December 2013 | Volume 16 Issue 4

INCOSE
International Council on Systems Engineering ®

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

What's Inside

What's Inside

From the President 3

Special Feature 6

AFIS Doctoral Symposium: Systems Engineering Research Challenges in French Universities 6

RobAFIS Student Competition: From Systems Engineering to the Engineering of One System 7

A Systemic Perspective for Mass Customization: An Approach to Defining Product Lines 10

Requirements Engineering Process according to Automotive Standards in a Model-Driven Framework 12

A Methodology for Defining Security Requirements using Security and Domain Ontologies 14

Evaluating Alternatives for Designing Mechatronic Systems in a Systems Engineering Context 16

Towards an Integrated Approach of Safety Analysis for Mechatronic Systems Design 19

Towards a Safe Systems Engineering 21

Model-Based Safety Assessment: The AltaRica 3.0 Project 24

A Metamodel for Knowledge Modeling and Maturity Integration in Systems Engineering 26

Model-Based Service Orchestration for Business Applications Re-Engineering 28

Creating a Common Vocabulary to Support the Exchange of Numerical Models between Suppliers and Users in a Complex System Design 30

An Integrated Approach for Designing an Agricultural Process Guided by Sustainable Evaluation: Application to Olive-Oil Production 32

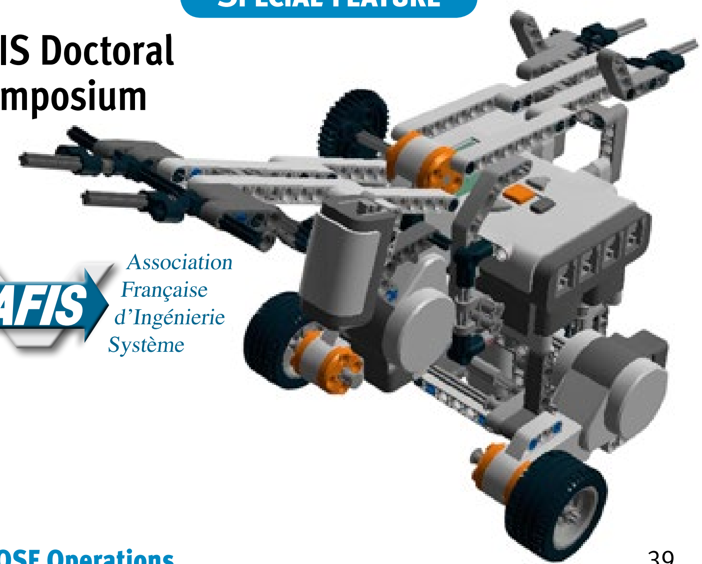
Technical Operations 37

Technical Directions: Setting the Stage for the 2014 International Workshop 37

SPECIAL FEATURE

AFIS Doctoral Symposium

AFIS Association Française d'Ingénierie Système



INCOSE Operations 39

Formation Meeting for INCOSE Denmark 39

Chapter News 39

A Common Language to Ensure More Effective Development of New Products 39

INCOSE Events 41

ISSS 2013 International Conference 41

Event Report: 25th Annual ICSSEA 42

INCOSE South Africa's Systems Engineering Winter School: A Collaborative Effort 43

Management in Practice in Krakow 43

Event Report: INCOSE Autumn Academic Forum 44

INCOSE Spotlight 46

INCOSE Spotlight on. . . Suja Joseph-Malherbe 46

Book Reviews 47

Just One More Thing 53

Final Thoughts 55

INSIGHT

Publication of the International Council on Systems Engineering

Chief Editor Bob Kenley
insight@incose.org 765 430 3774
Assistant Editor Andrew Cashner
andrew.cashner@incose.org
Theme Editor
herve.panetto@incose.org Hervé Panetto
Advertising Editor Roland Espinosa
respinosa@wiley.org +1 201 748 6819
Layout and Design Chuck Eng
chuck.eng@comcast.net
Member Services INCOSE Administrative Office
info@incose.org +1 858 541-1725
On the Web http://www.incose.org
Article Submission INSIGHT@incose.org

Publication Schedule. ***INSIGHT*** is published four times per year. Issue and article/advertisement submission deadlines are as follows: ***April 2014*** issue – 15 February; ***July 2014*** issue – 15 May; ***September 2014*** issue – 17 July; ***December 2014*** issue – 15 October
For further information on submissions and issue themes, visit the INCOSE website as listed above.

Advertising in *INSIGHT*. Please e-mail respinosa@wiley.org.

Subscriptions to *INSIGHT* are available to INCOSE members as part of their membership. Complimentary copies are available on a limited basis. Back issues are available on the INCOSE website. To inquire about membership or to order a copy, contact Member Services.

Who are we? INCOSE is a 9500+ member organization of systems engineers and others interested in systems engineering. 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, is sponsored by a corporate advisory board, and is led by elected officers and directors.

From the President

John A. Thomas, ESEP, john.thomas@incose.org

Greetings to my fellow INCOSE colleagues—this is my last ***INSIGHT*** article as president. I will be stepping down as your president on the 25th of January at our 2014 International Workshop (IW). The IW will be held in Torrance, California (US), and I hope to see you there. Please say hi or introduce yourself to me if you are attending the IW. I will have plenty of time to sit in on meetings and enjoy great technical conversations with you all. It will seem good to have the time to talk and not be attending so many business meetings.

I also want to say congratulations to David Long, our current president-elect who will so ably step into my place as your president. Hearty congratulations also go to election winners, Alan Harding, our new president-elect; Jean-Claude Roussel, our new director of the EMEA Sector, representing the chapters and members from Europe, Western Asia, and Africa; Jen Narkevicius, our new treasurer; and Art Pyster, who continues based on this election as our director of academic affairs.

Lastly, I want to reiterate my perspectives on INCOSE's business imperatives going forward. I shared these at our 2012 symposium in Rome and then in an article in December 2012 as well. We have made considerable progress on these imperatives since I started supporting our organization as your president-elect in 2010. I hope your Board of Directors continues to maintain and increase momentum in each of the areas.

Business Imperative No. 1 (This positively impacts your career growth and salary)

I believe it is in our best interests to advocate and promote:

- The system engineer as a multidisciplinary leader;
- The systems engineering discipline as a critical tool in the tool box of a systems engineer and of those who have systems problems;
- The value of the well trained system engineer—skilled in both the science of systems engineering and the art of leadership.

Business Imperative No. 2 (This positively impacts INCOSE's ability to make a difference)

To achieve our mission we must increase INCOSE's influence on worldwide systems issues. To increase our influence, we must deepen our leadership connections and form partnerships with sister organizations. INCOSE's relationship with these sister organizations mirror the relationship we have as system engineers with those whom we work with on a daily basis. There are too many examples to share a complete list. Additionally, the list will have to be prioritized. But a few include organizations involved with these areas:

- program management
- the engineering disciplines—mechanical, electrical, civil, chemical, computer science, software, and others
- safety and cybersecurity
- reliability and human factors
- test and evaluation, costing, and acquisition

Business Imperative No. 3 (This provides INCOSE with the resources required to influence decisions)

To achieve the first and second imperatives we require additional resources to implement the thought leadership agenda of our organization that is comprised mostly of volunteers. The breadth of increased resources includes these:

- Modernizing of our information technology that connects our distributed membership base and enables their ability to communicate and collaborate
- Transitioning to a full-time executive director for our organization, someone who (behind the scenes and in concert with the direction of the board) focuses on shifting our dependency in revenue away from membership dues through the delivery of high-value revenue generating services—services that will be seen as attractive to our

continues on next page

INCOSE
International Council on Systems Engineering

INCOSE PAST PRESIDENTS

Samantha Robitaille, 2010 – 11
Pat Hale, 2008 – 09
Paul Robitaille, 2006 – 07
Heinz Stoewer, 2004 – 05
John Snoderly, 2002 – 03
John Clouet, 2001
Donna H. Rhodes, 2000
Ken Ptack, 1999
William W. Schoening, 1998
Eric C. Honour, 1997
V. A. (Ginny) Lentz, 1996
James Brill, 1995
George Friedman, 1994
Brian Mar, 1993
Jerome Lake, 1992

From the President *continued*

members as well as the broader systems engineering community

- Adding professional staff to support operation and planning within our organization and to maintain day-to-day relationships and execution of joint agendas with sister organizations

Lastly, I would like to thank the directors of INCOSE's Board (existing and new) for their volunteer service and commitment to the welfare of our organization. I wish you all the best.

Corporate Advisory Board — Member Companies

Air Force Institute of Technology
Airservices Australia
Alliant Techsystems
AM General LLC
Analytic Services-Applied Systems Thinking Institute
Astrium, an EADS Company
ATKINS
Aviation Industry Corporation of China
BAE Systems
Bechtel
Beihang University School of Reliability and Systems Engineering
Boeing Commercial Airplane Co.
Boeing Defense, Space & Security
Boeing Research and Advanced Technology
Booz Allen Hamilton Inc.
C.S. Draper Laboratory, Inc.
Carnegie Mellon University Software Engineering Institute
Cassidian
Cranfield University
Cummins, Inc.
Dassault Systems
Defense Acquisition University
Deloitte
Deputy Assistant Secretary of Defense for Systems Engineering, US Department of Defense
DRS Technologies, Inc.
EADS N.V.
Exelis
Federal Aviation Administration US
Ford Motor Company
General Dynamics
General Electric
George Mason University
Georgia Institute of Technology
Honeywell International
IBM Corporation
JAXA (Japan Aerospace Exploration Agency)
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 and Technology
Mitsubishi Electric Corporation
Nanyang Technological University
National Aeronautics and Space Administration
National Geospatial–Intelligence Agency
National Reconnaissance Office
National University of Singapore
Naval Surface Warfare Center–Dahlgren Division
Northrop Grumman Corporation
PA Consulting
Pacific Northwest National Laboratory
Proctor & Gamble
Raytheon Corporation
Rockwell Collins, Inc.
Rolls-Royce
Saab AB
SAIC
Sandia National Laboratories
Scitor Corporation
SELEX ES, a Finmeccanica Company
Siemens
Singapore University of Technology and Design
Sotera Defense Solutions, Inc.
SRA International
Stevens Institute of Technology
Swedish Defence Materiel Administration
TASC, Inc.
Thales
The Aerospace Corporation
The MITRE Corporation
The SI
The University of New South Wales, Canberra
TNO Technical Sciences
UK MoD
United Technologies Corporation
University of Maryland
University of South Australia Defense and Systems Institute
University of Southern California
University of Texas at El Paso
US Army ARDEC
US Army TARDEC
Vitech Corporation
Volvo Construction Equipment
Woodward Inc
Worcester Polytechnic Institute

INSIGHT SPECIAL FEATURE

Association Française d'Ingénierie Système

Theme Editor's Introduction

AFIS Doctoral Symposium: Systems Engineering Research Challenges in French Universities

Introduction by theme editor Hervé Panetto, herve.panetto@incose.org, Professor, University of Lorraine, Research Centre for Automatic Control, CNRS, France

This issue is devoted to special coverage of the French Systems Engineering Academia-Industry Forum, organized by AFIS (Association Française d'Ingénierie Système), the French chapter of INCOSE, with the support of the University of Bordeaux. The forum was held 29–30 November 2012 in Paris, France, with the objective of developing strong relationships between industry and academia. It gathered 120 participants (of which one third were from industry).

The expected crossfertilization between academia and industry developed within 10 workshops on these topics:

- Teaching systems engineering: What and how?
- Model-based systems engineering
- Human-based cognitive systems
- Architecting systems and services
- System thinking
- Requirements engineering: Which research?
- Model-based systems interoperability from an organizational perspective
- System engineering scientific foundations: Open questions
- Innovation in systems engineering
- Safety from a systems engineering perspective
- The RobAFIS student robotics competition

Additionally, a preforum meeting was organized for attendance of teachers, students, and representatives of industry who were not members of AFIS, in order to disseminate system engineering practices, issues, and challenges.

This theme section begins with an account of the RobAFIS student competition "From Systems Engineering to the Engineering of One System." The 11 articles that follow are expanded from abstracts and poster presentations by doctoral students during the workshop on "Learning Systems Engineering while Doing Research," translated into English and improved through a peer-review process. The objective of publishing these articles is to disseminate current academic doctoral research done in French universities that is linked with industrial needs.

The first article, "A Systemic Perspective for Mass Customization: An Approach for Defining Product Lines," addresses systems engineering practices in the context of mass customization and, more specifically, an approach for defining product lines. The authors propose a model for formalizing the link between customer needs and system requirements and thus, improving the identification of the right product that meet functional specifications. Indeed, as an example, the embedded safety-critical systems industry is facing an exponential increase in the complexity and variety of systems and devices. The second article, "Requirements Engineering Process according to Automotive Standards in a Model-Driven Framework," explores challenges when following the model-driven engineering paradigm to develop automotive embedded systems that meet the user needs and the regulatory constraints of the domain and that further enhance the quality of developed product.

The third article, "A Methodology for Defining Security Requirements using Security and Domain Ontologies," proposes a method for exploiting both security knowledge and domain knowledge to guide the elicitation of domain-specific security requirements that is also applicable when requirements deal with safety. The next article, "Evaluating Alternatives for Designing Mechatronic Systems in a Systems Engineering Context," proposes a unified and interoperable data model that can be understood and shared by designers whatever may be their origin or domain. In the domain of mechatronics, knowledge from various disciplines such as computer science, mechanics, electronics, and automation, among others are needed to fulfill the complex requirements in the design of industrial products in various fields such as automotive, aerospace, and medical equipment. The increasing complexity of these systems makes their development and safety analysis more difficult. The authors of the article "Towards an Integrated Approach of Safety Analysis

» continues on next page

for Mechatronic Systems Design" proposes a method to integrate safety analyses within an MBSE (model-based system engineering) approach based on SysML that aids engineers by accounting for safety aspects through the whole design process. This leads to different approaches for taking into account safety in systems engineering. The article "Towards a Safe System Engineering" compares two approaches based on the model-based safety assessment and exchanges between system engineering and safety for identifying the requirements for a safe system engineering. The Altarica project, detailed in the article "Model-Based Safety Assessment: AltaRica 3.0 Project" proposes a set of modeling and assessment tools to perform preliminary safety analyses during the engineering of a system.

Modeling is one of the main activities in system engineering. However, how to assess the quality of the produced models is still an open issue. The authors of the article "A Meta-Model for Knowledge Modeling and Maturity Integration in Systems Engineering" are partially tackling this problem by proposing a metamodel to provide a tool that merges to merge data to ensure consistency and integrates a measure of maturity to support decision making in the engineering activities of a system-of-interest. The next article, "Model-Based Service Orchestration for Business Applications Re-Engineering," describes implementing business processes when reusing existing services. The authors propose a framework that helps business analysts in designing business processes using BPMN 2.0 specifications and that provides the description of each business task to select the most suitable services for implementing them.

In the automotive industry, computer-based simulation plays a crucial role in the design of modern products. The model-development process involves a number of parallel activities of experts in multiple domains. Model interoperability (which means also understandable by all stakeholders) is crucial to ensure that all requirements are taken into account at the right time by the right supplier. The article "Creating a Common Vocabulary to Support the Numerical Models Exchange between Suppliers and Users in a Complex System Design" then proposes especially for the automotive industry, a domain vocabulary for formally characterizing numerical models.

In the food industry, formalizing and developing knowledge of industrial production systems and their environmental impacts is important. The article "An Integrated Approach for Designing an Agricultural Process Guided by Sustainable Evaluation: Application to Olive-Oil Production" reports the results of the OiLCA project. This project, oriented on a lifecycle assessment (LCA) approach, aims at applying system engineering practices and tools for a better understanding of environmental assessment and thus proposing a knowledge model of an agro-industrial system.

I would like to thank all authors for their contributions and all reviewers for the tremendous work they have done to improve the articles published in this issue of ***INSIGHT.***

RobAFIS Student Competition: From Systems Engineering to the Engineering of One System

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

This article presents the RobAFIS competition, which has been organized every year since 2006 by AFIS. Each year this project benefits from feedback from the previous competitions that continuously improves the educational value of the event. Beyond the interest of competition, the RobAFIS repository is increasingly used as a model for deploying projects in support of systems engineering education.

1. Objectives

RobAFIS enhances the activities of AFIS, offering educational and research institutions a method to better understand and develop the use of systems engineering best practices, as recommended and formalized by AFIS. The recommended reference document for RobAFIS is the French book *To Discover and Understand Systems Engineering* (Fiorèse and Meinadier 2012).

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.

2. A Competition in a Pedagogical Framework

RobAFIS is both a comparative assessment between robots presented by competing teams, and an educational operation whose purpose is to lead student teams to implement collaboratively a systems engineering approach to design a solution and to write a development document.

Students and their supervising teachers have the opportunity to exchange with the jury of AFIS expert members who work in industry or teach systems engineering. During development, these experts answer questions via a frequently-asked-questions page on a dedicated collaborative space (http://www.robafis.fr). These questions concern competition rules and organization, and technical or methodological issues related to stakeholder requirements or to the development document.

After the competition, individual feedback is given to teams, during which jury members give detailed information about the evaluation, the

» *continues on next page*

methods and tools used, and about robot behavior and performance. The jury gives an educational dimension to its answers and comments. This feedback enables the comparison between learners' and practitioners' points of view.

3. RobAFIS: A Competition between Teaching Institutions

This competition is open to bachelor's or master's degree students in an engineering discipline (systems, electronics, software, mechanical, hydraulic, and others). Since the beginning of RobAFIS in 2006, about 20 different institutions participated at least one time, with an average of 10 registered teams each year.

4. RobAFIS: A Project, An Organization

4.1. Project Point of View

RobAFIS is organized as a project in which various stakeholders are involved. Each one has its own expectations and a well-defined role and contribution, as occurs in a real industrial organization for an acquisition project.

4.2. Organization Point of View

Competition management

The competition contracting authority remains the same every year. It is led by the "Training and Skills" AFIS Technical Committee and Jean-Claude Tucoulou, RobAFIS permanent manager. This contracting authority is in a "client" position for the competing teams, and is a "supplier" for AFIS members and for all those interested in systems engineering education. To that end, the contracting authority developed a generic repository in 2010 that codifies the RobAFIS operation and facilitates the annual work of organizing committees. For each new edition, it proposes a host institution for the final, sets up the organizing committee, and collects feedback to continuously improve the generic repository.

The competition prime contractor is organized by the RobAFIS organizing committee, supported by the assessment jury and by all involved in the event logistics, especially in the final phase. Its primary objectives are the definition of the overall schedule, the specification of the robot and its environment, the feasibility verification, and developing and disseminating the competition repository.

Competition participation

The competing contractors are all teams involved in the competition, participating in the development of a robot-solution and in the final phase of the competition.

5. RobAFIS: A Structured Engineering Approach in a Project Framework

The competition is subject to various phases: prior registration, development and implementation, free trials, configuration audit, operational validation, and project presentation.

Registration Phase

The Organizing Committee develops and disseminates the competition repository which includes the following:

- Competition rules, including participation conditions and overall schedule
- Specifications applicable to the system to be done and its operational environment
- Development-document template giving the development-document structure and the deliverables packaging

In a pedagogical manner, this phase reproduces current practices of contractual relationships between a contracting authority, a prime contractor and its first-tier cooperating contractors.

Development and Prototype Implementation

At the end of the development phase, each team provides a complete development document, consistent with the Robafis Development Document model. For the robot prototype (Figure 1), participants use only a LEGO kit, provided by the RobAFIS Organizing Committee for equity purposes regarding components supply.

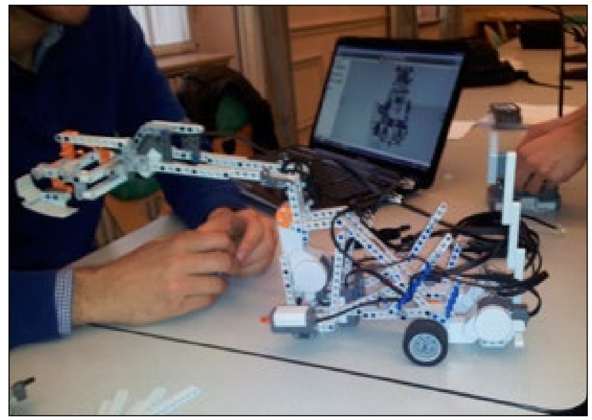


Figure 1. Engineering phase

Final (Part I): Configuration Audit

A group of three persons, external to the development team, has a limited time to assemble the robot using the configuration proposed in the development file. A jury supervises the assembly operation and ensures compliance with the development document.

Final (Part II): Tests and in situ *Verification:*

Each team has access to an assessment area during a limited time, to make on-site free trials and functional verifications. These tests are conducted privately, with the exception of members of the Organizing Committee. The team has the opportunity to modify the configuration of the product, but any change must be traced and discussed during the project audit.

Final (Part III): Operational Validation:

The qualification has the dual purpose of assessing the robot performance in an operational situation (Figure 2) and in a confrontation with opponents. This is the key step in the operational evaluation. The objective is to evaluate the functional coverage completeness of each solution and to compare the mission execution speed of competing solutions.

Final (Part IV): Project Audit:

The project presentation provides feedback after the operational evaluation. The objective is to explain reasons for success and to investigate the possible difficulties and technical problems encountered.

6. Document and Physical Deliverable-Driven Systems Engineering

Systems engineering education relies on the acquisition of knowledge and skills, based on methods and associated tools. Regarding the ISO 15288 standard, the knowledge and skills enable technical processes activities, implemented using project processes activities in a broader context of the company related business processes activities. The overall aim is to achieve a system, the result of acquisition and supply processes.

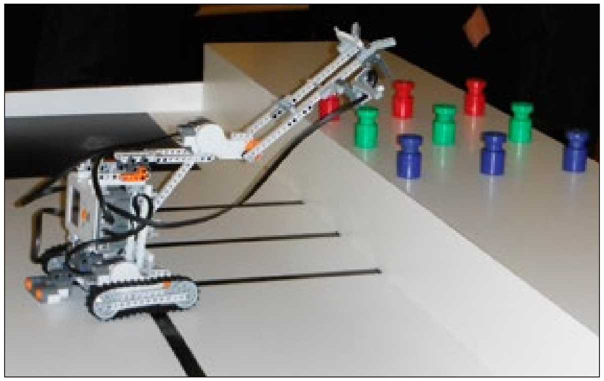


Figure 2. Deployment phase

The framework proposed by the standard does not really structure the sequencing and planning of activities and processes leading to an orderly realization of activities and project deliverables. The principle proposed for RobAFIS is to provide a development repository formalizing deliverable documents whose structure guides implementation of processes and activities, until the final stage of solution making, which includes validation and participation in a customer review. In this sense, we are achieving a deliverable-driven engineering.

7. Purpose of the Development Document

This framework guides teams in specifying the expectations for each elementary deliverable composing the development document. This document has three objectives for which students are particularly aware and guided throughout the operation:

Participate in the solution elaboration

This document is built up gradually during development, not afterwards as a documentary formalization of the prototype solution developed experimentally. It allows sharing of data between project stakeholders, as acquired during development.

Control the configuration of the developed solution

This document must contain study and definition data needed to identify the development result, for a definition deemed complete, justified, manufacturable, verifiable, operational, testable, reparable, and removable.

Document the project

This document is first established for the development team, to enable them to record and share data from stakeholder requirements to the justified solution. It helps, when asked in the contract, to share data with the customer (prime contractor) for its own needs. Above all, it allows for the company developing this product, to ensure technical archives for several reasons:

- If the project achievement is long-term and there is a need to ensure continuity with the resources (skills) that change (turnover)
- If the project is stopped and must return later with another team to continue without "starting from scratch"
- To capitalize on project achievements and to reuse results for developing future products (essential for product lines and derivatives treatments)

8. Development Document Architecture (downloadable at http://www.robafis.fr)

1) Requirement referential (Deliverable 10)
2) Architectural design (Deliverable 20)
3) Reference configuration (Deliverable 30)
4) Justification of architecture choices and definitions (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)

9. Best Practices for Engineering Quality

1) The analysis of the operational environment and the related systems, source of requirements and constraints complementary to those included in the initial functional specifications.
2) The study of the functional architecture, an essential step for the requirement analysis and the physical architecture definition.
3) The search for candidate architectures and the justified choice of the selected one.
4) Performance allocation to functions, subsystems and elementary components, with values and tolerances including component characteristic dispersions.
5) 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.

Tucoulou et al. continued

6) The realization of a robust solution incorporating functioning margins able to absorb dispersions related to specific constituents or resulting from the integration.
7) A justification guaranteeing the ability to achieve the mission, for all scenarios and for implementation boundary conditions.
8) The integration of the support system engineering in the system-of-interest engineering.

