December 2011 | Volume 14 Issue 4

INCOSE
International Council on Systems Engineering

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

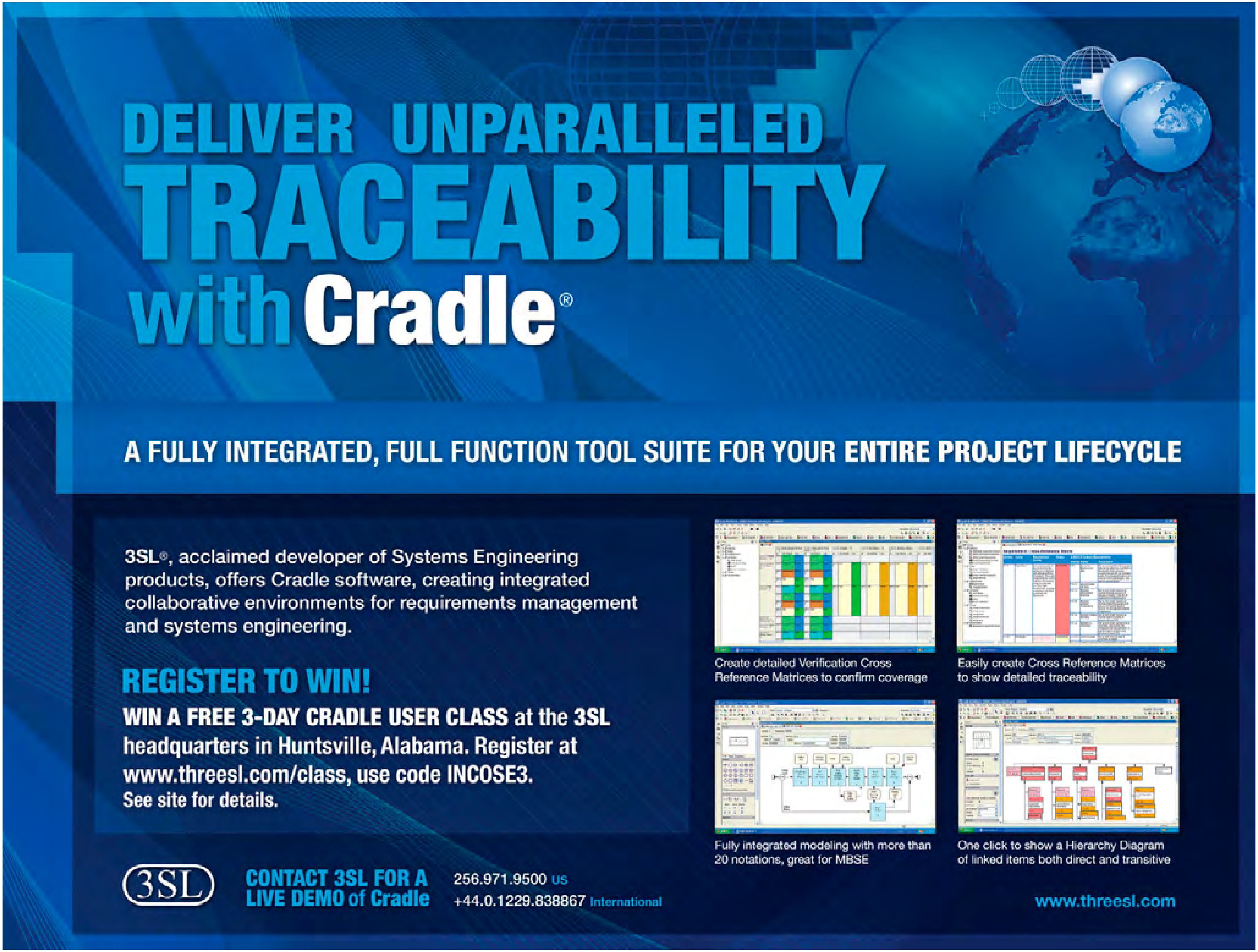
What's Inside

What's Inside

SPECIAL FEATURE

AFIS Doctoral Symposium: Systems Engineering Research Challenges in French Universities

AFIS Association Française d'Ingénierie Système


DELIVER UNPARALLELED
TRACEABILITY
with Cradle®
A FULLY INTEGRATED, FULL FUNCTION TOOL SUITE FOR YOUR ENTIRE PROJECT LIFECYCLE
3SL®, acclaimed developer of Systems Engineering products, offers Cradle software, creating integrated collaborative environments for requirements management and systems engineering.
REGISTER TO WIN!
WIN A FREE 3-DAY CRADLE USER CLASS at the 3SL headquarters in Huntsville, Alabama. Register at www.threesl.com/class, use code INCOSE3.
See site for details.
Create detailed Verification Cross Reference Matrices to confirm coverage
Easily create Cross Reference Matrices to show detailed traceability
Fully integrated modeling with more than 20 notations, great for MBSE
One click to show a Hierarchy Diagram of linked items both direct and transitive
3SL
CONTACT 3SL FOR A LIVE DEMO of Cradle
256.971.9500 US
+44.0.1229.838867 International
www.threesl.com

INSIGHT
Publication of the International Council on Systems Engineering

Chief Editor Bob Kenley
insight@incose.org +1 260 460 0054
Assistant Editor Andrew Cashner
andrew.cashner@incose.org
Theme Editor
herve.panetto@incose.org Hervé Panetto
Advertising Editor Christine Kowalski
advertising@incose.org +1 858 541 1725
Layout and Design Chuck Eng
chuck.eng@comcast.net +1 206 364 8696
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 2012*** issue – 15 February; ***July 2012*** issue – 15 May; ***September 2012*** issue – 26 July; ***December 2012*** issue – 15 October
For further information on submissions and issue themes, visit the INCOSE website as listed above.

Advertising in *INSIGHT*. Please see http://www.incose.org/ProductsPubs/periodicals/insight.aspx or e-mail advertising@incose.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 8000+ 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, directors, and member board representatives.

President's Corner

Samantha Robitaille, samantha.robitaille@incose.org

My desk at home in Michigan is in a lovely room with windows on three sides, looking out over the garden surrounded by tall trees. It is a wonderful vantage point from which to observe the changing of the seasons, brilliantly illustrated by the local flora and fauna. As I write, autumn has definitely arrived and there are as many leaves on the ground—in every shade of green, yellow, orange, and brown—as there are on the trees. Squirrels, especially the oversized and rather handsome eastern fox squirrel, have largely completed their ransacking of the oak trees but continue to bury their treasures all over the garden. The woodchuck trots along the edge of the lawn, mouth full of dried leaves to line her winter burrow. Migratory sparrows have arrived and join the throng under the bird feeder, which looks like a busy railway station at rush hour. When the birds depart, the chipmunk moves in to supplement his winter supplies, too. Everywhere there is activity, played out against a backdrop which changes daily. My seventeen-year-old son waits impatiently for the first snowfall.

And so I sit down to write my last "President's Corner," reflecting back not only on two years as president but on nearly eight as a member of INCOSE's Board of Directors. How busy we have been! Those eight years have seen our membership grow by nearly 50% and the number of chapters increase by a third, with typically four new chapters each year. Our Corporate Advisory Board membership has nearly trebled and in 2011 we will certify ten times the number of system engineering professionals than we did in 2004.

INCOSE is a truly remarkable organization. I have come to know many active members, and all have one thing in common: passion. We seem to attract people who, no matter what their personal or employment circumstances, give to INCOSE and to the profession with an extraordinary energy that enables our small organization to continue to grow and do great things.

Four years ago, I laid out my vision for INCOSE along four main themes, and I have been pleased with the progress we have made. Regarding the first theme, participation, we continue to increase the intellectual strength of our volunteer organisation by engaging the incredible breadth and depth of expertise of our membership. We have improved working-group participation outside the United States and made a number of products developed by individual chapters available to all members. Almost half of our new Fellows in the past eight years come from outside the US, reflecting a significant improvement in global engagement. We still have some way to go, however, and are reliant on improvements in our information-technology infrastructure to make my vision a reality. I am hopeful that our director for information technology, Ryan Mortimer, will finally make this happen in 2012.

The second theme was partnership: enhancing systems engineering by improving our relationships and interfaces with other disciplines, and our cooperation with other systems engineering organisations. We have formally established a number of these partnerships, across a spectrum from systems science to program management (as reported in the last issue of ***INSIGHT***), and from safety to SysML. We have also strengthened our relationships in a number of countries to support the growth of certification programs and the translation of INCOSE products, most recently in Taiwan and Japan.

The third theme was professionalism. Continuing to build both INCOSE's professional reputation and the reputation of individual systems engineers remains critical if we are to increase our influence on systems engineering practice, in both traditional and new domains. Certification has now exceeded its thousandth registrant, and the program continues to grow as new companies and countries adopt it as their "internal," but globally recognized, standard. Creating a systems engineering certification program was perhaps the

President's Corner *continued from page 3*

most important thing that INCOSE has done for the profession in the past 10 years. I applaud all those who had the vision and tenacity to create it, and all those who supported the initiative through its embryonic stage—especially those who have made a personal commitment and become certified.

My final theme was performance – for members to set and achieve realistic targets in their volunteer activities, recognising the external, organisational, and personal constraints. This continues to be a challenge for us as a volunteer organization, especially in these difficult times. I would like to thank all those who have chosen to give their time and energy to INCOSE in recent years—to working groups, chapter activities, administrative committees, planning events, and so many other activities. Without you they could not happen. I am painfully aware that for each service award we give at the International Symposium, there are dozens of volunteers who are never properly recognised. I would like to take this opportunity to thank each one of you for helping to make INCOSE what it is.

Finally, a change in the presidency is a time not just to look back, but to look forward. Last month, we held our annual elections. My congratulations to all those who were duly elected, in particular to David Wright who will join the 2012–13 leadership team as president-elect.

In addition to the leadership elections, we asked members to vote on a proposal to change the way in which members and chapters are represented at the Board of Directors, through the introduction of three sectors, each with a voting director. I am delighted to say that members voted overwhelmingly to support the proposal, with over 91% in favour. This is a historic moment for our organization, and is the culmination of several years' work by many in the leadership team to evolve our organizational structure that both continues to serve our existing membership and supports our international growth. My personal thanks go to everyone who helped to formulate the proposal by participating in our strategy sessions over the past three years, and to everyone who voted to provide such a clear mandate for the way forward. Preparations are already underway to identify potential candidates for sector director and for the chapter presidents of each sector to vote and appoint their board member.

The view from my window reminds me that the vibrant world around us is always active and the environment changes all the time. Over the years that it has been my privilege to serve you, INCOSE has been, and remains, a truly active organisation —for which I thank each one of you. The ballot outcome confirms that we are not only active, but evolving in response to the changes in our environment. As my term as president draws to a close, I am confident that our organisation will continue to grow, to evolve, and to thrive as we pursue our mission to serve humanity and the planet through systems engineering, in new domains, industries, and countries around the world.

Correction

The print edition of volume 14, issue 3, of ***INSIGHT*** had an inadvertent error on page 45. The recognition of new Fellows showed Dov Dori's photo was repeated in the position intended for Chandru Mirchandani. Our sincere apologies to Chandru. The digital edition had the correct photo.

Award Winners

INCOSE Fellows class of 2010. From left to right: Dov Dori, Regina Griego, Jerry Fisher, and Chandru Mirchandani.

STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY

The School of Systems & Enterprises

Real World Problems. Innovative Solutions.

APPLY TODAY FOR SPRING 2012

The School of Systems & Enterprises (SSE) at Stevens Institute of Technology is nationally recognized as a leader in the delivery of high impact graduate studies in systems and engineering. Leveraging our partnerships with industry and government our students are taught to be thought leaders and problem solvers, with the ability to find breakthrough solutions across a wide range of domains, including Defense, Homeland Security, Intelligence and Finance.

Discover the SSE Advantage

- We meet the demands of industry and government by developing engineers who possess inter and trans-disciplinary skill sets.
- The SSE curriculum covers a broad range of targeted subject matter from cyber security and software assurance to infrastructure architecture and systems engineering management.

Networks That Matter

- Partnerships with industry and government ensure that our research course designs are grounded in the state-of-practice for real-world environments and applications.
- Selected by the Department of Defense in 2008 to manage the Systems Engineering Research Center (SERC), a UARC focused on Systems Engineering Research

Online via Web Campus • On Campus (Hoboken, NJ or Washington, DC) • On site (at various industry locations)

Graduate Programs:

Systems Engineering
Systems Security Engineering
Software Engineering
Space Systems Engineering
Engineering Management
Enterprise Systems
Graduate Certificates in over 20 focus areas

FOR ADDITIONAL INFORMATION:
School of Systems and Enterprises
Stevens Institute of Technology
Castle Point on Hudson, Hoboken, NJ
sse@stevens.edu

www.stevens.edu/SSE

SSE School of Systems&Enterprises

INCOSE
International Council on Systems Engineering

Promote INCOSE

To obtain materials to promote INCOSE in the workplace and at events, such as regional conferences and chapter meetings, contact the INCOSE Administrative Office:

info@incose.com
+1 858 541-1725, or access the INCOSE website at
www.incose.org

7670 Opportunity Road
Suite 220
San Diego, CA 92111-2222

We supply INCOSE table signs, promotional items, and informational materials.

Corporate Advisory Board — Member Companies

Air Force Center for Systems Engineering
Alliant Techsystems
Analytic Services-Applied Systems Thinking Institute
Ansaldo STS
Astrium, an EADS Company
ATKINS
BAE Systems
Boeing Commercial Airplane Co.
Boeing Defense, Space & Security
Boeing Defense, Space & Security–East
Booz Allen Hamilton Inc.
C.S. Draper Laboratory, Inc.
Carnegie Mellon University Software Engineering Institute
Cassidian
Certification Training International
Cranfield University
Cummins, Inc.
Defense Acquisition University
Deloitte
Deputy Assistant Secretary of Defense for Systems Engineering, US Department of Defense
DRS Technologies, Inc.
EADS N.V.
Federal Aviation Administration US
Ford Motor Company
General Dynamics
General Electric
George Mason University
Honeywell International
IBM Corporation
ITT
JAXA (Japan Aerospace Exploration Agency)
Jet Propulsion Laboratory
Johns Hopkins University
Keio University Collaboration Complex Library
L-3 Communications
Lockheed Martin Corporation
ManTech International Corporation
MAP systeme
Massachusetts Institute of Technology
Medtronic, Inc.
Missouri University of Science and Technology
Mitsubishi Electric Corporation
National Aeronautics and Space Administration
National Geospatial–Intelligence Agency
National Reconnaissance Office
National Security Agency
National University of Singapore
Naval Postgraduate School
Naval Surface Warfare Center–Dahlgren Division
Northrop Grumman Corporation
Orbital Sciences Corporation
Pacific Northwest National Laboratory
Project Performance International
Raytheon Corporation
Rockwell Collins, Inc.
Rolls-Royce
Saab AB
SAIC
Sandia National Laboratories
Scitor Corporation
SELEX Sistemi Integrati SpA
Serco Defence, Science and Nuclear
Siemens–UGS PLM Software
SPAWAR Systems Center Charleston
SRA International
Stevens Institute of Technology
Swedish Defence Materiel Administration
Systems Engineering Innovation Centre
TASC, Inc.
Tectura Corporation
Thales
The Aerospace Corporation
The MITRE Corporation
The SI
UK MoD
United Technologies Corporation
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

Abbreviations Used in this Issue of *INSIGHT*

ASEP	Associate Systems Engineering Professional
BKCASE	Body of Knowledge and Curriculum to Advance Systems Engineering (Pronounced “bookcase”)
BOD	INCOSE Board of Directors
CAB	INCOSE Corporate Advisory Board
CAG	Certification Advisory Group
CSEP	Certified Systems Engineering Professional
DoD	United States Department of Defense
DoDAF	United States Department of Defense Architecture Framework
ESEP	Expert Systems Engineering Professional
EST	Eastern standard time (UTC-05)
GRCSE	Graduate Reference Curriculum for Systems Engineering (“Gracie”)
HSI	Human systems integration
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
ISO	International Organisation for Standardisation
IT	Information technology
MBSE	Model-based systems engineering
MoD	Ministry of Defense
MoDAF	Ministry of Defense Architecture Framework
NASA	United States National Air and Space Administration
OMG	Object Management Group
SE	Systems engineering or systems engineer
SEBoK	Systems Engineering Body of Knowledge
SEH	*Systems Engineering Handbook*
SEP	Systems engineering professional
SwEBoK	Software Engineering Body of Knowledge
SysML	Systems Modeling Language
UK	United Kingdom
UML	Unified Modeling Language
UPDM	Unified Profile for DoDAF/MODAF (United States Department of Defense Architecture Framework and United Kingdom Ministry of Defense Architecture Framework)
UTC	Coordinated Universal Time
v (e.g., v0.1)	Version
V&V	Verification and validation
WG	Working group

Two-letter abbreviations used for US states and Canadian provinces are listed in the *Chicago Manual of Style*, 16th ed. (Chicago, IL [US]: University of Chicago Press, 2010), §10.28–29, pp. 498–499; and (for US states), at the website of the US Postal Service, http://zip4.usps.com/ncsc/lookups/usps_abbreviations.html.

Three-letter abbreviations used for currency are determined by ISO and listed at http://www.iso.org/iso/support/faqs/faqs_widely_used_standards/widely_used_standards_other/currency_codes/currency_codes_list-1.htm.

We use the United Nations Statistics Division Standard Country and Area Codes Classifications for the names of global regions listed at http://unstats.un.org/unsd/methods/m49/m49regin.htm. Note that the United Nations nomenclature uses the neutral term *Western Asia* instead of the more common but Eurocentric term *Middle East*. Two-letter abbreviations used for countries determined by ISO are listed at http://www.iso.org/iso/country_codes/iso_3166_code_lists/country_names_and_code_elements.htm.

INSIGHT SPECIAL FEATURE

AFIS Doctoral Symposium: Systems Engineering Research Challenges in French Universities

Introduction by theme editor Hervé Panetto, herve.panetto@incose.org

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 2–3 December 2010 in Bordeaux, France, with the objective to develop strong relationships between industry and academia. It gathered 120 participants (of which one third were from industry).

The expected cross-fertilization between academia and industry developed within 10 workshops on these topics:

- Learning systems engineering while doing research
- Teaching systems engineering: What and how?
- Model-based safety systems
- Human-based cognitive systems
- Architecting systems and services
- Model-based systems interoperability from an organizational perspective
- System engineering scientific foundations: Open questions
- Patterns for MBSE
- Safety from a systems engineering perspective
- The RobAFIS student robotics competition

Four invited lectures completed this programme by addressing these topics:

- "BKCASE (Body of Knowledge and Curriculum for Advanced Systems Engineering): Where are we?" by Art Pyster (Stevens Institute of Technology) and Alain Faisandier (MAP système)
- "Systems Engineering: The 2020 vision of AFIS" by Catherine Devic (EDF)
- "INCOSE: Future and Perspectives" by INCOSE President-elect John Thomas (Booz Allen Hamilton)
- "Systems Engineering Virtualization" by Yannick Fourastier (EADS, Aerospace Valley)

Additionally, a preforum meeting was also 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. The 14 papers that follow are expanded from poster presentations by PhD 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 that is linked with industrial needs.

The first article, "Systems Interoperability Evaluation through Formalisation of Semantic Relationships" addresses, in the context of information-systems engineering, the issue of enterprise interoperability measurement. This approach allows any enterprise to fully evaluate its own capacity to interoperate, and therefore to anticipate possible problems before a partnership. Indeed, there is a need for a "Formalization and Exploitation of the Coupling between Systems Engineering Methods and Product Lines" for adapting systems engineering process and methods in order to apply them to the development of families of systems. Such structuring may be helped by "The Decomposition of a Process and the Definition of Interoperability Metrics." These activities allow one to distinguish between business activities and non–value-added activities. In order to cope with distributed enterprise information systems, the next article proposes "A Harmonized and Reversible Development Framework for HLA-Based Interoperable Application."

"Model-Based Systems Engineering with SysML for Reliable Systems Design" focuses on defining a method for improving the realization of reliability analysis during the systems engineering process and its early design phases. There is a need for "A Design Pattern Metamodel and Use Mechanisms for Systems Engineering," based upon a global systems engineering metamodel. The whole metamodel includes most of the entities needed to support a systems engineering process that will conform to the ISO 15288 standard. From a system perspective, the next article proposes

» continues on next page

"The Distributed and Interoperable Architecture SCEP-SOA as a Way to Manage Multisite Production Projects": this approach brings systems engineering into a multisite project's management process in a service-oriented market, based on a generic architectural framework.

In the context of product development, other authors suggest a "Framework for Product-Lifecycle Management Systems in Extended Enterprise: Application on Mechanical Small and Medium Enterprises" with an application on small and medium mechanical enterprises. This framework allows one to create a data model that manages the product data from an extended enterprise. At the same time, it is necessary to define "Abstract Constraints: A General Framework for Solver-Independent Reasoning on Product-Line Models." This research contributes to systems engineering with a representation of the semantics of product-line models as abstract constraints with a unique notation that encompass other constraint languages.

The supporting system, "Using Object-Oriented Bayesian Networks to Model an Industrial System: A New Approach to Assessing Maintenance Strategies" computes a set of multicriteria indicators that help one to assess, whether the functional architecture of the maintenance system and its associated strategies are able to satisfy the system of interest's objectives and the enabling system requirements. The article on a "A Model-Based Platform for Long-Term Knowledge Retention" proposes a knowledge engineering methodology dedicated to product-line models and a multilayer architecture for long-term knowledge retention, establishing a digital-preservation platform.

As the next article states, "The emergence of ever more technically advanced customer needs, combined with incremental competition on a global range, is fueling the demand for more innovative products." Indeed, "Process Modeling in Innovative Design using Systems Engineering" provides an important tool for companies by giving them a way to analyze the factors that have direct effects on innovation through system engineering.

Complex systems engineering needs "Semantic Alignment between Enterprise Repositories" that also aligns different engineering-project views. And finally, "A Systems Engineering Framework based on Eco-Design" may well be the future of the discipline.

We would like to thank all authors for their contributions and all reviewers for the tremendous work they have done to improve the papers.

