent the SoI. Finally, yet importantly, the engineers can use the simulation during the model run time. During this phase, the behavioral rules specified by the dynamic semantics specification of the used DSML are used as interpretation rules, in order to simulate created models. Such simulation is based on the gradual computation or the interpretation of these rules allowing model animation that is achieved as a result to the systematic visualization of changes driven by the computation of the interpretation rules.

CONCLUSION: THE XVIDSML DESIGN PROCESS

The proposed approach, under tooling and tests on Eclipse platform, for designing an xviDSML is based on the composition of three languages as illustrated in Figure 2: a metamodeling language, a behavioral modeling language, and a property modeling language. The underlying structure of our language is defined by a metamodel created by using the EMF's

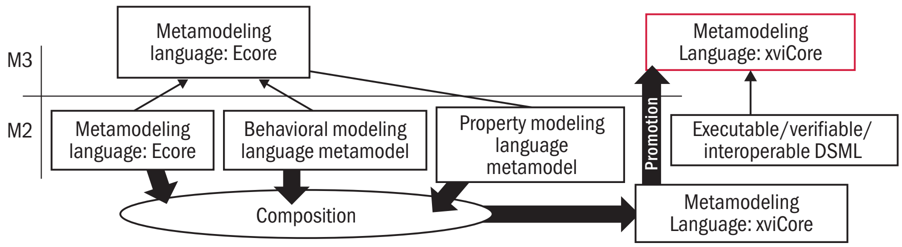


Figure 2. The xviDSML design process, inspired by Muller et al. 2005

metamodeling language Ecore (https://eclipse.org/modeling/emf/). It integrates the entire structure of the metamodeling language Ecore, the structure of the behavioral modeling language (here considered ISM), and the structure of the property modeling language (here considered CREI). Note that, Ecore is downloaded at the M2 (meta-modeling layer). The resulting DSML is furthermore promoted to the M3 (meta-meta-modeling layer) substituting the previous metamodeling language. The composition process is designed to ensure that the existing and already defined DSMLs, (conforms to the previous metamodeling language), remains fully compatible with the new metamodeling language. ■

REFERENCES

- Chapurlat,V., 2013. "Property Concept and Modeling Languages For Model-Based Systems Engineering Context (MBSE)." *Internal Research Report* (available on demand).
- Combemale, B., X. Crégut, and M. Pantel., 2012. "A Design Pattern to Build Executable DSMLs and Associated V&V Tools." In *Software Engineering Conference* (APSEC), 19th Asia-Pacific (Vol. 1, pp. 282-287). IEEE.
- Estefan, J. A., 2008. "Survey of Model-Based Systems Engineering (MBSE) Methodologies 2. Differentiating Methodologies from Processes, Methods, and Lifecycle Models." *Jet Propulsion*, 25:1–70.
- Larnac M., V. Chapurlat, J. Magnier, and B. Chenot. 1997. "Formal Representation and Proof of the Interpreted Sequential Machine Model." *EUROCAST'97*, Las Palmas, US-CA.
- Lee, E. A., 2015. "The Ptolemy Project." Home Page: http://ptolemy.eecs.berkeley.edu.
- Muller, P. A., F. Fleurey, and J. M. Jézéquel. 2005. "Weaving Executability Into Object-Oriented Meta-Languages." *In Model Driven Engineering Languages and Systems* 264-278. Berlin Heidelberg, DE: Springer.
- Nastov, B., 2014. "Contribution to Model Verification: Operational Semantics for Systems Engineering Modeling Languages." CIEL. Paris, FR.
- Nastov B., V. Chapurlat, C. Dony, and F. Pfister. 2014. "A Verification Approach from MDE Applied to Model Based Systems Engineering: xeFFBD Dynamic Semantics." In *CSD&M* 225-238. Paris, FR: Springer International Publishing,
- Vandermeulen, E., H. A. Donagan, M. Larnac, and J. Magnier, 1995. "The Temporal Boolean Derivative Applied to Verification of Extended Finite State Machine." *Computer and Mathematics with application*, 30 (2).

UNIVERSITY OF MARYLAND
MASTER OF SCIENCE
SYSTEMS ENGINEERING

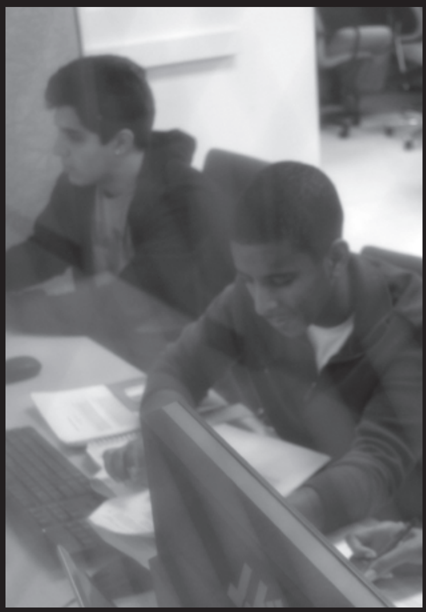
EARN THE DEGREE THAT PREPARES YOU TO:
- Create complex systems and services
- Become an effective technical leader-collaborator
- Reap the rewards of a career in systems engineering

MODEL-BASED SYSTEMS ENGINEERING
IN A SYSTEMS RESEARCH ENVIRONMENT

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

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

go.umd.edu/iimsse



INSTITUTE FOR SYSTEMS RESEARCH
A. JAMES CLARK SCHOOL OF ENGINEERING

ScOLA, a Scenario Oriented Modeling Language for Railway Systems

Melissa Issad, melissa.issad@ecp.fr; **Leila Kloul,** leila.kloul@prism.uvsq.fr; **Antoine Rauzy,** antoine.rauzy@ntnu.no; **and Karim Berkani,** karim.berkani@siemens.com

INTRODUCTION AND MOTIVATIONS

Complex railway systems rely on paper-based specifications written in a natural language. It results in lengthy and ambiguous documents integrating the systems architecture and its behavior.