10. Perspectives

The use of yearly feedback improves the competition and continuously strengthens the educational aspect of the operation. 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 (http://www.afis.fr) for viewing and analysis by the students and teachers.

Teachers increasingly use RobAFIS as an example included in training curriculums, allowing students to apply knowledge and expertise learned, an essential condition to assess their ability to implement them. This is the case from the beginning of the competition for master's degrees and engineering schools. New appropriate deployments are expected in 2014 for high schools, preparatory classes, and technician-level training.

References

Fiorèse, S., and J.-P. Meinadier, eds. 2012. *Découvrir et comprendre l'ingénierie système*. Toulouse, FR: Cépaduès éd.

A Systemic Perspective for Mass Customization: An Approach to Defining Product Lines

Antonio Giovannini, antonio.giovannini@univ-lorraine.fr**; Alexis Aubry; Hervé Panetto; and Hind El Haouzi**

Today's customer-oriented market and the stress on performances lead many enterprises to adopt the mass customization (MC) strategy (Davis 1987) as identified by Tseng and Jiao (2001, 684–709). Ideally mass customization is a business approach that provides a product-customization capability like an engineer-to-order organization (the product is designed at each order entry), preserving the mass-production efficiency. In this article, we cope with the formalization of the link between customer needs and the customized product variant that can be configured starting from them. The aim is to propose a way to build this link for formalizing the product variety in the customer domain. This article is based on industrial cases of Trane, a multinational company that develops, manufactures, and commercializes air-handling systems.

In mass customization, a commonly accepted solution for managing product variety is develop a configuration system in which customer can enter values for available options to define a product variant. A configuration system manages various product configurations accordingly with rules formalized by designers. These rules compose a model that represents the product variety. Usually, this kind of product model is named in many different ways, such as product family (Hong et al. 2008), configurable product model (Aldanondo and Vareilles 2008) or product line (Pohl, Bockle, and Van der Linden 2005; Mazo et al. 2012). The name product line (PL) will represent here all these types of models.

Current works on product line can be classified on the distance between the customer needs and the input of the product line (that is, the values of proposed options required by the system to configure a product). Three groups, based on the main system definition models (requirements, logical architecture, technical architecture; see Pyster et al. 2012) can be identified on the basis of this dimension:

- *Product component features* (technical architecture) as input (see Zhu et al. 2008). At this stage the customer (not supposed to be an expert) is not able to understand the interaction between components and is evidently not able to understand their impacts on her needs.
- *Product functions* (logical architecture) as input (see Li et al. 2006). The customer is not able to know how functions can satisfy her needs and especially about how functions interact for doing so.
- *Product specification* (requirements) as input (see Qin and Wei 2010; Hong et al. 2008). The customer is informed about the effects of how the product functions (and so product components) interact for performing the product behaviour, of which specifications are the description; in this case, the customer still is not able to know the impact of the product specification on her needs.

Starting from this classification, we can identify a gap to be filled (figure 1) in order to formalize the link between customer needs and system requirements. There are several ways to fill this gap in the configuration system:

- Manufacturers can use market research (see Helo et al. 2010), but therefore the manufacturer

» continues on next page

designs the products and afterward looks for potential customers.

- Manufacturers can optimize customer satisfaction (see Hong et al. 2008), but that implies they are able to represent and to assess the feeling of satisfaction of each customer.
- A seller is charged to fill this gap, performing a real requirement-analysis process (Haskins 2011).
- In the worst (but very usual) case, customers have to fill this gap.

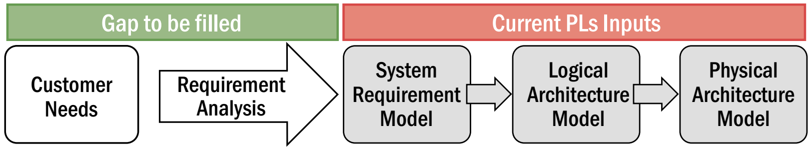


Figure 1. The gap of current PL approaches

Other approaches from the business sciences (Bergvall-Kareborn et al. 2009; Stahlbröst 2008) involve customers in each design stage. But, for instance, let us consider the case of an air-conditioner manufacturer: the question to put to the customer is, What kind of air do you want to breathe? In our opinion, customers are not able to answer this kind of question. We are convinced that customers have to be expert only on their environment. Therefore the question to be posed is, What kind of room do you want to heat? Therefore, in this work, we propose a method to build product lines that formalizes the variety directly in the customer domain, that is, the properties of the environment.

Here we propose a method based on definitions of the system, environment, interaction, purpose, and the "black box" coming from cybernetics and system theory (Giovannini et al. 2013). Starting from the system's purpose (a predefined air quality), an engineer can identify needed interactions (based on some principles of physics) between the purpose's fulfilment and the system's features (the goal of the configuration process). Here the system to be configured is seen like a black box, because in the mass customization scenario, we suppose that the link from the system specifications (features) and the manufactured product (how to manufacture offered products) is built during the design stage (and not at each order entry). In order to represent a PL we use a mathematical representation typical of control theory, state-space models (Levine 1996), as follows:

- The system (in state-space models) is the customer environment (customer room to heat).
- States functions model a parametric interaction (representing the PL variety) between the environment description (built taking into consideration information from the customer) and configured system specifications (stationary features of the air-conditioner).
- Input functions represent the system specifications that can be managed dynamically.
- The output function is based on state values in an instant of time.
- A set of defined state values (room temperature, humidity) or one output value (air comfort) defines the purpose of the configured system.

In order to define this model, a well-defined purpose (values for humidity and temperature of the customer room) has to be identified. By means of the domain knowledge, the engineer identifies the parametric interactions between the customer environment (e.g., room orientation and placement in the building, room usage) and the system specifications (features of the air conditioner to be configured). These interactions have to be formalized as state functions. At each order entry, starting from customer information, values for environment parameters are determined. Starting from these, system specifications are defined by the values that allow achieving the purpose taking into account the environment features formalized in the states functions.

Deploying this method, a customer has only to describe its environment to configure the right product (from the engineering point of view). Our future work will address for formalisation of the approach, based on an extension of the ONTO-PDM product ontology (Panetto et al, 2012), and validating this approach on a real case from Trane.

References

Aldanondo, M., and E. Vareilles. 2008. "Configuration for Mass Customization: How to Extend Product Configuration Towards Requirements and Process Configuration." *Journal of Intelligent Manufacturing* 19 (5): 521–535.

Bergvall-Kareborn, B., M. Hoist, and A. Stahlbrost. 2009. "Concept Design with a Living Lab Approach." In *42nd Hawaii International Conference on System Sciences HICSS'09*, 1–10. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4755508.

Davis, Stanley M. 1987. *Future Perfect*. Reading, US-MA: Addison-Wesley.

Giovannini A., A. Aubry, H. Panetto, and H. El Haouzi. 2013. "Mass Customisation in Sustainable Networked Enterprises." Paper persented at the 14th IFIP Working Conference on Virtual Enterprise, Collaborative Systems for Reindustrialization, Dresden, DE.

Haskins, C. 2011. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 3.2.1 Revised by M. Krueger, D. Walden, and R. D. Hamelin. San Diego, US-CA: INCOSE.

Helo, P. T., Q. L. Xu, S. J. Kyllönen, and R. J. Jiao. 2010. "Integrated Vehicle Configuration System—Connecting the Domains of Mass Customization." *Computers in Industry* 61 (1): 44–52.

Hong, G., L. Hu, D. Xue, Y. L. Tu, and Y. L. Xiong. 2008. "Identification of the Optimal Product Configuration and Parameters Based on Individual Customer Requirements on Performance and Costs in One-of-a-kind Production." *International Journal of Production Research* 46 (12): 3297–3326.

Levine, W. S., ed. 1996. *The Control Handbook*. Boca Raton, US-FL: CRC Press.

Li, B., L. Chen, Z. Huang, and Y. Zhong. 2006. "Product Configuration Optimization Using a Multiobjective Genetic Algorithm." The *International Journal of Advanced Manufacturing Technology* 30 (1): 20–29. doi:10.1007/s00170-005-0035-8.

» continues on next page

Giovannini et al. *continued*

Mazo, R., C. Salinesi, D. Diaz, O. Djebbi, and A. Lora-Michiels. 2012. "Constraints: The Heart of Domain and Application Engineering in the Product Lines Engineering Strategy." *International Journal of Information System Modeling and Design* 3 (2): 33–68.

Panetto H., M. Dassisti, A. Tursi. 2012. "ONTO-PDM: Product-Driven ONTOlogy for Product Data Management Interoperability within Manufacturing Process Environment." *Advanced Engineering Informatics* 26 (2): 334-348. doi: 10.1016/j.aei.2011.12.002.

Pohl, K., G. Bockle, and F. van der Linden. 2005. *Software Product Line Engineering: Foundations, Principles, and Techniques*. Vol. 10. New York, US-NY: Springer. http://cs5292.userapi.com/u11728334/docs/7ca820593ad0/Klaus_Pohl_Software_Product_Line_Engineering.pdf.

Pyster, A., D. Olwell, A. Squires, N. Hutchison, S. Enck, and J. Anthony. 2012. "A Guide to the Systems Engineering Body of Knowledge (SEBoK). Version 1.0." http://www.sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_%28SEBoK%29.

Qin, Y. H., and G. X. Wei. 2010. "Product Configuration Based on Modular Product Family Modelling." *Journal of Computational Information Systems* 6 (7): 2321–2331.

Staahlbröst, A. 2008. "Forming Future IT: The Living Lab Way of User Involvement." Doctoral diss., Luleå tekniska universitet, Luleå, SE. http://epubl.ltu.se/1402-1544/2008/62/index-en.html.

Tseng, M. M., and J. Jiao. 2001. "Mass Customization." *Handbook of Industrial Engineering*. 3rd ed. New York, US-NY: Wiley.

Zhu, B., Z. Wang, H. Yang, R. Mo, and Y. Zhao. 2008. "Applying Fuzzy Multiple Attributes Decision Making for Product Configuration." *Journal of Intelligent Manufacturing* 19 (5): 591–598. doi:10.1007/s10845-008-0132-2.

Requirements Engineering Process according to Automotive Standards in a Model-Driven Framework

Morayo Adedjouma, morayo.adedjouma@uni.lu; Hubert Dubois; and François Terrier

A leading objective for automotive manufacturers and suppliers today is the control of quality and the dependability of embedded and mechatronic systems. Standards such as the Automotive SPICE (Software Process Improvement and Capability Determination, http://www.automotive-his.de) and ISO 26262 (ISO 2011) particularly constrain this objective. The goal is to produce a well-defined and standard compliant system, but also the relevant system. As a consequence, achieving this goal leads to greater consideration of the requirements.

Our goal is to address, following a model-based approach, the challenge of developing automotive embedded systems that comply with standards and meet the user's needs. We proceed in various steps that are subsequently used jointly as shown in figure 1. In the first phase, a merging approach that integrates product and process quality assessment is defined. In a second phase, we define a metamodel for managing safety assets regarding these automotive standards at the product level. This metamodel defines how the requirements and architecture of a system can be captured in such a way that they can be traceable from each other and from origin specification documents. Finally, we develop a model-based approach to support project management, where the interaction of process and product models is managed to address requirements identified in the preceding phases. The approach uses tailoring and measurement to improve the control and the monitoring of project and to reduce the cost and frequency of replanning.

ISO 26262 and SPICE Standards in a Unified Process Certification Approach

The ISO 26262 and SPICE standards have the aim of standardizing the development of safety-critical automotive embedded systems to manage their increasing complexity. A study of both of them was performed to extract safety concepts and processes rules relevant to the automotive field (Petry 2009), in particular from the "Specification and Management of Safety Requirements" chapter of ISO 26262-8 (chapter 6), and we analysed the first Automotive SPICE Engineering Process Groups: requirements elicitation, system requirements analysis and system architecture design process.

We propose then an extended metamodel (Adedjouma et al. 2012) to describe the two standards in a common framework without altering their respective contents. For the purpose of assessment, we propose a generic methodology where a SPICE assessment and a functional safety audit are simultaneously performed in a certification perspective. The main undertaking is the definition of a framework where we apply an assessment method to the common metamodel defined first. We propose a SPICE-based model approach due to past experiences in the automotive industry (Lami, 2011). The notion of SPICE rating scale allows us to determine the maturity of a product regarding compliance with automotive standards. Thus, systems engineers can evaluate the adequacy of their systems to standards and assess the maturity level they have achieved (Adedjouma et al. 2012).

The major benefit of our proposal of such an integrated assessment process is that it reuses the practices already present in the industry and thus eliminates the effort necessary to introduce of the new standard. For modeling this metamodel, we choose SPEM (Software and Systems Process Engineering Metamodel Specification). SPEM allows us to use some general concepts as activity, process, task, tool, role, work product, and example, that match some of those present in our metamodel such as clause, requirement, tool, human resource, work product, and note. However, SPEM does not cover other specific concepts of the automotive standards, such as ASIL (Automotive Safety Integrity Level), BasePractice, the rating scale, methods tables or properties tables. By extending the SPEM metamodel with these concepts, we define a metamodel that includes all concepts of the standards common metamodel. The instantiation of the metamodel allows us having a generic organizational process compliant to automotive standards in the center of our efforts: it is a process model.

Process Configuration to Project-Specific Context

From the generic process model, we define an approach to tailor it into specific processes. We consider in the approach the context and characterization of specific projects in order to create effective and efficient products. An example of such characteristics is the ASIL level reached, the size and the complexity of the project, the technology involved, the engineers involved, and the recommendation level of methods and properties desired depending on the ASIL classification. In a particular tailoring, these characteristics use attribute values in the source process model element to determine the elements to be considered in a specific process.

Since this tailoring process is intended to be automatic, it is expected to achieve a reduction of the tuning time and cost with fewer adaptation errors as only process elements that are required for the particular project context are considered. In addition, high quality can be expected, because the process is adjusted with the goal in each particular project context.

Specification of Requirements Engineering Activities Considering Safety Aspects

Quality of product is also significant during the certification. One necessary objective was to identify in the different standards that are applicable as part of requirements management. We develop the UML-based ReMIAS profile to accomplish the identification. We define, through the profile, various means and methods of producing these automotive standards requirements that afford compliance between automotive standards and system requirement engineering. We use dedicated languages based on UML profiles and combine them into a more complete and still consistent language (Noyrit et al. 2010): SysML and DARWIN for the requirements specification part; parts of EAST-ADL2 extended for design specification and for verification and validation. The traceability is ensured using traceability links inherited from DARWIN (Adedjouma et al. 2010). We have also defined an Eclipse plugin for automatic import of requirements specification document requirements into the modeling environment: Office2Papyrus (Adedjouma et al. 2011).

Measurement of Product and Process Quality

Application of measurement is a prerequisite and an excellent tool to guide process improvement, because it provides feedback on the effects of process actions on product quality. An integration of the process and product properties was first managed using a work-product elements extension realized in SPEM. The extension

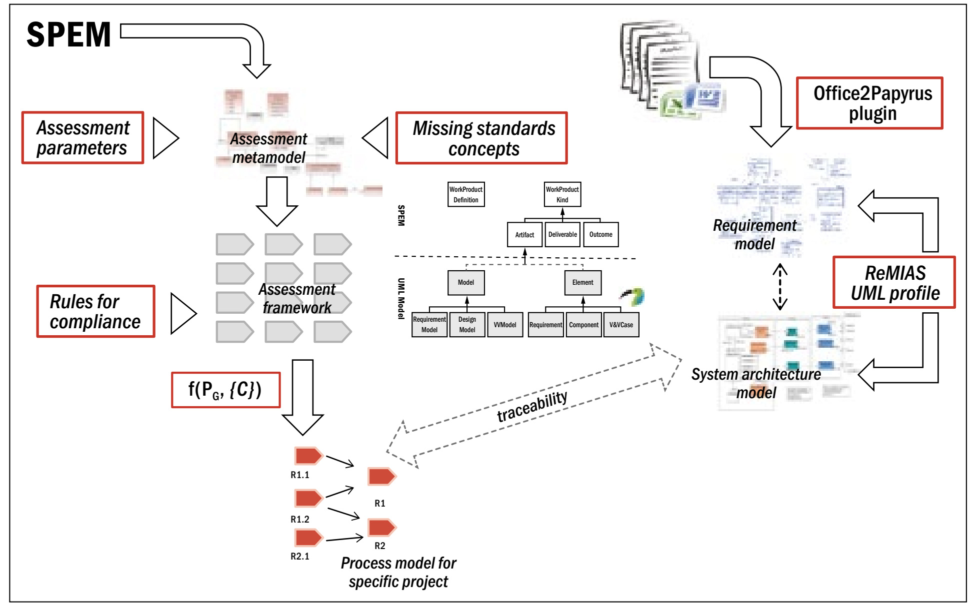


Figure 1. Merging, modeling, configuring, tracing in the automotive certification approach

allows us to consider the product-model elements as work products in the process model. For the assessment, they support the definition of measures as progress metrics that are compliant with measurement standards for different tasks in a development process as well as the definition of milestone goals for general milestone plans. A measure is linked to one or more work products and can be linked to a task. An indicator provides the possibility of using more complex assessment models to assess the progress of certain task. Measure and indicator are defined through a graphical interface as OCL (Object Constraint Language) queries that will be interpreted at runtime (EU Project CESAR 2010). Once a process including its measurement procedures is defined, it is possible to compose whole specific development processes. At this stage, the previously described approach to configuring the process to the project-specific context is applied. We map the resulting process obtained into project plans, enacting these plans with project planning and enactment systems. The main integral part between the project plans and the actual development is the concrete instance of the system or component the plan is defined for. For example, we can create a standards-compliant project plan for a selected system or subsystem according to a given configured process. Through an execution, these representations help to follow the completeness of process activities and the monitoring of the project. For that, the project plan, linked to the process and its measurement definition is exported to MS Project where we assign durations and milestone deadlines. The content of the different work products is calculated dynamically based on the concrete

» continues on next page

Adedjouma et al. *continued*

component and the semantics of the work product types. The assessment against the current development data gives results such as progress of tasks and milestone achievement, progress over time for metrics, quality status of different work products from the system model (artefact outputs from ReMIAS model), and traceability from process work products to concrete development artefacts.

References

Adedjouma, M., H. Dubois, K. Maaziz, and F. Terrier. 2010. "A Model-Driven Requirement Engineering Process Compliant with Automotive Domain Standards." Paper presented at the third workshop on Model-Driven Tool and Process Integration (in conjunction with ECMFA 2010).

Adedjouma, M., H. Dubois, and F. Terrier. 2011. "Requirements Exchange: From Specification Documents to Model." Paper presented at the ICECCS conference, Las Vegas, US-NV.

———. 2012. "Merging the Quality Assessment of Processes and Products in Automotive Domain". Paper presented at the PROFES conference, Madrid, ES

EU Project CESAR. 2010. "Requirement/Traceability Management: D_SP2_R3.2_M2." Vol. 3, V0.002. 2010.

ISO (International Organization for Standardization). 2011. ISO 26262. *Road vehicles – Functional safety*. Geneva, CH: ISO.

Lami, G. 2011. "ISO/IEC 15504-10 Safety Extension, Yet Another Safety Standard?" Paper presented at the 8th Automotive SPIN Italia Workshop, Milan, IT.

Noyrit, F., S. Gerard, F. Terrier, and S. Selic. 2010. "Consistent Modeling using Multiple UML Profiles." Paper presented at the MODELS conference, LNCS, Oslo, NO.

Petry, E. 2009. "Automotive SPICE® & ISO/CD 26262, Their Mutual Relationship." Paper presented at the 5th Automotive SPIN Italia Workshop, Milan, IT.

A Methodology for Defining Security Requirements using Security and Domain Ontologies

Amina Souag, amina.souag@malix.univ-paris1.fr; Camille Salinesi; and Isabelle Comyn Wattiau

Security is the discipline concerned with protecting systems from a wide range of threats (malice, error, or mischief) that break a system by exploiting a vulnerability. A vulnerability is a property of a system or its environment that, when faced with particular threats, can lead to failures (Anderson 2010). Security is a multifaceted problem; it is as much about understanding the domain in which systems operate as it is about the systems themselves. While it is important to develop security features such as encryption, identity control, or specific architectures, our attention should be drawn toward the sociotechnical context in which target systems will operate and threats that may arise and their potential harm, so as to uncover security requirements. Researchers have recently been debating the importance of considering security at the early stages of the information systems development process, and especially the need to consider security during requirements engineering.

An ontology, in the field of knowledge representation, is most often defined as "a representation of a conceptualization" (Gruber 1995). It should represent a shared conceptualization in order to have any useful purpose (Dobson and Sawyer 2006). Ontologies are useful for representing and interrelating many types of knowledge. Domain ontologies are formal descriptions of classes of concepts and relationships between these concepts that describe a given domain.

Several security ontologies have been proposed (Fenz and Ekelhart 2009; Herzog et al. 2007). In the literature, some methods consider the use of ontologies for security requirements engineering (Daramola, Sindre, and Moser 2012; Velasco et al. 2009). RITA (Salinesi, Ivankina, and Angole 2008) is a requirements elicitation method that exploits "just one threat" ontology. Our previous experience with RITA revealed that, "being generic, the threats in the RITA ontology are not specific to the target [bank] industry" (the case study was in the banking sector). Experts involved in the evaluation complained about "the lack of specificity of the types of threats to the industry sector and the problem domain at hand." The problem that remains open is therefore that we need to exploit both security knowledge and domain knowledge to guide the elicitation of domain-specific security requirements. Our research question concerns how to combine the use of security ontologies and domain ontologies to guide requirements elicitation efficiently.

This article presents an ongoing research project that aims to develop a method for exploring the use of security and domain ontologies applicable to security requirements engineering. The approach is generic in the sense that different security ontologies and different domain ontologies can be used with it. However it is domain-specific when it is applied in the sense that during its application only one domain ontology is used.

Our method guides the discovery of security requirements for a specific domain at the very early stages of system development, while defining security requirements according to stakeholders' security goals. This process is handled by a series of heuristic production rules. Starting from high-level security requirements (security goals), the rules produce a security requirements specification. Figure 2 shows an overview of our method. There are two subsets of rules. The first set of rules handles domain-specific analysis. The second set of rules performs a security-specific analysis. Each set of rules exploits different ontologies, respectively, domain ontologies and security ontologies. In order to be able to handle different security and domain ontologies, the rules were specified with so-called "upper ontologies" that handle concepts that (a) are common to most ontologies, (b) are sufficiently high-level to abstract many other concepts in the specific ontologies, and (c) more importantly, represent an important subject of interest for the method.

» continues on next page

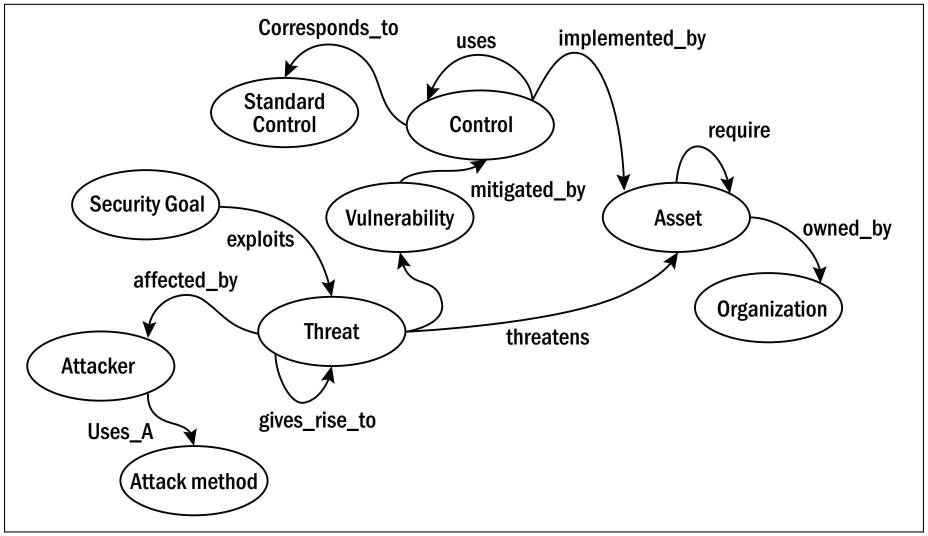


Figure 1. The upper security ontology

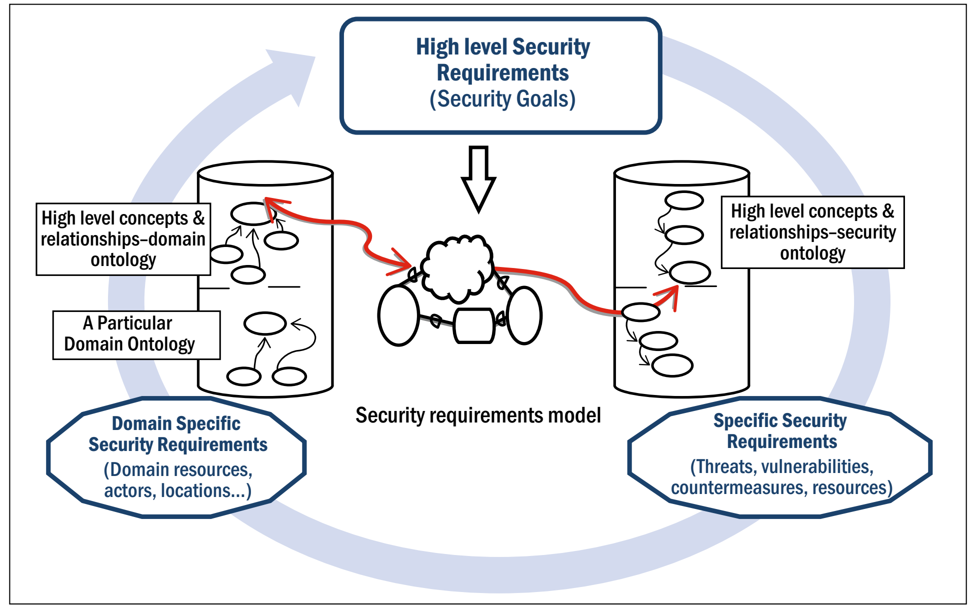


Figure 2. Overview of the method

Figure 1 presents a part of the upper security ontology used in the approach. A threat gives rise to follow-up threats; it represents a potential danger to the organization's assets and affects specific security goals (such as confidentiality, integrity, and availability) as soon as it exploits vulnerability in the form of a physical, technical, or administrative weakness; and it causes damage to certain assets. For each vulnerability, the asset on which the vulnerability could be exploited is assigned. Controls have to be implemented to mitigate an identified vulnerability and to protect the respective assets. Each control is implemented as an asset concept, or as combinations thereof. Controls are derived from and correspond to best practices and information security standard controls, such as the German IT Grundschutz Manual (BSI 2004; ISO/IEC 27001).