A Student Challenge in Systems Engineering: RobAFIS 2010

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

RobAFIS is a student engineering competition to design robots, organized by the French Chapter of INCOSE (AFIS) each year since 2006. The competition brings together approximately 10 teams of students coming from an engineering program of a French university or graduate engineering school. Those students may be pursuing either a bachelor's degree or a master's degree in engineering. In most cases, they will be fresh graduates with a bachelor's degree but little or no experience in systems engineering. Their lack of experience is a challenge in realizing the educational outcomes. Each team consists of at least three students, including a project manager, and is advised by a systems engineering professor who plays the role of "teaching reference."

Objectives

RobAFIS relies on a comparative evaluation of the robots produced by the concurrent teams and represents a teaching operation of AFIS. The objective is to lead students to implement methods of systems engineering, to work in a collaborative way to design their solution, and to produce and provide a development document.

In the preparatory phase of the competition, it is important to check the feasibility of the mission: several teams must succeed in performing the expected mission to maintain the students' motivation. During the competition, the students and their referent have the possibility of questioning AFIS systems engineering experts who work in industry or teach systems engineering. The questions can relate to the course of the competition, to its regulation, and to technical or methodological aspects. The answers or comments they provide enhance the systems engineering training.

After the competition, a debriefing is carried out with each team to emphasize the positive points and the drawbacks of their solution. A systems engineering expert gives a detailed evaluation of the project, considering whether the team's development document reflects good use of systems engineering tools and methods, and explains the results of the audits of configuration and the performance of the robot during the operational validation.

Organization and Schedule of the Competition

The place of the operational phase is within a higher-education establishment. The steering committee is composed of a team who is in charge of the management of the competition and provides the technical expertise that is necessary to answer students' questions. The schedule of the competition, covering about 10 months, is as follows:

- Communication about the general schedule of the competition (To – 8 months)

» continues on next page

Tucoulou et al. *continued*

- Publication of the regulation, the specifications, and the reference Development Document (To – 6 months)
- Registration of the teams (To –3 months)
- Free supply of a LEGO Mindstorms kit for the building of robots (To – 2 months)
- Reception of the Development Documents (To –15 days)
- Evaluation of the Development Documents by experts in systems engineering
- Final stage: Tests, configuration audit, operational validation, project audit (To)
- Detailed debriefing with each team (To + 1 to 2 months)

Organization of the Final Stage

Part I. Tests for setting up the robot and on-site checking

Before the operational evaluation, each team has an access to the zone of evaluation to perform tests for setting up the robot and functional checking in situ.

Part II. Configuration audits

The robot is completely disassembled and the parts are deposited on a table in front of the jury (figure 1). The team has 15 minutes to assemble the robot on the basis of the configuration written in their development document. A jury must make sure that the assembly conforms to the solution definition recorded in the development document. The audit is then completed by the feasibility checking of one of the preventive maintenance actions that are written in the development document, in a section on "aptitude for maintenance."

Figure 1. Configuration audit with systems engineering experts

Part III. Operational validation

The operational validation (figure 2) is performed as a succession of matches pitting two teams' robots against three of the others. Each robot must achieve the mission described in the specifications three times, so that the jury is able to check that the solution is robust (and to give greater chances of success for each team).

Figure 2. Operational validation

Part IV. Brief closing presentation (lessons learned): project audit

After the operational phase, each team has to present a final review in front of the jury (figure 3). The team explains the reasons for their success or failure during the operational validation and analyzes the possible engineering difficulties and problems.

Figure 3. A student team during project audit

Part V. Public announcement of the results and distribution of awards.

The organization committee presents the final ranking of teams, and distributes awards in the presence of AFIS systems engineering experts and representatives of the AFIS board.

Assessment Criteria of the Development Document

The appreciations that are given by the members of the jury do not consist of a judgment on the technical quality of the product because this aspect is evaluated by the results obtained during the operational validation. The judges only assess the quality of the implementation of the systems engineering processes and the quality of the development document, taking into consideration the initial specifications. The marking system of the competition is decomposed into four points:

- 40% for the quality of the development document
- 10% for the configuration audit
- 40% for the results obtained during the operational evaluation
- 10% for the quality of the brief closing presentation of the project

Key Interests of the Competition for the Organization Committee

The organization committee is in charge of the processes and activities concerning the management of the competition (the competition is structured like a professional project with milestones), the relationships between the contracting authority (here the main stakeholder is the organization committee) and the prime contractor (the student team), the development of the specifications for consultation, and the feasibility studies before consultation. This results in the production of the reference documents of the competition (regulation, specification, reference document for the development). The committee organizes and participates also in the evaluation of the development documents, and controls the final phase.

Major "Lessons Learned" for Student Teams

For student teams, the competition offers various interests:

- A comprehensive view of the major systems engineering processes and activities, through the realization of a concrete system and the writing of the development

» continues on next page

Tucoulou et al. *continued*

document.

- Understanding of the impact of the systems engineering process on the final product quality.
- Learning on key systems engineering topics:
  - Organization and management of a development project (WBS, risk management, cost control, planning, assignment of roles within the project team)
  - Requirements analysis and management
  - Configuration management and traceability
  - Architectural design, functional and physical modeling, and allocation of performances
  - Justification of the architectural choices, production of integration and validation plans
  - Guidelines for structuring of a development document
  - Verification and validation for the engineering phase and for the integration phase
- Meetings and exchanges between students and with systems engineering experts.

Systems Interoperability Evaluation through Formalisation of Semantic Relationships

Esma Yahia, esma.yahia@incose.org; Alexis Aubry, alexis.aubry@incose.org; and Hervé Panetto, herve.panetto@incose.org

To remain competitive, enterprises increasingly need to collaborate with each other and evolve into extended enterprises or networked enterprises. These organisation configurations incite the different enterprise systems to be interconnected in spite of their functional, structured, and conceptual heterogeneity. Typically, these features refer to the interoperability that can be defined as the capacity of systems or organisations to provide or to accept services and to use those services to effectively operate together (IEEE 1990). To promote added value creation through this collaborative activity, there is an increasing demand for information exchange and knowledge sharing among the various information systems (known as technical interoperability). Information and communications technologies and implementing standards can contribute to solving (at least partially) the barriers to technical interoperability. However, these efforts remain insufficient to guarantee interoperability at the conceptual level, where enterprises in a networked system can share information with each other, interpret this information correctly according to common business semantics and then use this information to achieve a global mission.

Interoperability Evaluation

Interoperability is not visible when it is effective, but the lack of interoperability poses a series of challenging problems to the industrial community. Indeed, it leads to significant costs, largely attributable to the time and resources spent when exchanging information. This increased cost and the resulting delays in providing expected services could severely hurt enterprises' global performance. For example, a 1999 study by Tassey, Brunnermeier, and Martin estimated that in that year the lack of interoperability drained at least USD 1 billion per year from the United States automotive supply chain.

We propose to address the measurement of enterprise interoperability in order to allow any enterprise to fully evaluate its own capacity to interoperate with others, and therefore to anticipate possible problems before a partnership. To do this we must first define the indicators and metrics to quantify, and then we must qualify the interoperation between the enterprise systems.

Formalizing the Interoperability Relationship

We propose at first to provide a formal mathematical definition of the interoperability relationship between the enterprise systems models (Yahia et al. 2011). For this reason we identify the necessary and sufficient model semantics by defining a semantically recursive structure composed of aggregates, what we call *semantic blocks*. These semantic blocks will eventually guide the formalization and then facilitate the evaluation of the interoperability relationship through mathematical functions.

Measures of the Potential Interoperability and its Effectiveness

Interoperability evaluation consists of discovering the semantic losses when interoperation occurs and then qualifying their effects on the interoperation. Based on the mathematical

» continues on next page

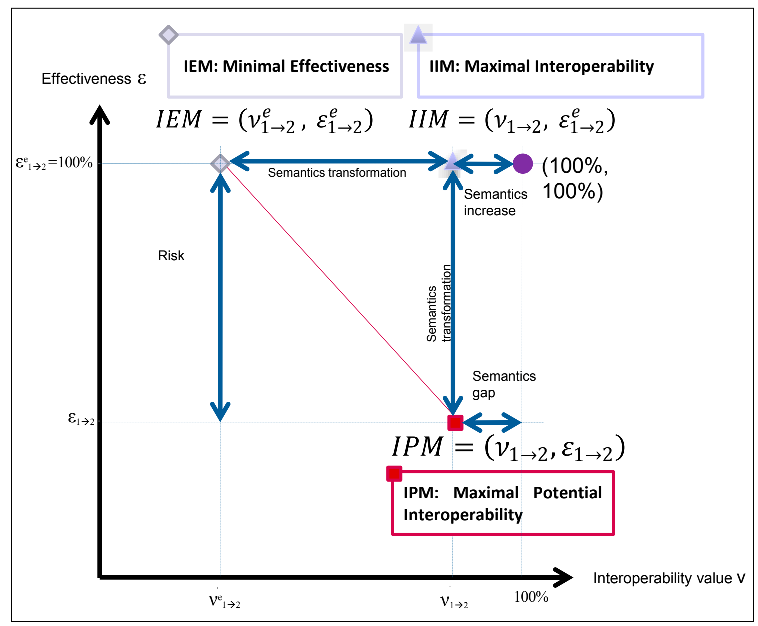


Figure 1. Map of interoperability measures

Yahia et al. *continued*

formalization of the semantic relationship, we proposed two measures, the potential (ν) and the effective (ε) interoperability, for assessing the interoperability between two information systems. A map of interoperability measures (figure 1) makes it easier not only to identify the semantic gap (the missing part of semantics that causes a problem of understanding by the receiving system), but also to evaluate what must be done to improve the interoperation.

Perspectives

We proposed a formal approach to interoperability assessment that provides measures in order to assess quantitatively and qualitatively the interoperation between collaborative information systems. One of our perspectives is to apply our formal approach with different maturity models that are mainly based on semiformal evaluation (Panetto 2007, Ford et al. 2007). In addition, we have defined the various actions that the collaborative enterprises could take to improve their potential and effective interoperability. Future research should detail how to add semantics when studying the interoperation from one way of exchange (of information) and what the effect of that transformation could be on the interoperation in the other way. We are studying the possible correlation between both interoperation directions.

References

Ford, T., J. Colombi, S. Graham, and D. Jacques. 2007. "The Interoperability Score." Paper presented at the Fifth Conference on Systems Engineering Research, Hoboken, NJ (US), 14–16 March.

IEEE (Institute of Electrical and Electronics Engineers). 1990. *Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, NY (US): IEEE.

Panetto, H. "Towards a Classification Framework for Interoperability of Enterprise Applications." *International Journal of Computer Integrated Manufacturing* 20 (8): 727–740.

Tassey, G., S. B. Brunnermeier, and S. A. Martin. 1999. "Interoperability Cost Analysis of the US Automotive Supply Chain." Research Triangle Institute Final Report. RTI Project 7007-03. Report to the US National Institute of Standards and Technology, Gaithersburg, MD, (US).

Yahia, E., M. Lezoche, A. Aubry, and H. Panetto. 2011. "Semantics Enactment in Enterprise Information Systems." *Proceedings of the 18th IFAC World Congress* 18 (1): 13064–13073. Milan, Italy: IFAC (International Federation of Automatic Control). Available through IFAC-Papers Online, doi:10.3182/20110828-6-IT-1002.03597, http://www.ifac-papersonline.net/Detailed/51763.html

Formalization and Exploitation of the Coupling between Systems Engineering Methods and Product Lines

Cosmin Dumitrescu, cosmin.dumitrescu@renault.com; **Camille Salinesi,** camille.salinesi@univ-paris1.fr; **and Alain Dauron,** alain.dauron@renault.com

As time to market continuously decreases, capitalization and reuse have a great impact on the development of reliable, customer-oriented systems. In the automotive industry, where product diversity represents a key factor in the success of the company, the need for sharing common assets in systems families is a major concern. The consequence for systems engineering is the need to adapt methods, techniques, and tools to meet the challenge of diversity.

The subject of creating customized products has been studied from multiple perspectives, such as software product lines, product platforms, modular design, or market-related studies. In the automotive industry this task is particularly complex, as it must take several points of view into account, and it has very high chances of generating erroneous configurations (Astesana and Dauron 2010). The purpose of our work is to adapt the systems engineering process and methods in order to apply them to the development of families of systems. This effort in part requires us to adopt a proper model for the representation of variability in the context of model-based systems engineering activities.

The reuse theme is extended to all engineering assets, such as requirements, generic components, or model fragments. This led us to a domain of recent interest in software development: software product-line engineering, which shares the common goal of reusing development assets. Numerous approaches to product-line engineering already exist. Among the different formalisms proposed in this domain we can distinguish: features (Kang et al. 1998), orthogonal variability (Pohl et al. 2005), aspect-oriented variability, constraints (Salinesi et al. 2010), views, and decisions.

However, we believe the core issue is the conceptual mismatch: bringing the different models used in system engineering together with those employed in engineering product lines, while they are conceptually different. The challenge is multiple: to specify the variability with systems engineering models, integrate the reusability of design processes, guiding reuse, and ensure consistency of documents produced during development.

In the automotive industry, it is difficult today to specify, integrate, and analyze variability from multiple sources such as the environment where the system will be used, vehicle features (external variability, observed by the client), technical vehicle characteristics that impact the system, and the internal behavior and structure of the system. Today, reuse in the automotive industry most often targets component reuse through vehicle architecture strategies (such as modular architecture or platform), production line flexibility, or interaction with the market and customer preferences through software-based product configurators.

An effective framework for reuse in systems engineering can bridge the gap between customers' needs and corporate strategies, down to the final vehicle-production phase in the factory.

Our research methodology is based on a series of engineering scenarios for a vehicle system that involves variability. The engineering scenarios and the collection of requirements for a variability model (presented in table 1), combined with a case study, will allow us to select, adapt, or create the appropriate model.

Table 1. Requirements for variability-management formalism in the context of systems engineering (non-exhaustive)

Generality	The approach must be applicable to all types of lifecycle and system artifacts.
Stakeholder participation	The approach must integrate all actors from the organizational structure that participate in the development. It must be comprehensible for all stakeholders.
Planning and Methodology	The approach must explicitly support the different kinds of reuse existing in automotive development and in the company's practices.
Traceability	The approach must support traceability of variations along the system's lifecycle and across the lifecycle of each artifact.
Scalability	The model must allow implementation of a scalable approach.
Recursion	The approach must be recurrent, applicable to subsystems.
Technical Integration	The approach must integrate to the existing organizational technical context external to the system engineering process (e.g., the existing means of describing diversity).

Based on the observation of current development practices, we have started to define a set of engineering scenarios:

1. *Synchronizing systems development*. Several systems within the same product line are developed in parallel, while all reuse related to commonalities between systems is anticipated and planned.
2. *Integrating a single system into a product line*. This is often the case of an innovation that enters mainstream development. A new system or option is proposed, but it is still possible to share lifecycle assets with existing systems (e.g., improvement of a braking system by adding safety capabilities like anti-lock braking or electronic stability control)
3. *Merging two product lines*. This represents a middle case that can be useful when different systems are at first developed independently, but common elements can be identified for future developments (e.g., development of similar products in different companies that eventually establish a partnership).

These scenarios are not yet supported by specialized tools, and can be found in ad-hoc activities such as creating generic requirements for a family of systems, for later specialization or identifying commonalities. Nevertheless, the scenarios provide a base for defining the general process for developing families of systems.

We are now in the process of further refining our collection of requirements for variability management, which will serve as a validation reference in a second case study. We expect that we will soon be able to define a proper variability formalism based on the models and concepts retained from the literature review; we will evaluate this formalism based on the scenarios defined in the first phase. The use of this formalism will be supported by a collection of transformation rules and proper tools to deal with multiple views in a complex cooperative environment.

References

Astesana, J.-M., and A. Dauron. 2010. "Spécification et configuration de la ligne de produits véhicule de Renault." Keynote speech presented at Journée Ligne de Produits, Université Paris 1–Panthéon-Sorbonne, Paris, France, 20 October 2010.

Kang, K. C., S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh. 1998. "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures." *Annals of Software Engineering* 5 (1): 143–168.

Pohl, K., G. Böckle, and F. J. van der Linden. 2005. *Software Product Line Engineering: Foundations, Principles and Techniques*. New York, NY (US): Springer.

Salinesi, C., R. Mazo, D. Diaz, and O. Djebbi. 2010. "Using Integer Constraint Solving in Reuse Based Requirements Engineering." Paper presented at the Eighteenth IEEE International Requirements Engineering Conference.

The Decomposition of a Process and the Definition of Interoperability Metrics

Mamadou Camara, mamadou.camara@ims-bordeaux.fr; **Yves Ducq**, yves.ducq@ims-bordeaux.fr; and **Remy Dupas**, remy.dupas@ims-bordeaux.fr

Our research proposes an approach to graphically represent and measure the interoperability in an enterprise or a supply chain so that it can be improved. This approach is based on two simple principles. The first principle is that we must distinguish between business activities and non–value-added activities. The second principle is that process-performance indicators can be used as interoperability measures.

If one can measure interoperability as this approach allows, then one can define a methodology that integrates several techniques and tools for analysis and evaluation, drawn from the business-process engineering or enterprise-engineering domains.

We begin with the first principle, distinguishing between business and non–value-added (NVA) activities. Business activities create value in a business process. IEEE (1990) defines interoperability as the ability of two or more systems or components to exchange information and use the exchanged information without special effort from either system. The definition was chosen among many possible definitions because it suggests that the efforts of the systems to exchange information should be reduced and then eliminated to achieve interoperability.

To reduce and eliminate the efforts, however, one must be able to represent them first. Non–value-added activities are the parts of business processes that represent the efforts between partners to achieve the interoperability of information exchanges. Business process engineers proceed in two steps: in the first step they identify each activity as being either an NVA or business activity, and in a second step, they check in order to detect some activities that have been identified as business but are in fact subprocesses. It is necessary to break these activities up into both business and NVA categories.

Figure 1 represents a simple sequential business process made up of three activities. In the first step, the activities labeled A1 and A3 are identified as a full business activities and the activity labeled A2 is identified as a full NVA activity. In the second step, the activity labeled A3 is broken up in two parts labeled A31 (business activity) and A32 (NVA activity).

Next we come to the second principle, that process performance indicators (PIs) can be used as interoperability measures. Individual PIs are associated to each activity (business or NVA) generated by the identification and decomposition just described.

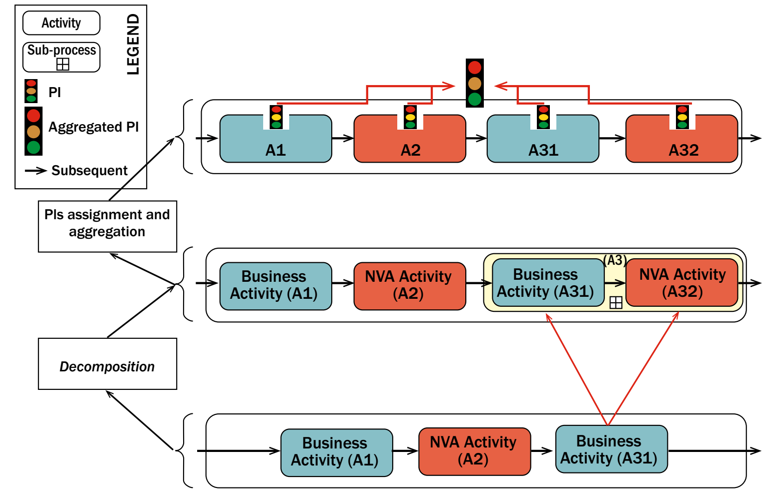


Figure 1. Representing and measuring interoperability

For each business or NVA activity within the process, the following PIs can be calculated: average elapsed time, average cost, and percentage of failure (Yaxiong, Zhen, and Huibin 2010). The process PIs are average elapsed time, average cost, and percentage of failure at the process level. These are aggregations of the PIs for the NVA and business activities.

When systems pairs need translation to interoperate, Ford, Colombi, Graham, and Jacques (2007) consider these instances as NVA and define a function that penalizes their measure of interoperability in these instances. Our measurement of interoperability uses the same logic in that, given two versions of the same process, each model is penalized by the part of its performance that comes from the NVA activity it contains. As proposed by Chen, Vallespir, and Daclin (2008) and by Lebreton and Legner (2007), our measurement of interoperability is realized exclusively through the evaluation of cost, time, and quality (that is, percentage of failure). This choice relies on the assumption that these characteristics encompass all other types of interoperability efforts in the physical system. This means that several interoperability measures defined in the literature can be converted into process performance indicators.

» continues on next page

Camara et al. *continued*

If one can measure interoperability as this approach allows, then one can define a methodology that integrates several techniques and tools for analysis and evaluation, drawn from the business-process engineering or enterprise-engineering domains. For example, business-activity monitoring and business-process simulation will be used to measure improvements in interoperability.

References

Blanc, S. 2006. "Contribution à la caractérisation et à l'évaluation de l'interopérabilité pour les entreprises collaboratives." PhD diss., University of Bordeaux, France.

Chen, D., B. Vallespir, and N. Daclin. 2008. "An Approach for Enterprise Interoperability Measurement." Paper presented at the International Workshop on Model Driven Information Systems Engineering: Enterprise, User and System Models, Montpellier, France.

Ford, T., J. Colombi, S. Graham, and D. Jacques. 2007. "The Interoperability Score." Paper presented at the Fifth Conference on Systems Engineering Research, Stevens Institute of Technology, Hoboken, NJ (US).

IEEE. 1990. *Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, NY (US): IEEE.

Lebreton, B., and C. Legner. 2007. "Interoperability Impact Assessment Model: An Overview." In *Enterprise Interoperability II: New Challenges and Approaches*, ed. R. J. Goncalves, J. P. Müller, K. Mertins, and M. Zelm, 725–728. London: Springer.

Yaxiong, T., X. Zhen, and X. Huibin. 2010. "Bpm Exception Monitoring based on Process Knowledge." Paper presented at the 2010 IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 28–30 June.

A Harmonized and Reversible Development Framework for HLA-Based Interoperable Application

Zhiying Tu, zhiying.tu@ims-bordeaux.fr; **Gregory Zacharewicz,** gregory.zacharewicz@ims-bordeaux.fr; **and David Chen,** david.chen@ims-bordeaux.fr

Enterprise collaboration is becoming more and more important because of the globalized economic context. The competitiveness of enterprises depends not only on their internal productivity and performance, but also on their ability to collaborate with others. This necessity leads to the development of interoperability, which makes it possible to improve collaborations between enterprises. Therefore, more and more networked enterprises are being developed. Further, enterprise interoperability is one of the most suitable solutions to total enterprise integration.