The design and development of a CBTC (Communication Based Train Control) (IEEE 2004) systems rely on the requirements and specifications defined using existing systems and their additional features. Its verification and validation rely on test cases coming from, on one hand the experience of engineers, and on the other hand, the systems behaviors described in documents.

Solely, in order to comply with the European railway standards EN50126 (CENELEC 1999) and EN50128 (CENELEC 2011) maintained by the European Committee for Standardization (CEN), it is important to provide a complete analysis of the risks of all the functions and components of the systems. Moreover, test, validation, development, and safety analysis rely on scenario-based specifications.

The main reasons driving our work: First, despite its large expressiveness, natural language generates ambiguous systems specifications that are not suitable for further use on the systems v-cycle like test and safety analysis. Second, in our knowledge, in the literature, there is no methodology or tool that fits exactly the methodologies in place in the context of CBTC systems.

Languages like UML/SysML (Friedenthal et al. 2014) are very effective modeling notations; they provide a large set of diagrams (activity, sequence diagrams, and more). Still, there is a need for formalization and clarification of methodologies, so that the information generated in those diagrams is verified and validated. There have been successful attempts at setting a formal semantics to SysML (Hamilton et al. 2007, Haugen et al. 2005, Ober et al. 2010). But, they were too specific to a particular domain. This leads us to define our own semantics for systems specifications.

Thus, after a thorough study of the systems specifications of CBTC systems, we define ScOLA (Issad et al. 2014), a novel formal modeling language based on scenarios concept. This language allows us

▸ Enforce Train Door Release

The on-board subsystem provides an enforce train door release operation. With this operation, door release can be enforced either on the left side, on the right side, or both sides. This function is for instance, intended to enable door opening in case the train has not stopped correctly.

<table>
<tr><td>Identifier</td><td colspan="5">#REQ-AS_TGMT_R2-enforce_door_release_HMI-01#</td></tr>
<tr><td>Trace from</td><td>#REQ-AS_TGMT_R2-doors_management-02#</td><td>CrossRef.</td><td colspan="3">#REQ-AS_TGMT_R2-doors_opening_train_side-2#.
#REQ-AS_TGMT_R2-remote_enforce_door_release-01#</td></tr>
<tr><td>Scope</td><td colspan="3">FR</td><td>Modifier</td><td>FWD</td></tr>
<tr><td>Product Class</td><td>–</td><td>Stability</td><td>–</td><td>Safety Related</td><td>NO</td></tr>
<tr><td>Allocation</td><td colspan="5">AS_OBCU_R2</td></tr>
<tr><td colspan="6">The on-board subsystem shall indicate whether the door release is currently enforced or not and the enforcement side to the HMI (HMI_O_Door_Release_Enforced).</td></tr>
</table>

Figure 1. Example of function in the system specifications

J3	Action/Event	Comment
1	The ATS sends a command telegram to “enforce stop at station” at the current platform to the WCU_ATP via the CDI of the interlocking (ATS_I_Enforce_Cancel_Skip_or_Stop).	#REQ-AS_TGMT_R2-setting_skip_stop_status-01#
2	The WCU_ATP receives the command telegram and sends the crresponding status indication for the platform track back to the ATS (via ATS_O_Enforce_Cancel_Skip_or_Stop).	#REQ-AS_TGMT_R2-indication_skip_stop_status-01#
3	In response to the command telegram the WCU_ATP sends an acknowledgement telegram (value 0 = positive) to the ATS.	–
4	The WCU_ATP forwards the :enforce stop” in a WOD telegram to the on-board subsystem.	#REQ-AS_TGMT_R2-providing_skip_stop_status-01##
5	The on-board subsystem modifies its operational stopping point to bring the train to a halt at the platform.	#REQ-AS_TGMT_R2-executing_stop_command-02#
6	The on-board subsystem reflects “hold” in the OOS telegram. The WCU_TTS forwards this information to the ATS (ATS_O_Train_Hold_States).	#REQ-AS_TGMT_R2-station_exit-01#

Figure 2. Example of scenario in the systems specifications [WCU: wayside control unit]

to thrive the concepts of the system, have a formal semantics and use the natural language in a more structured way. We chose scenarios because they are sets of organized requirements and they are very useful when validation of requirements is necessary. Also, a formally defined language sets clearly the way concepts interact in the system.

Moreover, scenarios are used in test and safety cases. Also, scenarios represent a good way of behavior modeling involving multiple artifacts of the system. The language formalizes the description of scenarios using concepts underlying the description of the behavior of the system.

In our context, CBTC specifications are defined using system requirements grouped under functions coming from the functional architecture (see Figure 1).

Figure 2 is an example of scenario as defined in the system specification. It involves multiple actions and components, comments and requirements. However, the description may be too detailed and does not really match any pattern. Steps are described in natural language and use components from multiple levels of the system architecture. Moreover, the actions do not clarify in which order they are realized and by which components.