The requirements-definition process starts with the elicitation step, where stakeholders express their needs about security in nonformal sentences. Then an analysis stage is carried out to discover more requirements and express these needs in semiformal requirements.

During the elicitation step, an initial requirements model is first constructed from the stakeholders' needs and concerns expressed about security at the beginning of the project. At this stage, the analyst defines initial actors, resources, and especially security goals (such as integrity, confidentiality, and availability). During the security-requirements analysis stage, the production rules will exploit the security-specific ontology to discover threats, vulnerabilities, countermeasures, and resources, and thus enrich the requirements model by adding new elements (like malicious tasks and vulnerability points). During the domain-specific security-requirements analysis stage, another set of rules explores the domain ontology to improve the requirements model with resources, actors, and other concepts that are more specific to the domain at hand; for instance: thieves in the banking domain, hijackers in the aeronautic domain, and pirates in the maritime domain.

Each rule is described under the form <S→ C>, where S is a situation and C a conclusion. <S→ C> means that if the situation S is true, then conclusion C can be drawn. The situation holds on an input model and input ontology. Situation is defined using first order logic predicate that relies on two kinds of functions:

1. EquivalentClass (X, Y) is true if X in the input model has the same semantics (meaning) as the concept Y in the input ontologies.
2. OntologyLink (Type, X, Y) is true if in the input ontology there is a link from X to Y that has the type <Type>. For instance, OntologyLink (IsAffectedBy, X, Y) is true if in the ontology, X and Y are related by an "affects" link from Y to X.

Conclusions indicate elements that should be added to the output model. They are specified with three kinds of functions:

1. CreateClass(ClassC, X) indicates that a concept X that instantiates the <ClassC> class can be created in the model.
2. CreateLink(LinkTypeL, X, Y) indicates that a link from X to Y, and of type <LinkTypeL>, can be created in the model.
3. ReplaceClass(ClassC, X), where ClassC belongs to a model, X to an ontology: indicates that ClassC will be replaced by the concept X. This function is used to get more precision from the domain ontology.

The originality of the method lies first in the fact that the combination of security and domain ontologies is not achieved *a priori*, but at runtime, while the method is applied (see Souag et al. 2013) and, being structured and equipped with reasoning features, they form a powerful tool to handle requirements. We believe that since security is a multifaceted problem, a single security ontology is not enough to guide security-requirements engineering. Second, the method is original in being generic, in the sense that it is designed to be used with any pair of security and domain ontologies.

Our preliminary evaluation conducted through a small, but real, case study and through critical analysis by three experts (domain, security, and requirements engineering). The evaluation shows that the method provides a good balance between the genericity with respect to the ontologies (which do not need to be selected in advance), and the specificity of the elicited requirements with respect to the domain at hand.

References

Anderson, R. J. 2010. *Security Engineering: A Guide to Building Dependable Distributed Systems*. New York, US-NY: Wiley.

BSI (Bundesamt für Sicherheit in der Informationstechnik). 2004. *IT Grundschutz Manual*. Bonn, DE: BSI.

Daramola, O., S. Guttorm, and M. Thomas. 2012."Ontology-Based Support for Security Requirements Specification Process." Paper presented at the "On the Move to Meaningful Internet Systems" OTM 2012 Workshops. Berlin, DE: Springer.

Fenz, S., and A. Ekelhart. 2009. "Formalizing Information Security Knowledge." Paper presented at the Fourth International Symposium on Information, Computer, and Communications Security.

Glen, D., and P. Sawyer. 2006. "Revisiting Ontology-Based Requirements Engineering in the age of the Semantic Web." Paper presented at the International Seminar on Dependable Requirements Engineering of Computerised Systems.

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.

Herzog, A., N. Shahmehri, and C. Duma. 2007 "An Ontology of Information Security." *International Journal of Information Security and Privacy* 1 (4): 1–23.

Lasheras, J., R. Valencia García, J. T. Fernández Breis, and A. Toval. 2009. "Modelling Reusable Security Requirements based on an Ontology Framework." *Journal of Research and Practice in Information Technology* 41 (2): 119.

Salinesi, C., E. Ivankina, and W. Angole. 2008. "Using the RITA Threats Ontology to Guide Requirements Elicitation: An Empirical Experiment in the Banking Sector." Paper presented at the First International Workshop on Managing Requirements Knowledge.

Souag, A., A. Salinesi, I. Wattiau, and H. Mouratidis. 2013. "Using Security and Domain Ontologies for Security Requirements Analysis". Paper presented at the Computer Software and Applications Conference Workshops (COMPSACW), 26–30 July.

Evaluating Alternatives for Designing Mechatronic Systems in a Systems Engineering Context

Mambaye Lo, Pierre Couturier, and Vincent Chapurlat, vincent.chapurlat@mines-ales.fr

The system-analysis process allows "developers to objectively carry out quantitative assessments of systems in order to select and/or update the most efficient system architecture and to generate derived engineering data" (Pyster and Olwell 2013). The goal is here to perform trade-off studies. We focus on the evaluation of costs, risks, or effectiveness of alternative system-design solutions provided by the technical systems engineering processes (ISO/IEC 2008) considering two main constraints. First, the iterative aspect of the design approach induces a growing but uncertain maturity level of detail of the proposed solutions. Second, each decision resulting from evaluations must be argued considering the stakeholder's requirements. It is particularly crucial to note here that these requirements can be contradictory. They can also be relative to various domains and cultures such as mechanics, electronics, or computer science. The proposed work aims at giving a methodological guide with embedded tools and innovative decision making methods to support the effectiveness evaluation of alternative system-design solutions. It is here applied in the field of analyzing mechatronic systems. This guide is the subject of methodological, conceptual, and technical contributions summarized in the next sections.

Methodological Contribution: A Generic Set of Evaluation Activities

The proposed activities are here considered as generic and iterative activities for evaluation purposes in the system-analysis process. They are gathered into four main sets of activities for evaluation:

- Defining the objectives of the evaluation and selecting the solutions to be evaluated
- Preparing the job to be performed by defining a decision model, selecting applicable multicriteria analysis methods (Fülöp 2005) and tools, and then selecting required data extracted from design models
- Performing the job including sensitivity and traceability analyses
- Delivering expected results, justifications, and recommendations

Conceptual Contribution: A Data Model for Evaluation

The goal is to provide a unified data model (figure 1) that can be understood and shared by designers regardless of their origin or business domain. This model formalizes concepts and relations between concepts that represent classes of data required during evaluation. The model integrates the perspectives of the stakeholders, system designers, and design evaluators, as promoted by Blanchard and Wolter (2011), Haskins (2011), and Maier and Rechtin (2009). Stakeholders are interested in technical requirements, measures of effectiveness, and measures of performance. Designers are interested in alternative system-design solutions, predictive models

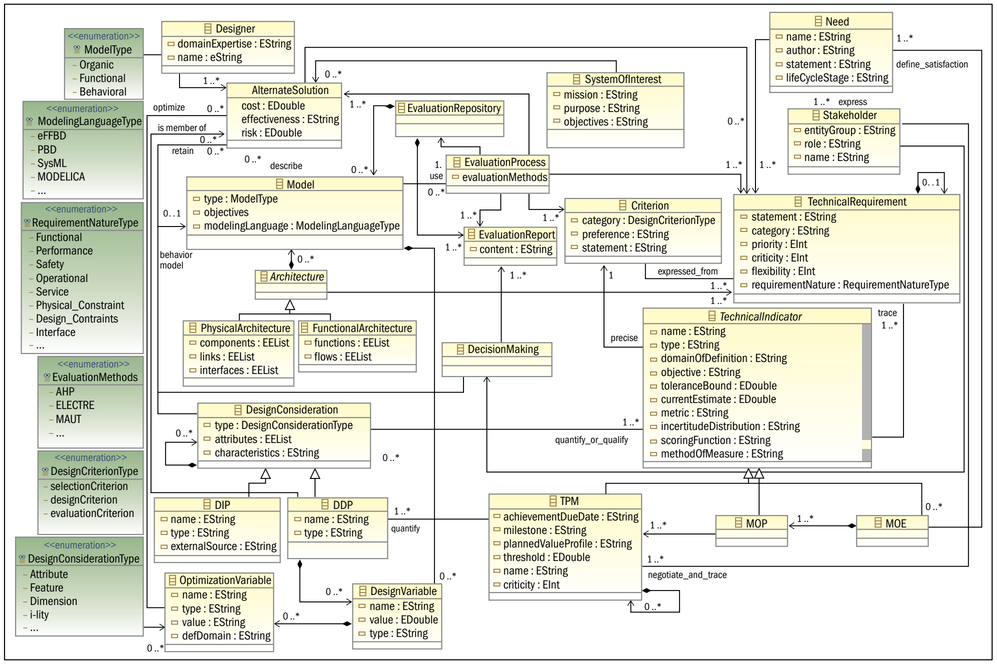


Figure 1. Evaluation data conceptual model (simplified)

as described below, design considerations, and design-dependent parameters (that is, parameters that can be split up into input and output parameters, abbreviated iDDP and oDDP). Last, evaluators are interested by system architecture, evaluation criteria, and associated technical indicators.

Conceptual and Technical Contribution: Impact Evaluation and Evaluation Traceability

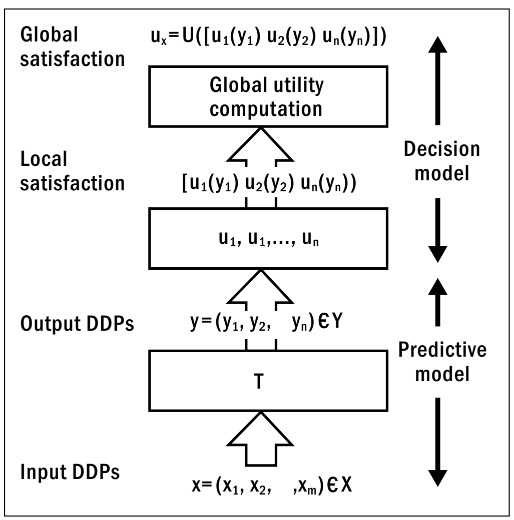


Figure 2. Merging decision model and predictive model

The goal is to support evaluation activities and systematically trace design choices with a certain level of automation. Relying on the relationships, it is then possible to identify the possible influence of design-dependent parameters (DDPs) on the criteria. Then, a decision model is merged with a predictive model (figure 2) in order to estimate the magnitude of such influences.

This figure shows how a "predictive model" is used in order to predict the values of oDDPs based on the values of iDDPs. The "decision model" computes the global satisfaction level of each alternative system-design solution, applying multicriteria analysis methods such as using utility functions. However, the transformation from iDDPs to oDDPs may be unknown or marred by uncertainty. This mainly arises during the preliminary (or conceptual) design phase. Therefore we propose an original approach based on qualitative influence analysis to deal with this uncertainty issue. This approach consists of asking experts to advise on qualitative influence (improvement or degradation) of iDDP choices on the value of utility function applied to each oDDP. It then becomes possible to highlight the most promising alternative system-design solutions since the earliest stage of the design. For this, we adopt aggregation and propagation operators from Imoussaten and others (2011) and Giorgini and others (2002).

Research and Development Perspectives

There is a notable lack of interoperability between systems engineering tools and multicriteria analysis tools. The proposed contributions have been developed in order to be interoperable with classical systems engineering tools. This is achieved by enriching the system engineering metamodel with the proposed conceptual model for evaluation data. The approach has been tested particularly on the CORE V8 tool (Long and Scott 2011). A software platform supporting the entire guide is under development. The goal is now to enrich contributions by taking into account other -ilities evaluation, as proposed by De Weck, Ross, and Rhodes (2012).

References

Blanchard, B. S., and J. F. Wolter. 2011. *Systems Engineering and Analysis*. 5th ed. Boston, US-MA: Prentice Hall.

De Weck, O. L., A. M. Ross, and D. H. Rhodes. 2012. "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)." Paper presented at the Third International Engineering Systems Symposium, Delft University of Technology, Delft, NL.

Fülöp, J. 2005. "Introduction to Decision Making Methods." Report to the Workshop on Biodiversity & Ecosystem Informatics, The Evergreen State College, Olympia, US-WA, 13–15 Dec. 2004. http://academic.evergreen.edu/projects/bdei/documents/decisionmakingmethods.pdf.

Giorgini, P., J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani. 2002. "Reasoning with Goal Models." Paper presented to the 21st International Conference on Conceptual Modeling, Tampere, FI.

Haskins, C. 2011. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 3.2.1 Revised by M. Krueger, D. Walden, and R. D. Hamelin. San Diego, US-CA: INCOSE.

Imoussaten, A., J. Montmain, F. Trousset, and C. Labreuche. 2011. "Multi-Criteria Improvement of Options." In *Proceedings of the 7th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011) and LFA-2011*, 1030–1037. Aix-les-Bains, FR: Atlantis Press.

ISO/IEC (International Organization for Standardization and International Electrotechnical Commission). 2008. ISO/IEC 15288. *Systems engineering—System life cycle processes*. 2nd ed. Geneva, CH: ISO.

Long, D. A., and Z. Scott. 2011. *A Primer for Model Based Systems Engineering*. 2nd ed. Blacksburg, US-VA: Vitech Corporation.

Maier, M. W., and E. Rechtin. 2009. *The Art of System Architecting*. 3rd ed. Boca Raton, US-FL: CRC Press.

Pyster, A., and D.H. Olwell, eds. 2013. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*. Version 1.1.2. Hoboken, US-NJ: The Trustees of the Stevens Institute of Technology. http://www.sebokwiki.org/.

Towards an Integrated Approach of Safety Analysis for Mechatronic Systems Design

Faïda Mhenni, faida.mhenni@supmeca.fr; Jean-Yves Choley; and Nga Nguyen

The combination of SysML models and well-mastered safety analysis techniques in a unified framework will enable system engineers to automate the more time-consuming activities for assessing and enforcing a specified level of safety, security, and reliability for complex systems.

Mechatronic systems involve, by definition, knowledge from various disciplines such as computer science, mechanics, electronics, and automation, among others. They take an increasing role in the design of industrial products in various fields such as automotive, aerospace, and medical equipment. The increasing complexity of these systems makes their development and safety analysis more difficult because of the various interactions between multidomain components exchanging different types of flows.

The increasing complexity of mechatronic systems as well as their multidisciplinary aspect, require a model-based systems engineering (MBSE) approach to manage the complexity, enhance consistency and allow modeling and simulation of the whole system at early design stages with different abstraction levels. MBSE is an efficient approach to specify, design, simulate, and validate complex systems. This approach allows errors to be detected as soon as possible in the design process, and thus reduces the overall cost of the product.

In an MBSE approach, engineers from different fields and with different technological backgrounds are asked to cooperate during the design process. This usually leads to misunderstanding and confusion. A unified language to model and specify the system is necessary to tackle communication issues between the different work team members.

SysML is known as a semiformal modeling language that allows engineers to document the properties from different disciplines to specify, analyse, design, verify, and validate complex systems. This OMG standard is becoming more and more supported by industry because it provides a consistent, well-defined, and well-understood language to communicate the requirements and corresponding designs among engineers. For all these reasons, SysML is chosen to be our support modeling language. It is worth noting, however, that the SysML model is used as a common reference for all stakeholders and the other tools are needed during the development process like CAD, simulation, and domain-specific tools at the component level. In the case of large set of requirements, other tools can be used for requirements management.

Human-made systems can have erroneous behaviour due to components faults or design errors. These malfunctions may lead to serious damage on both humans and equipment. Potential risks should be thoroughly addressed during design stages via safety analyses and eliminated or brought to acceptable levels. However, safety-analysis tasks are usually performed once the design is at an advanced stage and once most of the design choices are already fixed. This could lead to major design changes, implying both delay and cost increases. They are also made with separate tools and may be based on obsolete design models.

The combination of SysML models and well-mastered safety analysis techniques in a unified framework will enable system engineers to automate the more time-consuming activities for assessing and enforcing of a specified level of safety, security, and reliability for complex systems.

There has been some research into the integration of safety analysis in the early design stage, in both industrial and academic domains. Thomas and Belmonte (2011) use Eclipse as a common framework to integrate SysML and safety analysis. In this work, an independent tool called Obeo Designer that implements classical risk analyses is developed. Then the interoperability of this modeling tool and the SysML model is achieved through the Eclipse Modeling Framework.

David (2009) and Cressent and others (2013) present a methodology named MeDISIS (standing for "integration method of reliability analysis in the systems engineering process"). In these works, a functional FMEA (failure modes and effect analysis) report is automatically generated from system functional behaviour written in SysML models and is completed by a safety expert. A Dysfunctional Behaviour Database is created to support this methodology and to store and manage the relevant data. Dysfunctional models are then computed in specific tools or DSL such as AltaRica (Cressent 2013). Jaber, Yakymets, and Lanusse (2012) set the focus on the use of UML and SysML for further model checking and fault tree generation.

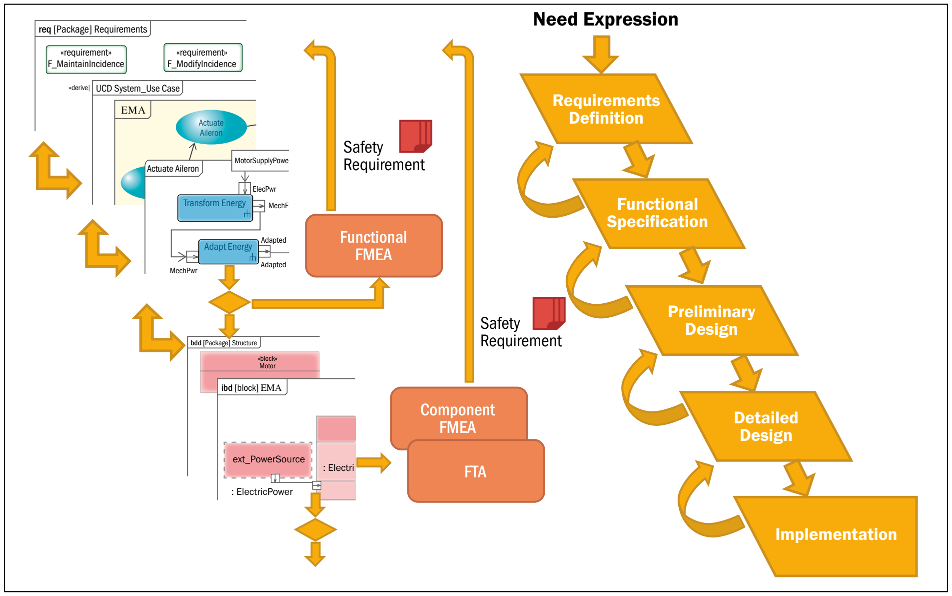


Figure 1. Integrated approach of safety analysis within the design process

In the following, an integrated methodology for the design and safety analysis of mechatronic systems is presented. The proposed methodology integrates both FMEA (at the functional and component level) and fault-tree analyses (FTA). At each stage of the design process, design model elements with SysML as well as safety analysis techniques to be used are briefly described.

Integrated Approach

The proposed integrated approach is presented in figure 1. In the right hand side, the design steps are given in the top-down phase of the V development process. In the left hand side, the SysML models for each step of the design and the corresponding safety analysis technique are given. As highlighted in figure 1, the whole process is iterative.

A new product development process usually starts with requirements definition where different stakeholder requirements are captured. In our approach, requirements are captured via SysML requirements and their dependencies are modeled in SysML requirements diagrams (figure 1). In the next step, system functions are modeled using SysML use cases and then a functional breakdown is performed with SysML activity diagrams that describe how input flows are progressively transformed into output flows. A list of functions is then available and, since this early stage, a functional safety analysis (using Functional Hazard Analysis or functional Failure Mode and Effects Analysis for example) can already be performed to assess the local and system level effects of each function failure. Critical systems functions are then identified in this step by ranking the functions according to the severity of their failure consequences. Design modifications can be performed at this stage to suppress or mitigate identified potential risks. Safety-analysis results are integrated in the main model as “safety requirements” added to the initial requirements model and traced to relevant requirements. The process then iterates to take into account the model modifications and check the consistency of the whole model.

Next, different system structures are defined by allocating components to functions, and alternative solutions are compared. Integrating safety analysis at this level allows system designers to consider safety among the criteria of the trade-off study. Component FMEA can be applied to each component of the system (usually included in a block definition diagram describing the system composition in the SysML model). Failure propagation among components can be described in a fault tree established based on the internal block diagram, which illustrates the interactions between the different components in the SysML model. In the same way, derived safety requirements are added to the requirements model and the process iterates to integrate the design changes, if they occur and check the whole model consistency.

Based on these studies, the design may be modified to take into account safety aspects from early design stages, minimizing late and costly design changes. Since they are based on the system model, safety analyses are performed on the up-to-date model. More detail about the first part of this process can be found in a paper by Mhenni and others (2012).

Conclusion and Future Works

In this article, a methodology to integrate safety analyses within an MBSE approach based on SysML was established. This methodology helps engineers taking into account safety aspects through the whole design process and thus reduces late design changes that are very expansive. First, safety critical functions are identified with the functional FMEA. Then, component failures are identified. The failure propagation is given by a fault tree.

In the next steps, this methodology will be tested on some real-life examples to be validated. Then a specification to extend SysML, in order to better support the methodology and automate parts of it will be developed. Further safety-analysis techniques can be added to this approach.

References

Cressent, R., P. David, V. Idasiak, and F. Kratz. 2013. “Designing the Database for a Reliability Aware Model-Based System Engineering Process.” *Reliability Engineering and System Safety* 111: 171–182. http://dx.doi.org/10.1016/j.ress.2012.10.014.

» *continues on next page*

Mhenni et al. *continued*

David, P. 2009. “Contribution à l’analyse de sûreté de fonctionnement des systèmes complexes en phase de conception : application à l’évaluation des missions d’un réseau de capteurs de présence humaine.” PhD diss., Université d’Orléans, FR.

Jaber, H., N. Yakymets, and A. Lanusse. 2012. “Model Based System Engineering for Safety Analysis of Complex Systems: The Benefits of UML Profile Mechanisms Implemented in Papyrus.” Paper presented at the Model-Based Safety Assessment Workshop, Bordeaux, FR, 11–12 Sept.

Mhenni, F., J.-Y. Choley, A. Rivière, N. Nguyen, and H. Kadima. 2012. “SysML and Safety Analysis for Mechatronic Systems.” Paper presented at the IEEE 9th France-Japan Congress, 7th Europe-Asia Congress, 13th International Workshop on Research and Education in Mechatronics, Paris, FR, 21–23 Nov.

Thomas, F., and F. Belmonte. 2011. “Performing Safety Analyses and SysML Designs Conjointly: A Viewpoint Matter.” Paper presented at the 2nd International Conference on Complex Systems Design and Management, Paris, FR, 7–9 Dec.