In the last decades, a great deal of research has focused on this problem, including the use of high-level architecture (HLA) to solve some interoperability problems. HLA is a software-architecture specification that defines how to create a global software execution composed of distributed simulations and software applications (IEEE 2000). It has succeeded in many aspects, especially in the areas of reuse and interoperability. However, with the rapid pace of technical change and further development of the IEEE standard, HLA faces many new challenges.

Our research focuses on reducing the time and cost of development, making federation more flexible and open while retaining adequate security and synchronization. Our work aims to contribute to the rapid and intelligent development of distributed enterprise information systems by proposing a harmonized and reversible development framework for HLA-based interoperable application, as figure 1 shows.

High-level architecture has many advantages, such as its generalized development process, distributed simulation engineering and execution process (DSEEP), synchronization standard, runtime infrastructure specification, and data standards. In order to keep these advantages,

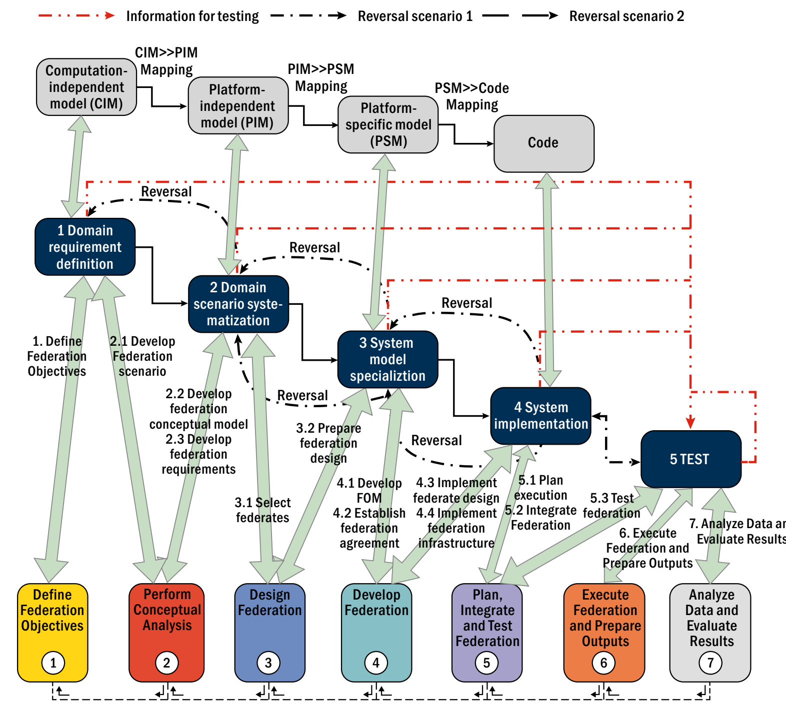


Figure 1. Harmonized and reversible development framework

HLA needs to benefit from developments in the commercial domain (Tolk 2002). Model-driven architecture (MDA) has the most to offer HLA, since MDA is popular, since it is a compatible development lifecycle with HLADSEEP, and since MDA can facilitate the construction of simulators and provide the standardized meta models to this integration. Elvesaeter and others proposed a model-driven interoperability framework in 2007 to provide a foundation (consisting of a set of reference models) for applying model-driven development in software-engineering disciplines that support the business-interoperability needs of an enterprise.

Basically, the harmonization of MDA and HLA is intended to simulate MDA-based systems using modeling and simulation. Therefore we need to consider the existing MDA-based systems.

» *continues on next page*

Zhiying et al. *continued*

As a result, finding a way to rapidly acquire the knowledge from legacy systems becomes the key point of reducing the time and cost of development. Fortunately, after MDA became well known as an important change in software-development practice, the Object Management Group launched another project called architecture-driven modernization. This approach, the opposite of model-driven architecture, aims to "rewind" the models from legacy systems. However, sometimes one would like to discover more specific models from a legacy system. This is why the architecture-driven-modernization group has defined several metamodels to this purpose, the best known being the Knowledge Discovery Metamodel and Abstract Syntax Tree Metamodel (Jouaultet al. 2009).

In order to adapt to the "Web 2.0" context, IEEE was published 1516TM-2010 in August 2010 (IEEE 2010), which benefits from web services such as support for numerous newer and older languages and operating systems as well as the ease of deployment across wide area networks. Because we are dedicated to developing an open framework, we have chosen an open-source RTI (run time infrastructure), poRTIco (Portico Project 2009), which does not provide web-RTI functionality. Thus, we implement a special federate, WebservicesFederate, as a bridge, which takes in charge of providing web services, connecting and synchronizing federates outside traditional federation with federates inside.

References

Elvesaeter, B., A. Hahn, A.-J. Berre, and T. Neple. 2007. "Towards an Interoperability Framework for Model-Driven Development of Software Systems." In *Interoperability of Enterprise Software and Applications*, ed. D. Konstantas, J.-P. Bourrières, M. Léonard, and N. Boudjlida, 409–420. London, UK: Springer.

IEEE. 2000. IEEE 1516.2. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Federate Interface Specification*. New York, NY (US): IEEE.

———. 2010. IEEE 1516TM. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules*. New York, NY (US): IEEE.

Jouault, F., B. Jean, and B. Mikaël. 2009. "Towards an Advanced Model-Driven Engineering Toolbox." *Innovations in Systems and Software Engineering* 5: 5–12.

Portico Project. 2009. "Developer Documentation." Website of the Portico Project. http://porticoproject.org/index.php?title=Developer_Documentation.

Tolk, A. 2002."Avoiding Another Green Elephant: A Proposal for the Next Generation HLABased on the Model Driven Architecture." Paper presented at the 2002 Fall Simulation Interoperability Workshop, Orlando, FL (US).

Model-Based Systems Engineering with SysML for Reliable Systems Design

Robin Cressent, robin.cressent@ensi-bourges.fr; **Vincent Idasiak**, vincent.idasiak@ensi-bourges.fr; and **Frederic Kratz**, frederic.kratz@ensi-bourges.fr

Our contribution to model-based systems engineering focuses on defining a method to improve the realization of reliability analysis during the systems engineering process and its early design phases.

Model-based systems engineering (MBSE) is becoming the predominant paradigm for systems engineering. Model-based representations of a system permit the engineer to obtain more consistent, reusable, and expressive views, thus helping the management and realization of the system-design process. However, RAMS (reliability, availability, maintainability, and safety) activities are generally forgotten in this engineering field.

Our contribution to MBSE focuses on defining a method to improve the realization of reliability analysis during the systems engineering process and its early design phases. This method, described in several publications, is called MéDISIS (David et al. 2010; Cressent et al. 2010; Cressent et al. 2011).

The MéDISIS framework (figure 1) relies on the use of SysML to model the functional aspect of the system. The framework offers the following process:

- Generation of failure-mode-effect analysis (FMEA) to deduct dysfunctional behaviour and identify impacted requirements
- Construction of a model integrating functional and dysfunctional behaviours with a formal language such as Altarica DataFlow.
- Translation to architecture analysis and design language (AADL) to perform analysis and quantification of the impact of dysfunctional behaviour on requirements and timing constraints.
- Partial generation of a Simulink model to simulate the system and realize fault injection.

Each model generated by a translation process needs to be completed by an expert, but using the MéDISIS framework permits a deductive and iterative approach to ease reliability analysis and enhance the use of the diverse tools and languages for reliability analyses. Each MéDISIS process requires dysfunctional information that is not at first modeled in SysML. Some of that information is deduced from the functional model, but most of it depends on expert feedback and knowledge. To address this issue, the MéDISIS

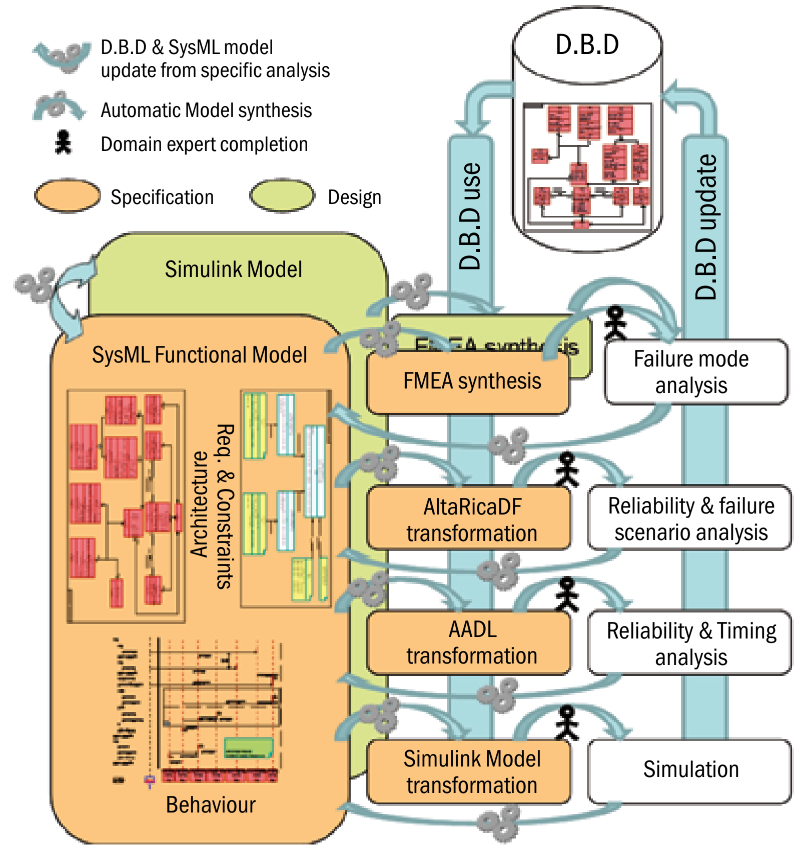


Figure 1. The MéDISIS framework

framework includes the Dysfunctional Behaviour Database (DBD), which is a model-based reliability repository. This database is structured by a metamodel based on SysML metaclasses and is designed to store dysfunctional data such as failure modes, failure laws, and other dysfunctional behaviours. In fact, each process consults the DBD to generate the target model and when the dependability model is completed and the study done, results are collected to update the DBD. This way, every use of a MéDISIS process participates in increasing MéDISIS effectiveness.

MéDISIS benefits at two different levels: the project level and the product level. At the project level, during the planning of a project, MéDISIS makes it possible to connect system engineering activities (such as needs definition, functional analysis, or requirements specifications) with reliability activities (FMEA, scheduling analysis, or fault injection). At the product level, MéDISIS effectively facilitates not only dependability studies but also studies performed after system evolutions (David et al. 2010). Indeed, the DBD tracks knowledge from previous studies to limit the work needed to complete a model generated again.

Our current work applies these methods to the design of the embedded controller of a hypersonic aircraft system called LEA. The LEA project started in 2003 by MBDA-France to test a dual-mode ramjet in the range Mach 4 to 8. Our team is charged with specifying, designing, testing, and validating the embedded system, which must control the flight from the launching of the craft to the final landing.

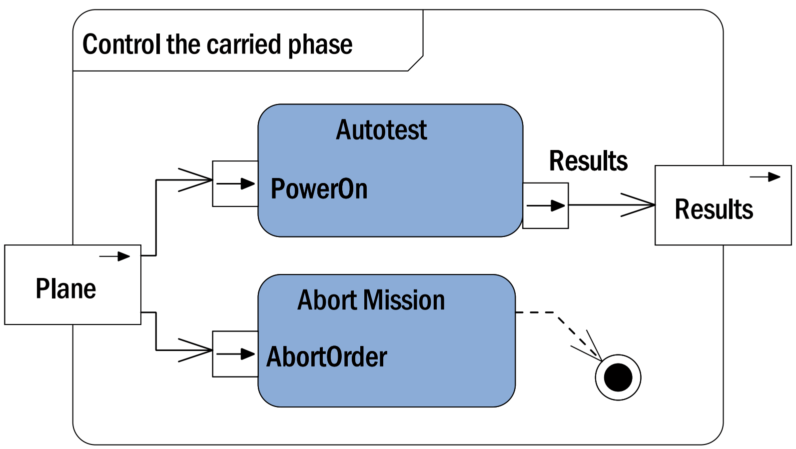


Figure 2. Excerpt of use-case diagram used to define needs

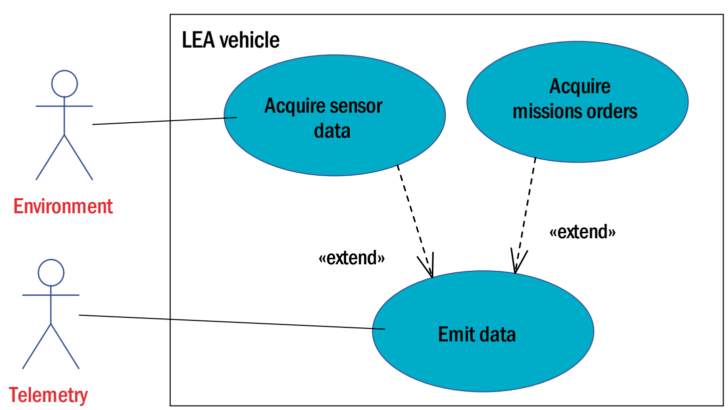


Figure 3. Example of activity diagram used to perform functional analysis

In this project, we first realized the needs definition and the functional analysis with our industrial partners, using SysML's use-case diagrams and activity diagram to support those studies (figures 2 and 3). These first analyses of our vehicle allow the generation of early FMEA based on the functions of the system. Then we model the architecture of our vehicle, supported by SysML's block definition diagram and internal block diagram. Again, we generate the FMEAs to include components. With these two dependability analyses we already obtain a classification of the critical functions and components of our system and it also highlight the risks related to temporal constraints.

To address these constraints we use the AADL translation process and then perform scheduling analysis using an appropriate tool. This step provides success criteria that refine the timing requirements and stores these criteria in the original SysML model using parametric diagrams (Cressent et al. 2010). Finally, by translating to Simulink and completing the resulting model, we enable the simulation of our system and its failure modes (Cressent et al. 2011). This fault-injection phase is driven using the FMEA to list the most critical failure modes to be studied. We also used the success criteria from the scheduling analyses for validation of our Simulink model merging functional and dysfunctional behaviour.

To conclude, MéDISIS was designed to be a tool for system engineers and reliability experts, handling the complex task of making the bridge between their domains efficiently. The LEA project represents our real case study to evaluate the benefits brought by MéDISIS. We manage to maintain a coherent and stable FMEA through multiple specification updates and design evolutions. We also expect some other gains such as an increased efficiency of the exchanges between system engineers and reliability engineers and a work time reduced from a project perspective, for the design and validation steps. We will now work on enhancing MéDISIS for the use of commercial off-the-shelf components in systems with high reliability requirements.

References

Cressent, R., P. David, V. Idasiak, and F. Kratz. 2010. "Increasing Reliability of Embedded Systems in a SysML Centered MBSE Process: Application to the LEA Project." Paper presented at the first M-BED workshop, during the Design, Automation, and Test in Europe conference, DATE 2010, Dresden, Germany, 12 March 2010.

Cressent, R., V. Idasiak, and F. Kratz, F. 2011. "Mastering Safety and Reliability in a Model Based Process." Paper presented at the Fifty-Seventh Annual Reliability and Maintainability Symposium, Orlando, FL (US), 24–27 January 2011.

David, P., V. Idasiak, and F. Kratz. 2010. "Reliability Study of Complex Physical Systems using SysML." *Journal of Reliability Engineering and System Safety* 95 (4): 431–450.

The Distributed and Interoperable Architecture SCEP-SOA as a Way to Manage Multisite Production Projects

Karim Ishak, karim.ishak@enit.fr; **Bernard Archimede**, bernard.archimede@enit.fr; **and Philippe Charbonnaud**, philippe.charbonnaud@enit.fr

Companies today are turning more and more to multisite production to remain competitive. They are centered on their core competencies and, when activities add little value, they share the activities with other companies. Multisite production e-markets are dominated by big companies or original-equipment manufacturers, which create their own e-marketplaces for the realization of their products. In these e-markets, original-equipment manufacturers often require strong constraints that penalize a large number of small and medium enterprises with limited capacities and means (OECD 2007). Dynamic service-oriented markets (Li et al. 2006) increase the presence of smaller enterprises in these e-markets, by reducing the strong requirements of original-equipment manufacturers and putting smaller enterprises in the foreground by displaying these enterprises' skills and know-how on production activities in term of services, thus obliging companies to look for the service providers that will satisfy their needs.

Our research proposes a multisite project-management process in a service-oriented market, based on a generic architectural framework that deals with the various aspects of such a market.

The dynamic aspect of collaborations highlights the necessity of a mechanism for dynamic discovery of partners as well as a multisite project's distribution management, allowing the participation of all partners in a project's management. This management must also take into account the heterogeneity between the various systems and applications used by the partners, which concerns not only the technologies and the functions deployed by the systems, such as technical models, data types, and platform, but also semantic and structural differences existing in the concepts manipulated by these systems.

Our research proposes a multisite project-management process in a service-oriented market, based on a generic architectural framework that deals with the various aspects of such a market (i.e., distributed management, dynamic discovery, and technical and semantic heterogeneities). The SCEP (supervisor, customer, environment, producer) model makes it possible to deal with the distributed management aspect. This model allows agents to cooperate via a shared environment between customer agents. This environment represents the projects and ambassador agents. These

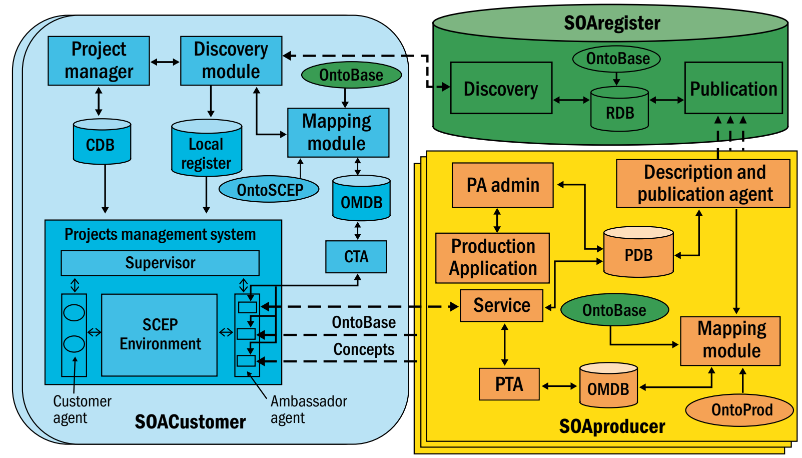


Figure 1. SCEP-SOA for multisite projects' management process in a service-oriented market

agents represent the remote production sites that have homogeneous systems, under the control of a supervisor agent (Archimede, Charbonnaud, and Firmin 2003; Enjalbert, Archimede, and Charbonnaud 2006).

The service-oriented architecture (SOA) model (Nickull 2005) permits dynamic discovery and cooperation that is independent of a platform and a technology. SOA is also used to enhance the communication and cooperation between heterogeneous systems. The architecture SCEP-SOA is proposed to insure the good understanding of the exchanged information and deal with semantic heterogeneity aspect (Aleksovski et al. 2006). An original ontology mechanism was defined and integrated in SCEP-SOA, as depicted in figure 1.

SCEP-SOA is organized around three kinds of actors: (1) SOAregister contains service descriptions, (2) SOAproducer provides the service, and (3) SOAcustomer uses services by invoking them at the corresponding SOAproducers. SOAregister contains a communication interface that allows SOAproducers to declare their know-how in term of SOA services, and allows SOAcustomers to discover the useful services. Producer services implement the production functions assured by the production application, and these functions manage the achievement of provided activities. Service description is stored in the producer database (PDB), which also stores information about SOAcustomers' project description according to the supplied services, and the established projects' management results. The description of services and its publication is realized by a description-and-publication agent, which interacts with SOAregister. The administrator of the Production Application (PA Admin) interacts with PDB to get the projects description received from SOAcustomers in order to process them.

At SOAcustomer, the Customer DataBase (CDB) contains descriptions of the projects defined by the project manager. The discovery module discovers the requested services as well as their SOAproducers. The local register is a limited copy of SOAregister. It stores information about the most frequently invoked services and their SOAproducers with which SOAcustomers had good collaborations. The management of the projects is made on the one hand by cooperation between the SCEP customer and ambassador agents and on the other hand by invoking services at the concerned SOAproducers by the corresponding ambassador agents. OntoBase represents the global and common ontology used as reference ontology to structure information exchanged between production systems and applications. Each SOAproducer (resp. SOAcustomer) has its own ontology denoted OntoProd (resp.OntoSCEP). Mappings between a local ontology and OntoBase are established by the mapping module and stored in the Ontology Mappings DataBase (OMDB). The Producer Translation Agent (PTA) (resp. Customer Translation Agent) uses these mappings during the translation phase.

SCEP-SOA masks the complexity of the networks of applications to be set up to realize the projects. It was implemented in a case study dedicated to the interoperable distributed planning between heterogeneous planning systems. Future works will focus on the other production functions, such as conception and transport, in order to evaluate their interoperability performances.

References

Aleksovski Z., M. Klein, W. Ten Kate, and F. van Harmelen. 2006. "Matching Unstructured Vocabularies using a Background Ontology." Paper presented at the Fifteenth International Conference on Knowledge Engineering and Knowledge Management, Podebrady, Czech Republic.

Archimede, B., P. Charbonnaud, and C. Firmin, 2003. "A Supervised Multisite Reactive Production Activity Control Method for Extended Enterprise." *Journal of Decision Systems* 12 (3–4): 309–328.

Li, Y., X. Lu, K. Chao, Y. M. Huang, and M. Younas. 2006. "The Realization of Service-Oriented e-Marketplaces." *Journal of Information Systems Frontiers* 8: 307–319.

Nickull, D. 2005. "Service Oriented Architecture (SOA) and Specialized Messaging Patterns." Technical white paper, Adobe System Incorporated. http://www.adobe.com/government/pdfs/SOA-technical-whitepaper.pdf.

OECD Group. 2007. "Enhancing the Role of SMEs in Global Value Chains." OECD background report, OECD global conference, Tokyo, Japan.

Enjalbert, S., B. Archimede, and P. Charbonnaud. 2006. "Feasibility Evaluation of Multisite Scheduling by Distributed Simulation of Workshops." Paper presented at the Twelfth IFAC INCOM Symposium, Saint Etienne, France.