SCOLA

ScOLA stands for Scenario Oriented LAnguage. It defines the description of system behavior using the following concepts:

a) Concept of scenario: It consists of a sequence of actions and events that describe how the system reacts and handles its main functions. Let S be the set of all scenarios of the system.

b) Concept of component: The system is composed of a set of components. We define C the set of all components of the system. Each component c in C executes actions of the scenarios either *individually* or *in cooperation* when the action is shared between two components.

c) Concept of action: An action is a scenario at its lowest abstraction level, also called atomic scenario. We define A the set of all the system actions. In the CBTC system, we distinguish three types of actions:
   - *Simple* Action: When an action is realized by only one component, it is called *simple*. It is defined using its unique identifier, its description and its corresponding component. We also define A_s as the set of all the system simple actions.
   - *Transfer* Action: When two components cooperate in order to transmit information, the action is called *transfer*. The action is characterized by its unique identifier, its description and the components that send and receive the information. We define A_t as the set of all system transfer actions.
   - *Choice* Action: Sometimes in a scenario process, a choice has to be made and conditions have to be verified to determine the next actions of the scenario. Such actions are called *choice action*. It is characterized by its unique identifier, its description and the component realizing it. We define A_q as the set of all the system choice actions.

d) Concept of precedence: Scenarios are sets of sub-scenarios. These sub-scenarios, regarding their relationship, can be triggered in precedence. This concept means that if a scenario s_1 follows s_2, then s_2 has to wait the completion of s_1 in order to proceed. It also means that both scenarios share either information or a component that does not allow them to occur simultaneously.

e) Concept of parallelism: When the order of realization of two scenarios is not important, we consider that both scenarios are in parallel. Meaning that if scenarios s_1 and s_2 are in parallel, s_1 can start before s_2 or vice versa.

f) Concept of refinement: It reflects the need of a definition of scenarios at multiple levels of abstraction or details.

At the highest abstraction level, noted $l = 0$, scenarios are behaviors of the system triggered and processed by components from the highest level of the system architecture.

The refinement of a scenario provides a set of sub-scenarios processed by components from the next lower level of the system architecture. Scenarios at the lowest level of abstraction represent *actions*.

Moreover, ScOLA provides a set of operators, graphical and textual, in order to represent scenarios. The following table (Figure 3) depicts the operators.

MODELING A CBTC SCENARIO USING SCOLA

A CBTC railway signaling system uses wired and wireless communications between the train and the track equipment for the traffic and infrastructure control. It significantly improves the way trains are

Operator	*Graphical*	*Textual*
Scenario	S	**Scenario** S
Component	C	**Component, Basic-component**
Simple Action	A ← C	**Action** A **By** C
Transfer Action	T; C_1 → C_2	**Transfer** T **from** C_1 to C_2
Choice Action	Q → S_1, S_2	**If** (Q) {S_1} **else** {S_2}
Parallelism	S_1 ↔ S_2	$S_1 \parallel S_2$
Precedence	S_1 → S_2	S1 → S2

Figure 3. Graphical and textual operators of ScOLA

localized. Old systems used the track occupancy to determine the position of a train while CBTC equipped trains determine independently their localization and forwards it to the track equipment. According to the IEEE 1474 standard (IEEE 2004), a CBTC system is a "continuous, automatic train control system using high-resolution train location determination, independent of track circuits; continuous, high-capacity, bidirectional train-to-wayside data communications; and trainborne and wayside processors capable of implementing Automatic Train Protection (ATP) functions, as well as optional Automatic Train Operation (ATO) and Automatic Train Supervision (ATS) functions."

In the following, we consider scenario S = "Enforced Train Stop." It is realized by the two main components of the system: the train and the wayside. The wayside commands the enforced stop for the next platform a train will reach. Due to the enforced stop, the train does not initiate a departure although all other departure conditions are fulfilled.

The scenario is depicted in specifications in Figure 2. It is a description at the lowest abstraction level. Thus, we formalize the description of the scenario and we define it at the highest abstraction level that involves the components train and wayside (see Figure 4). We use the notation $s_{a,b}$ for a sub-scenario of S at abstraction level a and sequence order b of all the sub-scenarios

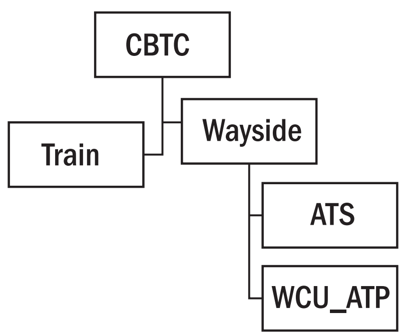


Figure 4. Partial description of the system architecture

of S. In the following, we describe the abstraction level l = 0 sub-scenarios of S.

$s_{0,1}$: *the* ***wayside*** *sends and acknowledges an 'enforce stop' command*
$s_{0,2}$: *the* ***wayside*** *sends the 'enforce stop' command to the train*
$s_{0,3}$: *the* ***train*** *modifies its operational stopping point*
$s_{0,4}$: *the* ***train*** *reflects a 'hold' command to the wayside*
$s_{0,5}$: *the* ***wayside*** *forwards the 'hold' information*
$s_{0,6}$: *the* ***train*** *stops at the platform*

According to the information provided by the functional specification and the structural architecture of the components, several refinements to lower levels are possible. We describe here the abstraction level $l = 1$ of sub-scenario $s_{0,1}$.