Towards a Safe Systems Engineering

Pierre Mauborgne, pierre.mauborgne@mpsa.com; **Samuel Deniaud; Eric Levrat; Jean-Pierre Micaëlli; Eric Bonjour; Pascal Lamothe; and Dominique Loise**

Faced with the increasing complexity of systems, model-based system engineering relies on SysML, one of the recognized languages for systems modeling. In the case of the automotive industry, the introduction of model-based systems engineering in the design process is considered as an efficient way to improve design performance and to master new regulations such as ISO 26262 (ISO 2009) concerning functional safety of automotive systems. Although research work exists on model-based safety assessments (see Cressent et al. 2012; Belmonte and Soubiran 2012), there remains a lack of an approach on integrating system engineering and safety analysis, two domains handling their own concepts, models, and methods. In this short article, we are presenting two types of approaches.

Elaborated Model-Based Safety Assessment versus Exchanges between Systems Engineering and Safety

Overall, we can distinguish at least two approaches to solve this problem: (1) Elaborated Model-Based Safety Assessment (EMBSA) and (2) exchanges between system engineering and safety. For the first approach, a functional model of the system is used to realize a safety analysis as in Papadopoulos and McDermid (1999) or Mauborgne and others (2013). The EMBSA is therefore an *a posteriori* approach.

In contrast, the second one is an *a priori* approach. Throughout the modeling of the system, there will be exchanges between the activities of system design and safety analysis. Thus the result of these exchanges will be a safe system. We can see this type of approach in the work of Cressent and others (2012).

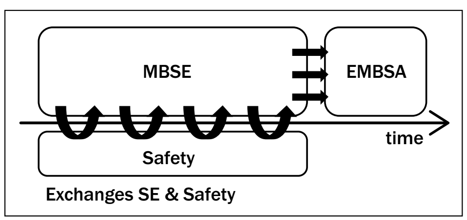


Figure 1. MBSA approaches

The interactions between model-based systems engineering and these two approaches are illustrated in figure 1.

Elaborated Model-Based Safety Assessment

As shown by Mauborgne and others (2013), there are different EMBSA approaches. Figure 2 proposes the steps of such an approach.

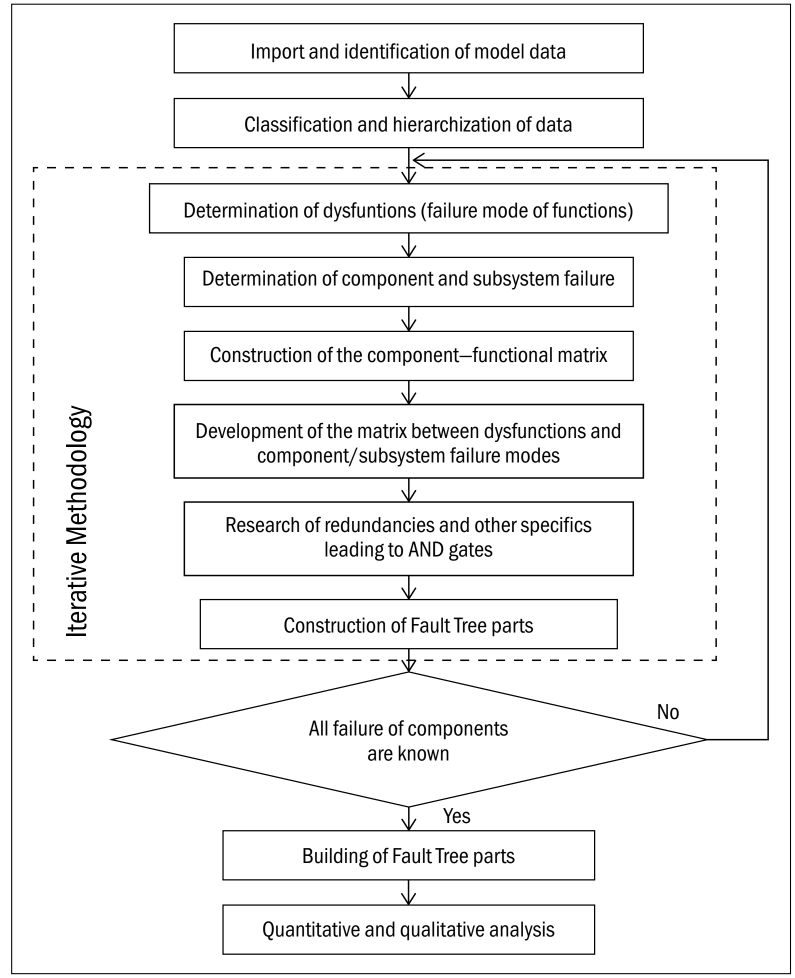


Figure 2. An example of EMBSA approach

To do a safety analysis using a functional model, we have to extract the functional and system architectures and the table of allocation between functions and components. By adding some dysfunctional information, we can construct a failure mode and effect analysis (FMEA), a fault tree, and an AltaRica model (Prosvirnova and Rauzy 2012). With these dysfunctional models, it is possible to perform a safety assessment.

Exchanges between System Engineering and Safety

In order to have a safe system, exchanges between system engineering and safety are required. Indeed, this type of approach supports the functional and component modeling of the system. So there may be iterations to improve the modeling and to reduce any subsequent returns.

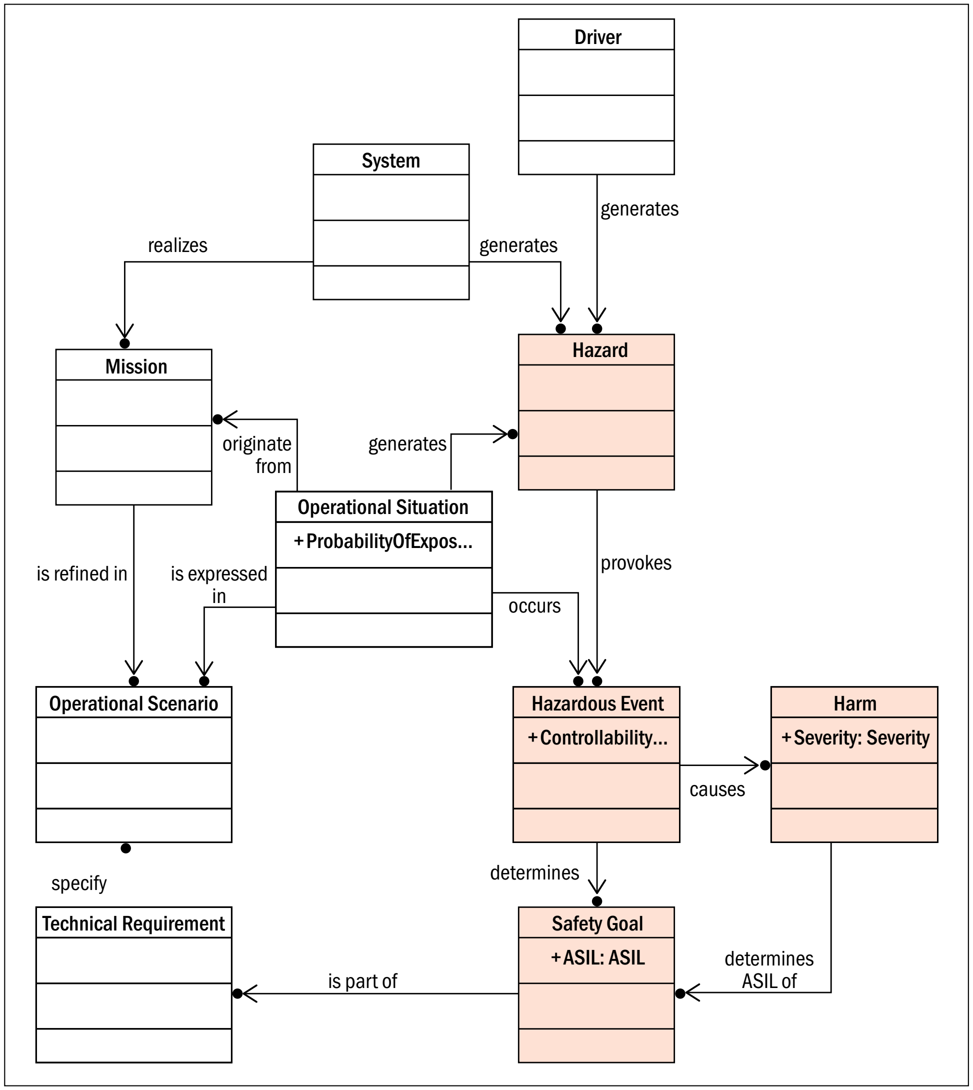


Figure 3. Conceptual model of the determination of safety goals in ISO 26262

As noted in the INCOSE *Systems Engineering Handbook* (Haskins 2010, 15), costs for changes become more and more important through the design of the system. So exchanges between system engineering and safety enable to increase the speed of changes in the earlier phases of design. One consequence is the reduction of the modeling time and therefore the design cost. Moreover, in functional safety standards like ISO 26262 (ISO 2009) for the automotive domain, system architects and safety engineers have to specify some safety requirements like safety goals.

Figure 3 shows that to determine a safety goal (a high-level safety requirement), there must be some exchanges between system architects and safety engineers. To determine hazards, hazardous events, safety engineers must have information about the system (its missions, its operational situations).

Determination of a hazardous event can provoke new operational scenarios. Iterations of this process will allow proper design of the system. So in the early stages of design, there must be exchanges between systems engineering and safety in order to design a safe system.

Conclusions

The first conclusion of this work is that at least two types of model-based safety analysis can be highlighted: elaborated model-based safety assessment and exchanges between system engineering and safety. Previous work with EMBSA has shown that there is a minimum of needed information to perform a functional safety analysis and to define system architectures. Obtaining a safe system requires an appropriate combination of these two approaches. Exchanges permit improving the design of a safe system and EMBSA verifies that the safety objectives are performed.

References

Belmonte, F., and E. Soubiran. 2012. "A Model Based Approach for Safety Analysis." *Computer Safety, Reliability, and Security* 7613: 50–63.

Cressent, R., P. David, V. Idaziak, and F. Kratz. 2012. "Designing the Database for a Reliability Aware Model-Based System Engineering Process." *Reliability Engineering and System Safety* 111: 171–182.

Haskins, C., ed. 2010. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 3.2. Revised by M. Krueger, D. Walden, and R. D. Hamelin. San Diego, US-CA: INCOSE.

ISO (International Organization for Standardization). 2009. ISO DIS 26262. *Functional Safety for Road Vehicles*. Geneva, CH: ISO.

Mauborgne, P., M. Labe, A. -S. Smouts, and N. Stojanovic. 2013. "Towards a Transition Approach from a Functional Model in SysML with Harmony Revisited Methodology to Fault Trees." Paper presented at the third International Workshop on Model Based Safety Assessment, Versailles, FR, 25–27 Mar.

Papadopoulos, Y., and J. A. McDermid. 1999. "Hierarchically Performed Hazard Origin and Propagation Studies." In *Computer safety, reliability and security*, ed. M. Felici and K. Kanoun, 139–152. Berlin, DE: Springer.

Prosvirnova, T., and A. Rauzy. 2012. "Système de Transitions Gardées: formalisme pivot de modélisation pour la Sûreté de Fonctionnement." Paper presented at Lambda Mu 18, Tours, FR, 16–18 Oct.

INSTITUTE FOR
SYSTEMS RESEARCH
A. JAMES CLARK SCHOOL OF ENGINEERING

FACULTY POSITION IN MODEL-BASED SYSTEMS ENGINEERING

The Institute for Systems Research (ISR) and the A. James Clark School of Engineering at the University of Maryland, College Park, invite applications for a tenure-track assistant professor position in Model-Based Systems Engineering (MBSE). We seek applicants with a doctoral degree who have research expertise in MBSE, formal approaches to engineering design and validation, requirements engineering, and multi-objective optimization and trade-off analysis, with appropriate connections to an engineering and/or biological domain.

This position is tenure track in a Clark School department (Aerospace, Electrical and Computer, Fire Protection, Mechanical, Civil and Environmental, Chemical and Biomolecular, Materials Science and Engineering, or Bioengineering) with a joint appointment to the ISR. The tenure home for this appointment will be in the department, and the ISR and its Director will also review the candidate's dossier.

The position holder will contribute to teaching systems engineering, advise systems engineering students for their theses, and develop new systems engineering courses. With regard to research responsibilities, the position holder will propose and conduct MBSE methodology and tools research that improves the ability to conceive, design and analyze complex multi-domain systems and will collaborate with other ISR faculty researchers to incorporate MBSE methods into ISR multi-disciplinary research.

The Clark School's (www.eng.umd.edu) graduate programs collectively rank 19th nationally (11th among public universities) in U.S. News & World Report's "America's Best Graduate Schools 2014." In 2013 the Academic Ranking of World Universities ranked the Clark School programs 16th in the world. ISR (www.isr.umd.edu), an original NSF ERC, celebrates 27 years of multi-disciplinary research in cooperation with universities, government, and industry. With annual expenditures exceeding $20M, it conducts inter-disciplinary research and provides education in systems engineering and sciences; and devises basic solution methodologies and tools for systems problems in disparate application domains. ISR has 38 joint appointment faculty; 27 affiliate faculty; and eight research scientists from four colleges and 14 units across the University of Maryland.

Prospective candidates should demonstrate the required expertise and apply with:

- A cover letter
- Curriculum vitae
- Teaching and research statements
- Three relevant scientific papers
- List of references

TO APPLY: Please visit (ejobs.umd.edu) and find posting #105839. For best consideration, submit applications by January 31, 2014; the position remains open until filled. The University of Maryland is an EEO/AA employer. We strongly encourage women and minorities to apply.

Model-Based Safety Assessment: The AltaRica 3.0 Project

Tatiana Prosvirnova, prosvirnova@lix.polytechnique.fr; Pierre-Antoine Brameret; and Antoine Rauzy

The model-based approach for safety and reliability analysis is gradually winning the trust of engineers but is still an active domain of research. Safety engineers master "traditional" risk modeling formalisms, such as failure mode, effects, and criticality analysis; fault trees; event trees; and Markov processes. Efficient algorithms and tools are available.

However, despite their qualities, these formalisms share a major drawback: models are far from the specifications of the systems under study. As a consequence, models are hard to design and to maintain throughout the lifecycle of systems. A small change in the specifications may require completely revisiting safety models, which is both resource-consuming and error-prone.

The high-level modeling language AltaRica Data-Flow (Rauzy 2002; Boiteau et al. 2006) has been created to tackle this problem. AltaRica Data-Flow models are made of hierarchies of reusable components. Graphical representations are associated with components, making models visually very close to process and instrumentation diagrams. AltaRica Data-Flow is at the core of several integrated modeling and simulation environments: Cecilia OCAS (Dassault Aviation), Simfia (EADS Apsys), and Safety Designer (Dassault Systemes). Successful industrial experiments were held using AltaRica Data-Flow (including the certification of the flight-control system of the Falcon 7X aircraft). In a word, AltaRica Data-Flow has reached an industrial maturity.

However, more than ten years of experience showed that both the language and the assessment tools can be improved. AltaRica 3.0 is an entirely new version of the language.

It improves AltaRica Data-Flow into two directions:

1. Its semantics are based on the new underlying mathematical model, Guarded Transition Systems (GTS).
2. It provides new constructs to structure models.

The new underlying formalism, Guarded Transition Systems (Prosvirnova and Rauzy 2012), makes it possible to handle systems with instantaneous loops and to define acausal components, that is, components for which the input and output flows are decided at run time. It is much easier to model systems with bidirectional flows, such as electrical systems.

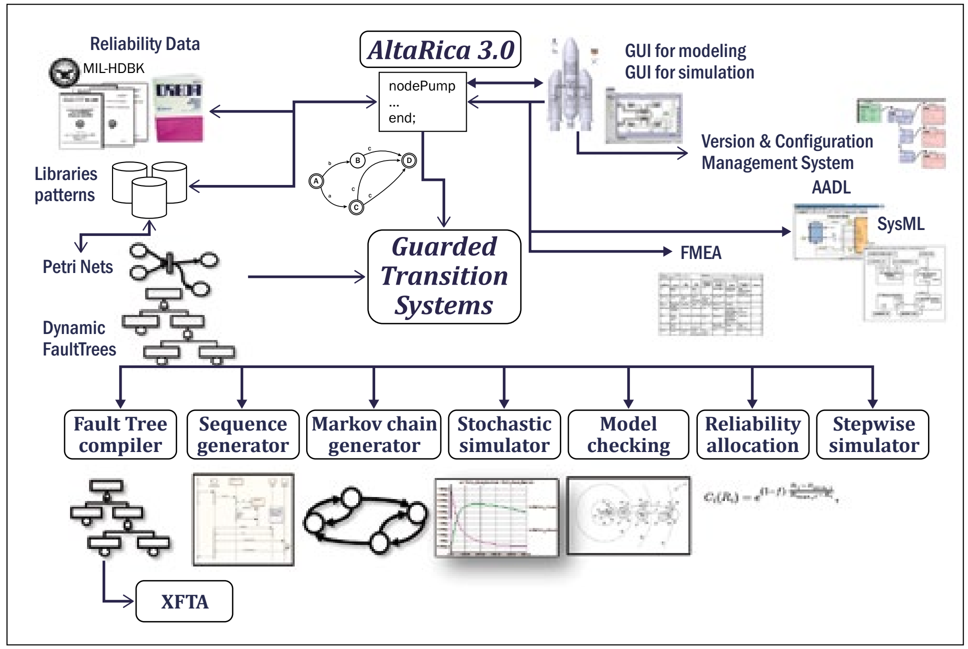


Figure 1. Overview of the AltaRica 3.0 project

AltaRica 3.0 is a prototype-oriented modeling language. Prototype orientation makes it possible to separate the knowledge into two distinct spaces: the stabilized knowledge, incorporated into libraries of on-the-shelf modeling components; the sandbox in which the system under study is modeled. With prototype-orientation, models can be reused in two ways: at the component level by instantiating on-the-shelf components; at the system level by cloning and modifying a model designed for a previous project.

The new version of the language is at the heart of AltaRica 3.0 project, which aims to propose a set of modeling and assessment tools to perform preliminary safety analyses. Figure 1 presents the overview of the project. Models are compiled into GTS, which is a pivot formalism for safety analyses: other safety models can be compiled into GTS to benefit from the assessment tools.

The assessment tools for GTS already include a fault-tree compiler to perform fault-tree analysis (FTA), a Markov-chain generator, a stochastic simulator, and a stepwise simulator. Other tools are under specification or implementation: these include a model-checker and a reliability allocation module.

These tools will be distributed under a free license to make them available to a wide audience, especially in the academic community. They enable users to perform virtual experiments on systems, to compute reliability indicators and, also, to perform cross-check calculations. With these tools AltaRica models can be used to perform preliminary system-safety analysis and system-safety analysis.

Fault-Tree Compiler

Fault trees are widely used to perform safety analyses and some regulatory authorities require their use to support the certification process. From a GTS model, it is possible to generate corresponding fault trees by transforming a states-transition model into a set of Boolean formulae. It may seem inefficient at a first glance to use a states-transition formalism to end up with a fault tree. However in practice, it is easier and less time consuming to automatically generate a fault tree from high-level models rather than create them from scratch. High-level models greatly improve the design, sharing, and maintenance of models.

The algorithm of compilation to fault trees for AltaRica Data-Flow, described in Rauzy (2002), can be extended to a general case of Guarded Transition Systems.

This algorithm includes three steps:

1. The Guarded Transition Systems model is partitioned into independent Guarded Transition Systems and an independent assertion.
2. Reachability graphs of each independent Guarded Transition System are calculated.
3. Reachability graphs and the assertion are separately compiled into Boolean equations.

Partitioning is a key point of the algorithm that ensures its efficiency. In practice, components of a system generally fail in a relatively independent way. In that case a partitioning is possible. If the Guarded Transition System is combinatorial, its compilation to fault trees is efficient and does not loose information.

The generated fault tree could be assessed with any fault-tree calculation engine supporting Open-PSA format. For example, a fault-tree engine developed for the Open Initiative for Next Generation Probabilistic Safety Assessment known as XFTA (Rauzy 2012) can be used to calculate minimal cutsets, events probabilities, and importance factors.

Markov-Chain Generator

The different reachable states of a system can be built from its GTS model. The state space of the system can be transformed into a Markov chain to compute the sojourn times or steady-state probabilities of the different states of the system. The Markov-chain formalism can be efficiently assessed by numerical methods such as developed by Rauzy (2004).

It is a very straightforward method to compute mean values of the observers defined in the AltaRica 3.0 model. For instance, the availability of the system can be built from an observer that takes value 1 when system is working and 0 otherwise.

This computation method has two limits:

1. The Markov hypothesis must hold for the system, that is, the transition rates between states must be constant.
2. The size of the Markov model is subject to exploding exponentially.

The second hypothesis is difficult to overcome. A method has been developed to limit the construction of the state space. It consists in selecting the most influential states toward the given reward, thus giving accurate results while drastically limiting the size of the state space.

The influence of the first hypothesis really depends on the system modeled. It is usually valid for safety assessment, and it gives good results to quickly assess an AltaRica model with the Markov-chain generator. It is particularly valid while designing the system architecture, when engineers need to assess and compare several models.

References

Boiteau, M., Y. Dutuit, A. Rauzy, and J. P. Signoret. 2006. "The AltaRica Data-Flow Language in Use: Assessment of Production Availability of a Multi-States System." *Reliability Engineering and System Safety* 91: 747–755.

Prosvirnova, T., and A. Rauzy. 2012. "Guarded Transition Systems: Pivot Modeling Formalism for Safety Analysis." Paper presented at the Lambda Mu 18 conference, Tours, FR, 16–18 Oct.

Rauzy, A. 2002. "Mode Automata and Their Compilation into Fault Trees." *Reliability Engineering and System Safety* 78: 1–12.

———. 2004. "An Experimental Study on Iterative Methods to Compute Transient Solutions of Large Markov Models." *Reliability Engineering and System Safety* 1: 105–115.

———. 2012. "Anatomy of an Efficient Fault Tree Assessment Engine." Paper presented at the International Joint Conference PSAM'11/ESREL'12, Helsinki, FI, 25–29 June.

A Metamodel for Knowledge Modeling and Maturity Integration in Systems Engineering

Nicolas Dremont, nicolas.dremont@utc.fr; Nadège Troussier; Nassim Boudaoud; Sébastien Castric; Ian Whitfield; and Benoît Eynard

Designing today's complex systems requires a significant number of models, specific for each discipline involved in systems engineering. In terms of data and process modeling, several models exist to support a multiple-view representation of the system, in order to support product-lifecycle management and collaborative decision-making (Belkadi et al. 2010). The goal is to develop a conceptual framework that enables defining heterogeneous knowledge models that integrate the maturity in order to support the decision-making.

The large range of product models identified in the literature survey has pointed out the importance of a robust metamodeling approach.

According to the CMMI approach (Beth et al. 2007), the concept of "maturity" is defined as the association of the knowledge and performance. To improve information management, the following question addressed in this article is then, how to model product information and uncertainties in early phases of systems engineering? A research survey has been carried out on both uncertainty modeling (Kreye et al. 2011) and product or knowledge models. Product models supporting collaborative decision making and taking into account of uncertainties have been assessed as shown in table 1.

The product and knowledge models identified make it possible to manage, to structure, and to take into account the different design tasks of systems engineering in order to support product lifecycle management. However, it should be underlined that none of them considers uncertainties.

Table 1. Survey synthesis

Uncertainty modeling	Qualitative approaches	Sustainability (Gaudin, 2001) Variation (Grebici et al., 2007) Sensitivity (Krishnan, 1996) Completeness (Yassine et al. 1999) PEPS: Precision, Accurancy, Parsimony, Specialisation (Sebastian et al. 2005)
	Quantitative approaches	Fuzzy sets (Zadeh,1999) Possibility theory (Choi et al. 2004) Evidence theory (Dempster, 1967) (Shafer, 1986)
Product and knowledge models	PPO: Product Process Organisation (Noel et al. 2005) KCM: Knowledge Configuration Management (Badin et al. 2011) CPM: Core Product Model (Sudarsan et al. 2005) MOKA: Methods and tools Oriented to Knowledge Acquisition (Moka 2000) PPR: Product Process Resource	

A metric allows us to assess the maturity of a system under design by calculating the maturity of each system's component at each design iteration. The following equation presents how the maturity of a component Ci is defined. The metric evolves at each design iteration, as an updated result of design parameters. Design iteration stops when the stakeholder's requirements are fulfilled by the complete technical specification.

$$C_i = \frac{1}{Co_i} \times \frac{\dfrac{\sum\left[\left(1-\dfrac{\text{interval}}{\text{value}}\right) \times \text{SusSen}\right]}{n} + \text{Perf}}{2}$$

The factors are "n," "value," "interval," "SusSen," "Perf," and "Co_i."