Abstract Constraints: A General Framework for Solver-Independent Reasoning on Product-Line Models

Raul Mazo, raulmazo@gmail.com; Camille Salinesi, camille.salinesi@univ-paris1.fr; and Daniel Diaz, daniel.diaz@univ-paris1.fr

Product-line engineering (PLE) is a paradigm for developing reuse-based complex systems that has become well established in industry. Among the proven benefits are reduced time to market, better asset reuse, and improved software quality (Clements and Northrop 2001). To be successful, product-line engineering must efficiently manage the variability—the capacity of a product line's artifacts to vary—that is present in the products that form a product line. Several modeling approaches have been proposed to represent the correct combinations of artifacts of a product line, their properties, and their relationships in a single product-line model (Salinesi et al. 2010).

In this context, being able to analyze and verify the product-line model (PLM) is an important success factor in the strategy of product-line engineering. Reasoning about product-line models is achieved by querying the models in order to verify, analyze, or configure them (Salinesi, Mazo, Djebbi, et al. 2011). For instance, these models can be verified to guarantee that they do not have undesirable properties that affect the correctness of the products they help develop.

Several approaches are available in the literature to support automatic reasoning about product-line models (Mazo, Salinesi, Diaz, and Lora 2011). Some of these consist of transforming the models into a constraint program that can be executed by a solver. For example, satisfiability (SAT) solvers are used to analyze product-line models that are specified as Boolean constraints (Mendonça et al. 2009). Others use SAT or constraint-over-finite-domains solvers to find the number of solutions that can be configured on a product-line model. Interestingly, though, it seems that for this task binary-decision-diagram (BDD) solvers are more efficient (Mendonça et al. 2009). Some authors therefore seem to undermine the efficiency of certain reasoning operations to prioritize others. One cause for this error might be that the transformation is guided by the choice of solver and not by the nature of the product-line models or the relative efficiency or limitations of using a particular solver.

To overcome these limitations, we propose to represent the semantics of product-line models as abstract constraints. We use a unique notation that encompasses other constraint languages (such as over Booleans, integers, reals, trees, and lists). As figure 1 shows, once a product-line model is specified as abstract constraints, it can be compiled with the platform in any constraint language depending on the desired analysis and the preferred solver.

Constraint System to Represent Product-Line Models

In order to implement the framework of figure 1, our first concern is to define a notation that consists of a system of constraints system that represents product lines. As defined by Saraswat (1992), a constraint system can be defined as a tuple $(D, \vdash)$ where D is a set of first-order formulas closed under conjunction and

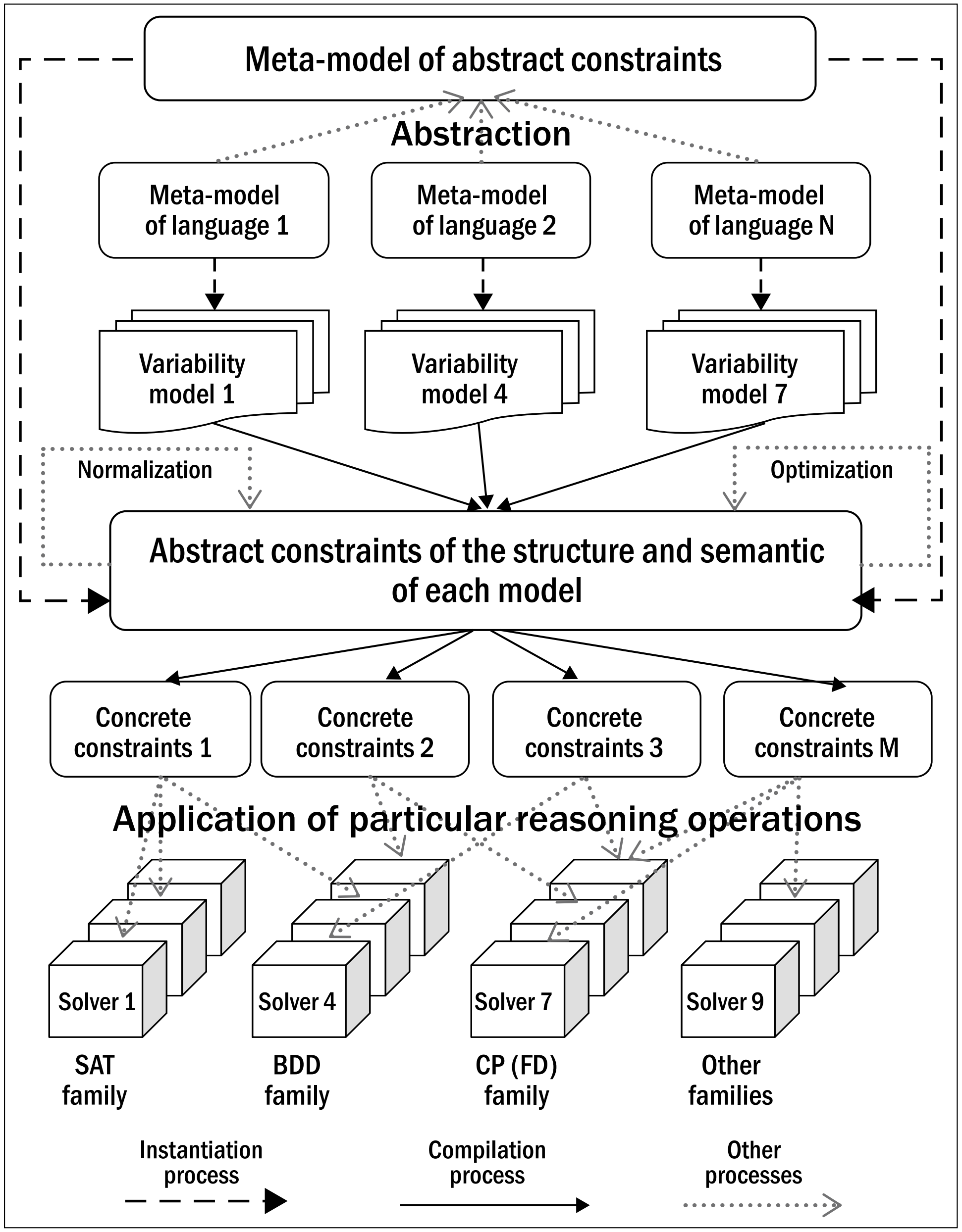


Figure 1. Constraint-based configuration overview

existential quantification, $\vdash$ is an entailment relation between a finite set of formulas (taken from D) and a single formula, and $\vdash$ must be generic (that is: $S[t/X] \vdash d[t/X]$ whenever $S \vdash d$, for any term t).

A constraints system for representing product lines over a domain X that can be parameterized (for example, X = finite domain, X = reals, or X = Booleans) is a tuple of the minimal set of first-order formulas allowed to represent product lines. Traditionally in the product-line literature, the minimal collection of complete variability constraints to represent a product line is {*mandatory, optional, requires, excludes*}. However, others can be added and an entailment relation between these constraints can be defined by rules. We can therefore define a kind of operational semantic of entailment between constraints adapted to the domain of the solver on which the constraints system will be executed. Therefore these rules can be reduced to conjunction operators among the variability constraints on the domain of the product line. Indeed, any product to be configured from the product-line model must satisfy all the constraints of the product line (such as *mandatory, optional, requires, excludes*, and other constraints), which are entailed by means of conjunctions.

The first-order formulas representing the variability constraints of a product line are as follows, where *Variable(x)* means that *x* is a variable in a non-specified domain:

mandatory: $\exists$ a, b (Variable(a), Variable(b) $\wedge$ (a $\Rightarrow$ b) $\Leftrightarrow$ (b $\Rightarrow$ a))
optional: $\exists$ a, b (Variable(a), Variable(b) $\wedge$ (a $\Rightarrow$ b) $\vee$ (b $\Rightarrow$ a))
requires: $\exists$ a, b (Variable(a), Variable(b) $\wedge$ (a $\Rightarrow$ b))
excludes: $\exists$ a, b (Variable(a), Variable(b) $\wedge$ (a $\oplus$ b))

Now, our next issue is to identify a proper form for the components that allows us to transform constraints that were specified with the generic notation into some kind of constraints in a particular domain, and the other way round. In order to achieve this, we are developing a series of transducers. The difficulty in developing the transducers is that they must be monotonic and continuous in the transformation, because one defined solver will execute each transformed model at a time. Because of the first-order structure of the constraints, the transducers should be generic in all the variables. To be generic in a variable V means that if the transducer can produce the information d on input c, then it can also produce the information $d[t/V]$ for input $c[t/V]$ for any t.

In the context of product-line models, the design of these transducers depends on the target back-end solvers that will be used to achieve the analyses. The details are not provided in this paper for the sake of space, but examples are given in two 2011 articles by Mazo and others cited in the reference list.

Mazo et al. *continued*

References

Clements, P., and L. Northrop. 2002. *Software Product Lines: Practices and Patterns*. Boston, MA (US): Addison-Wesley.

Mazo, R., C. Salinesi, D. Diaz, and A. Lora. 2011. "Transforming Attribute and Clone-Enabled Feature Models into Constraint Programs over Finite Domains." Paper presented at the Novel Approaches to Software Engineering (ENASE) conference, Beijing, China, 8–11 June.

Mendonça, M., A. Wasowski, and K. Czarnecki. 2009. "SAT-Based Analysis of Feature Models Is Easy." Paper presented at the Software Product Line Conference (SPLC), San Francisco, CA (US), 24–28 August.

Salinesi, C., R. Mazo, D. Diaz, and O. Djebbi. 2010. "Solving Integer Constraint in Reuse Based Requirements Engineering." Paper presented at the Requirements Engineering Conference, Sydney, Australia, 26–30 September.

Salinesi, C., R. Mazo, O. Djebbi, D. Diaz, and A. Lora. 2011. "Constraints: The Core of Product Line Engineering." Paper presented at the Fifth IEEE International Conference on Research Challenges in Information Science, Guadeloupe, France, 19–21 May.

Saraswat, V. 1992. "The Category of Constraint Systems is Cartesian-Closed." Paper presented at the Seventh Annual IEEE Symposium on Logic in Computer Science, Santa Cruz, CA (US), 22–25 June.

Using Object-Oriented Bayesian Networks to Model an Industrial System: A New Approach to Assessing Maintenance Strategies

G. Medina-Oliva, gabriela.medina-oliva@cran.uhp-nancy.fr; **P. Weber**, philippe.weber@esstin.uhp-nancy.fr; **E. Levrat**, eric.levrat@cran.uhp-nancy.fr; and **B. Iung**, benoit.iung@cran.uhp-nancy.fr

In keeping with the standard ISO/IEC 15288 (ISO and IEC 2002), the implemented interactions between a maintenance system and a system of interest have to be built on the definition of the bilateral requirements and constraints between the engineering cycle of the system of interest and its enabling systems. For example, some requirements of the system of interest (such as availability) induce requirements on the maintenance system (such as mean time between failures to warranty the availability rate). In the same way, the maintenance system's requirements constrain the requirements on the system of interest. For example, equipment reliability will impact the quality of a product.

In a recursive way, these interactions become the objectives to be satisfied for the studied system at a lower abstraction level. Indeed the maintenance system itself has a set of constraints to be controlled in relation to its enabling systems such as the logistic system or human-resource system in order to achieve the expected objectives. According to Parida (2006), these requirements and constraints for the maintenance system could be represented as multicriteria key performance indicators.

Within this context, we propose a methodology wherein we compute a set of key performance indicators and use them to assess whether the functional architecture of the maintenance system and its associated strategies—defined mainly through the maintenance plan and its organization—satisfy the objectives of the system of interest and its enabling requirements. If the objectives are not achieved, one must identify the causes with the most impact on the deviations from requirements (that is, a diagnostic process). In that way, in order to help the decision-making process for maintenance managers, we have developed a "unified" generic model. This model integrates (a) the interactions of the maintenance system with its enabling systems, (b) the impact of the maintenance strategies through the computation of some key performance indicators, and (c) different kinds of knowledge regarding the maintenance system and system of interest, including quantitative and qualitative knowledge.

According to the justifications given by Weber and others (2010) and by Langseth and Portinale (2007), we selected Bayesian networks for developing the unified model. A structured representation, supported by an extension of Bayesian networks (that is, a probabilistic relational model) allows a modular representation that eases the modeling and improvement of the model while enabling its inference.

For knowledge formalization of both the system of interest and maintenance systems, the proposed modeling approach consists, from functional systemic analysis in representing some complementary points of views such as abnormal operation or malfunctioning (Léger et al. 1999), and informational and organizational points of view (figure 1A). Next, it is necessary to use this formalized knowledge to establish a coding semantic with the required knowledge and programming rules (figure 1B) to obtain a generic maintenance model than can allow one to assess performances (figure 1C).

To do the functional modeling of both system of interest and maintenance system, one uses qualitative causal relationships (flows) according to the structured analysis and design technique method to formalize the interactions between the functions performed by each of the subsystems (as opposed to their activities) down to the component level of elementary functions.

The modeling objectives for a dysfunctional system are to identify the degraded and failure states of the components and of the flows, and to determine the causes and consequences of these states on the system of interest's behavior through failure modes and effects analysis and through hazard and operability study. This view requires one to identify the logical links between essential components in order to perform the system mission supported by fault trees or reliability block diagram. The informational point of view takes into account the flow attributes or properties within a class diagram. In this way, for measuring the performance of

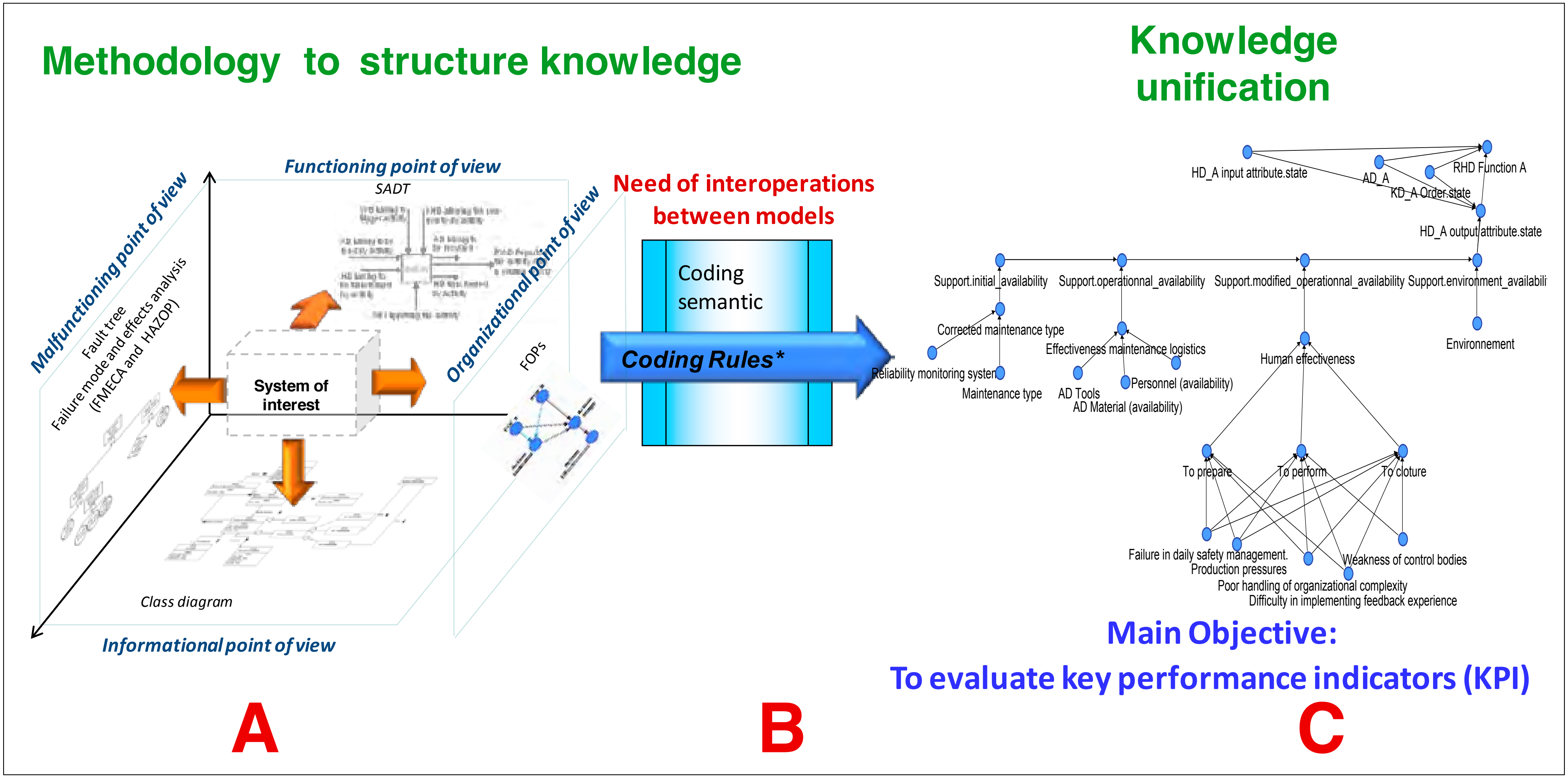


Figure 1. Methodology to formalize knowledge within a probabilistic relational model

a function, we assume that performances can be evaluated directly from the flow attributes. Finally, the organizational and human points of view focus on operators' performances and their context. These operators are influenced primarily by their organizational context. In that sense, the organizational point of view is built based on a study of pathogen factors (such as production pressure) that impact operators' performance, as modeled by Leger (2010).

Afterwards, one must establish "interoperation" between the different models. We have done this by transforming all the model knowledge into a unified probabilistic relational model (PRM). In this model, variables represent the nodes, and causality relationships or dependences are defined by the relationships described in the informational and functional modeling. We based our modeling on PRM patterns while taking into account flow exchanges between functions and using specific coding rules (figure 1B).

The scientific contribution of our work, then, is that this approach makes it possible to create modular and generic patterns for a PRM on relevant aspects of the system of interest, the maintenance system with its main enabling systems (include maintenance logistic, maintenance organization, and others), in order to assess the impact of maintenance actions on the performance of a component as well as the system of interest. For example, in figure 1C, the first impact of the node "component initial availability" is impacted by the maintenance intervention plan.

Medina-Oliva *continued*

References

ISO and IEC (International Organization for Standardisation and International Electrotechnical Commission). 2002. ISO/IEC 15288. *System Engineering: System Life Cycle Processes*. Geneva, Switzerland: ISO and IEC.

Langseth, H., and L. Portinale. 2007. "Bayesian Networks in Reliability." *Reliability Engineering & System Safety* 92 (1): 92–108.

Léger, A., P. Weber, E. Levrat, C. Duval, R. Farret, and B. Iung. 2009. "Methodological Developments for Probabilistic Risk Analyses of Socio-Technical Systems." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 223 (4): 313–332.

Léger J.-B., B. Iung, A. Ferro de Beca, and J. Pinoteau. 1999. "An Innovative Approach for New Distributed Maintenance System: Application to Hydro Power Plants of the REMAFEX Project." *Computers in Industry* 38 (2): 133–150.

Parida, A. 2006. "Development of a Multi-Criteria Hierarchical Framework for Maintenance Performance Measurement." Doctoral thesis, Luleå University of Technology, Luleå, Sweden.

Weber, P., G. Medina Oliva, C. Simon, and B. Iung. 2010. "Overview on Bayesian Networks Applications for Dependability, Risk Analysis and Maintenance Areas." *Engineering Applications of Artificial Intelligence* (forthcoming); published electronically 16 July. doi:10.1016/j.engappai.2010.06.002. http://hal.archives-ouvertes.fr/docs/00/40/20/92/PDF/DCDS2009_Medina_Weber_Simon_Iung_Review_HAL.pdf

A Design Pattern Metamodel and Use Mechanisms for Systems Engineering

Francois Pfister, francois.pfister@mines-ales.fr; **Vincent Chapurlat,** vincent.chapurlat@mines-ales.fr; **Marianne Huchard,** huchard@lirmm.fr; **Clémentine Nebut,** clementine.nebut@lirmm.fr; **and Jean-Luc Wippler,** jean-luc.wippler@cassidian.com

In facing repetitive classes of problems during their projects, engineers need to combine practitioners' experience in design with solutions that are already capitalized, approved, and standardized. Sharing, interpreting, and applying this experience and these solutions allows engineers to improve their performance (including comprehensiveness and relevance) and their reliability (since they are using proven solutions that have been justified and argued contextually), while also raising their economic value (through time savings). In this way enterprises can capitalize on their engineers' experience.

The idea of using such "design patterns" is intended to help an engineer improve the nonfunctional features, quality of service, and "ilities" (Manola 1999) of a system under design. A design pattern is a simple and small artifact, linking a model of a problem noticed in a given context with a model of a well-known solution that has been already used to solve the problem in another but quite similar context on which the interest of the solution has been validated. This solution must then be imitated and adapted to another context. This approach is currently used in several engineering fields, such as traditional architecture (Alexander et al. 1977), software engineering (Gamma et al. 1994; Coplien and Schmidt 1995; Harrison 1999; Fowler 1996), and process management (Appleton 1997; Van der Aalst et al. 2003). More recently it has been applied to systems engineering (Barter 1998; Haskins 2003, 2005; Schindel and Rogers 2005; Cloutier 2006; Cloutier and Verma 2007). Despite this literature, however, the design-pattern concept remains poorly formalized for the systems engineering domain. Our research promotes a formalized metamodel for design patterns.

A System Pattern Metamodel

The proposed metamodel is a domain-specific language defined in *ecore* format shown in figure 1 and built taking into account a global systems engineering metamodel. So, the whole metamodel includes all entities needed to support a systems engineering process, as specified in the ISO 15288 standard.

The main concepts of the metamodel concern system-of-interest models and a catalog of existing design patterns called system patterns. A system pattern is designed as a parameterized functional microarchitecture; in other words, a function graph in which some elements play given roles (pattern roles) linked by a parameter metaclass to roles (concrete roles) played by elements belonging to the model under study.

A system pattern identifies and describes a solution that addresses a problem in a given context. More precisely, a system pattern is characterized by a unique identifier, a short but evocative name, alternative aliases, a creation date, a textual description, and an author.