$s_{1,1}$: *the* ***ATS*** *sends an 'enforce stop' command to the* ***WCU_ATP****.*
$s_{1,2}$: *the* ***WCU_ATP*** *sends the corresponding status indication of the platform track to the* ***ATS****.*
$s_{1,3}$: *the* ***WCU_ATP*** *sends an acknowledgment to the* ***ATS****.*

Figures (5, 6 and 7) on the following page depict the textual and graphical representation of the scenario at level $l = 0$ and level $l = 1$, respectively.

CONCLUSION

In this article, we gave a description of the ScOLA modeling language. The language is formally defined and allows a textual and graphical representation. Moreover, it allows modeling a railway system and can easily be extended to model any other complex system since it relies on generic system architectures. This specification transformation will ease the linkage with formal analysis such as test and safety analysis. ■

REFERENCES

- CENELEC (European Committee for Electromechanical Standardization). 1999. EN 50126-1 Railway applications– The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) Part 1: Basic Requirements and Generic Process. Brussels, BE: CENELEC.
- CENELEC (European Committee for Electromechanical Standardization). 2011. EN-50128 Railway applications-Communication, Signaling and Processing Systems-Software for Railway Control and Protection Systems. Brussels, BE: CENELEC.
- IEEE. 2004. IEEE 1474.1-2004 (Revision of IEEE 1474.1-1999). Standard for Communications Based Train Control (CBTC) Performance and Functional Requirements. Piscataway, US-NJ: IEEE.
- Friedenthal, S., A. Moore, and R. Steiner. 2014. *A Practical Guide to SysML: The Systems Modeling Language*. 3rd ed. Burlington, US-MA: Morgan Kaufmann OMG Press.
- Hamilton, M. H. and W. R. Hackler. 2007. "A Formal Universal Systems Semantics for SysML." Paper presented at the 17th Annual International Symposium of INCOSE, San Diego, US-CA, 24-28 June.

REFERENCES *(continued)*

- Haugen, Øystein, K. E. Husa, R. K. Runde, and K. Stølen. 2005."STAIRS Towards Formal Design with Sequence Diagrams. *Software & System Modeling*, 4(4): 355-367.
- Issad, M., L. Kloul, and A. Rauzy, 2014. "A Model-Based Methodology to Formalize Specifications of Railway Systems." In *Model-Based Safety and Assessment* 28-42. New York, US-NY: Springer International Publishing.
- Ober, I. and I. Dragomir. 2012. "OMEGA2: A New Version of the Profile and the Tools." In: *Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference on IEEE*, 373-378.


```
System TGMT {
     Architecture TGMT {
          Component Train {
               Basic-Component OBCU
          }
          Component Wayside {
               Basic-Component WCU_ATP
               Basic-Component ATS
                         }
               }
        Block TGMT1 {
        TGMT.Train train ;
        TGMT.Train.OBCU obcu ;
        TGMT.Wayside wayside
        TGMT.Wayside.WCU_ATP wcu_atp ;
        GMT.Wayside.ATS ats ;
        }

        Scenario S with TGMT1 {
          Scenario s01 = "the wayside sends and acknowledges an enforce stop command" {
           Transfer s11 = "The ATS sends an enforce stop command to the WCU_ATP" from TGMT1.ats to TGMT1.wcu_atp {
           Transfer s12 = "The WCU_ATP sends the corresponding status indication of the platform track to the ATS" from TGMT1.wcu_atp to TGMT1.wcu_ats ;
           Transfer s13 = "The WCU_ATP sends an acknowledgment to the ATS" from TGMT1.wcu_atp to TGMT1.ats ;
           Script s11 -> s12 -> s13 ;
          }
          Scenario s02 = "the wayside sends the enforce stop command to the train"
          Scenario s03 = "The train modifies its operational stopping point"
          Scenario s04 = "The train reflects a hold command to the wayside"
          Scenario s05 = "The wayside forwards the hold information"
          Scenario s06 = "The train stops at the platform"

          Script s01 -> s02 -> s03 -> s04 -> s05 -> s06    ;

          }
        }
```

Figure 5. Textual representation of the scenario S

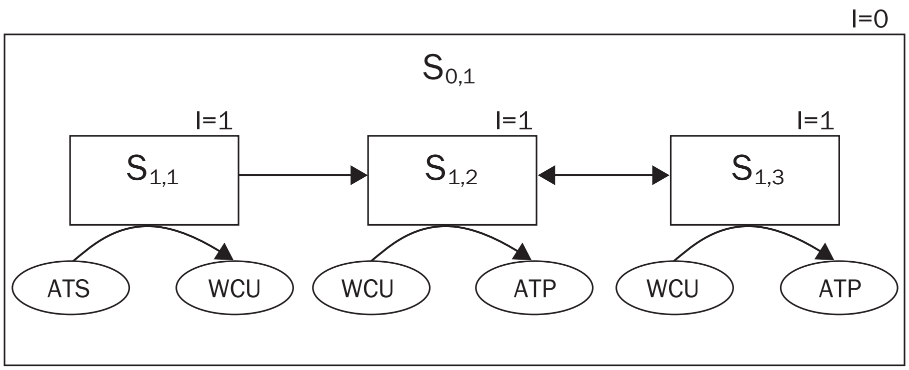


Figure 6. Graphical representation of the level 1 of S

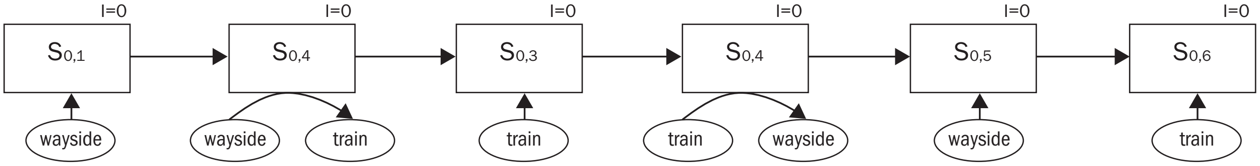


Figure 7. Graphical representation of the level 0 of S

Realizing the Potential of Connected Fitness Technologies: a Case for Systems Engineering Involvement