- "n" is the number of design parameters for a component (diameter, length, etc.).
- "value" is the nominal value of the design parameter, for example diameter = 25 mm.
- "interval" is the authorized variation domain of the value, for example diameter = 25 +/- 5 mm.
- "SusSen" represents the user's point of view. The user is placed in the center of the metric, because in the upstream phases of design, the main problem is the lack of knowledge retained by designers.
- "Perf" is the performance level defined by the percent of requirements achieved at the end of the design iteration compared to total number of requirements for the studied component.
- "Co_i" is the maturity level defining the achievement at the end of the design iteration. This is a constant that allows the adjustment of the level of maturity.

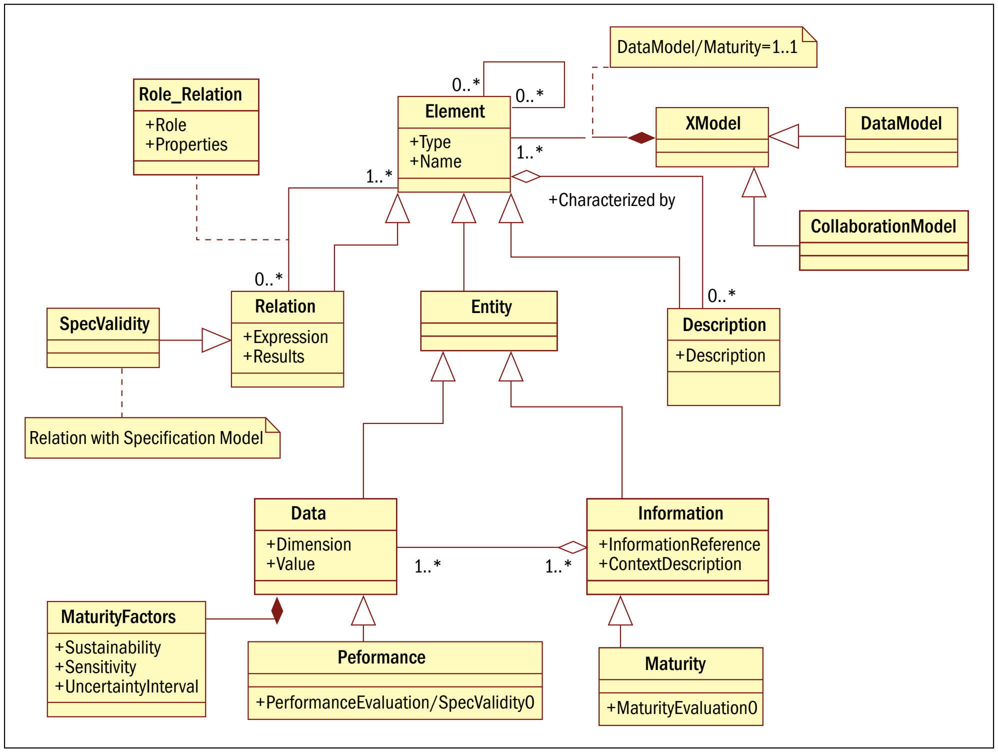


Figure 1. Metamodel of knowledge

Since the above metric is partly based on design parameters, the definition of product information can be considered as the key medium for knowledge modeling and maturity integration.

We propose metamodels to provide a tool for merging information and for ensuring their consistency as a support of collaborative decision making. The data metamodel and the collaboration metamodel are instances of the so-called knowledge metamodel.

The data metamodel provides concepts that allow for modeling of the disciplinary knowledge within a common and simplified semantic. It includes the parameters, their relationships, and the maturity information. The collaboration metamodel proposes the concepts representing the collaboration between disciplinary models in the sense of flipping from one to another. This includes interdisciplinary parametric relationships and model changes.

The knowledge metamodel is a conceptual framework allowing the creation of knowledge models, such as KCM and PPO in table 1, through instantiation of the knowledge metamodel. Therefore, the knowledge metamodel must be user-friendly and generic for the purpose of bringing consistency within one conceptual representation. It makes it possible to combine different models and then build the most appropriate one.

To conclude, the large range of product models identified in the literature survey has pointed out the importance of a robust metamodeling approach. It supports the consistency of heterogeneous knowledge produced by the involved disciplines during complex systems design. Accounting for the results of the literature survey and industrial feedback, future work will focus on implementation of the proposed metamodels in product-lifecycle management for managing the maturity of product information and for supporting collaborative decision-making based on managing uncertainties.

References

Badin, J., D. Monticolo, D. Chamoret, and S. Gomes. 2011. “Using the Knowledge Configuration Model to Manage Knowledge in Configuration for Upstream Phases of the Design Process.” *International Journal on Interactive Design and Manufacturing* 5 (3): 171–185.

Belkadi F., N. Troussier, B. Eynard, and E. Bonjour. 2010. “Collaboration based on Product Lifecycles Interoperability for Extended Enterprise.” *International Journal on Interactive Design and Manufacturing* 4 (3): 169–179.

Beth, C., M. Konrad, and S. Shrum. 2007. *CMMI: Guidelines for Process Integration and Product Improvement*. 2nd ed. Upper Saddle River, US-NJ: Addison Wesley.

Dempster, A. 1967. “Upper and Lower Probabilities Induced by a Multivalued Mapping.” *The Annals of Statistics* 28: 325–339.

Choi, K., L. Du, and B. Youn. 2004. “A New Fuzzy Analysis Method for Possibility-Based Design Optimization.” Paper presented at the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, US-NY, 30 Aug.–1 Sept.

Grebici, K. 2007. “La Maturité de l’Information et le Processus de Conception Collaborative.” PhD thesis, Institut National Polytechnique de Grenoble, FR.

Kreye, M. E., Y. M. Goh, and L. B. Newnes. 2011. “Manifestation of Uncertainty: A Classification.” Paper presented at the International Conference of Engineering Design, Lyngby, DK, 15–19 Aug.

MOKA. 2000. “Moka User Guide.” MOKA Project - ESPRIT 25418. http://web1.eng.coventry.ac.uk/moka/Documents/consortium/mig_def.pdf

Noel, F., L. Roucoules, and D. Teissandier. 2005. “Specification of a Product Modelling Concepts Dedicating to Information Sharing in a Collaborative Design Context.” In *Advances in Integrated Design and Manufacturing in Mechanical Engineering*, ed. A. Bramley, D. Brissaud, D. Coutellier, and C. McMahon, 135–146. London, GB: Springer.

Sebastian, P., J. Nadeau, X. Fischer, and R. Chenouard. 2005. “Knowledge Modelling in Mechanical Embodiment Design for Real Time Simulation and Decision Support.” Paper presented at the Virtual Concept conference, Biarritz, FR, 8–10 Nov.

Shafer, G. 1986. “The Combination of Evidence.” *International Journal of Intelligent Systems* 1: 155–179.

Sudarsan, R., S. Fenves, R. Sriram, and F. Wang. 2005. “A Product Information Modelling Framework for Product Lifecycle Management. *Computer-Aided Design* 37 (13): 1399–1411.

Yassine, A., D. Falkenburg, and K. Chelst. 1999. “Engineering Design Management: An Information Structure Approach. *International Journal of Production Research* 37 (13): 2957–2975.

Zadeh, L. 1999. “Fuzzy Sets as a Basis for a Theory of Possibility.” *Fuzzy Sets and Systems* 100 (Supplement 1): 9–34.

Model-Based Service Orchestration for Business Applications Re-Engineering

Sophea Chhun, sophea.chhun@univ-lyon2.fr; Néjib Moalla; and Yacine Ouzrout

Our model allows business analysts to test the feasibility of their business process designs. It also helps to reduce the development cost by reusing the existing web services stored in the service registry.

The Workflow Management Coalition provides this definition of *business process*: "It is a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships" (1999, 10). A service task is an element of a business process, and it can be performed by a web service or a composite service. A web service is a software module created to perform a specific business task, and it is described by a service description language (such as WSDL, WSDL-S, and OWL-S). Using the current design capabilities, business analysts cannot validate the feasibility of their business-process designs. The existing solutions do not provide a complete solution for the automatic implementation of business processes from their design specifications, or they require more intervention from users at the design stage (Talantikite, Aissani, and Boudjlida 2009; Frece and Juric 2012).

This research study aims to implement the business processes automatically by reusing the existing web services stored in the service registry. At first, the business analysts design their business processes by using BPMN 2.0 specifications and provide the description of each service task. After that, our model performs the service selection and composition to choose the most suitable services to execute the corresponding service tasks. Finally, our model validates the syntax of the generated business process and produces an executable business process. A composite service is a service created by combining many web services together (May and Weber 2008, 3). The existing web services are used to implement the new business processes and to re-engineer the existing processes, because this reuse approach can reduce the development cost (García-González, Gacitúa-Décar, and Pahl 2013, 1). Correia and Abreu (2012, 2) state, "A BPMN2 process model diagram has around 100 different modeling constructs, including 51 event types, 8 gateway types, 7 data types, 4 types of activities, 6 activity markers, 7 task types, 4 flow types, pools, lanes, etc." However, our proposed model only performs the automatic implementation of the service task (one kind of task type in business-process specifications).

This work targets first the semantic representation of the existing web services and users' requirements. It is because the service registry UDDI (Universal Description, Discovery, and Integration) supports only keyword matching and does not store the nonfunctional properties of web services. However, the nonfunctional properties of services are the important criteria of the service selection algorithm. Second, this work creates a semantic service selection and composition algorithm. This algorithm matches between the users' requirements and services' descriptions in order to find the most suitable services. Third, this project provides a solution to solve the problems of synonyms and homonyms because organizations usually use their own specific terms to name business elements and web services.

This research study is expected to produce the following outcomes:

- A model to perform the automatic implementation of business processes from users' business design specifications
- A semantic representation of the users' specifications and existing web services with ontologies
- A semantic service selection and composition algorithm that reuses the existing web services
- A method to generate and evaluate the resultant business processes

Proposed Architecture

A global overview of our approach is proposed in figure 1. Our proposed model requires as input the specifications of business processes, and it outputs the specifications of business processes with the implementation of service tasks. The proposed model works as follows:

1. The business analysts provide the specifications of business processes throughout an interface. They specify each service task by its context, inputs, outputs, and weights that describe the importance of each property of the quality of service (QoS). In our work, QoS is specified by service's performance (availability, execution time, and number of

calls) and service's security (authentication and encryption). In addition, the performance values of quality of service are calculated from the tracking data of services' execution on the server.

2. The Semantic Transformer takes the user's specifications with business process and builds the corresponding business-process ontologies to represent it.
3. The Content Extractor extracts the web services' information from the service registry and WSDL (Web Service Description Language) files. After that, it passes the data into Ontology Builder to generate a web-service ontology that represents the semantic of the existing available web services.
4. The Semantic Matching Engine matches the business-process ontologies and web-service ontologies to obtain the most suitable services for executing the service tasks. In case the Semantic Matching Engine cannot find an atomic service that can answer to a user's request, it performs a service composition algorithm to create a new service by combining the existing web services together.
5. After finding the matched services, the Business-Process Transformer integrates the matched services into the business-process specifications and generates an executable business process.
6. The Validator evaluates the syntax and checks the consistency of the generated business processes from the Business-Process Transformer.
7. Finally, the business processes with the implemented service tasks are provided to the users as output.

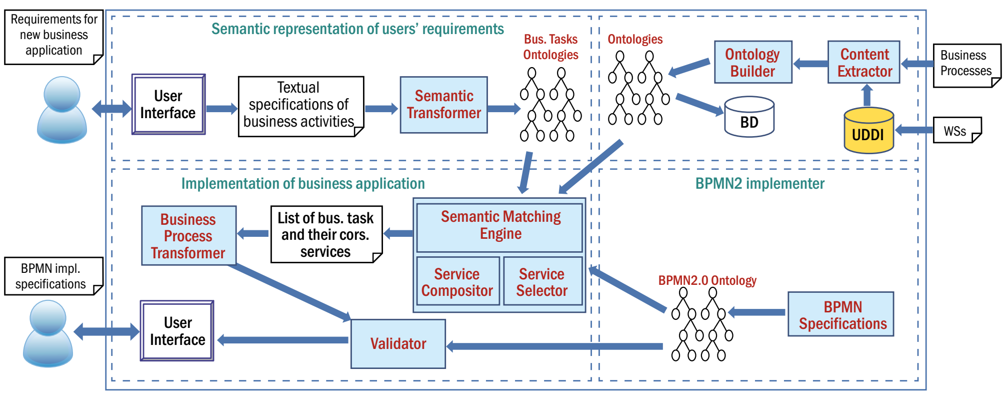


Figure 1. Global overview of our approach

Web Service Ontology

As presented in figure 2, our proposed web-service ontology groups the services into categories. The services' categories are defined by the values of tModel in the UDDI (Universal Description, Discovery, and Integration) because UDDI uses the tModel value to categorize the services. For each service, the web-service ontology stores the functional and nonfunctional properties of services. The functional properties of services are defined by their service interface and operations. The non-functional properties of services are defined by the quality of service values. Every service and services' category link to a list of keywords. Those keywords are extracted from the category's description, service's description, service's name, operation's name, operation's description, input's name, and output's name. They are obtained by using some methods such as POS (Part Of Speech) tagger, split combined terms, remove stop words, and word stemming.

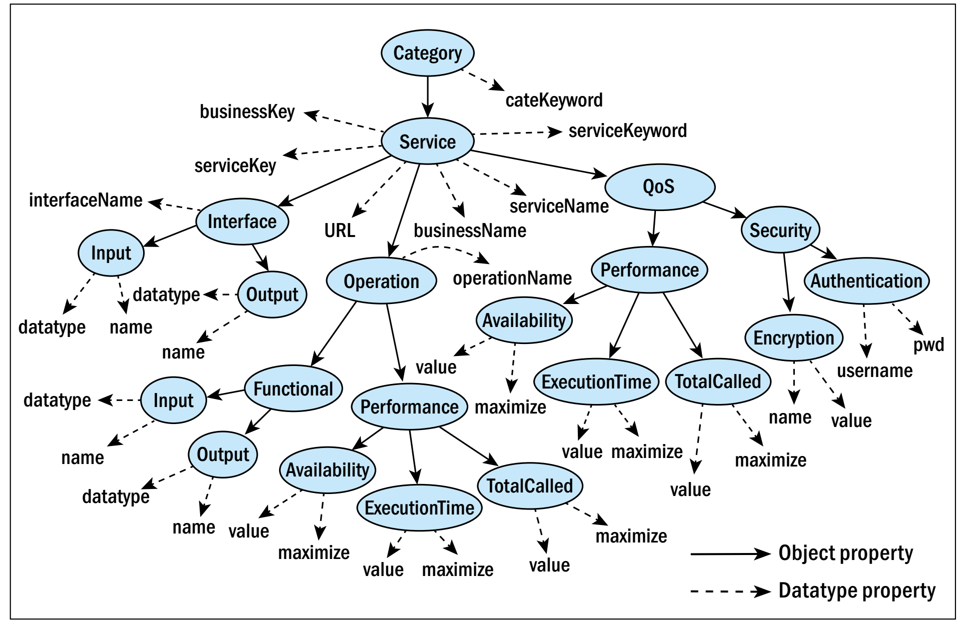


Figure 2. Web-service ontology

Conclusion and Future Work

Our model allows business analysts to test the feasibility of their business process designs. It also helps to reduce the development cost by reusing the existing web services stored in the service registry. In addition, we proposed a web-service ontology structure that can store both the functional and non-functional properties of services.

Our future work aims to define a semantic service selection and composition algorithm that selects the most suitable services to execute service tasks. The semantic meaning is considered because the companies usually use their own specific terms to name the business elements and web services.

» continues on next page

Chhun et al. continued

References

García-González, J. P., V. Gacitúa-Décar, and C. Pahl. 2013. "Service Registry: A Key Piece for Enhancing Reuse in SOA". ACSIJ Advances in Computer Science 2 (1): 1–8.

Workflow Management Coalition. 1999. *Workflow Management Coalition Terminology & Glossary*. Hampshire, GB.

Correia, A., and F. Brito e Abreu. 2012. "Adding Preciseness to BPMN Models." *Procedia Technology* 5: 407–417.

Frece, A., and M. B. Juric. 2012. "Modeling Functional Requirements for Configurable Content-and Context-Aware Dynamic Service Selection in Business Process Models." *Journal of Visual Languages & Computing* 23: 223–247.

Ikite, H. N., D. Aissani, and N. Boudjlida. 2009. "Semantic Annotations for Web Services Discovery and Composition." *Computer Standards & Interfaces* 31 (6): 1108–1117.

May, N., and I. Weber. 2008. "Information Gathering for Semantic Service Discovery and Composition in Business Process Modeling." In *4th International Workshop EOMAS, held as CAiSE 2008, Montpellier, France, June 16–17, 2008 Proceedings*, ed. J. L. G. Dietz, A. Albani, and J. Barjis, 46–60. Berlin, DE: Springer.

Creating a Common Vocabulary to Support the Exchange of Numerical Models between Suppliers and Users in a Complex System Design

Göknur Sirin, goknur.sirin@ecp.fr, Bernard Yannou, and Eric Landel

Computer-based simulation plays a crucial role in the design of modern products. Since the early 1990s, automotive companies have been using increasingly sophisticated, intensive, and integrated numerical models—behavioral models (Mocko, Malak, Paredis, and Peak 2004), expert or simulation models—to improve product quality and performance by intensive early design exploration (Sinha et al. 2012). The simulation-based design process typically involves a large number of disparate numerical models (thermal calculations, fluid calculations, spreadsheets, drawings). The process includes some actors such as the external or internal model provider (such as the creator or analysts), the model user and decision-maker (the simulation analyzer and product-model modifier). Managing the data within these models and these actors to maintain consistency, communication, and reuse is an important but complicated task (Paredis et al. 2001).

In this kind of complex development situation, we decompose numerical models of complex systems into three layers: product (object model), process (model activity), and organization (its dynamic environment). According to Eppinger and Salminen (2001), each of these three layers is decomposed in order to manage the complexity. In this article, we focus basically on the product level (object model). "Product layer" refers here to the simulation object model, which is iterative and hierarchical in nature. To solve product-level complexity problems, a design team typically handles the problem at different levels of abstraction, ranging from very high-level system decompositions to very low-level detailed specification of components (Vriesand and Breunese 1995). This is particularly important for the design of multidisciplinary systems in which components in different disciplines (such as mechanical, electrical, and embedded control) are tightly coupled to achieve optimal system performance (Sinha et al. 2001).

The model-development process involves a number of parallel activities in which experts in each domain (such as engine, transmission, chassis) create subsystem or component models based on model fidelity and interface requirements defined by model use cases. These component and subsystem models must finally be assembled and integrated together within a modeling framework to build up a full-system level model (Branscomb et al. 2013). Integration of these disparate domain models is challenging and error prone. Today, the model supply especially from an external provider is a bottleneck activity. Automotive manufacturers request a new model or a customized existing model from the supplier. In the case of a new model supply (from the requirements-elicitation phase to model- integration tests), as there is not a common vocabulary, the probability of failing during the model integration is very high.

The source of this problem is mostly based on wrong or insufficient knowledge transmission from automotive manufacturer to model supplier (see figure 1). As the assumption of common understanding is incorrect, the provided model does not totally conform to the requirements.

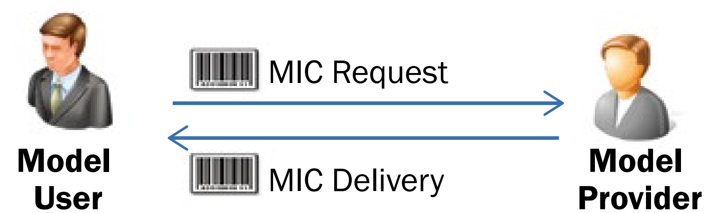


Figure 1. Knowledge capture between model user and provider

Thus, it is necessary to create a semantically rich model characterization support to reduce the knowledge gap between the model provider and user. This work aims to introduce the first necessary step—creating a domain vocabulary for formally characterizing the numerical model. Based on this common vocabulary, a Model Identity Card (MIC) is developed as an intermediate support in automotive context. The vocabulary presented in this paper is intended to

- decrease the ambiguity between stakeholders,
- facilitate the model (0D, 1D, 2D) characterization,

- reduce the time to get a correct model from external model provider, and
- track the model traceability.

MIC characterizes a model into five main classes: Physical Object, Interface, Methods, Usage, and Validation and Verification (see the first column of table 1).

Table 1. MIC object classes and their attributes

CLASS	ATTRIBUTES	SUBATTRIBUTES	EXAMPLE
Object Physics	Component name (generic)		Engine
Object Physics	Specific name		Compressor
Object Physics	Granularity	System, subsystem, components	Submodel
Object Physics	Causality		Yes
Object Physics	Author		F. Ravet
Object Physics	Model version		0.1
Object Physics	Chosen method	Finite volumes, elements, difference, Runge Kutta	Runge Kutta
Method	Precision	1, 2, 3, 4, 5, 6	2
Method	Solver	Chemistry, Navier Stokes, Strength of materials, Maxwell, Dynamic behavior of materials	Navier Stokes
Method	Time step	Millisecond, second, minute, hour	Second
Method	Linearity		No
Method	Continuity		Yes
Method	Model dimension	0D, 1D, 2D, 3D	1D
Usage	Compilability		Yes
Usage	Time computation	Real time, time computation	Time computation
Usage	Scalability		Yes
Usage	Software name		GT-POWER
Usage	Software version		7, 3
V&V	Code verification	Development, candidate, reference, previous	Candidate
V&V	Solution verification		
V&V	Validation		

Each main class consists of numerous attributes. In table 1 we identify classes and attributes of all physics based on analyzed numerical models. The first column is the term employed to represent this modeling knowledge concept (attributes), the second and the third column is the attributes of related domain and subdomain, and the fourth column lists constraints to give some example. As shown in table 1, the Object Physics class consists of some basic attributes such as Specific name, Granularity, Causality, Author, and Model version. Some of the attributes have also subattributes (granularity, for example, has system, subsystem, and component subattributes). The Method class consists of Chosen method, Precision, Solver, Time step, Linearity, Continuity, and Model dimension. The Usage class consists of Compilability, Time computation, Scalability, Software name, and software version. The Verification and Verification (V&V) attribute is under development. V&V will be developed in future work. A working group composed of fifteen engineers from five different disciplines (thermal comfort, motor, acoustic, electric, vibration) developed the beta version of MIC. They met more than ten times in four months to facilitate and standardize inputs during a data collection phase.

The attributes of the Interface class are developed based on respecting laws of conservation. The workgroup creates first a tree of Object Interface description. We distinguish the nature of interface as parameters, control, and physics, and each main interface class attributes can be divided into domain, subdomain, and unit. This tree provides the good level of abstraction for domains and subdomains (see figure 2).

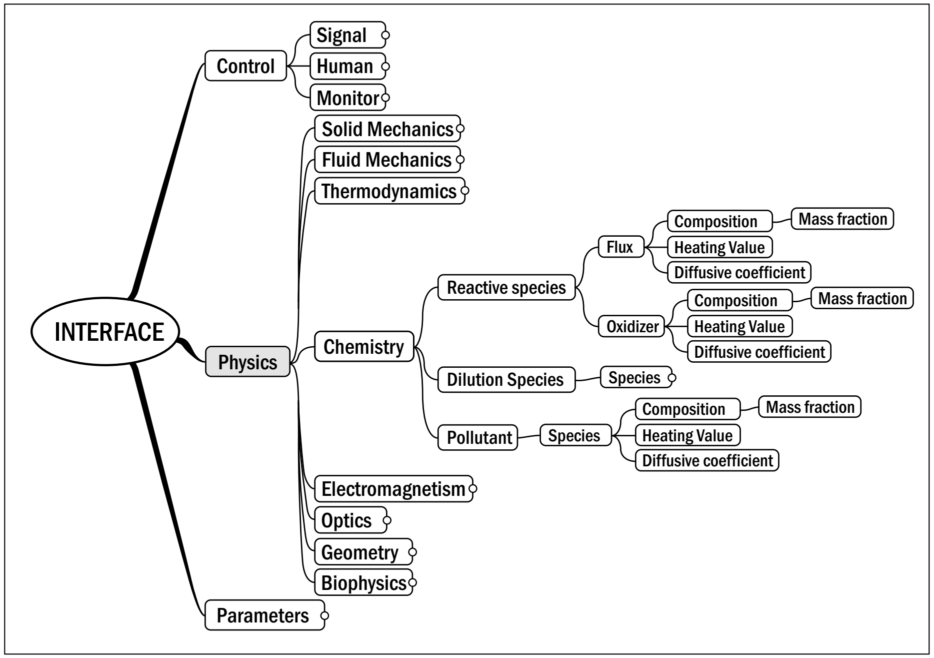


Figure 2. Interface definition

Sirin et al. *continued*

Model exchange and reuse problems have already been addressed by well-defined interfaces such as the ISO STEP AP 233 and 239-model exchange standard (Eckert, Mansel, and Specht 2005). The AP233 standard was developed for communication between similar tools in systems engineering usage (ISO 2003). Thus, systems engineering software vendor companies have been developing and testing AP233 interfaces in order to ensure interoperability. The MIC concept is a common vocabulary for facilitating the knowledge capture and communication between stakeholders. A graphical user interface is developed to support editing of model characterization.