In the terms of the system pattern, a Problem describes the specified design problem that is motivating the system pattern. It is characterized by an informal description, a Feature to optimize, a set of competing Forces, and a use-case Model showing an elementary functional or organic architecture. A Force is a competing constraint which, when put in conflict with another constraint, is the cause from which the Problem arises. So the decision to apply a system-pattern depends from arbitration between the Forces. A Force is described by a Challenge, a Constraint and a Problem Type (which might include Fluid, Field, Structure, Security, or others).

A Feature is an extra functional characteristic identified as an "ility." A Solution holds a Pattern Model, which is parameterized system architecture. It configures a design solution as a response to a Problem considering the given Context. There is only one solution for one pattern, but one Problem may have many solutions through several patterns by using equivalent patterns or related patterns.

A Solution is illustrated by a use case showing a more relevant architecture and an Impact quantified by a Variation Sense (increase, decrease, equals) and a value on a scale. Impact is measured on a feature and allows one to quantify the influence of

Pfister et al. *continued*

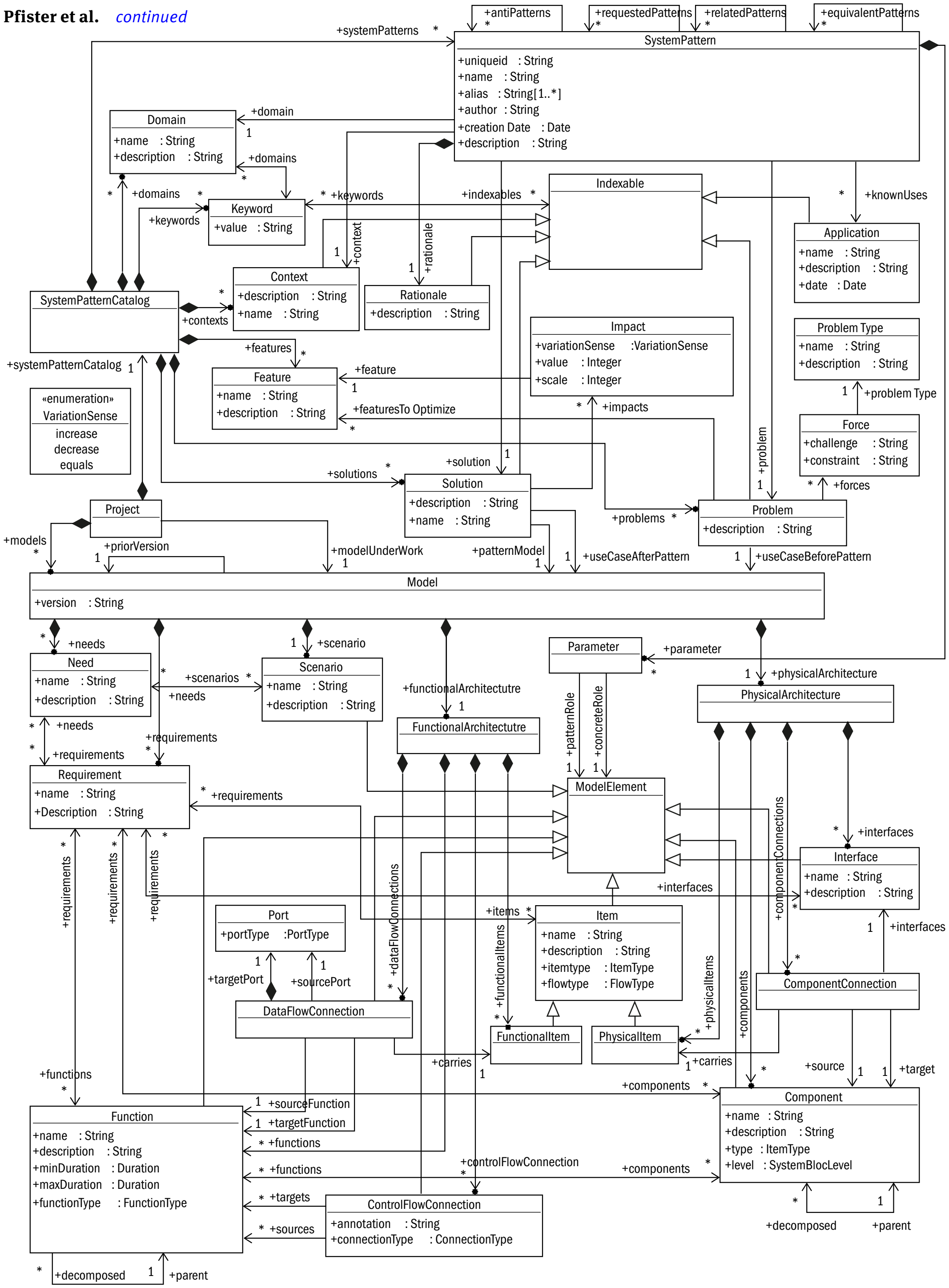


Figure 1. A system-pattern metamodel

a system pattern on a system-of-interest model by detecting the optimized and degraded Feature. The Context is interpreted as a set of preconditions that define in what cases and in which conditions the System Pattern may be applied.

The Rationale justifies the system pattern by an explicit description and the associated argumentation; thus the rationale allows one to determine whether or not the system-pattern can be applied. This approach is different from a statistical observation which inventories known uses in several Applications. Last Problems, Solutions, Contexts, Applications, and Rationales are indexable objects, described by Keywords.

A system pattern has a Domain that identifies a specific area in which a system pattern can be applied or is relevant, such as mechanics, electronics, software, civil engineering, organization and service, security, or pedagogy. System patterns are related to each other by several relationships: requested, related, and equivalent patterns, and antipatterns. Each system pattern references well-known cases (known uses).

Conclusion and Outlook

The proposed metamodel contributes to INCOSE's Model-Based Systems Engineering initiative (Estefan 2008) by describing the required language that allows one to implement a catalog of systems engineering design patterns. This metamodel is currently under validation. The goal is now to provide mining techniques and models of alignment mechanisms to identify applicable design patterns in a given context. We have designed software to support this metamodel. The editor has to be fully interoperable with the main tools used by system architects. Mining and alignment mechanisms (based on model transformations) are under development.

References

Alexander, C., S. Ishikawa, and M. Silverstein. 1977. *A Pattern Language: Towns, Buildings, Construction*. New York, NY (US): Oxford University Press.

Appleton, B. 1997. "Patterns for Conducting Process Improvement." Paper presented at the fourth annual conference of Pattern Languages of Program Design, Monticello, IL (US), 3–5 September.

Barter, R. H. 1998. "A Systems Engineering Pattern Language." Paper presented at the Eighth Annual International Symposium of INCOSE, Vancouver, BC (Canada), 26–30 July.

Cloutier, R. J. 2006. "Applicability of Patterns to Architecting Complex Systems." PhD diss., Stevens Institute of Technology, Hoboken, NJ (US).

Cloutier, R. J., and D. Verma. 2007. "Applying the Concepts of Patterns to Systems Architecture." *Systems Engineering* 10 (2): 138–154.

Coplien, J. O., and D. C. Schmidt. 1995. *Pattern Languages of Program Design*. Reading, MA (US): Addison-Wesley.

Pfister et al. *continued*

Estefan, J. A. 2008. "Survey of Model-Based Systems Engineering (MBSE) Methodologies." Revision B. Paper for INCOSE MBSE Initiative. Available online at http://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf.

Fowler, M. 1996. *Analysis Patterns: Reusable Object Models*. Menlo Park, CA (US): Addison-Wesley.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA (US): Addison-Wesley.

Harrison, N. B. 1999. "The Language of Shepherding: A Pattern Language for Shepherds and Sheep." Paper presented at the seventh conference of Pattern Languages of Program Design, Monticello, IL (US), 13–16 August.

Haskins, C. 2005. "Application of Patterns and Pattern Languages to Systems Engineering." Paper presented at the Fifteenth Annual International Symposium of INCOSE, Rochester, NY (US), 10–15 July.

Haskins, C. 2003. "Using Patterns to Share Best Results: A Proposal to Codify the SEBOK." Paper presented at the Thirteenth Annual International Symposium of INCOSE, Crystal City, VA (US), 29 June–3 July.

Manola, F. 1999. "Providing Systemic Properties (Ilities) and Quality of Service in Component-Based Systems." Draft. Object Services and Consulting, Inc. http://www.objs.com/aits/9901-iquos.html.

Schindel, W. D., and G. M. Rogers. 2000. "Methodologies and Tools for Continuous Improvement of Systems." *Journal of Universal Computer Science* 6 (3): 289–323.

Van der Aalst, W. M. P., et al. 2003. "Workflow Patterns." *Distributed and Parallel Databases* 14 (3): 5–51.

A Model-Based Platform for Product Knowledge Preservation

Fei Teng, fei.teng@univ-lyon2.fr; **Néjib Moalla**, nejib.moalla@univ-lyon2.fr; and **Abdelaziz Bouras**, abdelaziz.bouras@univ-lyon2.fr

Digital information plays a significant role in manufacturing industry when digital preservation has become one main objective of the construction and the maintenance of a production-information system. However, regarding product-lifecycle management, product lifecycles are often far longer than the expected lifetime of a manufacturing software application used to interpret the data: an aircraft might last fifty years but software rarely lasts three years without the need for upgrades. In spite of the application of traditional document-based engineering methods, the problems of long-term knowledge retention have been mostly neglected in traditional implementations of information lifecycles.

As our research work is about a new architecture for long term knowledge preservation, we propose a methodology for knowledge engineering. The main challenge of designing such a methodology and architecture is how to deal with long-term changes in information systems for product-lifecycle management. In our methodology, we propose to establish a multi-layer knowledge preservation architecture, which has dynamic features that will continually adapt for the length of the product's lifecycle. We propose a model-based architecture in order to enhance the flexibility of the evolution of the target system. We aim to cover long-term issues in knowledge preservation.

From the view point of the operators in enterprise production and business processes, the proposed architecture is implemented as a long-term knowledge preservation platform, which handles knowledge management and knowledge retention. This platform acquires knowledge from product-lifecycle-management information systems, formalizes this knowledge, and preserves it. Besides, the platform would dynamically adapt to organizational and technical changes in the enterprise. Following previous research into long-term knowledge retention (such as the LOTAR: Long Term Archiving and Retrieval of Product Data within the Aerospace Industry project), we choose to use the Open Archival Information System (OAIS) reference model as the fundamental base of our long term preservation platform. The OAIS was developed under the Consultative Committee for Space Data Systems (CCSDS), and this system standardizes metadata schema (that is, the information package model) and the information package's engineering workflow (including information package design, composition, transfer, storage, sharing process, preservation planning, and management process). Besides these characteristics of knowledge archiving, which are provided by OAIS and are functionalities of the Digital Preservation Layer, knowledge workers in the enterprise perform knowledge-management approaches.

Thus, in general, the multilayer architecture proposed in our project consists of three logical layers:

- The enterprise layer performs the knowledge-management approach, in order to collect digital data and knowledge from the information systems or domain experts in product lifecycle management.
- The digital preservation layer keeps collected knowledge and manages the retention of the knowledge objects (digital objects).
- The mediation layer connects the previous two layers and provides communication and knowledge-transfer services.

We need a knowledge-transfer approach between the digital preservation layer and the mediation layer, because in these two layers knowledge appears in different forms, and in the enterprise layer there exist many kinds of forms due to the spatial and temporal variety. Spatial variety means that the knowledge management results would be different in knowledge structure even in the same enterprise due to the diverse objectives in different departments. Temporal variety means the structural changes of the knowledge model in an enterprise due to the long term organizational or technical changes.

To establish the platform for long-term knowledge retention, we have designed several models (figure 1):

- Structural model: The multilayer architecture we have discussed.

- Functional models: Functionalities for achieving the designated missions of each layer. The function models are designed based on the knowledge-management approach and the OAIS reference model, which is an information-package-workflow process model for knowledge archiving and reusing.
- Data models: Knowledge objects' schemas in the knowledge-management approach and a comprehensive information-package model, which is the data object in the workflow of the digital preservation layer.

We utilize business process modeling (BPM) for the purpose of validating the predesigned models (that is, the structural, functional and data models) and implementing the knowledge preservation platform. In the business-process models, the functions that concern dynamic features are defined and deployed, using the SOA principle and components, which collaborates with the BPM approach.

The platform is deployed and operated by a preservation engineer. The dynamic communication functions that the platform configures are the preservation plan, the connections between the digital-preservation layer and the enterprise layer, and the knowledge mapping configurations. The preservation plan is a project plan that aims to instruct the whole preservation process. The preservation plan identifies the format and content of messages and reports, which are used for communications between the platform's functions. Moreover, the preserved knowledge would be checked periodically in case there may be redundancies in knowledge storage, which would prevent or lower the efficiency of reusing knowledge. These connections between layers should not only be technical connections between systems, but also collaborations between the operators of digital preservation and knowledge domain experts in the enterprise. The knowledge mapping would be realized by using an enterprise service bus or mediators. In this way, the functionalities in different layers (i.e., the enterprise layer and the digital preservation layer) handle knowledge in simplex form. The spatial or temporal changes in the enterprise layer will have the least effect on the knowledge repositories in the digital-preservation layer. Therefore knowledge would be maintained well in the long-term. Besides the functions that we have illustrated above, the knowledge-preservation platform should also provide other basic functions such as authorization, versioning, hierarchical storage, and data searching.

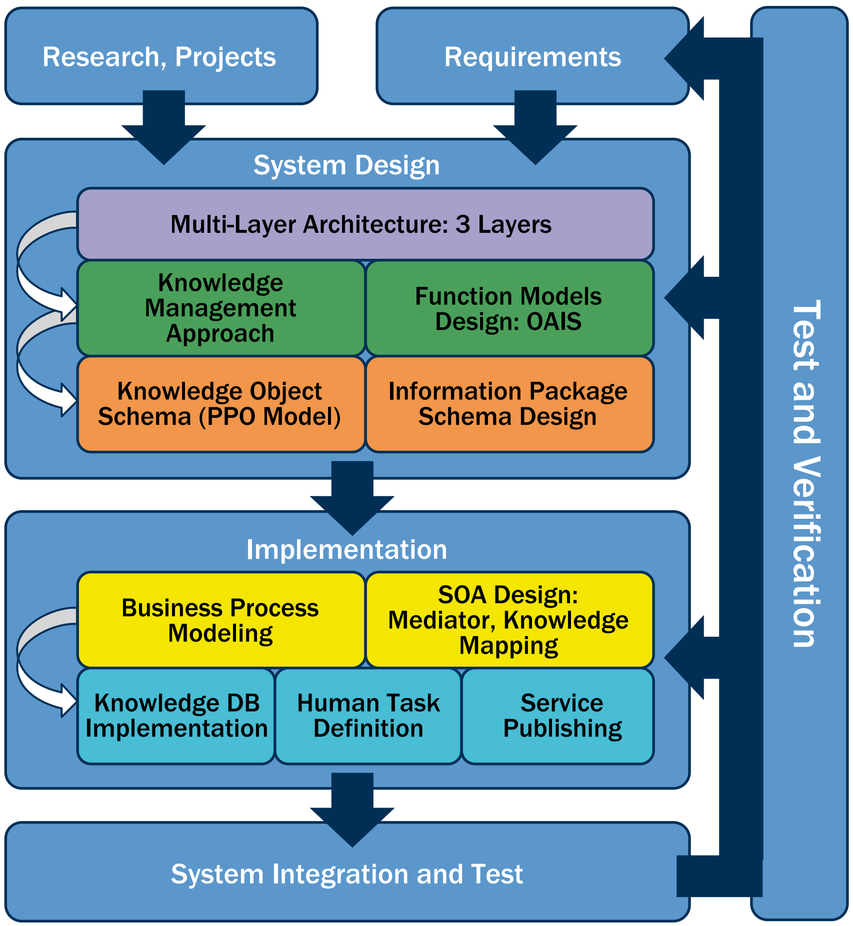


Figure 1. Model-based design process and designated models for long-term knowledge-retention platform

References

Ahmadian, M., Z. Nazari, N. Nakhaee, and Z. Kostic. 2005. "Model-Based Design and SDR." Paper presented at the second IEEE/EURASIP Conference on DSP Enabled Radio, Southampton, UK, 19–20 September.

CCSDS (Consultative Committee for Space Data Systems) Secretariat. 2002. CCSDS 650.0-B-1. *Reference Model for an Open Archival Information System (OAIS).* Blue Book (i.e., "CCSDS recommended standard"). Issue 1. Washington, DC (US).

Glandney, H. M. 2007. *Preserving Digital Information*. Berlin: Springer.

Lubell, J., B. Kassel, and S. Rachuri, 2009. "Descriptive Meta-Data Requirements for Long-Term Archival of Digital Product Models." Paper presented at the Indo-US Workshop on International Trends in Digital Preservation, Pune, India, 24–25 March.

Timsina, P., F. Teng, N. Moalla, and A. Bouras. 2010. "Classification Framework for Digital Preservation Platforms: An Evaluation Approach for Digital Preservation Platforms in Long Term Preservation Scope." Paper presented at the International Conference on Software, Knowledge, Information Management and Applications, Paro, Bhutan, 25–27 August.

Process Modeling in Innovative Design using Systems Engineering

Qiang Zhang, qiang.zhang@insa-strasbourg.fr; **Deniaud Ioana**, deniaud@unistra.fr; **Emmanuel Caillaud**, emmanuel.caillaud@unistra.fr; and **Claude Baron**, claude.baron@insa-toulouse.fr

The emergence of ever more technically advanced customer needs, combined with exponentially growing competition on a global range, is fueling the demand for more innovative products. Innovative design, however, involves many variables whose characteristics and interactions are not well understood (Pavitt 2006). Meanwhile, the complexity of requirements, design activity, and organization are increasing the difficulty of controlling the process (Suh 2005). Systems engineering, of course, has much wider concerns than addressing the product system, and also encompasses social interaction and organizational systems.

Is it possible, then, for companies to utilize system engineering to control design activities while also encouraging innovative activities at the same time? How does innovation actually develop from concept to implemented detail design? Which factors of systems engineering can have positive or negative impact for innovative design? These questions guide our research. In this brief summary of our research, we will first discuss the characteristics of innovative design needed to build a descriptive model, and propose the relevant definitions. We then analyze the factors that directly affect innovation through systems engineering. Finally we propose a prescriptive model of innovative design that can balance between control and innovation.

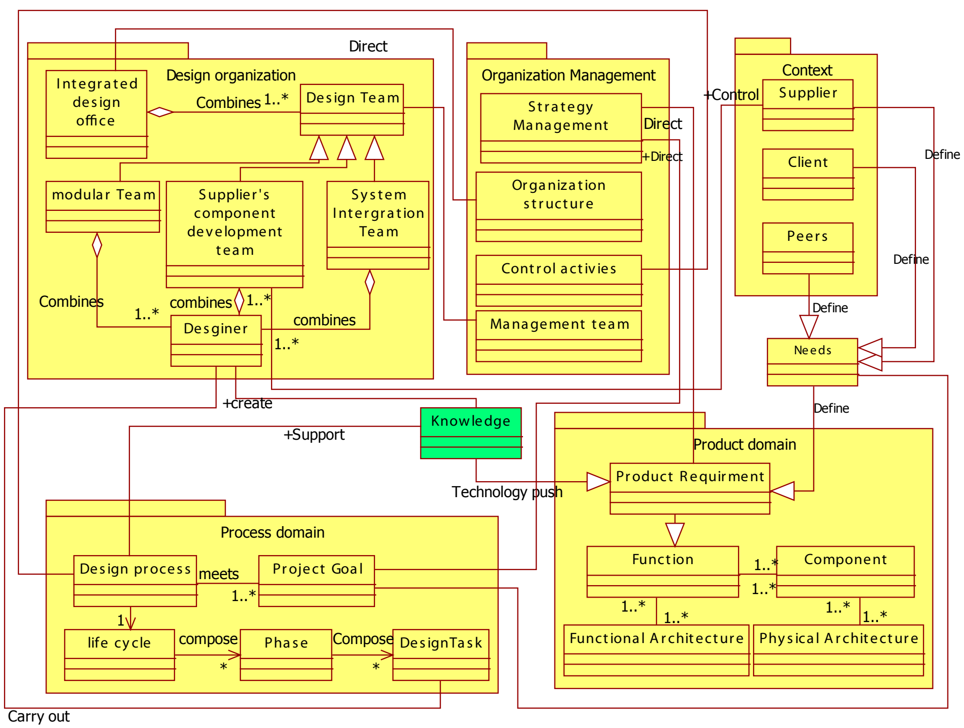


Figure 1. Conceptual model of innovative design

What Is Innovative Design?

Defining innovative design necessarily requires one to understand the concepts of innovation and design. Design has not been adequately defined: researchers agree that it is a process, but disagree on what kind of process it is. Some have considered it as a rational problem-solving process (Simon 1969), others as a reflective process (Schön 1983), and still others as an evolving process between knowledge and concept (Hatchuel and Weil 2003). There is more agreement about innovation, namely, that it consists of two components: creativity and implementation (Stamm 2008). Creativity refers to how to generate more and better ideas; implementation concerns how to translate the ideas into products.

We can define innovative design, then, as some kind of process that applies a creative idea to create a product, process, or service for a customer and market. An innovative design should break away existing forms, and demonstrate three characteristics:

1. *Novelty*. The result of innovative design is different from all previously existing products.
2. *Value*. The value of innovative design is related to a human purpose, and should be judged by the customer and society.
3. *Commercialization*. Innovative design is distinguished from the term *creative design* because it involves commercial transactions.

Conceptual Model of Innovative Design

Different dimensions, such as organization, design process, and product and market context, influence the creative ability of innovative designers (Galanakis 2006). Based on Bonjour and Micaëlli's (2010) core competence framework for design, we propose a conceptual model to describe the links between key dimensions involved in innovative design (figure 1).

In figure 1, knowledge plays a key role as the transformer that connects with other dimensions. The product should satisfy the goal of innovative design—creatively meeting customer demands and greatly improving the ability of innovation—and the innovative strategy of a company through existing and outside knowledge. The innovative design process provides the solution step by step with an iterative model, which improves efficiency and innovation. Design organizations carry out the design process and design tasks. The culture of an organization (such as shared values) and its structure (particularly its degree of flexibility and freedom) will influence the whole process of innovative design. The supplier, client, and peers in the market are not only the sources of innovation, but also the measure criteria of innovative design.