Woodrow W. Winchester, III, wwinches@kennesaw.edu**, and Valerie Washington,** vwashin8@students.kennesaw.edu

CONNECTED FITNESS: THE SPACE AND OPPORTUNITIES

A new report from WinterGreen Research predicts that the market for fitness, sports, and performance wearables market will increase from just $3.5B in 2014 to $14.9B in 2021 ("Smart Wearables For Sports And Fitness: Market Shares, Market Strategies, And Market Forecasts, 2015 To 2021."). Fitness tracking devices alone will be a $5B+ industry by 2019, according to Parks Associates, a research firm from Dallas, TX, USA (http://www.marketwired.com/press-release/parks-associates-global-revenues-from-connected-fitness-trackers-exceed-5-billion-2019-2003533.htm).

This growth of activity and fitness trackers in the consumer marketplace speaks to their transformative potential in enabling overall health behavioral change. To echo Yves Béhar, designer of the Jawbone and the HiveTech fitness trackers, "the largest potential is for wearable technology to change health and healthcare" (https://www.wareable.com/meet-the-boss/yves-behar-is-a-wearable-tech-rockstar-7632).

CONNECTED FITNESS: THE CHALLENGES

While the mission potential of connected fitness is great, design, development, and deployment challenges exist in this complex and evolving space. Most recently, frequent technological developments compound these challenges. For example, in 2014, Nike announced it would discontinue its FuelBand fitness tracker, the data collection device, in light of market factors and emerging innovation and leadership in the wearable fitness device space by Apple with the Apple Watch. Nike decided to instead focus on the service component of its connected fitness ecosystem, centering efforts on realizing a device-agnostic service architectural design. Kevin Plant, Under Armour's founder and CEO, expressed similar sentiments in a 2015 Forbes article, regarding their development efforts with the now defunct Armour39, "we learned that making hardware was incredibly cumbersome and difficult...Someone would always be coming out with a better mousetrap (http://www.forbes.com/sites/parmyolson/2015/09/30/kevin-plank-under-armour-apps-technology/)."

Beyond the hardware components of connected fitness ecosystems, software challenges also exist; as an example, the translation of collected data into meaningful and actionable insights on health. This is paramount in realizing the potential of connected fitness technologies, as this would enable individuals to "[go] beyond

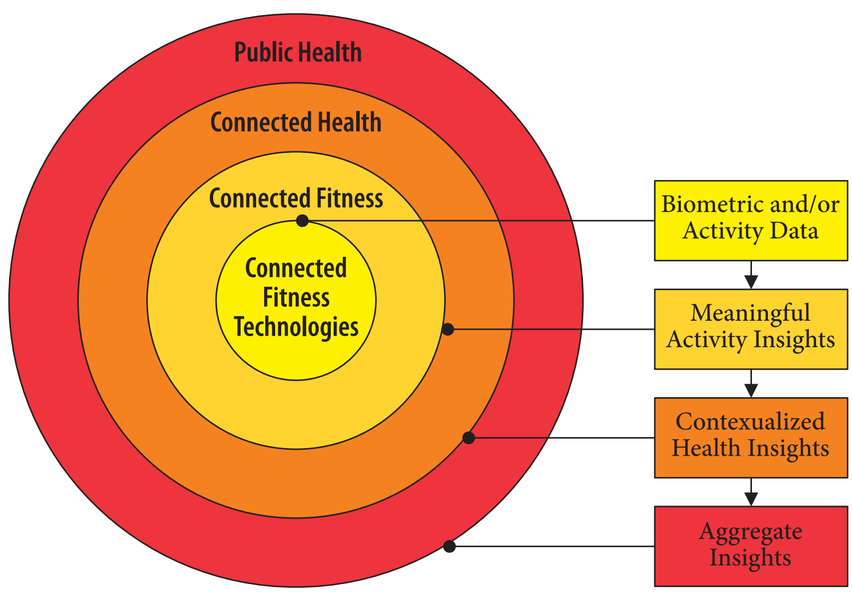


Figure 1. Fit of connected fitness as a system

athletic performance to something that's more [of a] lifestyle (http://www.wareable.com/smart-clothing/where-are-our-smart-clothes)."