The MIC concept are locally integrated in each company and tested by different engineering teams. According to test results, MIC attributes are accurate and sufficient for characterizing especially 0D reduced models, 1D, and 2D models. MIC is generic and thus it is important to extend it to support different specific domains of interest. Future work includes increasing the amount of MIC testing with different engineering teams and models to extend its usage.

References

Branscomb, J. M., C. J. J. Paredis, J. Che, and M. J. Jennings. 2013. "Supporting Multidisciplinary Vehicle Analysis using a Vehicle Reference Architecture Model in SysML." Paper presented at the Conference on Systems Engineering Research, Atlanta, US-GA, 12–22 March.

Eckert, R., W. Mansel, and G. Specht. 2005. "STEP AP233 + Standard PDM = Systems Engineering PDM?" Paper presented at the international Conference on Concurrent Engineering ICE, Munich, DE, 20–22 June.

Eppinger, S. D., and V. Salminen. 2001. "Patterns of Product Development Interactions." Paper presented at the International Conference on Engineering Design ICED 01, Glasgow, GB, 21–23 Aug.

Mocko, G., R. J. Malak, C. J. J. Paredis, and R. Peak. 2004. "A Knowledge Repository for Behavioral Models in Engineering Design." Paper presented at the ASME International Design Engineering Technical Conferences, Salt Lake City, US-UT, 2 Oct.

Paredis, C. J. J., A. Diaz-Calderon, R. Sinha, and P. K. Khosla. 2001. "Composable Models for Simulation-Based Design." *Engineering with Computers* 17 (2): 112–128.

Sinha, R., V. C. Liang, C. J. J. Paredis and P. K. Khosla. 2001. "Modeling and Simulation Methods for Design of Engineering Systems." *Journal of Computing and Information Science in Engineering* 1 (1): 84–91.

Sirin, G., B. Yannou, E. Coatanéa, and E. Landel. 2012. "Analysis of the Simulation System in an Automotive Development Project." Paper presented at the Complex Systems Design & Management conference, Paris, FR, 12–14 December.

Vriesand, T. J. A., and A. P. J. Breunese. 1995. "Structuring Product Models to Facilitate Design Manipulations." Paper presented at the International Conference on Engineering Design, Prague, CZ, 22–24 August.

An Integrated Approach for Designing an Agricultural Process Guided by Sustainable Evaluation: Application to Olive-Oil Production

Guillaume Busset, guillaume.busset@ensiacet.fr; Jean-Pierre Belaud; Mireille Montréjaud-Vignoles; and Caroline Sablayrolles

Scientific and Industrial Context

Industrial production systems and their consequences on the world constitute an important scientific issue. To consider technical, economic, quality, and environmental dimensions, the development of an industrial production system is based on models that simplify the real system in order to represent it with different objectives.

a. Process Approach

In terms of quality, the ISO 9000 standard recommends a process approach. This approach represents the system at any level (technical, tactical, or strategic) as a set of processes with inputs, outputs, controls, and constraints (ISO 2005). The process approach is also the basis for environmental-impacts assessment of a system within the lifecycle-assessment methodology.

b. Enterprise Models, Levels, and Views

At a technical level, process design is organized by process systems engineering based on the modeling of physical, chemical, and biological processes. At a tactical level, business process management is based on modeling the business process. At a strategic level, enterprise engineering supplies models of the enterprise. A system—typically an enterprise—can be seen from four main views: organizational, functional, resources, and information (Ulmer 2011). The different models from the different levels take elements from the different points of view of the system.

c. Sustainability Assessment

At a transversal level, the lifecycle analyst is considered as a specialist in lifecycle assessment. The evaluation of environmental, social, and economic impacts of a system is usually modeled with the lifecycle assessment (LCA) method. Until recently, only environmental and economic LCA have been well applied, even though social LCA is being undertaken. The integration of the three methods is a critical issue in order to assess any system in terms of sustainable development.

d. System Engineering for Multilevel Sustainability

Whatever the level of enterprise, systems and processes are modeled in order to improve the engineer's knowledge of the system, and the system's agility. Modeling also helps to design a system considering future impacts on environment, society, and economy at the early stages of the lifecycle. Lifecycle thinking is intrinsic to the process approach. As a consequence, sustainable lifecycle assessment of a system can be undertaken at the different levels of a system study. Lifecycle assessment can be coupled to process systems engineering for "process eco-design," to business process management for "business eco-design" and finally to enterprise engineering for "enterprise eco-design." System engineering is usually and successfully applied to aerospace, aeronautic, information, and mechanical systems. We think that it can be also applied to agro-industry for a multilevel, multicriteria analysis of sustainability.

e. Integration of LCA with the Different Levels of Enterprise Models

At a technical level, Azapagic (2006), Gilliani (2010), and Jacquemin (2012) have studied integration of the design of chemical and agrochemical processes with sustainable lifecycle assessment. However, no generic integrated approach has yet been formalized within a systems-engineering-based framework, neither for chemical process, nor for agro-industrial process. Furthermore, any work has not been led to integrate lifecycle assessment, process and system engineering, and business-process modeling (BPM). Systems engineering and the process approach are common to all these domains; using tools from systems engineering is proposed in order to integrate them. It seems crucial to know how technical and tactical levels of modeling are linked and how sustainable lifecycle assessment can model interactions between a system and its environment. In this context, a comprehensive approach is proposed for designing an agricultural process with a sustainable perspective.

Methodology for Definition, Application, Verification and Validation of the Approach for the Design of Agricultural Processes

The proposed approach has the objective to design for sustainability, which is chosen as the center of the approach. The question is, how will the methodology of lifecycle assessment integrate data and models from other domains?

a. LCA as an Evaluation Approach

Lifecycle assessment was first developed to reduce environmental impacts of a product, a process, or a service (ISO 2006). Since then, lifecycle costing and social lifecycle assessment have been developed to account for economic and social aspects in the lifecycle assessment of a system. However, there are no mutually agreed norms, but only guidelines from the United Nations Environment Program (UNEP). General LCA is a semiformalized framework that includes system modeling and focuses on a functional point of view with elements taken from many points of view such as resources. The originality of LCA is to link the system and its environment through impact assessment. Does the environment have to be internalized into system boarders or does it have to be studied as a new point of view? Furthermore, LCA needs to describe all the inputs and outputs that pass through every process that constitutes the system. LCA is only a descriptive method for characterizing the interactions between the system and its external environment. LCA needs other disciplines and other business competencies to analyze and explain the link between the system and its impacts on the environment. Lifecycle assessment is defined as a product-oriented approach, but the product is only the result of the process. Consequently, LCA must be considered more as a process approach.

LCA uses several factors to characterize inputs and outputs that cross the system. Environmental and economic factors are well known and easy to apply, but social factors remain more complicated. Nevertheless, the principle is to qualify and quantify impacts of input consumption and output emission of processes of the system. Environmental indicators are, for instance, global warming, ozone-layer depletion, acidification, eutrophication, and depletion of nonrenewable resources. Economic indicators include investments, salary and benefits, and spending. Social indicators include the number of employees, job creation, and human rights. The limits of the LCA methodology concern the modeling of the system. No formalism defines the way to represent the system, but in other disciplines with other objectives, there exist some formalisms to establish models of processes at the levels described above.

b. Enterprise-Architecture Framework

Considering the limits of lifecycle assessment, we propose an approach based on the LCA method but using architecture frameworks for modeling agricultural processes. Considering the ISO 15704 norms that define an enterprise-architecture framework, we place lifecycle assessment, process and system engineering, and business-process modeling within three dimensions: different levels of specificity (generic, partial, particular), different points of view (organization, information, resources, function), and at different lifecycle stages (ISO 2000). These lifecycle phases are identification, concepts, needs definition, specification and design, implementation, operation, and dismantling (Vernadat 1999, 136).

c. Lifecycle Assessment, Business-Process Modeling, and Process System Engineering in an Enterprise-Architecture Framework

The LCA method focuses on elements taken from the different points of view and is usually applied to the implementation, operation, and dismantling phases. Nevertheless, LCA can be applied during earlier stages in order to consider the future impacts cited above as soon as possible. It can also be applied to any level of specificity. BPM supplies a representation of the system using functional, resources, and information points of view. It may represent the system at any lifecycle phase. It is applied at any degree of specificity. PSE represents the system using functional and operational points of view, at any level of specificity and at any phase of the lifecycle. We propose to map the different results and models given by PSE, LCA, and BPM in order to align models and to get an overall consistency.

d. Experimental-Data Collection to Approach Validation and Verification: The OiLCA Project

The approach relies on the OiLCA project results that are based on LCA, lifecycle costing, and ecolabel design. The project consists in "enhancing the competitiveness and reducing the carbon footprint of the olive oil sector through waste management optimization and the establishment of an ecological label"

(OiLCA 2011). The latter is a way of communicating to consumer the efforts to take into account and to limit environmental footprint of a product. This project is a partnership with several industrial research agents: the Centre Technnique de l'Oliver in France (CITOLIVA), the Instituto Andaluz de Tecnologia and Fundació Centre Tecnològic de Manresa in Spain, and the Centro para a Valorização de Residuos (CDR) in Portugal. It led to a database with actual economic and environmental data from 59 companies that produce olive oil and to the development of a software application for carbon footprint and cost calculation, available on the website (www.oilca.eu/oilcatool).

Conclusion and Perspectives

Systems engineering was used for a better understanding of sustainable assessment of an agricultural-industrial system. It was used to develop a methodology within enterprise architecture framework. Finally, the integrated approach could be completed with optimization tools in order to find best, most sustainable solution.

References

Azapagic, A., A. Millington, and A. Collett. 2006. "A Methodology for Integrating Sustainability Considerations into Process Design." *Chemical Engineering Research & Design* 84 (A6): 439–452.

Jacquemin, L., P. Y. Pontalier, and C. Sablayrolles. 2012. "Life Cycle Assessment (LCA) Applied to the Process Industry: A Review". *International Journal of Life Cycle Assessment* 17: 1028–1041. doi:10.1007/s11367-012-0432-9.

Gillani, S. T., J. P. Belaud, C. Sablayrolles, M. Vignoles, and J. M. Le Lann. 2010. "Review of Life Cycle Assessment in Agro-Chemical Processes". *Chemical Product and Process Modeling* 5 (1): 1–26.

ISO (International Organization for Standardization). 2000. ISO 15704. *Industrial Automation Systems—Requirements for Enterprise-Reference Architectures and Methodologies*. Geneva, CH: ISO.

———. 2005. ISO 9000. *Quality Management Systems. Fundamentals and Vocabulary*. Geneva, CH: ISO.

———. 2006. ISO 14040. *Environmental Management—Life Cycle Assessment Principles and Framework*. Geneva, CH: ISO.

OiLCA (Olive Oil Life Cycle Assessment). 2011. "Enhancing the Competitiveness and Reducing the Carbon Footprint of the Olive Oil Sector through Waste Management Optimization and the Establishment of an Ecological Label." Website of OiLCA (Mengíbar, ES-J), http://www.oilca.eu/en/.

Ulmer, J. S., J. P. Belaud, and J. M. Le Lann. 2011. "Towards a Pivotal-Based Approach for Business Process Alignment". *International Journal of Computer Integrated Manufacturing* 24 (11): 1010–1021.

Vernadat, F. 1999. *Techniques de modélisation en entreprise: applications aux processus opérationnels*. Paris, FR: Economica.

Technical Operations

Technical Directions: Setting the Stage for the 2014 International Workshop

William Miller, INCOSE Technical Director, william.miller@incose.org

In the September 2013 issue of ***INSIGHT***, I described the challenges in bringing together our different camps within INCOSE so that all might contribute to the forthcoming *Systems Engineering Handbook* version 4, the BKCASE Systems Engineering Body of Knowledge (SEBoK) sandbox, and the *Systems Engineering Vision 2025*. In previous columns, I identified six camps: two empirically based process-focused schools, i.e., (1) technical and (2) systems engineering management; and industrial outreach-inspired, nontraditional domains, (3) transportation, energy, and biomedical/healthcare; as well as (4) model-based systems engineering; and (5) soft systems, systems thinking, and systems science. I also identified a potential camp (6) focused on systems engineering leadership. At the 2014 International Workshop, we intend to (a) adjudicate comments to the version 4 handbook, (b) engage working groups to contribute to the SEBoK, (c) transition the *Systems Engineering Vision 2025* to the next stage of engagement through the wider systems community, and (d) socialize the mainstreaming of model-based systems engineering into our working groups.

The theme of this December 2013 issue of ***INSIGHT***, "AFIS Doctoral Symposium: Systems Engineering Research Challenges in French Universities," points to another INCOSE challenge: how to highlight and leverage the contributions of chapter working groups. We will be working at the 2014 International Workshop to engage working groups supported by Technical-Operations with those of the chapters beyond the handful of already established relationships.

Paul Schreinemakers, Deputy Technical Director, and I look forward to working with you at the workshop this coming January in Torrance, California (US).

To SE or Not To SE (With apologies to William Shakespeare)

To SE, or not to SE, that is the question:
Whether 'tis Nobler in the mind to suffer
The Slings and Arrows of outrageous Customers,
Or to take Arms against a Sea of Risks,
And by mitigating end them: to Architect, to Design
No more; and by an Architecture, to say we end
The heart-ache, and the thousand Natural shocks
That Systems are heir to? 'Tis a consummation
Devoutly to be wished. To Architect to Design,
To Integrate, perchance to Deliver; Ay, there's the rub,
For in that Integration of the Architecture, what Delivery may come,
When we have shuffled off this Project,
Must give us pause. There's the respect
That makes Calamity of System Life Cycles:
For who would bear the Whips and Scorns of Design Reviews,
The Customer's wrong, the "Real Systems Engineer's" Contumely,
The pangs of despised Non-Compliances, the Schedule's delay,
The insolence of the Program Manager, and the Spurns
That patient merit of the unworthy takes,
When he himself might his Quietus make
With a New Assignment? Who would Fardels bear,
To grunt and sweat under a System Engineer's life,
But that the dread of something after V&V,
The undiscovered Stakeholder, from whose bourn
No Traveler returns, Puzzles the will,
And makes us rather bear those Requirements we have,
Than fly to others that we know not of.
Thus Systems doth make Cowards of us all,
And thus the desire for New Features
Is sicklied o'er, with the pale cast of Maintenance and Disposal,
And enterprises of great pitch and moment,
With this regard their Profits turn away,
And lose the name of Action. Soft you now,
The fair Ophelia? Nymph, in thy Orisons
Be all my Change Requests remembered.

— S. E. Hamlet (a.k.a. Dave Walden, ESEP)

INCOSE Operations

Exam for Acquisition Extension to be Offered for Final Time in March 2014

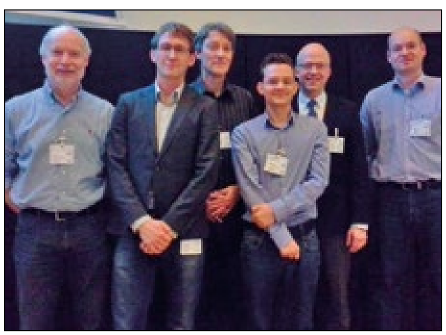
The INCOSE Certification Program will be sunsetting the Acquisition (Acq) extension. Those who have earned this extension will retain it as long as their base certification (ASEP, CSEP, or ESEP) is active. Any lapse in certification requires re-starting the process, which means the Acq extension will no longer be possible to regain once it is lost. Applications for the Acquisition extension will be accepted through 1 February 2014, and the exam offered through 1 April 2014.

Other extensions are under consideration using lessons learned from Acq experience. All questions should be directed to certification@incose.org.

Formation Meeting for INCOSE Denmark

Cecilia Haskins, cecilia.haskins@incose.org; **and Henrik Balslev,** henrik.balslev@incose.org

The last day of October was a momentous day in Copenhagen as the INCOSE DK members ratified their bylaws and elected their first slate of officers and directors in anticipation of joining the community of INCOSE chapters. The administrative portion of the meeting included welcoming presentations from Cecilia Haskins, coordinator for new chapters, and from the director of the EMEA Sector, Asmus Pandikow. INCOSE Ambassador Terje Fossnes was also present. GN ReSound graciously hosted the meeting, chaired by Rene Mortensen, their vice president of systems engineering and test. Over 40 attendees included member and potential members representing the broad industry base of this chapter that includes pharmaceutical and medical, construction, manufacturing, and defense firms, as well as universities and incubators. Two keynote presentations rounded out the day by challenging the audience to look systematically at innovation via new technology insertion and creating standards for the built environment. More information is available on http://www.incose.dk.



Newly elected officers and directors (left–right): Niels Christian Jensen, Mikkel Vestergaard Hansen, Jens Christian Andersen, Claus Ballegaard Nielsen, Henrik Balslev, Claus Broch

Chapter News

A Common Language to Ensure More Effective Development of New Products

Henrik Balslev, henrik.balslev@incose.org

The development of new products is becoming increasingly challenging for many companies. Products are becoming more complex, covering the spectrum of mechanics, electronics, and software. This means that the risk of making expensive mistakes during the development phase rises sharply. Hence, there is an increasing need to ensure better cohesion during the development process.

Against this backdrop, many Danish companies have chosen to increase focus on the discipline of systems engineering. Just recently, representatives from a number of companies and educational institutions have formed a Danish chapter of the international systems engineering network INCOSE, in order to disseminate knowledge about systems engineering and to establish a forum for exchanging experiences on how systems engineering is applied in Danish companies.

Systems engineering is a holistic approach to development processes. Systems engineering provides any company with a common language, which helps to ensure an efficient development process, bearing the goal for the new products in mind. In this manner, systems engineering effectively does away with the silo mentality, which often permeates development projects, where the companies' separate development functions sit isolated on each of their own islands, working more or less in a vacuum with no or very little interaction.

One of the Danish companies that makes extensive use of systems engineering is Terma, headquartered in Lystrup, close to Århus. The company develops advanced systems for defence departments, including aviation, aeronautics, and security.

Systems engineer Claus Broch works at Terma's division in Herlev. "With SE we are able to identify the actual needs before we initiate a development project," Broch says. "Additionally it ensures that we are continually up to date with regard to our goals throughout the project."

One of the areas where Terma has applied systems engineering is for the development of the command-and-control system for the new Royal Danish Navy frigates of the Iver Huitfeldt Class, including integration with the missile-based air-defence system.

"We are convinced that a suitable dose of systems engineering has been a contributing factor for a successful project outcome in terms of customer satisfaction, fulfillment of operational requirements, and the financial and time-related matters," says Broch.

Terma has also used systems engineering during the development of a self-protection system for the British Royal Air Force Tornado jets. The project was run on a very tight time schedule and crossed the finishing line on time and became a huge success. This was achieved by using mechanical, electronic, and software components from different projects and integrating them into a new solution, which met the customer's requirements.

"Once the pilots had got acquainted with the system, they refused to fly without it when flying on serious missions," recounts Mikkel Vestergaard Hansen, who heads Terma's Systems Engineering Division for Airborne Systems.

GN ReSound in Ballerup develops and produces hearing aids. Five to six years ago, there was a sharp rise in the complexity of the products, primarily due to the introduction of wireless communication with related accessories and software features. For this reason, GN ReSound decided to introduce systems engineering into its development work, which has been a key factor in the company regaining its lead in the industry.

Niels Christian Jensen, requirements manager at GN ReSounds Systems Engineering Department, explains, "We quickly became aware that it was crucial for the projects that mechanics, electronics, and software were not developed in separate silos, and that it was paramount to have a system-based approach to prevent grievous errors being discovered late in the development process, or that we even might be forced to reduce the functionality in order not to delay the launch of new products."

The GN ReSounds Systems Engineering Department is cross-disciplinary and deals with requirements specifications, architecture and design, and test processes. The group also serves as a link between product development and marketing.

"SE has given us a bird's eye view and greater cohesion with regard to our development work," Jensen says. "Our development projects are more predictable, and we avoid being surprised by grievous mistakes. We manage to develop the products that make sense to the end customers."

Systems engineer Jens Christian Andersen from Novo Nordisk in Hillerød is, among other things, a visiting lecturer on the development of safety-critical systems at the Technical University of Denmark. In his opinion, it could also be advantageous for small companies to apply systems engineering:

"Small companies, whose products have become increasingly complex with built-in electronics and software, would do well to tap into the experience that larger companies have with SE, and here INCOSE Denmark can play an important role," he says.

Systems engineer Henrik Balslev, partner in the consulting engineering firm Balslev & Jacobsen in Copenhagen, points out that the construction industries in Denmark are discovering the potentials inherent in systems engineering:

"It all began with the construction of the Opera House in Copenhagen, where SE elements, which turned out to be highly effective, were included in the project. These days, the construction industry is introducing fundamental SE elements as the new common industry language, which forms the basis for far more efficient construction. Regardless of the realm in which a company moves, SE will provide it with a simple and precise cross-disciplinary language. Furthermore, the built-in SE management tools ensure increased efficiency, improved finances, and quality throughout development projects."

Educational institutions are also showing increased interest in teaching systems engineering to engineering students. The Technical University of Denmark and the University of Aarhus are in the process of strengthening their efforts in the systems engineering area. According to Jan Madsen, professor at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark, "There is a clear industry need for graduates with SE skills."

"To us, it is interesting that we are dealing with a discipline that covers several different domains. As it stands today, we offer a few courses that are related to SE, but are planning to establish an actual degree programme. It is the plan to make SE a pivotal part of several different programmes. In that way, the graduates will still become specialists within their own area, and at the same time acquire a set of skills to make them understand how to build complex systems."

Madsen hopes that the Technical University of Denmark will be able to launch a systems engineering programme starting next year, focussing on information-technology and electronics development. Furthermore, he confirms that the Technical University of Denmark wishes to strengthen its systems engineering research, in particular concentrating on the early phases of development, where many of the central parameters suffer from significant uncertainties. A model-based approach makes it possible to model and simulate a system at an early point of the development process, so that systems engineering can be applied as a tool in the decision-making process.

PhD fellow Claus Ballegaard Nielsen from the Department of Engineering at the University of Aarhus says, "We are seeing increased industry interest in people who can think in systems and have interdisciplinary understanding, and there is also significant international focus on systems engineering. For this reason we would like to strengthen the SE programmes. We already offer two systems engineering classes, one of which is an introduction to the subject, and the other an advanced course for engineers at master's degree level."

INCOSE Events

ISSS 2013 International Conference
Haiphong, Vietnam

Timothy L. J. Ferris, tim.ferris@incose.org

The 57th World Conference of the International Society for the Systems Sciences was conducted in Haiphong, Vietnam, from 14–19 July 2013. Approximately 200 people, half local and half international, attended the conference. The Haiphong city government was the sponsor and facilitated many complex logistical actions related to the conference.

Haiphong is located near to the Ha Long Bay and hosts one of the major ports in the region, well upstream along a river. At the mouth of the river there is a large island, Cat Ba, which was declared a UNESCO biosphere reserve in 2004 because of the considerable biodiversity and number of exceptionally rare species native to the island and the much smaller surrounding islands in Ha Long Bay. The bay is famous for the limestone islands covered in tropical forests that rise out of the water vertically (figure 1). But the island is also home to a large number of people, including aquatic farmers whose livelihood comes from farming in floating villages in the bay.

The amazing scenery of the bay and islands led to the establishment of a number of tourist ventures which in turn had potential for bringing a large number of tourists who would over-stress the environment through their presence. Clearly this situation needed a major management initiative to ensure a balanced solution.

Figure 1. Floating farms around Cat Ba island (Photo by Timothy L.J. Ferris)

Factors present in the underlying situation were the delicate nature of the environment, the poor economic condition of the people, the attractiveness of the location for tourists, and the need for development in order to enable the people to have a reasonable share in their country. The initial attempts to address the situation were organised in the common piecemeal fashion of individual projects developed to provide for particular intended outcomes. Just as systems engineers would expect, such projects were found to be detrimental to the overall situation. In fact, the very success of the projects in delivering their individual objectives caused additional problems because of the distortion in prior relationships that was caused by the use of resources for different purposes which advantages some and disadvantaged others in the community. Overall, such individual aid projects did not overcome the community-wide recognised problems and increased pressure on the environment.