Dynamic Management Model for Innovative Design

These multiple requirements of innovative design often lead to contradictions in different steps of an innovative design project. Compromise and contradictions overcome are thus strategic choices for an innovative design project (Deniaud et al. 2011). Therefore, to manage the requirements and the different contradictions, we propose a model in unified modeling language, based on the V-cycle (figure 2), which concerns arbitration through an existing knowledge base.

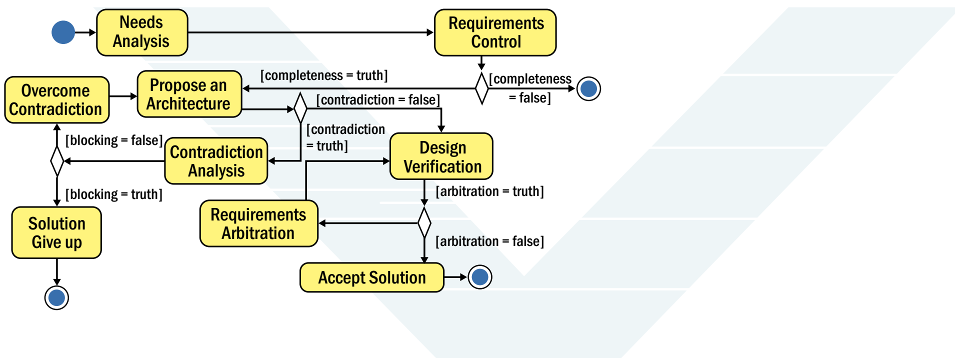


Figure 2. Dynamic management model

References

Bonjour, E., and J.-P. Micaelli. 2010. "Design Core Competence Diagnosis: A Case From the Automotive Industry." *IEEE Transactions on Engineering Management* 57 (2): 323–337.

Deniaud, I., C. Lerch, and E. Caillaud. 2011. "Eco-Design of Complex Products using System Engineering." Paper presented at the Fourth International Conference on Industrial Engineering and Systems Management: Innovative Approaches and Technologies for Network Manufacturing Enterprise Management, Metz, France, 25–27 May.

Galanakis, K. 2006. "Innovation Process. Make Sense using Systems Thinking." *Technovation* 26 (11): 1222–1232.

Hatchuel, A., and B. Weil. 2003. "A New Approach of Innovative Design: An Introduction to CK Theory." Paper presented at the 14th International Conference on Engineering Design, Royal Institute of Technology, Stockholm, Sweden, 19–21 August.

Pavitt, K. 2006. "Innovation Processes." In *The Oxford Handbook of Innovation*, ed. J. Fagerberg, D. C. Mowery, and R. R. Nelson. Oxford Handbooks Online. Oxford, UK: Oxford University Press. doi:10.1093/oxfordhb/9780199286805.001.0001. http://www.oxfordhandbooks.com/oso/public/content/oho_business/9780199286805/toc.html

Schön, D. A. 1983. *The Reflective Practitioner*. New York, NY (US): Basic Books.

Simon, H. A. 1969. *The Science of the Artificial*. Cambridge, MA (US): MIT Press.

Stamm, B. von. 2008. *Managing Innovation, Design and Creativity*. Chichester, UK: Wiley.

Suh, N. P. 2005. "Complexity in Engineering." *CIRP Annals: Manufacturing Technology* 54 (2): 46–63.

Semantic Alignment between Enterprise Repositories

Omar Sakka, omar.sakka@insa-lyon.fr; **Pierre-Alain Millet,** pierre-alain.millet@insa-lyon.fr; **and Valérie Botta-Genoulaz,** valerie.botta@insa-lyon.fr

Information exchanges between enterprise applications are never only a technical issue that requires different levels of technical protocols. According to Chen and Doumeingts (2003), interoperability has to be addressed also at the business and the functional layer. Information exchange is challenged by business and knowledge factors involving actors and their understanding capabilities, and depending on cultures, practices, and trust. Moreover, problems of misunderstanding and inconsistency occur with a risk of loss of semantics during the exchange between heterogeneous systems.

Meaningful and effective interoperability between enterprises requires that interoperability be applied to all levels of an enterprise. This is a semantic alignment challenge between heterogeneous business and technical knowledge. But how to use automatic approaches to solve semantic alignment, for example using ontology matching, is still an open research question. Considering the context of our applied research with business actors, we propose a decision-aided approach based on a standard business repository. This business repository is a framework for this semantic alignment between different engineering project views, particularly between business reference-model views that contain knowledge, and software views that support this knowledge.

To answer to software editors and integrators involved in the project, we focus on formalizing the knowledge of business processes and objects—knowledge that will be supported by applications and their interactions. This knowledge is heterogeneous across business, and is implemented in different ways (and often only partially) in each software package. Therefore, different actors may have different interpretations of the semantics exchanged. It is therefore necessary to have one unique, standardized representation for all business knowledge to ensure semantic interoperability for enterprises.

In this context, we built a Manufacturing Execution System (MES) business repository as a set of business process and objects, based on existing business standards and business reference models. The structure is in accordance with the IEC 62264 standard (IEC and ISO 2003). Next we enhanced this IEC-62264–oriented MES business repository using business expertise involved in the "MES project," along with the business semantic contents of web services defined by partners of the project, and some parts of the SCOR (supply chain reference) model (Supply Chain Council 2009). By doing this, we propose an alignment

Sakka et al. *continued*

methodology between the IEC 62264 standard and the SCOR model. In this article, we focus on this enhancement by the SCOR model as a case study of semantic alignment between business repositories.

Regarding the construction of MES business repository, we proposed as a first stage a general alignment framework based on ISO 19439 (IEC and ISO 2006) to define different alignment levels. We also proposed two levels of "genericity" (repository and project) and two model views (information view and function view).

To enable a coherent reading of the different models of the business repository, a definition of a meta-model is essential. We proposed, as a second stage, a metamodel for the rigorous construction of repository models (Bigand et al. 2004).

To achieve the semantic alignment between our "IEC 62264 oriented MES business repository" and the SCOR model, we proposed the following approach:

1. Comparison between the reference models for the two businesses to identify scope or structural differences, regarding different aspects (syntactic, terminological, conceptual, and pragmatic).
2. Extraction and modeling of the contents to be aligned. The extraction results from the comparison step. This step requires one to select the content to be aligned with the MES business repository. This choice is usually done by business experts to meet specific business needs. Selected content is then modeled according to our metamodel. This stage of modeling allows one to ensure a single reading of both reference models, because they are presented by using the same constructs. Moreover, using the same modeling language, the ARIS language in our case, ensures syntactic interoperability between the two repositories.
3. Alignment. This step defines the correspondence relationships between the components of IEC 62264 and SCOR model. In our approach, the alignment is unidirectional: it occurs only from the SCOR model elements to those of IEC 62264. Indeed, we seek to keep the IEC 62264 orientation of the MES business repository. This step cannot be automatic. As Klein (2001) has mentioned, the alignment step cannot be completely automated; the effort requires the knowledge and decisions of a domain expert. In this step, we align the two repositories by helping the expert to solve the heterogeneity problems identified in the first-phase comparison. We mechanize the expert work using rules defined to check consistency, completeness, and conformity to the IEC 62264.
4. Refinement. Interpretation of the models, objects, and business processes semantics of each repository varied from one expert to another, causing semantic conflicts in alignment phases. We suggest a step of refinement assisted by several business experts, each of whom works on business cases that are expressed by using the MES business repository. This step directly engages our MES business repository with the real business needs of companies. The heterogeneity of the experts in this step improves the quality of our repository.

The expert-oriented approach to semantic alignment between business repositories that we have outlined has the potential to support, and then to accelerate business projects in which interoperability issues are mainly caused by heterogeneous knowledge. Further research might enhance the rules defined for expert validation, especially to detect inconsistency between experts.

References

Bigand, M., O. Korbaa, et al. 2004. "Integration of FMS Performance Evaluation Models using Patterns for an Information System Design." *Computers & Industrial Engineering* 46 (4): 625–637.

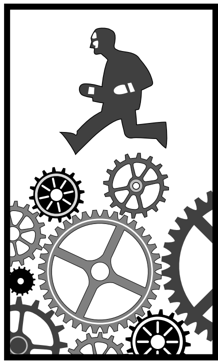
Chen, D., and G. Doumeingts. 2003. "European Initiatives to Develop Interoperability of Enterprise Applications: Basic Concepts, Framework and Roadmap." *Annual Reviews in Control* 27 (2): 153–162.

IEC and ISO. 2003. IEC 62264 (AFNOR 231). *Enterprise-Control System Integration*.

IEC and ISO. 2006. ISO 19439 (AFNOR 47). *Enterprise integration: Framework for enterprise modelling*.

Klein, M. 2001. "Combining and Relating Ontologies: An Analysis of Problems and Solutions." Paper presented at the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA (US).

Supply Chain Council. 2009. "Supply-Chain Council, SCOR Supply Chain Operations Reference Model." Version 9.0. http://www.supply-chain.org/scor/9.0.



A Systems Engineering Framework based on Eco-Design

Mourad Messaadia, mourad.messaadia@utc.fr; **Flore Vallet**, flore.vallet@utc.fr; **and Benoît Eynard**, benoit.eynard@utc.fr

In order to reduce their impact on the environment and to improve their product development process, original-equipment manufacturers integrate sustainable development and eco-design principles as an effective part of their industrial strategy at every stage. Eco-design can be categorized into three elements based on the product lifecycle (Takagi 2001). The first is the cost of the product, which represents economic value and concerns original-equipment manufacturers and suppliers. The second one is impact, which represents environmental value and the influence on the global environment. The third is product performance, which represents the fulfillment of consumer requirements and is related to safety, benefits, and convenience.

Eco-design should be considered as an added-value skill of suppliers, which clearly enables the systems engineering process of original-equipment manufacturers.

As the adoption of environmental rules and legislation becomes an unquestionable challenge to the industrial performance of the global supply chain and extended enterprise, original-equipment manufacturers have to involve their partners and suppliers in the deployment of their green business processes. These business processes are characterized by the partners' and suppliers' involvement in the whole lifecycle of original-equipment manufacturers' products. Thus the supply chain evolves from traditional subcontracting to a close partnership called vertical cooperation (Le Dain 2007).

In such a context, designers assess environmental impacts of manufactured products using numerous tools, but it is not quite sufficient to consider the environmental task completed (Vallet 2009). An important issue in this approach is the sharing of environmental features and information between engineering designers, production engineers, and other project team members (Zhou 2009). Any process or product has impact on the environment at various stages of lifecycle, related to numerous stakeholders (original-equipment manufacturers, suppliers, users). Systems engineering has to integrate these impacts to minimize them in a "design for environment" approach. Manufacturing industry is seen as the main cause of ecological destruction, since it produces waste, but it also has the potential to become a creator of products that generate ecological, social, and economic value (Mc Donough 2002).

In the proposed framework for eco-design–based systems engineering, eco-design is not just a way of thinking; rather it is an effective method supported by tools (Messaadia 2010). In this framework, according to a product's lifecycle each business process is identified, characterized, and assessed based on eco-design method and tools.

Researchers have noted in particular that small-to-medium supplier enterprises rarely implement eco-design in the product development process or production process (Schischke 2006). The question is which dedicated business process allows the enterprise to deliver end products in a sustainable way, all the way from the requirements-clarification phase until the end of lifecycle.

Currently systems engineering offers the possibility to link the development of end product and the development of enabling products (such as production and testing) in a unified framework (Sahraoui 2004). This work addresses two issues: the first one concerns eco-design and the second concerns the right organization for improving industrial performance. Eco-design is a subsystem in the development system of the enabling product shown in figure 1.

In figure 1, the system is broken down initially into the end product (the operating system itself) on the one hand and the enabling products (all products that enable the production, testing, operation, and support of the end product) on the other hand (ANSI and EIA 2003). Then the end product is broken down into subsystems, and each subsystem is broken down into end product and enabling products. The refinement process is repeated until obtaining elementary parts or commercial off-the-shelf components. The fulfillment of a functional specification and environmental specification requires information sharing. Once the specification is made, areas for improvement are identified. The improvement may include, for example, a choice of materials with reduced environmental impact or further energy optimization of production processes. The product is fully designed; it will imply a

Messaadia et al. *continued*

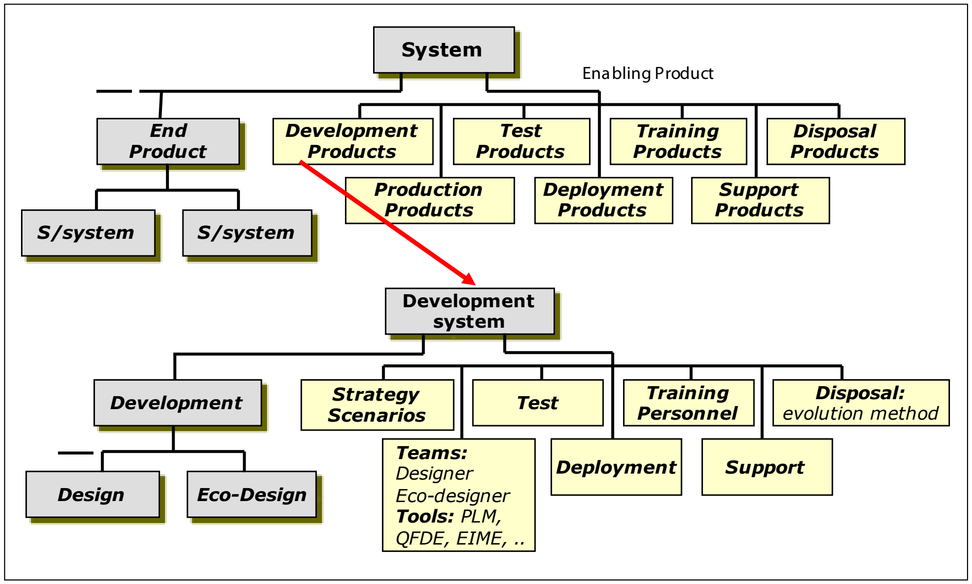


Figure 1. Eco-design in the systems engineering breakdown structure

new global assessment of environmental impacts.

The stakeholder's definition starts before the system development: it allows the engineer to identify who should be involved in the development of the system and the project. So when the process of identifying stakeholders starts, the first question to be answered is: "Who does what?"

For such a purpose, it is necessary to characterize the application among the set of applications fields, such as manufacturing, building, finance or business, and critical systems. Specific enabling products as testing, operation, maintenance, and disposal can be addressed as subsystems. The steps that need to be covered are these:

1. Identification of main attributes of the application
2. Hierarchical and strategic needs
3. Planning and analysis
4. Implementation
5. Verification of the deployment process

In the global supply chain and extended enterprise, the original-equipment manufacturer needs to integrate suppliers to operate the enabling products (figure 1). A database of the potential partners and suppliers is established, and the partners and suppliers are classified according to a list of key performance indicators. This database will evolve over time and integrate the changes of suppliers' expertise (new acquired tools, developed skills and know-how, or offered manufacture technology). The eco-design abilities of suppliers will be a key performance indicator to select and involve them in new product development and manufacturing processes.

Eco-design should be considered as an added-value skill of suppliers, which clearly enables the systems engineering process of original-equipment manufacturers (Fargnoli 2008). Supplier integration during the requirements clarification phase led by original-equipment manufacturers will allow the supplier to be proactive. It anticipates the environmental and functional needs, proposes the appropriate green solutions, properly manages the technical changes expected by the original-equipment manufacturers and, further, it contributes along with the final user toward developing the systems that are enabling and end products.

References

ANSI and EIA (American National Standards Institute and Electronic Industries Alliance). 2003. ANSI/EIA 632. *Processes for Engineering a System*. Approved 7 January 1999, reaffirmed 2003.

Fargnoli, M., and T. Sakao 2008. "Coordinating Ecodesign Methods in Early Stages of Industrial Product Design." *International Journal of Environmentally Conscious Design and Manufacturing* 14 (2): 35–65.

Le Dain, M. A., R. Calvi, and C. Sandra. 2007. "Proposition d'un modèle d'évaluation de la performance fournisseur en conception collaborative." Paper presented at the seventh Congrès International de Génie Industriel, Trois-Rivières, Québec, Canada, 5–8 June.

Messaadia, M., F. Vallet, and B. Eynard. 2010. "Sustainability as Requirement in System Engineering Approach." Paper presented at the International Conference on Advances in Production Management Systems, Cernobbio, Italy, 11–13 October.

McDonough, W., and M. Braungart. 2002. *Cradle to Cradle: Remaking the Way We Make Things*. New York, NY (US): North Point Press.

Schischke, K., J. Mueller, and H. Reichl 2006. "Ecodesign in European Small and Medium Sized Enterprises of the Electrical and Electronics Sector." Paper presented at the IEEE International Symposium on Electronics and the Environment, Scottsdale, AZ (US), 8–11 May.

Sahraoui, A.-E.-K., D. M. Buede, and A. P. Sage. 2004. "Issues for Systems Engineering Research." Paper presented at the Fourteenth Annual International Symposium of INCOSE, Toulouse, France, 20–25 June.

Vallet, F., D. Millet, and B. Eynard. 2009. "Investigating the Use of Eco-Design Guides: Presentation of Two Case Studies." Paper presented at the Seventeenth International Conference on Engineering Design, Palo Alto, CA (US), 24–27 August.

Takagi, T. 2001. "Eco-Design Towards Green Productivity." *In Greening Supply Chain: Enhancing Competitiveness through Green Productivity*, 79–89. Tokyo, Japan: Asian Productivity Organization.

Zhou, C. H., B. Eynard, and L. Roucoules. 2009. "Interoperability between PLM and RoHS Compliance Management based on XML and Smart Client." *Journal of Computing and Information Science in Engineering* 9 (3): 312–329.

Forum

An Introduction to Project and Product Traceability

Brian Berenbach, brian.berenbach@incose.org; and **Ren-Yi Lo**, ren-yi.lo@siemens.com

Untraced requirements lead to projects at risk, including delayed delivery, overrun budget and resources, and delivered systems that fail and cause injuries. This article will explain the need for requirements traceability, distinguish the different types of traceability, and discuss typical problems that occur when implementing traceability processes.

Tracing is mandatory for some types of projects and organizations to demonstrate compliance to regulatory mandates or to demonstrate business goal achievement.

Traceability is the ability to describe and follow the life of a requirement, from its origins to development and specification, to its subsequent deployment and use, and through iterative refinement in any of these phases (Gotel and Finkelstein 1995). During project execution, people in different roles need traceability for different purposes. The three most common needs are coverage, impact, and derivation analysis. *Coverage analysis* determines if all requirements in a contract or specification have been met. In practice, traces from originating requirements to product or system test cases are placed in a matrix and checked for coverage. *Impact analysis* is the technique of using requirements to design traces; these traces will be used to determine the cost impact of making a change by analyzing the impacted parts of the design. Finally, *derivation analysis* helps determine why a feature is in a product. Test or design elements associated with the feature are traced back to the originating requirements.

When the scale of a project becomes too large, processes break down. System engineers sometimes define tracing processes at the beginning of a project without thorough consideration of the total number of requirements. For example, a project with 50 features may easily have 5,000 high-level requirements, or 50,000 low-level requirements (assuming a proportion of 1 feature to 100 high-level requirements, and 1 high-level requirement to 10 low-level requirements). There will be at least the same number of traces to develop and maintain. In fact, the actual number of traces may be higher. One Siemens transportation project reported an average of about seven to eight traces per requirement in the database. In any case, tracing can be easily classified as a large project task.

Finally, projects will require different traces for various reasons. A buyer of a product or services will need traces that are different from those of the vendor or contractor providing the services. To create a product that will fulfill a need defined by marketing, a manufacturer will require yet a different trace mechanism. The process involving buyer and contractor is illustrated in figure 1.

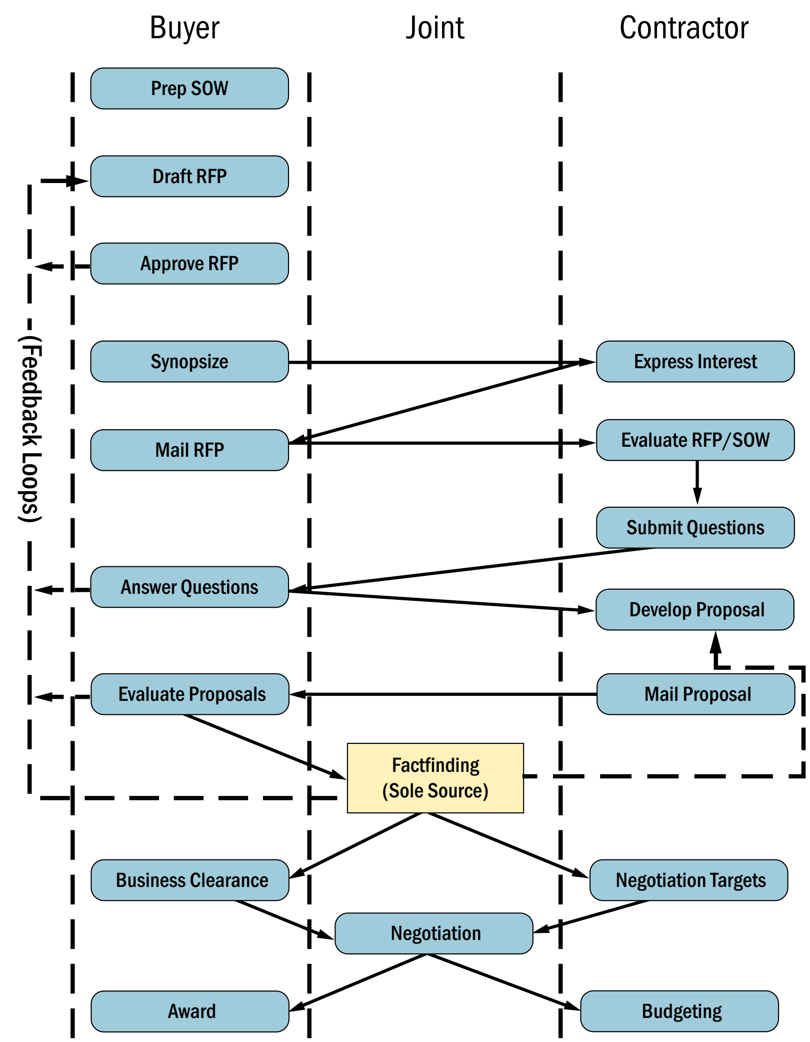


Figure 1. Baseline contracting process

Role-Based Traceability Needs

Several components define a traceability strategy:

- Resources and tooling for setup and management of traces
- Justification for effort to define and manage traceability
- Size of project

Project roles and needs often influence a traceability strategy, as shown in table 1.