CONNECTED FITNESS: DEFINED IN A SYSTEMS CONTEXT

With the intent of offering meaningful physical activity performance insights, connected fitness, as a concept, is a product ecosystem typically comprised of both a hardware product and software service. The product hardware is a wearable connected fitness technology or device that collects biometric data, such as heart rate, and/

Table 1. Example of Systems Engineering Challenges and Opportunities

PRODUCT/SERVICE

ARCHITECTURE:

- **Lack of interoperability**: Facilitating data/information to/from appropriate health/wellness constituencies: "There is no standard of entering data from medical or commercial wearables, which is huge burden for physicians": http://www.forbes.com/sites/jenniferelias/2015/11/30/an-overburdened-physician-system-is-yet-another-obstacle-to-getting-wearables-to-those-who-need-them/
- **Towards Technology Agonistic Designs:** Current **closed source platforms that do not allow for user to collect and store data from various sources in varying settings**: "Using Connected Fitness, prototype solution from FocusMotion, a user with the Samsung Gear and their smartphone can move seamlessly from a treadmill or elliptical directly onto strength training, without any loss of data. The result: not only is all cardio information — calories, distance, time, and more — collected by the Connected Fitness app, but also all of the strength training data — exercise performed, number of repetitions. And the best part is, **a gym-goer doesn't have to manually input any of this**; it happens automatically": http://www.sporttechie.com/2015/09/03/fitness-industry-falling-behind-wearables-race/
- **More appropriate enabling decisions through recognizing and leveraging core competencies:** Responsiveness to advances in enabling technologies as reflected in Polo Ralph Lauren's partnership/collaboration in with OMsignal: http://montrealgazette.com/technology/tech-biz/ralph-lauren-deal-is-a-game-changer-for-smart-wear-startup-omsignal

EXPERIENCE DESIGN:

- **Offer meaningful insights not just data:** "Consumers don't want to just see numbers and data…they want insights that are more meaningful, something that gives them a picture of whole health. **Going beyond athletic performance to something that's more lifestyle.**": http://www.wareable.com/smart-clothing/where-are-our-smart-clothes
- **Engendering behavioral change:** "Meaningful products enable us to do something better… to achieve this, we must go beyond the fad of just quantifying ourselves. Instead we should use the data to prompt us to act in a way that makes us healthier, stronger, better—a principle of 21st-century design.": http://www.fastcodesign.com/1672107/3-ways-to-make-wearable-tech-actually-wearable

PROCESS

DESIGN PROCESSES:

- **Process Design:** Processes and approaches not amenable to an integrated (hardware/software) interdisciplinary design context: http://designerfund.com/bridge/fitbit-design-an-inside-look/
- **More appropriate effort leadership decisions:** "Another forward-thinking solution might be for leading (fitness equipment) manufacturers to jointly create their own technology hub device… our (health clubs) industry may have to give up some level of control when partnering with a technology leader: http://clubindustry.com/forprofits/industry-should-develop-technology-partnerships
- **Building of privacy and security into development processes:** Associated use and privacy concerns exist and must be understood and resolved: http://www.techrepublic.com/article/the-dark-side-of-wearables-how-theyre-secretly-jeopardizing-your-security-and-privacy/

PERSPECTIVE

SITUATING THE PROBLEM AND SOLUTION SPACES:

- **The need for a more holistic viewpoint to design** – the connection to **"connected health":** "This isn't just about knowing that I took 8,000 steps yesterday," Plank said. "But because I took 8,000 steps, how did it make me feel? And more importantly, how did that work with how I slept or what I ate that day? And we think that having that information is going to allow them to make better decision to live healthier and enriched lives — affecting ultimately, obviously, fitness, but we think there's an outlook for us to affect global health, which gets us so excited.": http://mobihealthnews.com/45618/under-armours-connected-fitness-apps-now-have-140-million-users/
- The need to be responsive to varied stakeholders and their, often, evolving, changing, & conflicting needs and/or requirements: Must take into consideration other stakeholders such as the trainer and the gym: "Health and fitness professionals can provide value to their clients when they are not in the gym by using the data to inspire and motivate them to stay active even when they are traveling or otherwise unable to have a training session": http://www.acefitness.org/prosourcearticle/5570/ace-sponsored-research-how-will-wearable
- **Viewing the System as a solution to a problem vs. a solution searching for a problem (function over form):** "smart apparel makers need to start with improving a daily function through technology and then figure out how that technology can be worn in a unique and safe way.": http://fortune.com/2015/09/11/smart-wearables-fitness/

or physical activity data. The service typically is a wireless software platform that aggregates, analyzes, and presents the data. As a systems of systems, connected fitness technologies, and therefore connected fitness, can also situate as a subsystem of connected health (Figure 1).

As operationalized by the authors, connected health, conceptually, is the ability to enable health and wellness decision-making via the sharing of health and wellness-related data. These data can be collected, analyzed, and aggregated both inside and outside a clinical setting. In application, an individual could use a fitness device to track biometric and/or activity data, such as number of steps or heart rate throughout the day. This data feeds into a connected fitness platform, such as Under Armour's UA Record, Wahoo RunFit, Apple Health, and others, which translates the data into something more meaningful such as calories burned from the perspective of weight loss and/or management. This information can be further aggregated and/or contextualized and feed into one's electronic health record (EHR) and leveraged by a clinician in assisting one in managing a chronic disease, such as Type 2 diabetes. Further, insurance companies could use these insights to lower one's health insurance premium based on this better understanding of health & wellness behaviors and associated outcomes. Moreover, the data can be aggregated and analyzed to better inform responses to public health issues such as obesity.

SYSTEMS ENGINEERING: ADDRESSING THE CHALLENGES IN THE CONNECTED FITNESS SPACE

Systems engineering, as an approach to evolving a successful system, could address challenges and offer value in the realization of the potential of connected fitness technologies. The tools and techniques offered by systems engineering can truly advance developments in this space. A role for systems engineering in addressing the design and development challenges is clearly situated when using a framework of product/service (the system), process (the system realization process), and perspective (how the product, as a solution, is viewed), as indicated in Table 1.