In 2008 Ockie Bosch and Nam Nguyen, both of the University of Adelaide Business School, began to study the situation using systems-thinking methods to understand the interaction of the diverse factors. Their approach recognised that the situation was truly multidimensional and that any solutions generated through isolating one or a few dimensions of the issues from the complex whole would prove to be inadequate, and probably to be solutions to the wrong problem. For example, they recognised that any appropriate solution would need to address the economic and occupational needs of the people at the same time as addressing the environmental needs and opportunity, and that any solution would need to be sustainable in all ways to be worthwhile.

Their investigation methods sought to structure the problem using system dynamics to identify the interactions between the wide array of factors identified through issue-surfacing investigations. The issue-surfacing activities were necessary to make the set of relevant factors explicit, so that the factors could then be incorporated into discussions and models. However, the issues were so serious that it was necessary for the factors to be organised using the system dynamics modeling method of the causal loop. This methodology enabled identification of the linkages in the complex array of factors and tools to work with the strength of relationships identified as present.

One of the crucial system-design factors is to determine what goals the various stakeholders have for an intervention, or potential intervention. In a systemic intervention in a complex situation it is necessary for the intervention to be designed so that it will satisfy the variety of stakeholders in order that all will be willing to cooperate with the intervention. If the goals are identified by an oligarchy to impose those objectives on all, then the majority are likely to be uncooperative, even if the goals identified are objectively beneficial, simply for lack of being involved in the process of identifying the goals. The project has now achieved several tangible outcomes, including reduction in the floating aquaculture farms, establishment of several tourist resort sites around the bay and on the island to manage tourist activity to reasonable impact on the environment and to enable long-term desirability of the tourist destination.

I have discussed the Cat Ba island project at such length because it is instructive of the value of holistic investigation of what is desirable at the front end of a project. The project has confronted the need for development to support the people of the island in the context of a World Heritage listed site of environmental significance. The methods used to explore the situation and the need and potential solutions are all known to systems engineers as methods to approach significant challenges. These include the soft-systems methods for exploring the perceptions of people about aspects of the situations that they face and more formalised methods, most notably constructing the relationships identified through the soft systems approaches into system dynamics models that enable exploration of the relationships of the factors in a manner that enables prediction of outcomes.

The Cat Ba island case study presents the application of systems thinking approaches to deal with a complex problem, with both pluralist or even potentially coercive relationship structures and a complex arrangement. It emphasizes that known methods and techniques enable significant progress by proposing and developing solutions that are appropriately designed and which receive the support of stakeholders. This shows the importance of the front-end activities in systems engineering: the activities that enable clarification of what should be made in order to satisfy the underlying need.

Sixty-seven papers were presented at the conference, of which six were related to systems engineering topics. The systems engineering papers addressed subjects which sit at the interface between the traditional subject matter of systems engineering, the methods of working with engineered systems, and the interests of the systems-thinking community, in which issues such as the investigations required for the implementation of soft systems methods are the major focus.

The major take-home insight I gained through attendance is something which all of us in systems engineering can derive value from:

We need to practice systems engineering reflectively, considering

- the appropriateness of the methods we use,
- the assumptions embedded in our approaches,
- the assumptions, and their implications, that underlie the methods we use,
- and we need to be willing to critique, analyse, and build upon the contributions of the great contributors to systems engineering methods and thought.

Reflective and engaged practice is necessary to provide the benefits of systems engineering to society so that the rest of society will value what we do.

Event Report: 25th Annual ICSSEA

Cecilia Haskins, cecilia.haskins@incose.org

In November, with the support of AFIS and other sponsors, the 25th Annual International Conference on Software and Systems Engineering Applications took place in Paris at the site of the Telecom ParisTech campus. Over 100 participants took part in three days of intense and often insightful exchanges on topics as varied as a geopolitical perspective on the future of the Internet by Louis Pouzin to Barry Boehm setting the record straight on the use of the spiral model to integrate systems and software engineering with incremental commitment. Tony Wasserman opened the conference with his keynote on community and commercial strategies in free and open-source software where he hypothesized that the half-life of software knowledge may be as short as five years.

This conference is the brainchild of Jean-Claude Rault who is also the general manager of a quarterly journal *Génie Logiciel* published since 1995 with contributions in both English and French. For more information about the conference visit the websites http://icssea.enst.fr.

The systems engineering track included three half-day tutorials: one by the author on an overview of systems engineering, a tutorial on SysML versus UML by Pascal Roques, who is author of the first book on SysML in French, and a tutorial by Jean-Michel Bruel on SysML and requirements engineering which followed a paper session on requirements engineering. The program for systems engineering was rounded out by a paper session chaired by Joe DeRosa of MITRE who also delivered an invited lecture on systems engineering patterns, and separate paper sessions on testing and on verification and validation.

INCOSE South Africa's Systems Engineering Winter School: A Collaborative Effort

Alwyn Smit, alwyn.smit@incose.org, chair, Systems Engineering Training Working Group, INCOSE South Africa

Monday, 24 June 2013, marked a major milestone in the calendar of INCOSE South Africa's Systems Engineering Training Working Group. This was the opening of a winter school in systems engineering, which had been planned over a long period in collaboration with the Project Performance International (PPI) and the University of Stellenbosch. For our presenter, Robert Halligan from PPI, this was also a different challenge to lecture undergraduates as opposed to the normal graduate students attending his courses.

Since inception of the Systems Engineering Training Working Group, our main aim has been to bring together the industry requirements for systems engineering training with the current academic offerings in an attempt to reconcile the perceived mismatch between the two. We started this process by interviewing key personnel from various companies and education providers. During these discussions the topic of systems engineering training at undergraduate versus postgraduate level often came up and it became clear that there were significant differences in opinion about teaching systems engineering at undergraduate level. That is where the idea for the winter school was born. It was intended to be an experiment in teaching basic system design concepts to undergraduates in their final year to test their level of insight and capability to absorb and apply these concepts.

The winter school was organized in collaboration with the Department of Mechanical and Industrial Engineering and marketed to the whole of the final-year class. Having made only two short marketing pitches, we were not entirely sure what the response would be. We were pleasantly surprised to receive 35 paid applications, one of which was a philosophy student who heard about the course from one of her engineering friends!

Students at the Winter School hard at work on one of the many class exercises (Photo by Alwyn Smit)

As Robert's assistant taking a back seat in the class, I had the privilege of experiencing a master at work. The ease with which Robert is able to elaborate on a topic by quoting one relevant real-life example after another is simply amazing. This is where years of experience makes the difference, when the lecture is not simply a recital of theoretical knowledge, but it is backed up with real-life facts on what works and what does not work.

Initial feedback received from students was positive and encouraging. By the end of the week it was clear that the students believed in the value of the systems engineering course and many expressed this personally to Robert and me. I guess we can say the experiment was a huge success!

Management in Practice in Krakow

Cecilia Haskins, cecilia.haskins@incose.org

A two-day conference with the theme Management in Practice — Case Studies attracted over 100 people in Krakow on 11–12 October 2013. The conference was jointly sponsored by the Tischner European University, the European Union, Kapital Ludzki, and INCOSE via the emerging chapter in Poland. Cecilia Haskins, INCOSE Director for Communications and New Chapters Coordinator, was the invited keynote speaker. She addressed this audience of project manager students and practitioners on "PMI and INCOSE: A Common Standard for Project Management," building on the recent survey results with an emphasis on opportunities for teamwork between project managers and systems engineers in projects. By way of introduction to

Henryk Metz, president of the emerging Polish chapter, summarizes for the Polish audience the systems engineering concepts presented by Cecilia Haskins (left).

INCOSE, Henryk Metz gave a multimedia presentation that consisted of 20 slides each shown for 20 seconds—a very impressive feat of timing and effective communication. INCOSE members from Wroclaw University attended on the second day, which provided an ideal opportunity for additional discussion and networking.

The Associate Dean of Management, Grazyna Urbanik-Papp opened the conference and introduced the ambitious program that included seven presentations with highly diverse topics. Professor Bipin Indurkhya spoke on "Thinking Like a Child: The Role of Surface Similarities in Stimulating Creativity." Krzysztof Niec reported on "Change of Organizational Culture and Its Impact on the Implementation of Projects at Santander Bank." Tischner alumnus Michal Paluch was especially effective in his presentation on "Knowledge Management of Human Capital based on the Author's Model of Adequate Education and Community Centered Learning Strategy." His model built on many themes that we espouse in systems engineering, including the breaking down of silos, and the importance of asking questions that lead to root problem definitions to achieve change. Przemyslaw Stanisz gave an energetic presentation on the value of crowd marketing and ways that his firm, Mintia, uses it to help clients earn money and save money. The conference concluded with a moving and heartfelt presentation by Katarzyna Lubas based on her own journey to master "Team Energy and Time Management."

Speakers, faculty, students, and local business comprised the more than 50 participants each day.

Chapter business, discussions, and networking in the break with (from left to right) Bart Czerkowski, Michalina Moscicka, Natalia Klimecka, Chris Ciesielski, Katarzyna Odynokow, Michal Kosiorek, and Cecilia Haskins (All photos by Terje Fossnes)

Event Report: INCOSE Autumn Academic Forum

Rick Adcock, richard.adcock@incose.org**; and Cecilia Haskins,** cecilia.haskins@incose.org

This autumn, the International Council on Systems Engineering's Academic Council sponsored an Academic Forum event hosted at Cranfield University School of Engineering in the United Kingdom on 22–23 October 2013. Rick Adcock, INCOSE associate director for education, was the general chair of the forum, which welcomed academics from universities in Europe and elsewhere, both those already affiliated with INCOSE and those with an interest in systems and systems engineering.

This is the first in a series of forums which aim to create an arena for discussion focused on issues of interest to the broader systems engineering academic community. It is the intention that ideas, relationships, activities, and potential products emerging from this forum will have significant value for those who attend and more widely. The theme of the first forum was the weaving of systems thinking and aspects of systems engineering into the curricula of a broad range of university students, including all engineers.

Day 1 of the forum began with short position statements on current activity relevant to the theme:

- Alan Harding, president of INCOSE UK, reported on work with the Engineering Council, the UK regulatory body for the engineering profession, to recognize the importance of a systems approach for all engineers.
- Ariela Sofer, professor and chair of systems engineering and operations research at George Mason University, runs the GMU undergraduate program in systems engineering, and is involved in courses on both Systems Engineering and Engineering Systems in a Complex World. She talked about the general issues and advantages of teaching systems and systems engineering to undergraduates.
- Stuart Arnold, a retired systems engineer and INCOSE Fellow, is currently working as a visiting professor at the University of Hertfordshire under a Royal Academy of Engineering Visiting Professor scheme. Stuart talked about his work developing a systems engineering module for all final-year engineering master's graduates, and of his experiences as the "lone systems engineer" in an engineering faculty with no history of explicit systems engineering teaching.

After a break, the morning closed with a panel discussion on the challenges and innovative approaches of teaching systems ideas.

- Cecilia Haskins from the Norwegian University of Science and Technology (NTNU) talked about systems engineering at NTNU focused in four areas: (1) knowledge-based systems engineering; (2) Lean systems engineering, as practiced by the Shell Eco-marathon (SEM) team since 2011; (3) learn-by-doing approaches for teaching systems engineering to master's students, which includes a laboratory project based on Lego Mindstorms; and (4) systems engineering methods integration in PhD research, where the PhD students are organizing their work, and expanding the range of the application of systems engineering and model-based system engineering in their research projects.

- Hillary Sillitto, INCOSE Fellow and visiting professor at the University of Bristol Systems Centre, presented five key skills for systems thinking applied to practice, and proposed that they cannot be taught, but that educators must instead create the conditions in which they can be learnt. Neil Carhart, a Bristol research associate and recent EngD graduate, complemented Hillary's perspective by spending a few minutes describing his experiences both learning and teaching on the Systems Centre's Sustainable Systems Module.
- Harold Lawson, INCOSE Fellow, independent consultant, and author of the book *A Journey through the Systems Landscape* (London, GB: College Publications, 2010), discussed his experiences of producing an overview of systems engineering for his own teaching efforts, and interesting project results from course participants.

After lunch, the participants divided into smaller groups to meet and discuss the ideas raised in the morning and consider a number of key questions:

1. What are the benefits of teaching systems thinking and systems engineering to a wider population of university students?
2. What current barriers or issues would need to be considered to help achieve those benefits?
3. How can the INCOSE Academic Council, working with others, help to tackle some of the barriers and hence realise the benefits?

The conclusions of this discussion can be briefly summarized as follows:

It is essential for the success for future complex problems that all those involved can understand and apply the basic concepts of systems thinking and that all engineers are familiar with and able to use the key concepts and principles of systems engineering.

While we have many good examples of the above and how to do it, the kinds of break-through success being discussed are hampered by a lack of a clearly defined and agreed description of what these key concepts and principles are. An expression of system thinking, its basis in systems science and its links to systems engineering expressed in a language accessible to engineers and non-engineers would greatly improve our ability to sell the value of these ideas.

Teaching a systems approach, and the tools and techniques of systems engineering, to multidisciplinary groups of students has a number of unique challenges. There are some very good examples of how to do this within the academic community, but these are not repeated everywhere. To compound these problems, both systems thinking and systems engineering have struggled to be accepted as valid academic disciplines and to gain traction within traditionally organized university departments.

Presenters at the INCOSE Autumn Academic Forum (from left to right): Areila Sofer, Hillary Sillitto, Cecilia Haskins, Bud Lawson, Stuart Arnold, Rick Adcock, Alan Harding, and Neil Carhart (Photo by John Deane)

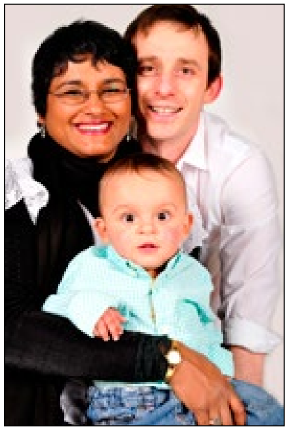
On the second day of the forum, some of the issues raised on day 1 were further elaborated. A number of existing INCOSE activities are already working on these issues. These include the Systems Engineering Body of Knowledge (SEBoK) and its associated Graduate Reference Curriculum; and the work of a number of working groups in particular the collaborations between the INCOSE Systems Science Working Group and the International Federation of Systems Research (IFSR). It is also clear that the academics within INCOSE could begin to work together on some of the issues directly related to them. A short summary of this Forum will be created and used to help frame the discussion for future meetings.

Participants in the first INCOSE Autumn Academic Forum in the atrium of the Vincent Hall on Cranfield University Bedford campus (Photo by John Deane)

INCOSE Spotlight

INCOSE Spotlight on. . . Suja Joseph-Malherbe

Sandy Young, info@incose.org



A family photo with Suja Joseph-Malherbe, her husband Daniël Malherbe, and son Jeevan Malherbe. (Photo by Rozanne Herbst)

Name: Suja Joseph-Malherbe
Title/Organization: Senior Engineer and Parliamentary Grant Theme Manager, Council for Scientific and Industrial Research (CSIR) — Defence, Peace, Safety and Security
Place of Birth: Ettumanoor, India
Current Residence: Stellenbosch, Republic of South Africa
Domain: Power and energy
Studied: Electrical and electronic engineering
Year joined INCOSE: 2008
Role in INCOSE: Cochair of the Ways and Means Committee
Years in systems engineering: I have been in engineering for more than a decade and more officially in systems engineering over the past six years.

How would you describe systems engineering to an eight-year old? If I was explaining what systems engineering was to my son, I would say, "Imagine you wanted to design and build a tree house. Systems engineering helps you to describe the tree house and what it needs to do for you, like keep you dry when it's wet and rainy. Systems engineering helps you decide on the best solution based on all that you want the tree house to be, how much money you have, the tools you need, the space you have available and so forth."

What did you want to do for a job when you were a little kid? I wanted to become an astronaut so that I could go into space, and I believed the way to get there was through engineering.

When people ask "What do you do at your job?," what do you say? I don't say much except that at work I feel like "Q" in the James Bond movies.

What are you working on currently? More recently, I worked on the development of a power-management system and lithium-ion battery pack for the dismounted soldier.

What trends do you see in portable power-management and energy-storage systems? There is an increasing need for power monitoring so that we can optimize power consumption and achieve long-term power savings. With more energy being harvested from alternative energy sources, demand for storing this energy is growing rapidly. There is also a need for a modular design approach to portable power management and energy storage systems that can allow us to meet a wide range of possible applications.

What work accomplishment are you most proud of? I was given the opportunity to be the portfolio manager of our department's Parliamentary Grant fund. It is directed towards building and strengthening our company's science, engineering and technology base, developing new knowledge, and applying this knowledge. We achieve this through investing the fund in equipment, human-capital development, and projects.

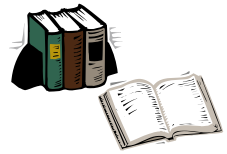
Since your husband (Daniël Malherbe, current treasurer of the INCOSE South Africa chapter) is also a systems engineer, do you ever collaborate with your husband? Yes, we have collaborated: Both of us served on the INCOSE South African Chapter Management Committee for three years. We also publish together in conference proceedings. Daniël and I always keep an eye out for opportunities to collaborate, and when the environment is right, we'd love to work together on a more permanent basis. Just to make sure we're both well-rounded engineers, we attend art classes together.

How has INCOSE benefitted you? INCOSE offers a fantastic opportunity to engage with the finest minds in the profession, both locally and internationally. When you become an active member, INCOSE also offers a platform for developing leadership skills. For the South African chapter, I served as branch coordinator in the Western Cape for two years and as the chapter membership officer for one year.

What do you like to do outside of work? I am an ardent reader. Currently I am interested in leadership skills to deal with complexity and working knowledge in organizations. I am in the process of self-publishing *Ammachi's Quilt*, a children's book about a character named Lyka and her grandmother. I enjoy the outdoors tremendously and try to go camping and hiking as much as I can. I also enjoy art, especially making abstract pieces.

What is the biggest challenge you face? Being a woman, born in India, married to a Caucasian South African, raising my son, developing engineering solutions in a male-dominated world, completing this questionnaire…and loving the challenge.

Book Reviews



Insight into Insight

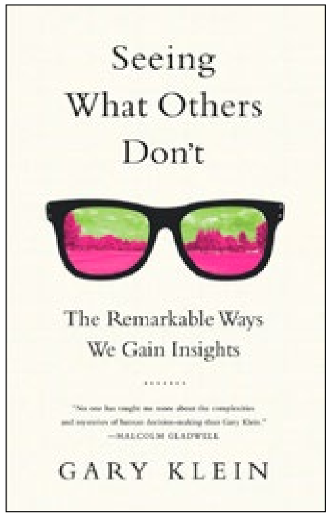
Seeing What Others Don't: The Remarkable Ways We Gain Insights

By Gary Klein

New York, US-NY: Public Affairs, 2013 (ISBN-13: 978-1-61039-251-8)

281 pp., including notes, story index, and index

Reviewed by Denise Howard, s.denise.howard@incose.org



Aerospace engineer Burt Rutan found himself at an impasse with the difficult problem of enabling a spacecraft to safely re-enter the atmosphere, as he worked on the architecture for what would become SpaceShipOne. Rutan later recalled yelling, "I've got it! I've got it!" when the resolving insight came to him in the middle of the night.

Daily experience suggests that insights often inform successful engineering projects, from the top level to the smallest details. Insights are often sought whenever there is a need to make things better or to do something that has never been done before. Thus, the ability to have and recognize insights should be valuable to systems architects and engineers.

Just what is this phenomenon commonly referred to as insight? Author and experimental psychologist Gary Klein began collecting accounts of what he thought of as insight. *Seeing What Others Don't* is a relaxed, informal account of this exploration, based on his collection, a qualitative top-level sifting and revision of the notion of insight, from which he expands the concept and derives a new model of insight, the "Triple Path Model."

Klein, currently a senior scientist at MacroCognition, is probably best known for his work on naturalistic decision-making. His career has taken him through the academic world, the United States military, and into his own research and consulting business. He has given seminars on cognitive systems engineering and has coauthored papers appearing in IEEE publications and in *Systems Engineering*.

Insight is often identified with suddenness, with the aha! moment, similar to what Rutan described. Klein traces this identification to the Wallas model of insight, probably familiar to many readers, where the "stages of insight" are categorized as preparation, incubation, illumination, and verification. Even psychologists who study insight tend to accept this model without question. However, as Klein coded and categorized the 120 accounts of insight in his collection, he began to think that an alternative to the Wallas model might be needed. He enlarges the notion of insight to include insights without any identifiable preparation or incubation phase and insights that emerge slowly.

The three ways to insight in Klein's model are contradiction, connections, and creative desperation. It may help to note that the categories are not mutually exclusive: For instance, the Rutan example might be primarily an insight of creative desperation, but also secondarily of connection, arising from his childhood experiences with the way airplane models landed. The arrival of the insight subsequently changes understanding and creates an outcome affecting acting, seeing, feeling, and desiring. In three parts, respectively, the book addresses how the model originated, how insights are often discouraged, and how they can be encouraged.

Part II includes an entertaining cautionary tale of how decision-support tools can draw awareness away from the very areas needed for insight. It continues with a compelling explanation for the conflicted relationship organizations often have with insight, seeking it to solve problems, yet rejecting it when such solutions involve disruption, the nemesis of most organizations. Klein elaborates in this section on a dichotomy he has set up early in the book, between reducing errors and promoting insight, often one at the expense of the other. Both are fully needed in most projects.

In Part III, on how to encourage insights, Klein walks the reader through each of the major pathways in his model. He also describes how one individual might help another achieve insight, by artfully raising "faulty" assumptions to awareness, an infrequently discussed skill. Interestingly, he does not point out how more familiar strategies for producing insight seem to correspond neatly with the outcomes part of his model, strategies like triggering insight by deliberately changing action, seeing, feeling, or desiring.

For some, the casual presentation may conceal the importance of the book. For instance, a bibliography providing background for readers unfamiliar with the existing work on insight might be expected. And a small source of confusion will undoubtedly disappear with the second printing: in recounting Jocelyn Bell Burnell's discovery of pulsars (an instance of slow insight), speeding up the data tape to enhance detail, rather than slowing the tape down (p. 48), is probably what the author intended.

» continues on next page

Two Delightful Books from One Great Fellow

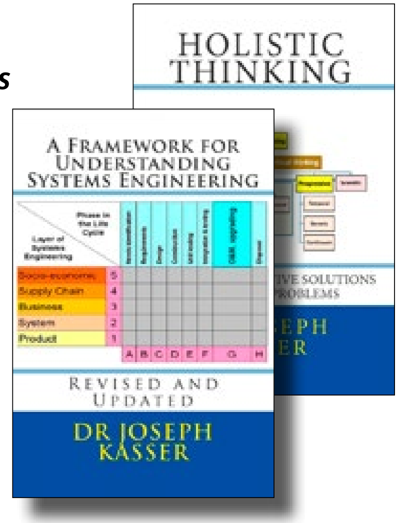
Holistic Thinking: Creating Innovative Solutions to Complex Problems

By Joseph E. Kasser
Cranfield, GB: The Right Requirement, 2013. ISBN: 978-1482539011. 471 pp.

A Framework for Understanding Systems Engineering (2nd edition)

By Joseph E. Kasser
Cranfield, GB: The Right Requirement, 2007. ISBN: 978-1482758160. 510 pp.

Reviewed by Cecilia Haskins, cecilia.haskins@incose.org



Long-time INCOSE members will recognize Joe Kasser from the cover of the ***INSIGHT*** April 2012 issue. In his interview published in that issue, he tells us that his pre-academic background includes 30 years of professional systems engineering experience, and that he uses magic and references to his hobby as an amateur radio operator to keep his lectures interesting. The books reviewed here encapsulate decades of original papers on problem solving, systems thinking, and systems engineering. Both books are well organized, filled with a blend of practical suggestions and quotes from the literature, and sprinkled with illustrations and cartoons. Each book concludes with an extensive list of references, which is helpful to novice and expert alike.

Holistic Thinking

This book opens with background chapters on systems thinking and tools that enable holistic thinking, such as concept maps and active brainstorming. There are also chapters on the nature of systems and decision-making. But this reviewer's favorite chapter was chapter 9 with its "no holds barred" approach to the challenges of creating problem solutions and a classification of types of problems, and the use of an insomniac named Fred as an illustrative case in point. The book closes with chapters containing concrete examples of the application of systems engineering to problem solving, personal insights, and a summary of the whole book in chapter 12. This last chapter is a great way for readers who need a refresher to come in and "reread" the 420 pages.