Table 1. Traceability needs based on role

Role	Manufacturer	Buyer	Contractor
Marketing and Sales	Is development implementing all features? What features are in the current product or planned release?	Has the contractor fulfilled the contract? Are all product features appropriately implemented and operational?	Is work executed in compliance with the contract? Has every contract item been successfully implemented per test cases?
Requirements Analyst	Has each feature been sufficiently explored? What requirements are associated with a feature? Do the requirements have sufficient detail?	Do all features and related requirements in the Request for Proposal trace to the contract? Does the contractor understand each feature?	Are all requirements in the contract feasible and testable? Are there any cost drivers? Do requirements trace to test cases?
Architect	Why is this feature needed? What is the cost to change a feature? Does the architecture of the product support all the features that are in scope?	(not applicable)	Do all contract requirements trace to correct design elements? Does the design fully meet the contract? What is the impact of a change request?
Developer	What is the context of this requirement? Why is it relevant? Do I have enough information to develop?	Will this product be maintainable after deployment? Will the trace mechanisms support an impact analysis?	What is the context of this requirement? Why is it relevant? Is there sufficient information for development?
Testing	Are all product features covered by test cases? Are all test cases traceable back to product features?	Are contract requirements traced to test cases? Does the product reflect the contract? Is there any non-compliance?	Is the work executed in compliance with the contract? Will the delivered product pass the acceptance test?

Traceability based on the V Model

The strategy for tracing requirements has two starting points: (1) creating a project V model or (2) creating a project metamodel. Figure 2 describes the V model approach.

An alternative approach is to define a metamodel (see figure 3). The V-model approach to defining traceability is used when regulatory codes are significant and when the product is outsourced. The metamodel approach is used when there are disparate sources of requirements or the requirements engineering process is complex.

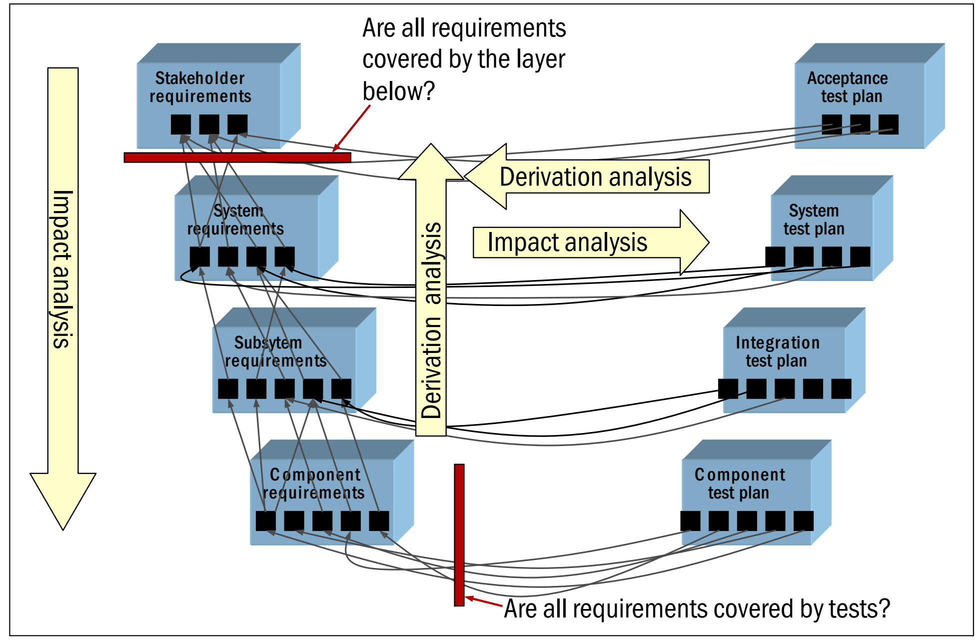


Figure 2. V-model traceability

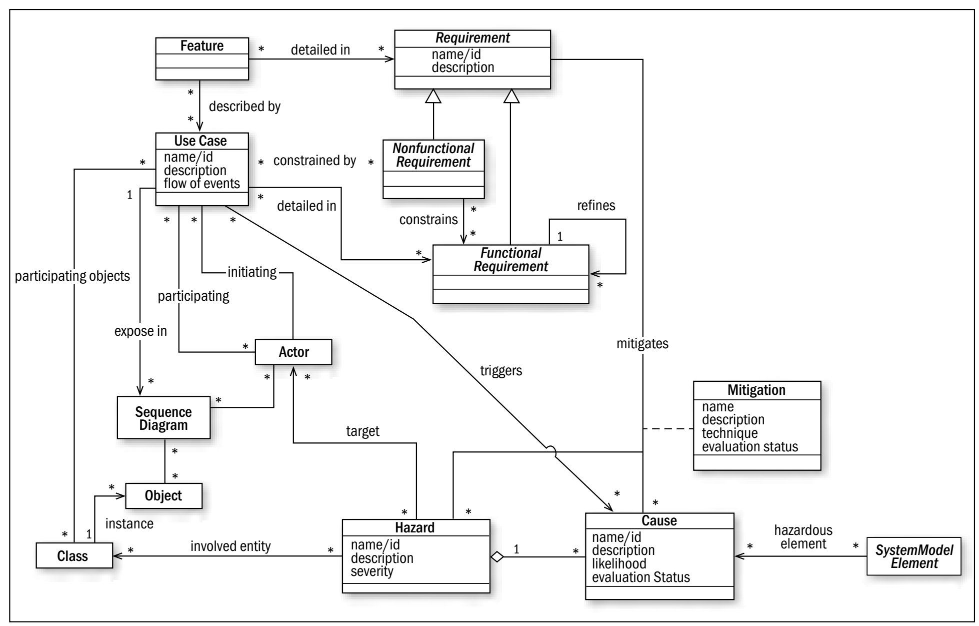


Figure 3. Sample metamodel subset showing a hazard to a requirement relationship

Impact Analysis

Analyzing the impact of a change request corresponds to tracing down the left side of the V model. Traces go from features to requirements, and then from detailed requirements to design. The objective is to determine the cost of a change request or new feature. The architect traces from the impacted features to the actual system design to determine how significant the modifications would be, including the cost impact.

Derivation Analysis

Derivation analysis (see Hull et al. 2002) help discover the origin and rationale of a function. Figure 2 illustrates two types of derivation analysis. The test case associated with the product feature is traced back to the stakeholder requests, market demand, or business goals that led to the decision to put the feature in the product. Alternatively, a component requirement is traced back to the original rationale for creating it. From the perspective of the V-model, requirements are traced directly across the V from right to left.

Coverage Analysis

Coverage analysis requires one to trace from product features or contract requirements to test plans; the goal is to ensure that every deliverable feature or contract requirement is in the end product. Figure 2 shows the top bidirectional trace. From the perspective of the V model, traces start from the left side of the V and go directly across to the right side.

Coverage analysis determines the following:

- If all requests for features are in the delivered product
- If test cases exist for all functional and nonfunctional requirements

Meeting Contractual Obligations

So that contractors can track whether or not they have met their contract obligations, every line item in the contract that represents a functional or nonfunctional requirement must trace to one or more test cases in the test plan. Hence, via coverage analysis, it is possible to determine that every contract requirement has been addressed. By means of derivation analysis, each requirement drives test cases that can check for correctness.

Using Traces to Support the Bidding Process

During bid evaluation, the architect is responsible for identifying requirements in the request for proposal and tracing each requirement to an analysis to determine cost, risk, and feasibility. The requirements from the request for proposal are then traced to the proposal. Traces exist between the requested requirements and the requirements' cost and risk analysis, and between the analysis and the line items in the proposal. A synthesis of these items becomes the foundation of the contract and is then traced to the design elements.

Failure to meet contractual obligations leads to lawsuits or personal risks. Furthermore, full requirements traceability for a contract is usually demanded by government agencies, especially those that mandate regulatory codes. For example, the US Food and Drug Administration requires full traceability for medical products.

On one project known to the authors, during the negotiation sessions between the buyer and contractor it was mutually agreed that the high performance requested by the buyer would require additional hardware. It was duly noted in the minutes of the project meeting that the buyer would pay for the added equipment (about USD 500,000). However, despite being written in the signed minutes, the agreement never made it into the contract. When the additional equipment was purchased, the buyer refused to pay for it, claiming that the contractor had agreed to bear the burden. As there were no traces from the design that linked the new hardware back to the meeting minutes (the rationale), the contractor was held liable for the amount, which exceeded the entire project contingency. The project manager of the contractor was then fired by his management for failure to exercise due diligence.

Traces are required from product features through defined testable requirements and to relevant design and implementation artifacts. The exact trace mechanism used is left to the vendor to decide.

Using Traces to Meet Regulatory Codes

Failure to meet regulations can become a criminal act. Since human lives depend on the performance of certain software (in particular in the medical and transportation domains), full traceability is required. In addition, where product failure may result in injury or death, hazard analysis is mandated, and traces are required from the original feature to the hazard analysis to mitigating requirements.

Failure to follow "best practices" can become criminal when dealing with systems that have hazards to the user. For example, a product feature might state, "The operating-room software will show a picture of the patient to ensure that the correct patient is operated on." But if there is limited or incorrect traceability, this feature may not have made it into the final product. At run-time of the product, when a wrong patient is operated on or there is a death, the individuals responsible for tracing may be held criminally and civilly liable.

Delivering to Market

Delivering on time and in budget is a goal that covers the entire product lifecycle. Tracing is critical to performing impact analysis. For example, functionality may have been added to the product that did not trace back to agreed features (this is called "gold plating"). The extra effort to add unnecessary features leads to significant delays in the time-to-market. Failure to trace every high-level feature to an analysis where effort and feasibility are determined may lead a project team to discover too late that they cannot meet their deadline.

Traces are required from scoping documents through product features. To prioritize features, it is necessary to trace the features back to the stakeholder request's rationale.

Failure to trace business needs to product features and requirements may result in late delivery to market. On a project to create a new payroll software system for automobile dealerships, one of the software product features was the ability to run on both UNIX and Windows platforms. The need for compatibility resulted in much added complexity and late delivery. During a project audit, it was discovered that there was no trace from the compatibility requirement back to a stakeholder. Upon further investigation it was determined that the requirement was created by a developer who simply thought it would be "cool" to run on both UNIX and Windows. The projected late delivery (after the official product announcement) resulted in the project being canceled and restarted. The entire project team lost their jobs, right up to the vice president in charge of product development.

Hazard and Threat Analysis

Hazard analysis is done on product features that have the potential for injury or death to people. Traces for hazards are described as follows:

- Any product requirement that may be a hazard to the user traces to a hazard analysis (Berenbach and Wolf 2007).
- The hazard analysis traces to any new requirement that mitigates the risks associated with the hazard.

For medical devices, it must be possible to demonstrate that every high-level requirement or feature has been checked for a potential hazard, that an analysis was done for those hazards, and that additional mitigating requirements have been added where necessary.

A threat is the potential for misuse of a system, associated with either financial risk or bodily harm. The types of traces are the same: a product feature leads to threat analysis, which leads to a mitigating treatment.

Traceability Techniques and Traps

Many requirements-engineering processes do not scale well. What works well with 100 requirements may implode with 10,000. The following problems are common on large-scale projects.

- Requirements hierarchies are not defined (for example, the parent–child relationships between feature and requirement are unclear). Failure to create a hierarchy leads to an overwhelming number of traces that are not maintainable or useable. Defining requirement hierarchies in the database schema permit database trace queries to return a meaningful subset of traces.
- No glossary of terms is defined. Failure to define a glossary results in different terms being used for the same thing, causing ambiguity and difficulty in mining tracing relationships.
- The project-artifact metamodel is not defined at project initiation. Creation of a metamodel will reveal all possible trace types that may exist and enable an automated (or manual) trace strategy.
- The requirements-database schema is not defined. Requirements databases are a good source of project metrics, but only as good as the metrics defined and the adherence to record them.
- Requirements are often ambiguous. This indicates that analysts are not adequately skilled and need additional training.

Nonfunctional Requirements

The senior architect is typically responsible for managing nonfunctional requirements (McBride 2007). This task includes the following:

- Tracing nonfunctional requirements to implementation requirements and design and test cases
- Analyzing nonfunctional requirements
- Tracing nonfunctional requirements to the analysis, rationale for the functional requirements, and design artifact that derives from the nonfunctional requirements.

Summary

Tracing is mandatory for some types of projects and organizations to demonstrate compliance to regulatory mandates or to demonstrate business goal achievement. In some rare instances, a lack of viable tracing strategy can result in severe project risk, including civil or criminal penalties. Since the architect (or systems engineering equivalent) is the only technical manager that oversees the project from negotiation with the client to final acceptance testing and deployment, she or he may have to accept responsibility for ensuring that effective tracing mechanisms are in place across the lifecycle.

» continues on next page

Berenbach et al. *continued*

Defining processes that enable tracing and finding adequate tooling to support tracing is not easy. Moreover, as the project's size and scope increases, simplistic processes can disintegrate. Often process breakdowns are only visible after a project has reached a critical phase, such as integration testing.

References

Berenbach, B., and T. Wolf. 2007. "A Unified Requirements Model: Integrating Features, Use Cases, Requirements Analysis, and Hazard Analysis." Paper presented at the IEEE International Conference on Global Software Engineering.

Gotel, O. C. Z., and C. W. Finkelstein. 1995. "An Analysis of the Requirements Traceability Problem." Paper presented at the Second IEEE International Symposium on Requirements Engineering.

Hull, E., K. Jackson, and J. Dick. 2002. *Requirements Engineering*. London, UK: Springer.

McBride, M. R. 2007. "The Software Architect." *Communications of the ACM* *50*:75–81.

Technical Operations

The Evolution and Future Direction of INCOSE's *Systems Engineering Handbook*

Kevin Forsberg, kevin.forsberg@incose.org**; and Garry Roedler,** garry.roedler@incose.org**; Cochairs of the Knowledge Management Working Group**

INCOSE's *System Engineering Handbook* (SEH) has become a useful resource to many organizations for creating their internal system engineering process documents. It has also been an essential document for those studying for INCOSE's system engineering certification exam, since it is the basis of the exam for both Certified Systems Engineering Professional and Associate Systems Engineering Professional. Certification applicants especially want to know that the handbook remains current. It is relevant, therefore to review the evolution of the handbook.

Table 1 shows the series of revisions to the handbook. This table indicates that there has been an average of three years between handbook revisions that impact the certification exam. The latest version, SEHv3.2, has had two subsequent releases focused on correcting spelling errors, minor text duplication, and editorial mistakes. Neither of these updates had impact on the exam, so a third decimal place (v3.2.1 or v3.2.2) was used in the version number to distinguish that these were only minor updates to version 3.2. To help those studying an earlier version of the handbook, the revision team will post on the INCOSE website the log of the changes to go from version 3.2 to 3.2.1, and from version 3.2.1 to 3.2.2. Again, version 3.2 is the basis for the current exam, and will remain so for at least one more year.

The last line in table 1 shows the September 2011 release of version 0.5 of the *Systems Engineering Body of Knowledge* (SEBoKv0.5), which is a product of the BKCASE project (Body of Knowledge and Curriculum to Advance Systems Engineering). The SEBoK, a wiki-based reference, is now publicly available for review and comment. On 18 September 2011 David Olwell, colead of the BKCASE development team, announced its release with this statement:

> SEBoK version 0.50 is released for world-wide review. Over the next three months (to mid December 2011), we are soliciting feedback through the discussion tabs on each article, and through a general form found in the left margin of each page under the tab, "Note to Reviewers." On December 15th, this feedback will close, and we will again compile comments and begin another adjudication round. We are tentatively planning to address immediate issues raised by the reviewers in a version 0.75, if necessary, to be released late March or early April 2012.

The website for the SEBoK, as noted in table 1, is http://www.sebokwiki.org/. Over 60 volunteer authors worldwide contributed to the creation of the SEBoK wiki. Currently the project is jointly managed by Art Pyster at Stevens Institute of Technology and David Olwell at the US Naval Postgraduate School. This three-year development project is partially funded by the US Office of the Secretary of Defense. At present all copyrights are held by Stevens Institute of Technology,

Table 1. Evolution of the INCOSE Systems Engineering Handbook

INCOSE Handbook version	Release Date	Comments
SEHv0.0	Jun 1994	Internal NCOSE use only; based on military standards from the US Department of Defense (DoD)
SEHv2.0	Jun 2004	Based on US DoD standards; Original basis for certification exam
SEHv3.0	Jul 2006	Restricted to 150 pages; New text, aligned with ISO/IEC 15288 (2002)
SEHv3.1	Aug 2007	Added 170 pages in appendices to fill in missing detail; New basis for certification exam
SEHv3.2	Jan 2010	Significant reorganization; Alignment with ISO/IEC 15288 (2008); **Current basis for updated certification exam**
SEHv3.2.1	Jan 2011	Minor corrections (spelling and typos); **No impact on certification exam**
SEHv3.2.2	Nov 2011	Minor corrections (more spelling and typos); No impact on certification exam
BKCASE SEBoK Wiki v0.5	Sep 2011	Version 0.5 publicly available for comment as of September 2011 at http://www.sebokwiki.org/

but upon project completion in the fourth quarter of 2012, joint stewardship of the SEBoK will be assumed by INCOSE and IEEE Computer Society, and the copyright for the SEBoK will be transferred to them. The joint management team will consist of two people from INCOSE (William Miller and Kevin Forsberg) and two people from IEEE Computer Society (Dick Fairley and Tom Hilburn). In addition, access to the Wiki version is to be freely available to all interested parties worldwide.

The open question is how the BKCASE SEBoK will influence future versions of the *Systems Engineering Handbook*. At present the plan is to develop and release a new version of the handbook (v3.3) in about a year, as shown in table 2. Several INCOSE working groups have asked to expand topics for their area in the current handbook, and these contributions, after INCOSE review, will provide clarification and amplification of existing material. The Knowledge Management Working Group encourages all members and working groups in INCOSE Technical Operations to provide input for this next version. The meeting at the International Workshop in January 2012 will focus on development of the outline for SEHv3.3, with the target of releasing the completed document by the fourth quarter of 2012. The INCOSE certification team expects to update the certification exam in the second quarter of 2013, based on SEHv3.3.

Version 1.0 of the SEBoK will continue to be in a monitored wiki format, and hopefully it will converge to a robust, stable baseline. At that time, consideration will be given to the scope and content of SEHv4.0, which may transition to reflect the stable core of the SEBoK, or it could be an independent document drawing on and contributing to the wiki-based SEBOK. SEHv4.0 will also be influenced by further evolution of ISO/IEC 15288 and other key developments in the field. At present the certification program requires a stable reference from which questions can be created. Since the SEBoK wiki will continue to evolve and it is actually a guide to the knowledge base with pointers to many other resources, including the INCOSE handbook, it is probable that the two resources will exist in parallel. It is the hope of the handbook development team that all interested professionals in INCOSE and IEEE will constructively comment on and participate in the evolution of this body of knowledge.

Table 2. Future direction of the INCOSE Systems Engineering Handbook

INCOSE Handbook version	Release Date	Comments
SEHv3.3	Planned release third or fourth quarter 2012	Expansion planned in several key areas. **Call for authors and contributors:** Meeting at International Workshop, January 2012; Will become basis for updated certification exam in 2013.
BKCASE SEBoK Wiki v1.0	Planned release fourth quarter 2012	In the fourth quarter of 2012, control and copyright will pass from Stevens Institute of Technology to joint four-person management team representing INCOSE and IEEE Computer Society.
SEHv4.0	To be determined	Evolution from SEHv3.3 and the stable sections from SEBOK Wiki. Will eventually become basis for updated certification exam.

Advances in the Integration of Program Management and Systems Engineering:
Progress of the INCOSE-PMI-LAI Community of Practice on Lean Principles in Program Management

Josef Oehmen, josef.oehmen@incose.org

In January 2011, a group of over 50 industry and government practitioners of program management and systems engineering from more than 20 organizations set out to lessen the gap between program management and systems engineering. As a joint working group between INCOSE, the Program Management Institute (PMI), and the Massachusetts Institute of Technology's Lean Advancement Initiative (LAI), the group develops best practices for systems engineering programs (and beyond). Their work is based on the six lean-management principles: value, value stream, flow, pull, perfection, and respect for people. The use of the lean-management principles is particularly powerful, as they heavily emphasize the need for overall integration of the value delivery across all process and organizational boundaries (including the boundary between program management and systems engineering).

The goal is to produce a handbook that (1) identifies the most significant challenges for systems engineering programs, and (2) presents corresponding enablers to overcome these challenges. The first draft of this handbook is to be released in January 2012 at the annual INCOSE International Workshop. While the group's main focus is on systems engineering programs, the group is also attempting to generalize their findings so they can be applied to other types of programs as well, for example information-technology programs, business-transformation programs, or community- and society-focused programs.

The collection, validation, ranking, and aggregation of the challenges were finished during the summer. The findings were consolidated into 10 major challenges:

1. Reactive program execution
2. Lack of stability, clarity, and completeness of requirements
3. Insufficient alignment and coordination of the program enterprise
4. Value stream not optimized throughout the enterprise
5. Unclear roles, responsibilities, and accountability
6. Mismanagement of team competency and knowledge
7. Insufficient program planning
8. Improper metrics, metric systems, and key performance indicators
9. Lack of active risk management
10. Poor acquisition and contracting practices

The latest face-to-face meeting of the group occurred on 22 October during the PMI Global Congress. The interest from the PMI community in the half-day presentation and workshop was so strong that registration had to be closed weeks ahead of the meeting. The 75 participants were very interested in the preliminary results, as well as contributing their own insights and feedback in break-out sessions.

The main focus was on the 101 best practices in 23 categories that the community of practice had collected so far. The group currently focuses on completing this collection of best practices, and the workshop at the PMI Global Congress has provided valuable input and guidance. The major activities of the group are as follows:

- Validation of the best practices through a number of methods (consensus of subject-matter experts, community feedback from INCOSE and PMI, surveys, mapping to successful—and unsuccessful—programs)
- Drafting the first version of our handbook including the group findings
- Development of additional materials with a particular focus on teaching our findings

If you are interested in bringing program management and systems engineering closer together, or in the application of the lean-management philosophy to programs, this community of practice is for you! There are two ways to get involved:

1. *As a subject-matter expert*. The group is very dynamic, and we always welcome additional members. Our subject-matter experts devote at least one hour per week to working for the group, often more. The group meets online via Webex every Monday, 2–3 p.m. Eastern Standard Time (14:00–15:00 UTC–05)
2. *As a general member of the group*. You will be invited to participate in monthly reviews of the group's progress in one-hour online meetings. These meetings occur every first Tuesday of the month, 2–3 p.m. Eastern Standard Time

To join our group in either capacity, and for any other questions, please contact Josef Oehmen at the e-mail address above.