SYSTEMS ARCHITECTURE: TOWARDS RESOLVING CONNECTED FITNESS CHALLENGES THROUGH SYSTEMS ENGINEERING

An analysis of these sorts of challenges and opportunities truly defines a role for systems engineering and suggests a possible lack of appropriate front-end rigor in the product development processes. This rigor, in the initial stages of the product development process (PDP), represents activities, such as identifying stakeholders, identifying and characterizing their often conflicting needs, and interpreting those needs as systems goals. Outcomes from these activities could inform more appropriate architectural decision-making.

It is important to note that INCOSE's Systems Engineering Vision 2025 highlights the importance of systems architecting, specifically highlighting the growing need to architect systems that address multiple stakeholder viewpoints, which, as detailed, is of import in the design of connected fitness technologies. Moreover, in the newly published text, System Architecture: Strategy and Product Development for Complex Systems, the authors amass and offer notions, tools, and techniques in informing more appropriate architectural designs. In the forward of the text, Norman R. Augustine states that "*much of the power of idea originates with the potential to trade among several architectures early, to look downstream and identify which constraints and opportunities will be central to value. It isn't possible to trade among early ideas if the architecture encompasses all details, nor is it a meaningful exercise if important drivers of value are missing.*"

The need for systems engineering in realizing the potential of connected fitness is clear. In particular, growing and evolving systems architecting competencies within the connected fitness space can advance this involvement. The development of these competencies and the engagement of analogous tools and approaches in this space would be a two-fold benefit. From a practitioner perspective, systems engineering involvement would equip the product management and development function within the connected fitness space with more robust techniques to appropriately execute front-end design activities such as conceptual designing. From a systems engineering disciplinary perspective, involvement in this emerging space provides a more contemporary case study for the application of systems engineering methodologies that could further motivate and inspire the growth of the systems engineering discipline. ■

INCOSE®

INCOSE Certification

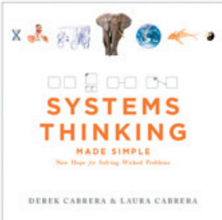
See why the top companies are seeking out
INCOSE Certified Systems Engineering Professionals.

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

Apply for INCOSE Certification Today!

Visit www.incose.org or
call 800.366.1164

INCOSE ASEP · INCOSE CSEP · INCOSE ESEP



Systems Thinking Made Simple

New Hope for Solving Wicked Problems

By Derek and Laura Cabrera
Odyssean Press 2015 (ISBN-978-0-9963493-0-7)

Reviewed by Richard F. Emerson, remerson9@gmail.com

INTRODUCTION

I was initially frustrated with the approach taken by the authors, but they convinced me that to create a world of systems thinkers the approach was necessary. Having completed it, I still have some frustration, particularly about the subtitle. Yet, with the new perspectives presented, I realize that I was too narrow in my expectations. This book is not about solutions for specific wicked problems, but presents a method of creating, changing, and evolving mental models for the class—wicked problems—that will lead to their resolution if not a "once and for all" solution.

I am a practicing systems engineer and have been for at least 30 years. I like to think that I was destined to be a systems engineer from a very early age. To quote a Reader's Digest quip, "To every complex problem, there is a straightforward, simple, and obvious solution, that is **wrong**. (Circa 1950)." In a way, that quip could be a motto for this book, but as a motto, it only captures the need for systems thinking. The authors' aim is to explain how to become systems thinkers, develop systems thinking organizations, and solve wicked problems.

As the authors explain, the book presents the material in a way that can be used to teach systems thinking to elementary school children, teenagers, CEOs, all types in between, and beyond. The style is not scholarly, yet, it achieves a well-defined scholarly position, and there are numerous footnotes for those needing the background material. The intent is to be accessible by all. Where jargon creeps in, it is due to the not-systems-thinking approaches.

My recommendation: "relax and enjoy this book, but be ready to be challenged and enlightened."

The organization of the book is into three sections of several chapters each. Here is a brief summary of each.

SIMPLE RULES OF SYSTEMS THINKERS – SECTION 1

With a minimum of introduction the authors present the foundation of their work—four simple rules: every thing is distinct (D), everything is a system and a part of a system (S), everything relates to other things (R), and everything can be perceived from a viewpoint and is therefore an object for that viewpoint (P). Applying these rules in a manner similar to fractiles leads to increasing understanding of the subject, whatever it may be.

The authors distilled these four rules from a massive collection of material on systems thinking. An additional concept creates the context for the application of the rules—for every distinction, that is a definition of what the item is, there is the "other." It is the same for systems-parts, relations, and perspectives. Application of the "other" instantly opens the field of discussion. For customer, there is not-customer, for target there is not-target, for transportation there is not-transportation.

The authors describe how to use the simple rules (DSRP) and how the use helps develop better mental models of the world. A better model in this sense is not necessarily more detailed, but is more faithful to the way the world actually works.

They also illustrate how a system of simple rules can result in Complex Adaptive Systems (CAS). This naturally leads to the reverse inquiry—what are the simple rules that engender a CAS?

BECOMING A SYSTEMS THINKER – SECTION 2

Just as information is a product of data and structure, mental models are a product of information and structure. It is these mental models, when in alignment, that promote collaborative and cooperative interaction. The authors describe the numerous ways that visual presentations have attempted to communicate mental models with varying success. They then present four simple conventions for visually presenting information, which address the four aspects (DSRP) of items—things or ideas. Presented in this way, the reader develops the understanding of the things and ideas and the structure associated with them, as well as seeing the information.

Extending the mental modeling by using pattern templates, the authors call them "Jigs," further facilitates the model development and subject understanding.