A Framework

A preview of the framework is colorfully situated on the cover of this book but does not appear until page 310, after laying a foundation for its appreciation. Chapter 7 makes a the case for the certification of systems engineers and many chapters discuss a systems engineering body of knowledge, both of which are better established in the INCOSE community than they were at the time of the first edition. A number of chapters also deal with the systems-software debates, and the book closes with a chapter that affirms the systems engineering is a discipline, despite the existence of seven camps or different worldviews on systems engineering. In his April 2012 interview, Joe confessed his desire to one day resolve the conflicts between these camps. This chapter makes a start by suggesting that "the approach to reconcile the camps is to distinguish between two systems engineering paradigms"—namely, the role of the systems engineer in the workplace as separate from the activities associated with systems engineering, which essentially can be performed by anyone. Only time will tell if this approach succeeds.

Taken Together

This reviewer is personally glad to have both books on the shelf, and ready for handy reference. Each fills a unique need, the former gives some good examples of problem solving techniques and problems solved using these techniques. The latter, contains a handy reference of thought-provoking papers on the history and practice of systems engineering.

Insight into Insight *continued*

Seeing is of interest though not for what it details or settles but for the possibilities it opens up. The researcher selected the cases according to his own criteria, but what might another researcher select? Would new cases fit the model or might some adjustments be needed? How, for instance, would this study accommodate the quintessential insight that brought language to the young Helen Keller? In this case, there was no narrative because there was no prior language, so if insight is a changing of the narrative, how should we regard the source of this emergence?

Along with the more expansive definition of insight and the new model, *Seeing What Others Don't* also reinforces several important ideas. First, insight can often result from thoughtfully examining our fundamental models and understandings, especially those that have long been accepted without question. Second, simple qualitative exploratory methods, like collecting cases and coding them can yield helpful new categorizations, which can serve as the basis for later, more rigorous studies. Finally, it reminds us that insight restructures our understanding, which may be the most powerful architecture of all.

Design Structure Matrix Methods and Applications

By Stephen D. Eppinger and Tyson R. Browning
Cambridge, US-MA: The MIT Press, 2012

Reviewed by Joseph Kasser, joseph.kasser@incose.org

A Design Structure Matrix (DSM) chart is a tool used to represent the interactions on the connections between the elements of a system. The book states that Professor Don Steward of California State University, Sacramento coined the term DSM in the 1970s (p. 12). This means that it was developed at about the same time that Robert J. Lano developed the N^2 chart that should be familiar to most systems engineers, while working at TRW. Both charts are tools that are more useful in analysing and designing complex systems (defined as systems with large number of parts and interconnections between the parts) than node-connection diagrams. The chart takes the form of the matrix shown in figure 1. The difference between the N^2 chart and the DSM is the way that the connections are shown:

- The N^2 chart shows outputs in rows, and inputs in columns.
- The DSM chart shows outputs in columns, and inputs in rows.

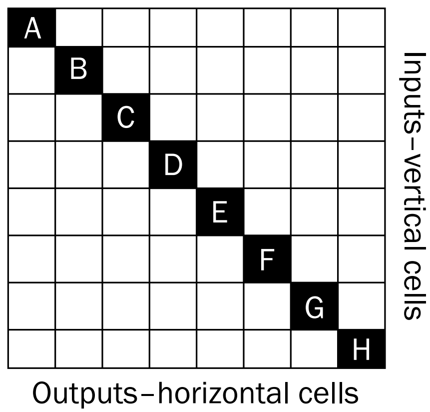


Figure 1. N^2 chart matrix

To create a chart, one draws a table with an equal number of rows and columns where the rows and columns represent the elements of the system. Since an element does not connect to itself, the square in the table where the row and column for the same element meet can be blocked out and the element name inserted, as represented by the letters in figure 1. In an N^2 chart, outputs from A to the other elements of the system are represented by O's or X's in the square in the A row and the column associated with the element as shown in figure 2. In DSM chart, outputs from A to the other elements of the system are represented by O's or X's in the square in the A column and the row associated with the element.

A	O		O	O	O		
O	B	O	O	O	O		
	O	C	O			O	O
O	O	O	D			O	O
O	O			E	O	O	
O	O			O	F	O	O
		O	O	O	O	G	O
		O	O		O	O	H

Output

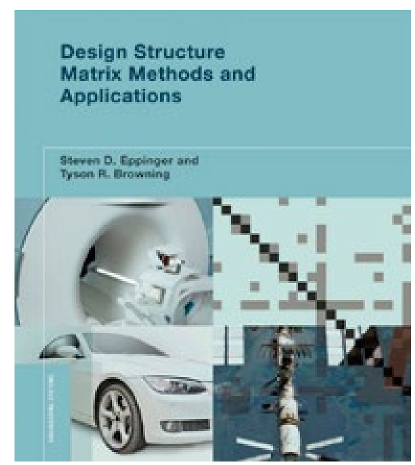
Input

Figure 2. Populated N^2 chart

In general, the N^2 chart is taught as a tool for identifying connections without providing information about the nature of the connection. However, there is no reason why the N^2 chart cannot. The N^2 chart is also used with unused parts of the matrix abstracted out, such as in the waterfall representation of the system-development process. The DSM on the other hand is described in this book with many examples of how to show the different type of connections between the elements of a system using, and always using the full matrix.

The book describes a variety of application examples of the use of the DSM in different process and product domains and in dealing with different problems in various industries. The book uses colour to make the connections clear. These are typical examples:

- Using a DSM to represent and analyse the architecture of complex systems
- Applying a DSM to represent and analyse organizations that develop engineered systems and the types of insights gained through these DSM applications
- Applying a DSM to represent and analyse product development processes for engineered systems and the types of insights gained through these DSM applications
- Using a DSM in matrix models that represent two or more domains at once



The book has a large number of colourful charts that illustrate the different types of information that can be shown in the connection between the system elements.

Is this book useful? Well, it depends. The book does deserve a place in an organizations' library to provide junior systems engineers with ideas and lots of examples of how DSM and N^2 charts can be used in various situations. This is a book on a specific set of methods that will be extraordinarily useful to some readers more than others.

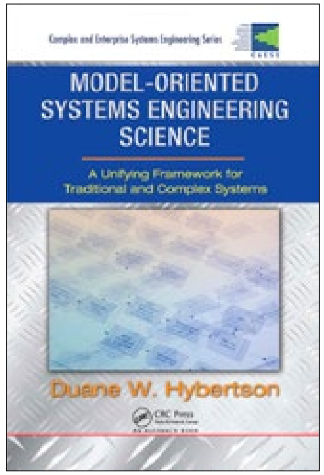
Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems

By Duane W. Hybertson

Boca Raton, US-FL: CRC Press, 2009 (ISBN 978-1-4200-7251-8)

379 pp including index, 69 illustrations

Reviewed by Janet Singer, jsinger@soe.ucsc.edu



The Need For and Aim of This Book

This book is the culmination of over two decades of work on how to unify and leverage existing knowledge and best practices from systems engineering and a wide range of other disciplines into a coherent science capable of supporting 21st-century systems engineering practices. It discusses the strengths and weaknesses of "traditional systems engineering" and presents both a vision for the systems engineering of the future and a fairly detailed proposal for how to get there:

> The opportunities are the possible areas of expansion in the field. The challenges are to start with an SE that has strengths and weaknesses within its current scope, and to define an SE that is more robust, covers a significantly expanded scope, and constitutes a coherent whole of the old and the new. Fortunately, many of the solution elements for an expanded and robust SE already exist, in some cases within systems engineering but in other cases in related or supporting disciplines. What is needed is to bring the solution elements together into a unified whole. That is what this book aims to do: define a robust and unifying foundation for an expanded SE of the future.

This "robust and unifying foundation for an expanded systems engineering" is the "systems engineering science" of the title. The substance of systems engineering science is provided by the bodies of knowledge in the contributing sciences, drawing on biology, sociology, cognitive science, organizational theory, management science, computer science, and network science—any field that can provide knowledge of complex systems useful to systems engineering. Systems science (including general systems theory, systems thinking, and complex-systems science) provides support for extending and unifying systems engineering by identifying the common and distinct patterns of all systems, from mechanistic to organismic or autonomous, whether natural or artificial.

The qualifier "model-oriented" in the title refers to the means proposed for accomplishing this massive task. The model orientation (MO) does the "heavy lifting" in unification by bringing a common perspective to the whole enterprise, including both engineering and management:

> In this book we extend existing modeling approaches into an MO that views all science artifacts (such as theories, laws, and observed patterns) and all engineering artifacts (such as requirements specifications and designs) as models of systems. Based on this orientation, all traditional and complex systems engineering methods, processes, and artifacts and all supporting sciences, are organized into a virtual structured repository called the systems engineering model space. In effect, the model space is envisioned as a container for the unified SES and unified systems engineering body of knowledge of the future.

About the Author

Duane Hybertson is a researcher and member of the technical staff at the MITRE Corporation in McLean, Virginia (US). He has a broad background in software and systems engineering, including architecture, modeling, patterns, service orientation, security, foundations of systems, complex systems, and enterprise engineering. He has extensive experience applying these systems engineering principles and practices in large, complex systems for several of MITRE's government sponsors. Dr. Hybertson has also conducted research and published in the areas of foundations of architecture, security patterns, rapid system acquisition methods, and applying models of systems science and complex systems to systems engineering. Together with Markus Schumacher and Eduardo Fernandez-Buglioni he was coauthor of the book *Security Patterns: Integrating Security and Systems Engineering* (Chichester, GB: Wiley, 2006).

Content and Structure

The book begins with a description of how current and anticipated challenges and opportunities lead to specific requirements for an extended and unified systems engineering. Of the 24 requirements identified, traditional systems engineering is judged to be strong in seven, such as "Support mechanistic characteristics of systems," "Support a mix of matter, energy, and information elements in systems (i.e., a mix of conceptual and physical elements in systems) in a seamless way, including the dominant role of information and computation," and "Support multiple views and perspectives of a given system or class of systems." Extended and unified systems engineering must build on those strengths, while supplementing nine areas that are currently only partially addressed, such as "Support balance of simplicity and complexity: everything should be as simple as possible, but no simpler;

recognize imperative of complexity" and "Support clear communication among all systems engineering stakeholders but recognize that not all clear communication is explicit; support both explicit and implicit communication." Eight requirements are new and include "Support organic and agent characteristics of systems," "Support arbitrary, fuzzy, and uncertain system boundaries," and "Support tensions, dualities, contradictions, contrasts, and paradoxes in a unified way that includes both their separation and their integration in the manner of yin-yang principles."

The remainder of the book shows in detail how these requirements can be satisfied by the proposed approach. The contribution of system science and a view of the relation of system science to systems engineering are presented. Chapters 3 through 5 introduce some of the core concepts, definitions, and key features of the new systems engineering science, including model orientation and the context and structure of the "model space," the structured repository of knowledge of all systems engineering-related disciplines. Chapters 6 through 10 present more details on each dimension of the model space: composition, commonization, conceptualization, time, and views. The specification approach of "model-oriented systems engineering science" (MOSES) and mapping of traditional systems engineering artifacts to the MOSES structure are described in detail. Finally, an expanded and unified systems engineering process called "collective actualization" is discussed.

Usability

This is a very rich—nearly encyclopedic—book, with extensive contextualized citations and crossreferenced interconnections throughout, supported by many helpful figures and tables. The density of material and singularity of the author's vision mean multiple passes will be required to absorb and appreciate all that is presented. But it is clear that the author imposed exacting standards on himself, resulting in an exceptionally high level of scholarship, coherence, and clarity of presentation throughout. Repeated reading, whether straight through or jumping to sections of particular interest, will be rewarded with deepening insights.

The book would be a valuable reference if all it did was identify and coherently organize the many knowledge and practice areas that need to be integrated for future systems engineering. But it also provides guidance on a mindset that enables the unities within this material to be appreciated by seeing the complementary roles played by contradictory elements and paradoxes in the "dualities" of unified wholes. (The chapter on Views is particularly helpful in this regard.) At a third level of reading, the detailed presentation of the proposed MOSES framework will be immediately of use to anyone directly involved in related integration work. And, given the systematic care with which the MOSES structure is described and explained, even casual readers will likely find it becomes natural and helpful with increased familiarity.

Should You Buy This Book?

This book presents a contribution to systems engineering foundations arguably on par with the work of Wayne Wymore (1967; 1993) and John Warfield (1976; 1994). The remarkable achievement in MOSES is that the author does not allow his awareness of the massive complexity of the material to deter him from proposing a novel "good enough" model-oriented unification platform for going forward. His respect for the many who have come before—grounded in extensive reading, decades of practical experience, and obvious deep reflection on the issues—allows him to avoid the usual traps and pitfalls of "old wine in new bottles," "reinventing the wheel," and looking for a "silver bullet" that plague so much of the literature on complex systems theory and practice. Anyone working on or just interested in transformational systems engineering, the science foundations of systems engineering, or the promise of unified systems engineering will want to have this book as a resource.

References

Warfield, J. N. 1976. *Societal Systems: Planning, Policy, and Complexity*. New York, US-NY: Wiley.

———. 1994. *A Science of Generic Design*. Ames, US-IA: Iowa State University Press.

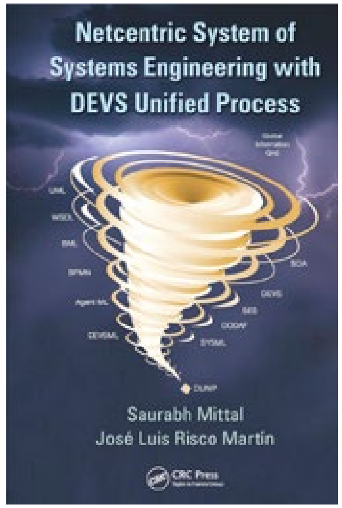
Wymore, A. W. 1967. *A Mathematical Theory of Systems Engineering*. New York, US-NY: Wiley.

———. 1993. *Model-Based Systems Engineering*. Boca Raton, US-FL: CRC Press.

Netcentric System of Systems Engineering with DEVS Unified Process

By Saurabh Mittal and José Luis Risco Martin
Boca Raton, US-FL: CRC Press, 2013 (ISBN 978-1-4398-2706-2)
684 pp., including list of acronyms and index

Reviewed by Sarah Sheard, sarah.sheard@incose.org



A long-standing conflict between the systems engineering and the software engineering definitions of the "system of systems" is eroding because all systems of note today contain some software (at least in development tools, but almost always in the system), and nearly all software today must be considered in terms of its system functions. Nevertheless, there is still a significant difference between systems engineering approaches that take a big-picture, functions-first view, and software engineering approaches that are much more interested in the details of how to make the software work.

This book clearly takes the second approach. A randomly selected 25 pages in the book contained 31% code (or program output), 42% text, 6% equations, and 12% figures (plus references and exercises), which seems consistent with the book as a whole. This translates to over 200 pages of code and about 280 pages of text.

In this book, system development is development of a discrete-event simulation model of the logical behavior of a distributed system of computers and software. INCOSE readers who are not interested in this view of systems of systems would be better off reading other books.

That said, and given the trends in systems engineering suggested by the opening sentence, there is interesting content in this book for systems engineers who are interested in model-based systems engineering and in different kinds of simulation, as well as those interested in how the software for distributed systems of computer systems can be modeled. In particular, by reading sections I, IV, and V, INCOSE systems engineers whose work intersects heavily with software engineering can profit from

- learning terminology used by software simulation experts;
- reading the hundreds of pages of Java programs, in lieu of taking a Java class;
- the list of personnel needed to be brought together to create a good system-of-systems-simulation, including the role of the systems engineer, according to software simulators (p. 185); or
- learning what the software engineers expect a systems engineer to do regarding interviewing system-of-systems personnel to create system simulation scenarios and use cases for the software for a distributed system of systems (section IV, beginning on page 499).

The contents of the book are as follows. Sections II and III will not appeal to "generalist" systems engineers, as they are intended more for graduate students in system-of-system simulations.

Section I (chapters 1–6, 188 pages) describes the "Basics." Chapter 1 defines simulation, object-oriented software systems engineering (the authors' term), and introduces the Java programming language (which constitutes most of those 200 pages of code). Chapter 2 describes systems of systems and complex systems without seeming to understand INCOSE's uses for the terms, even though the references cited are consistent with INCOSE's uses. Instead, after four pages of high-level description, chapter 2 dives right into a hierarchy of system-simulation methods and model formalisms. Chapters 3 and 4 list the formalism associated with discrete event simulations (DEVS) and then the modeling and simulation metamodels used in the book. Chapter 5 discusses the DEVS language in a manner that was incomprehensible to me. (Sample sentences: "The atomic DEVS formalism has deltint, deltext, deltcon, and lamda functions to specify the atomic behavior," and "All the above behavior specifications are code-assisted and validated, as behavior is specified in the editor.") Chapter 6 discusses the DEVS Unified Process. This is not a process like systems engineering processes or CMMI. This process defines stacks and transformations among kinds of models, as well as how to align with a software-development framework called OpenUTF.

Section II (chapters 7–12, 148 pages) is called "Modeling and Simulation-Based Systems Engineering." This is not INCOSE's "systems engineering" except possibly as a working group called "Model-based software systems engineering" would see it. According to the preface, this section is for graduate students and advanced practitioners of DEVS, and for industry professionals who want to "learn the advanced capabilities of DEVS-based systems engineering methodology [sic] and to use [model-driven engineering] in their efforts." Chapter 11 describes, and recommends changes to, the United States Department of Defense Architecture Framework, version 1.0. (Note that this was replaced by version 1.5 in 2007, and the framework is now on version 2.02.). Chapter 12 suggests how that framework and other architecture frameworks can be tested.

Section III (chapters 13–17, 160 pp.), is intended for people "who are interested in building DEVS virtual machines and netcentric SoS." This section has 50% code, 27% figures and tables, and only 15% text (again sampling 25 pages). I can't imagine many INCOSE systems engineers making it all the way through this section, much less using it. I can only see software engineers using it.

Section IV (chapters 18–22, 140 pages) is about "Case Studies," which sounds like where INCOSE systems engineers would find value. Chapters 18–20 indeed discuss designing a software simulation from informal scenarios. These chapters contain primarily figures and code. They would certainly help teach a senior INCOSE systems engineer what simulation software engineers on a distributed software system might be looking for, or better, serve as a conversation starter between the systems and software engineers along the lines of, what do you need and what should I be trying to create for you? Chapter 21 describes executable UML. Much of this content has been made available by the INCOSE Model-Based System Engineering Working Group in a way that is more digestible by INCOSE systems engineers; however, the advantage of this chapter is that it appears in a book with the other content, with similar language and formalisms. INCOSE systems engineers who have survived chapters 7–20 will find this chapter easy to read and relatively familiar. Chapter 22 addresses using these techniques to model an enterprise's business processes.

Section V (chapter 23) purports to address Netcentric (sic) Complex Adaptive Systems. The background sections are in agreement with the definitions used by INCOSE's Complex System Working Group. Particularly valuable is the table of questions to ask about the complex adaptive system to be able to model it correctly (p. 650 and following). However, the chapter falls short of providing easily understood advice about modeling such a system.

Weaknesses include a tendency to jump right into Java coding detail after minimal context and explanation, failure to explain the acronyms before using them (and not all are in the acronyms list at the back), and the use of confusing and unidiomatic English such as the following: "To the contrary of the structured programming, centered on functions, object-oriented programming is centered on data," and "When we define a class that does not extend other class, Java uses the Object class, which is the root of the class hierarchy in Java."

INCOSE Author News

China Aviation Publishing & Media Co. Ltd of Beijing, China, has announced that it will be publishing a Chinese translation of INCOSE Fellow Scott Jackson's book *Systems Engineering for Commerical Aircraft*, first published in 1997 (Aldershot, GB: Ashgate).

Just One More Thing

Andrew Cashner, cashner@uchicago.edu, assistant editor, *INSIGHT*

This is my final issue as assistant editor of ***INSIGHT***. My heartfelt thanks go to Chief Editor Bob Kenley for a productive and supportive collaboration over the last seven years. I thank Bob, Chuck Eng, Cecilia Haskins, and all those throughout the Council, for their efforts to make this periodical an open forum for international discussion among systems engineering practitioners across subfields and specialties. I have enjoyed the opportunity to work with the regular contributors and theme editors. With apologies to the many others not listed, these include Rick Dove, Denise Howard, Samantha Robitaille, John Thomas, Dave Walden, and Holly Witte. I especially wish to thank Jack Ring for his original and thought-provoking articles. I am a musicologist, not a systems engineer, but I have been inspired by Jack Ring's concept of knowledge as dynamic exchange within a community, to be measured not in quantity but in velocity—and I have been privileged to see that exchange happening in these pages. There are many other cases in which my encounter with systems engineering has influenced my work as a musician and musicologist: from inspiring me to think in terms of systems (my dissertation in part traces networks of musical influence and exchange across seventeenth-century Mexico and Spain), to motivating me to approach the research process itself as a systems engineering challenge.

Editing ***INSIGHT*** has given me a special vantage point into the INCOSE community, and I have admired the warmth and collegiality within the discipline. In the smallest of ways, I have sought to help this publication represent your vibrant, highly dynamic community in all its breadth. Going forward, I hope ***INSIGHT***, and INCOSE in general, continues to seek a truly global perspective. I hope the INCOSE members will continue to foster a culture of straightforward communication that will make discoveries and ideas in specialized domains widely accessible. Very best wishes to my successor Lisa Hoverman and to all of INCOSE.

Final Thoughts From the Chief Editor

Bob Kenley, insight@incose.org

In a previous issue (vol. 16, no. 2), I reported on the new role for Wiley in handling the printing, distribution, and advertising for ***INSIGHT***. In this issue, I am reporting on another change to our operations.

Andrew Cashner has served as assistant editor since the July 2006 issue that featured "The Use of Systems Engineering in Large-Scale Emergencies." When Andrew joined our staff, his primary occupation was serving as a music director at a church. He has since earned a master's degree in sacred music from the University of Notre Dame and is completing his PhD in music history and theory at the University of Chicago. While he was engaged in his principal profession of performing music and researching music history and theory, he enhanced the presentation and content of our contributors' articles, developed a tailored application of a standard US style guide to account for our diverse international readership, and reported on keynote speakers at two international symposia. Many, many thanks to Andrew for his outstanding service to INCOSE, and best wishes to him in future endeavors.

Lisa Hoverman will be stepping in to fill the role of assistant editor for the April 2014 issue on standards. She has a bachelor's degree in biology with minors in chemistry and psychology from Carlow University in Pittsburgh, Pennsylvania (US), and a doctorate in molecular, cellular, and developmental biology from the University of Pittsburgh. She served as a postdoctoral researcher at Pennsylvania State University, and subsequently has been an independent medical communications specialist. She has done grant and proposal writing for industry and academia that has included supporting healthcare divisions of Microsoft and General Electric. She is very excited at the opportunity to learn more about systems thinking and systems engineering while working with our contributors to prepare their articles for publication.

The upcoming issues on standards and agile systems engineering are progressing well, and the theme editors and their team have finished selecting articles. If you have an idea for an article for the issue on model-based conceptual design, resilient systems, agile security, and model-based systems engineering, please contact the theme editor. Hervé Panetto and his colleagues will be doing a reprise of this issue and will present us a new collection of articles by French doctoral students in the December 2015 issue of ***INSIGHT***.

It is my privilege to announce that the two emerging chapters appearing in this issue, namely, Denmark and Poland, have been chartered as INCOSE chapters in the EMEA sector by the board of directors just before going to press. We welcome them and look forward to future reports appearing on these pages.

Finally, I invite you to participate in the *2012–13* ***INSIGHT*** *Readers' Choice Survey** and to cast your vote for the issue and the article that you wish to be recognized as the readers' favorites at the 2014 International Workshop.

* https://connect.incose.org/admincomm/comm2/insight/Lists/201213%20INSIGHT%20Readers%20Choice%20Survey/overview.aspx

Upcoming submission deadlines and themes for INSIGHT

Issue	Submission Date for General Articles	Theme	Theme Editors
April 2014	15 February 2014	Standards	Ken Zemrowski
July 2014	15 May 2014	Agile Systems Engineering: Active Management of Unpredictable Risk	Rick Dove
September 2014	17 July 2014	2014 International Symposium Coverage: Las Vegas, Nevada (US)	LIsa Hoverman
December 2014	15 October 2014	Model-Based Conceptual Design: Engineering the Problem Space	Kevin Robinson, Quoc Do, and Michael Waite
April 2015	15 February 2015	Resilient Systems	Katri Hakola, David Yarbrough, Paul Cleary, and Scott Jackson
July 2015	15 May 2015	Model-Based Systems Engineering	Joachim Fuchs
December 2015	15 October 2015	AFIS Doctoral Symposium: Systems Engineering Research Challenges in French Universities	Hervé Panetto
July 2016	15 May 2016	Agile Security	Rick Dove

INSIGHT
International Council on Systems Engineering
7670 Opportunity Road, Suite 220
San Diego, CA 92111-2222