INCOSE International Council on Systems Engineering

PMI Project Management Institute

LAI MIT ENABLING ENTERPRISE EXCELLENCE

INCOSE Operations

INCOSE Certification Agreement Signed with KCOSE

Dave Walden, ESEP, david.walden@incose.org

INCOSE and the Korean Council on Systems Engineering (KCOSE) signed a memorandum of understanding to collaborate on professional certification of systems engineers. This agreement was signed at the Fifth Annual Asia Pacific Conference on Systems Engineering (APCOSE) held in Seoul, Korea, in October 2011.

The memorandum will make available subject-matter experts in certification to jointly develop a certification-recognition scheme by which KCOSE's own certification program can be internationally recognized through INCOSE. KCOSE and INCOSE will also work together in a larger forum to develop the certification recognition scheme for use with other national and local certification programs.

Signing the INCOSE and KCOSE agreement, seated from left to right: John Thomas, INCOSE president-elect, and Sung Kyou Choi, KCOSE president. Also present, standing left to right: T.S. Yeo, INCOSE director of international growth; Dave Walden, INCOSE certification program manager; Chul Whan Kim, former KCOSE president; Sung Ki Min, former KCOSE president; and Yong Soo Kwon, KCOSE president-elect. (Photo by Cecilia Haskins.)

INCOSE Certification Agreement Signed with Jacobs Technology

Dave Walden, ESEP, david.walden@incose.org

INCOSE and Jacobs Technology have signed an agreement in October 2011 that allows Jacobs to leverage the INCOSE Systems Engineering Professional certification program. Through the terms of this agreement, Jacobs and INCOSE are collaborating in certifying appropriately experienced Jacobs systems engineers.

In making the announcement, Jacobs Chief Executive Officer Craig Martin stated, "Employee talent is the cornerstone of our success. This agreement provides an excellent framework to collaborate with INCOSE and continue improving our systems engineering capability. Providing our employees with opportunities to hone their expertise contributes to value-added performance and promotes client confidence in our ability to partner with them for success."

INCOSE President Samantha Robitaille commented, "INCOSE's professional certification designations are the worldwide reference for systems engineering professionalism. INCOSE is pleased to enter this agreement with Jacobs to advance the practice of systems engineering."

Organizations interested in leveraging INCOSE's certification program should contact Certification Program Manager Dave Walden at the address above. For more information on INCOSE's certification program, please visit http://www.incose.org/educationcareers/certification/.

Updated Logos for INCOSE's Systems Engineering Professional Program

Dave Walden, ESEP, david.walden@incose.org

The INCOSE Systems Engineering Professional (SEP) program continues to receive increased recognition and experience impressive growth. Since 2004, INCOSE has recognized more than 1,000 professionals. We are seeing the SEP designations appear in numerous job postings and have begun to see them being referenced in requests for proposals. In addition, nine businesses, two universities, and several other groups have already partnered with INCOSE through agreements to better leverage the SEP program within their organizations.

» continues on next page

Updated Logos for INCOSE's Systems Engineering Professional Program *continued*

The SEP program has been implemented with three tiers of credentials and optional extensions. The original foundation-level Certified Systems Engineering Professional (CSEP) credential was complemented by the addition of the entry-level Associate Systems Engineering Professional (ASEP) in 2008 and the senior-level Expert Systems Engineering Professional (ESEP) in 2010. An extension for Acquisition (Acq) was added in 2008 that provides recognition of additional systems engineering knowledge in the United States Department of Defense acquisition environment.

To reflect the global nature of the organization, the INCOSE Certification Advisory Group and the INCOSE Board of Directors approved an update of the program's image with new logos that align the program with the organization. The new SEP logos add rings, orbiting around the INCOSE globe logo. The outer rings spell out the various credentials and retain the color linkages with the existing logos: green for ASEP, purple for CSEP, and gold for ESEP. In the center of the globe are the INCOSE name and the abbreviation for the relevant certification level. The extension logo is now a "tab style" format retaining the gray color of the current Acq logo. In addition, a new "generic" SEP logo has been created. This will be used when discussing the entire SEP program, rather than any individual credential. The generic SEP logo uses the standard INCOSE blue color.

The changeover to new logos also affords INCOSE the opportunity to protect the logo usage. All of the SEP logos also include the trademark symbol (™). The Certification Advisory Group has developed usage guidelines for the logos, so people or organizations wishing to use the new logos should contact the INCOSE Certification Program Office.

INCOSE SEP SYSTEMS ENGINEERING PROFESSIONAL

CERTIFIED INCOSE CSEP SYSTEMS ENGINEERING PROFESSIONAL

ASSOCIATE INCOSE ASEP SYSTEMS ENGINEERING PROFESSIONAL

EXPERT INCOSE ESEP SYSTEMS ENGINEERING PROFESSIONAL

INCOSE ACQ

INCOSE Events

INCOSE Representation at the International Society for the Systems Sciences 2011 Annual Conference

Stuart Arnold, stuart.arnold@incose.org

Systems sciences are key contributors to the mix from which systems engineering has been synthesized. Accordingly, INCOSE's System Science Working Group is committed to highlighting the linkages between systems-science theory and the empirical practices of systems engineering (see http://www.incose.org/practice/techactivities/wg/sysscìwg/). To support the working group's objective of encouraging the advancement of systems-science principles and concepts as they apply to systems engineering, the group has begun a dialogue with the International Society for the Systems Sciences (ISSS; http://www.isss.org/). Since its establishment in 1956 (originally as the Society for General Systems Research), the ISSS has included some illustrious members, such as INCOSE Pioneer Peter Checkland, who have helped to build today's body of systems-sciences knowledge.

Both groups viewed it as a natural step in this developing relationship for each group attend each other's annual conferences and, in accordance with the INCOSE/ISSS memorandum of understanding, INCOSE was formally invited to attend the 2011 ISSS Conference held in July in Hull, UK. The occasion helped launch the cooperation between the international focus of the systems-sciences community and the international focus of the systems engineering community. INCOSE President Samantha Robitaille represented INCOSE along with Len Troncale and Stuart Arnold, and to signify the importance of the emerging relationship, President Robitaille delivered a keynote speech. The conference provided an opportunity for members of both organization to "test the waters" regarding common understanding and the opportunities for cooperation.

For INCOSE this relationship should help establish a better, more explicit appreciation of the systems sciences' contributions to systems engineering, and thereby help to progress our discipline. Conversely, the principles and practices that have evolved from systems engineering's multidisciplinary synthesis can bring something to the way ISSS members think about and evolve systems sciences. The synergy of this situation holds great promise.

A workshop period at the ISSS conference was dedicated to exploring the similarities and contrasts between the domains of interest of both organizations. Although the language and concerns of the two communities have different emphases, participants concluded that there is considerable mutual benefit to be gained from the new relationship. There is significant commonality of purpose and several topics of interest on which to cooperate, and the meeting considered ways in which ISSS members might become actively engaged with INCOSE. Areas offering immediate interest were BKCASE, education, research funding, standards, and architecture.

Collaboration: INCOSE at the Fifty-Fifth Annual ISSS Conference (University of Hull, UK, 16–22 July 2011)

Len Troncale, len.troncale@incose.org**; and James Martin,** james.martin@incose.org

INCOSE has entered into a collaborative agreement with the International Society for the Systems Sciences (ISSS). Details of the agreement can be found at https://sites.google.com/site/syssciwg/meetings/workshop-jun-2011/ISSS_SystemsScience_MOU_2011.pdf.

Building on the interchange that occurred during the International Workshop in Phoenix through discussions of the INCOSE Systems Science Working Group (SSWG), the most recent activity of the collaborative effort was at the ISSS Annual Conference in the United Kingdom. The team representing INCOSE included Samantha Robitaille, Stuart Arnold, Mike Yearworth, and Len Troncale.

INCOSE President Samantha Robitaille delivered a plenary presentation during which she described why systems engineering had difficulty being widely accepted. Instead of systems engineering she might just as well have said systems science. This shows why the two societies need each other's contributions and could benefit from each other's support. The last question asked of the INCOSE president focused on exactly this point: "Mechanical engineers prepare for their profession by taking many courses in physics; chemical engineers prepare for their profession by taking many courses in chemistry; do systems engineers prepare for their profession by taking courses in systems science?" President Robitaille subsequently agreed that there was a lack of courses in systems science that reflect a consensus view of the field that also had credibility and relevance to systems engineers—an area of possible focus for this collaboration.

That is precisely why INCOSE's Systems Science Working Group initiated four projects, two particularly relevant to this collaboration with ISSS, because they parallel two existing ISSS special integration groups. The first of these (here, Project One) aims to identify, unify, and integrate the many existing systems theories. Project Two would develop a new, top-down, "systems pathology," at the level of systems architecture. Both projects aim to produce tools and deeper understanding to improve systems design. Both address key problem areas and provide excellent foci for future collaboration.

» continues on next page

Troncale et al. *continued*

Both INCOSE and ISSS participants attended a four-hour workshop on the collaboration effort. (Please see Arnold's report on the outcome of this workshop in this edition of ***INSIGHT***). In addition to the workshop, the ISSS Special Integration Group on Research Toward a General Theory of Systems and its Special Integration Group on Systems Pathology featured four presentations that also contributed directly to this collaboration effort. For Project One, Len Troncale spoke on "Towards A Science of Systems I: Proving Isomorphy" and "Towards A Science of Systems II: Types of Non-Linear Causality for Taming Complexity." For Project Two, the same author spoke on "A Manifesto On Systems Pathology" and "Comparing Systems Pathology Treatments Across Systems Processes Theory, Living Systems Theory, and Soft Systems Methodology."

Fourteen research posters relevant to the collaboration were mounted for the entire conference period. The first seven posters provided an introduction to a unified systems process theory (SPT), while posters eight to fourteen were on applications of this theory to sustainability. The latter seven posters were authored by seven graduate students from California State Polytechnic University (Pomona, CA, US), supervised by Professor Troncale. All of these posters are possible contributions to the Systems Science Working Group's Projects One and Two.

In his capacity as past president of ISSS and current member of its Board of Trustees, Troncale also represented and explained the INCOSE collaboration and the two INCOSE-SSWG projects at the annual trustee's luncheon, the annual meeting of the ISSS Council, and the planning session for the next two ISSS annual conferences in San Jose, California (US) in 2012, and Vietnam (2013).

En route to the ISSS Conference, Troncale also presented two talks to the Systems Centre, Faculty of Engineering University of Bristol (UK): both are contributions to the INCOSE-SSWG Projects One and Two. These were titled "Would a Rigorous Knowledge Base in Systems Pathology Add to the Systems Engineering Portfolio?" and "The Need for A Unified Systems Science to Inform Systems Engineering." Both presentations included discussions with regional industrial representatives.

Altogether, INCOSE representatives contributed to 12 different ISSS Conference events. The abstracts, presentation slides, and posters will be uploaded to the INCOSE-SSWG project sites at http://groups.google.com/group/syssciwg/

INCOSE-ISSS participants recognized three shortcomings to remedy next time: (1) too little time was allowed to discuss proposals and projects in sufficient depth, (2) posters need to have scheduled times for conference participants to visit and discuss, and (3) contact information for attendees at collaborative events must be collected and used for follow-up. The first shortcoming may be remedied, in part, by the SSWG workshops scheduled for INCOSE's 2012 International Workshop in Jacksonville, Florida (US).

Seven new professionals joined the INCOSE-SSWG projects cited above in addition to several new graduate students. Four ISSS members pledged to revive and expand activities of the ISSS Special Integration Group on Research toward a General Theory of Systems and this will make direct contributions to Project One. A group of ISSS and INCOSE members intend to found a new professional society called the International Society for Systems Pathology that would directly contribute to INCOSE-SSWG Project Two.

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

Call for Papers

The *Systems Engineering* journal is intended to be a primary source of multidisciplinary information for the systems engineering and management of products and services, and processes of all types. Systems engineering activities involve the technologies and system management approaches needed for

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

Systems Engineering is the archival journal of, and exists to serve the following objectives of, the International Council on Systems Engineering (INCOSE):

- To provide a focal point for dissemination of systems engineering knowledge
- To promote collaboration in systems engineering education and research
- To encourage and assure establishment of professional standards for integrity in the practice of systems engineering
- To improve the professional status of all those engaged in the practice of systems engineering
- To encourage governmental and industrial support for research and educational programs that will improve the systems engineering process and its practice

The journal supports these goals by providing a continuing, respected publication of peer-reviewed results from research and development in the area of systems engineering. Systems engineering is defined broadly in this context as an interdisciplinary approach and means to enable the realization of successful systems that are of high quality, cost-effective, and trustworthy in meeting customer requirements.

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

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

The journal will accept and review submissions in English from any author, in any global locality, whether or not the author is an INCOSE member. A body of international peers will review all submissions, and the reviewers will suggest potential revisions to the author, with the intent to achieve published papers that

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

Editorial selection of works for publication will be made based on content, without regard to the stature of the authors. Selections will include a wide variety of international works, recognizing and supporting the essential breadth and universality of the field. Final selection of papers for publication, and the form of publication, shall rest with the editor.

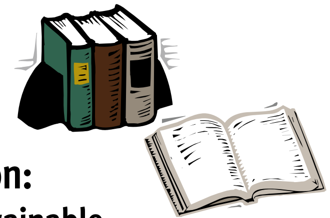
Submission of quality papers for review is strongly encouraged. The review process is estimated to take three months, occasionally longer for hard-copy manuscript. Five copies of your manuscript should be submitted for review purposes to

Professor Andrew P. Sage
Editor in Chief, *Systems Engineering*
School of Information Technology and Engineering
George Mason University
Fairfax, VA 22039-4444, USA

TEL: 703-993-1506
FAX: 703-993-1521
E-MAIL: asage@gmu.edu

Alternatively and preferably, electronic submission of manuscripts for review purposes is strongly encouraged, as this will speed up the review process considerably. Please send a copy of your complete manuscript in a single file with art and tables in approximately the proper location to asage@gmu.edu.

Book Reviews



Beyond the Lean Revolution: Achieving Successful and Sustainable Enterprise Transformation

By Deborah J. Nightingale and Jayakanth Srinivasan
New York, NY (US): Amacom, 2011 (ISBN 978-0-8144-1709-6)
270 pp., including end notes, key terms, and index

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

Good technology management requires a combination of humanities skills with science and engineering skills, and as such it will always be challenging. Management methods that become popular present a simple, implementable strategy for defining and improving the efficiency, effectiveness, competitiveness, and financial status of companies, usually but not always honing the recipe via a number of successful case studies by a management consultant.

Henry Ford implemented Frederick Taylor's (1911) "scientific management" theory on automobile assembly lines a hundred years ago. Since then the management conundrum between focusing on people and on technology has been addressed from both sides. In the 1950s Abraham Maslow proposed that managers should identify which incentives would meet the needs of individual employees at different levels in a hierarchy of needs. Peter Drucker, a management guru for over sixty years, infuriated his own manager when he focused on social analysis starting in 1943. Of the more recent theories, some have highlighted more quantitative methods and tools (such as operations research, reengineering, balanced scorecard, capability maturity, Six Sigma, and lean) and some focused more on the people (management by objectives, total quality management, and agile).

In general there is a great deal of commonality among these methods, including advice to address people issues as well as technical issues, and to understand before prescribing. Given the plenitude of such literature, then, why this new book? In particular, since *Beyond the Lean Revolution*'s first author coauthored a book describing the "lean revolution" only nine years ago, why is it now time to move beyond lean? The authors explain: "Traditional lean tools [. . .] tend to be applied in a rather prescriptive, cookie-cutter manner [. . .]. Their scope is limited [. . .]. Only when the company adopted a holistic view, did an end-to-end analysis of all its processes, and mapped its value stream did it see the possibilities not for incremental change alone, but for enterprise transformation" (6–7). In other words, the problem lies in how the method is applied.

This book integrates well-known and important theory from a variety of movements and into a roadmap that avoids a "prescriptive, cookie-cutter" approach. Their "seven principles of enterprise transformation" and their four "lenses" combine elements of the following:

- Systems thinking (Checkland 1993): Adopt a holistic approach.
- Process improvement (Sheard 2001): Ensure senior leadership commitment.
- Enterprise architecture (Spewak and Hill 1993): Assess the enterprise's current state and create a plan for transformation, process architecture lens, and the alignment integrative lens.
- Lean (Murman et al. 2002): Analyze stakeholder values and waste lens.
- Vision- and mission-based management theory (Peters 1987): Develop a future vision.
- Measurement (ISO and IEC 2007): Performance measurement lens
- Capability maturity: The maturity integrative lens
- Systems engineering: Stakeholder lens is needs elicitation plus stakeholder value analysis.

The book discusses the theory and provides detail in the related case studies. For purposes of clarity, each chapter concludes with "takeaways." The "Enterprise Transformation Roadmap" and chapter 12 ("Enterprise Transformation from Inception to Implementation") show steps to transform an enterprise, including how the strategic, planning, and execution cycles interact.

Some of this material is useful, but little of it is new. The authors specify their intended audience as "anyone interested in transforming an enterprise, whether you are a senior leader or middle manager" (10). This is a broad and weak definition. Most systems engineers, project managers, or even middle managers reading the book will find that they are already aware of most of what the book espouses that is in their control. The newer integration advice can only work if the executives are reading and implementing it.

» continues on next page

Book Review *continued*

Moreover, while any systems engineer benefits from understanding what theories their stakeholders are using (and executives are clearly stakeholders), the book does not address systems engineering per se—it's not even in the index. Which leads to my final question: As a systems engineer, why should I read this book?

References

Checkland, P. 1993. *Systems Thinking, Systems Practice*. Chichester, UK: Wiley.

ISO and IEC. 2007. ISO/IEC 15939. *Systems and Software Engineering: Measurement Process*.

Murman, E., T. Allen, K. Bozdogan, et al. 2002. *Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative*. New York, NY (US): Palgrave.

Peters, T. J. 1987. *Thriving on Chaos: Handbook for a Management Revolution*. New York, NY (US): Knopf.

Sheard, S. A. 2001. "What Is Senior Management Commitment?" Paper presented at the Eleventh Annual International Symposium of INCOSE, Melbourne, Australia, 1–5 July.

Spewak, S. H., and S. C. Hill. 1993. *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology*. Boston, MA (US): QED.

Final Thoughts

From the Chief Editor

Bob Kenley, insight@incose.org

The plenary speakers at the International Symposium in June addressed the challenges of several sociotechnical systems, and all of them mentioned that INCOSE and its members have much to offer toward understanding and managing these challenges. We systems engineers might have much to offer, but those who allocate private and public resources generally do not come from our ranks. In this issue of ***INSIGHT***, INCOSE Fellow Sarah Sheard reviews a book coauthored by INCOSE member Deborah Nightingale (*Beyond the Lean Revolution: Achieving Successful and Sustainable Enterprise Transformation*) that is aimed at senior leaders who do make these allocation decisions. The book describes various tools and techniques known to most system engineers that can be used to help an enterprise to transform itself and cope with its challenges. In our next issue of ***INSIGHT***, we will publish articles about other books written INCOSE authors who have responded to an online survey prepared by theme editor Cecilia Haskins. Whether the books are aimed at senior leaders who sponsor our work or aimed at system engineers, the articles should be compelling reading about the authors' books and other topics covered in the survey.

Rick Dove is in the midst of collecting contributions so that we can all have a close encounter with systems "of the third kind." In addition to his duties as theme editor for his upcoming theme section for ***INSIGHT***, Rick joined with other previous theme editors to offer some tips and tricks future theme editors, which I have compiled and will be sharing with "rookie" theme editors like Biomedical Working Group cochair Melissa Masters. She is requesting that you send her abstracts for the theme issue on health care to be reviewed by the working group at the upcoming International Workshop.

Abe Raher has agreed to be the theme-section editor for coverage of the upcoming International Symposium in Rome, Italy. Abe is a technical writer from Silicon Valley and will be attending his third symposium.

*Upcoming submission deadlines and themes for **INSIGHT***

Issue	Submission Date for General Articles	Theme	Theme Editors
1st Qtr 2012	15 Feb 2012	INCOSE Authors	Cecilia Haskins
2nd Qtr 2012	15 May 2012	Systems of the Third Kind	Rick Dove
3rd Qtr 2012	26 Jul 2012	2012 International Symposium Coverage: Rome, Italy	Abe Raher
4th Qtr 2012	15 October 2012	Health Care	Melissa Masters

Just One More Thing

Andrew Cashner, assistant editor of *INSIGHT*, andrew.cashner@incose.org

We always welcome submissions to ***INSIGHT***, especially from those who have never contributed. A complete style guide and template for authors is available on our website at http://incose.org/ProductsPubs/periodicals/editorialguidelines.aspx. As a helpful aid for those wishing to submit articles, the following is a brief summary of our requirements for document formatting.

File format: Microsoft Word (.docx preferred, or .doc)

File name: Author's last name + First few words of title
E.g., Cashner-Just-One-More-Thing.docx

Layout: One-inch margins on all sides

Font and style: Times New Roman, 12-point font (for *entire* document)

Paragraph: Single space, No space before or after paragraphs,
Left alignment
First line hanging indent .5 inches

Headings: Aligned to the left margin and set in bold

Captions: Aligned to the left margin and set in italics

Lists: Use Word's automatic list feature to create simple bulleted or numbered lists (Likewise for tables and other features: but keep it simple)

Citations: ALL citations follow author-date format according to *The Chicago Manual of Style*, 16th edition (Chicago, IL [US]: University of Chicago Press, 2010).
Author, date, and page number are cited in text in parentheses.
Full information is given in reference list, organized alphabetically by author's last name and date. See "***INSIGHT*** Citation Guide" on our website for examples.

Things to avoid: Alternative citation formats such as numbered references or footnote citations.
Multiple fonts, sizes, colors; special fields, elaborate tables.
Excessive usage of abbreviations (in particular, always spell out *systems engineering*).

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