The details presented in this section are worth in-depth consideration. Pick a wicked problem near and dear to your heart as you reread the details. Be ready for some surprises in what happens.

7 BILLION SYSTEMS THINKERS – SECTION 3

It is an easy step from the consideration of the plethora of wicked problems to the realization that we need many more systems thinkers. In this section, the authors address the development of systems thinkers and the development of systems thinking organizations. A great deal of systems thinking has gone into the material for this section. One result is that organizations have an additional set of four rules or principles—vision (V), mission (M), culture (C), and learning (L). These combine in the development of an organization to create a CAS organization.

CONCLUSION

Do not pass up the opportunity to read this book. The authors' exposition of the concepts of systems thinking is both simple and profound. Do not let it become shelf-ware either. Apply the ideas. ■

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 lifecycle, 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 lifecycle 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.

PENN STATE | ONLINE

Deandrea Thompson
Systems Engineering Graduate

Earn a Master's Degree in Engineering—Entirely Online

Systems Engineering

Software Engineering

Engineering Management

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

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

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

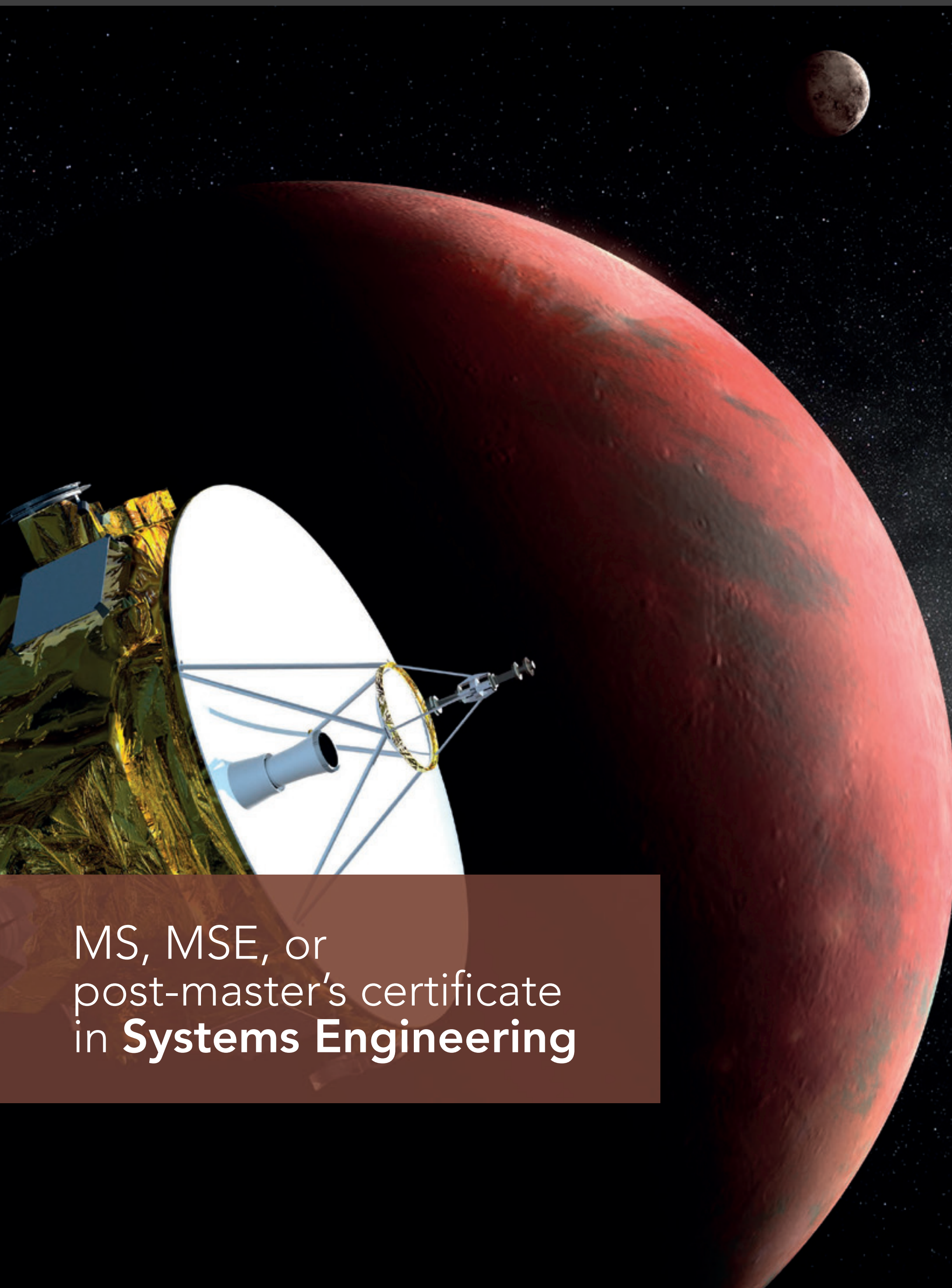
PENNSTATE
1855
World Campus

Apply Now: worldcampus.psu.edu/INCOSE

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

Johns Hopkins Engineering

Online and On-Site Part-Time Graduate Programs



LARGEST SE MASTER'S PROGRAM IN THE WORLD

GOOD LEADERSHIP MEETS SOLID TECHNICAL SKILLS

1 of **3**

ABET-ACCREDITED SYSTEMS ENGINEERING PROGRAMS IN THE NATION

LEARN MORE AT EP.JHU.EDU/SE

800-548-3647

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

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

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING