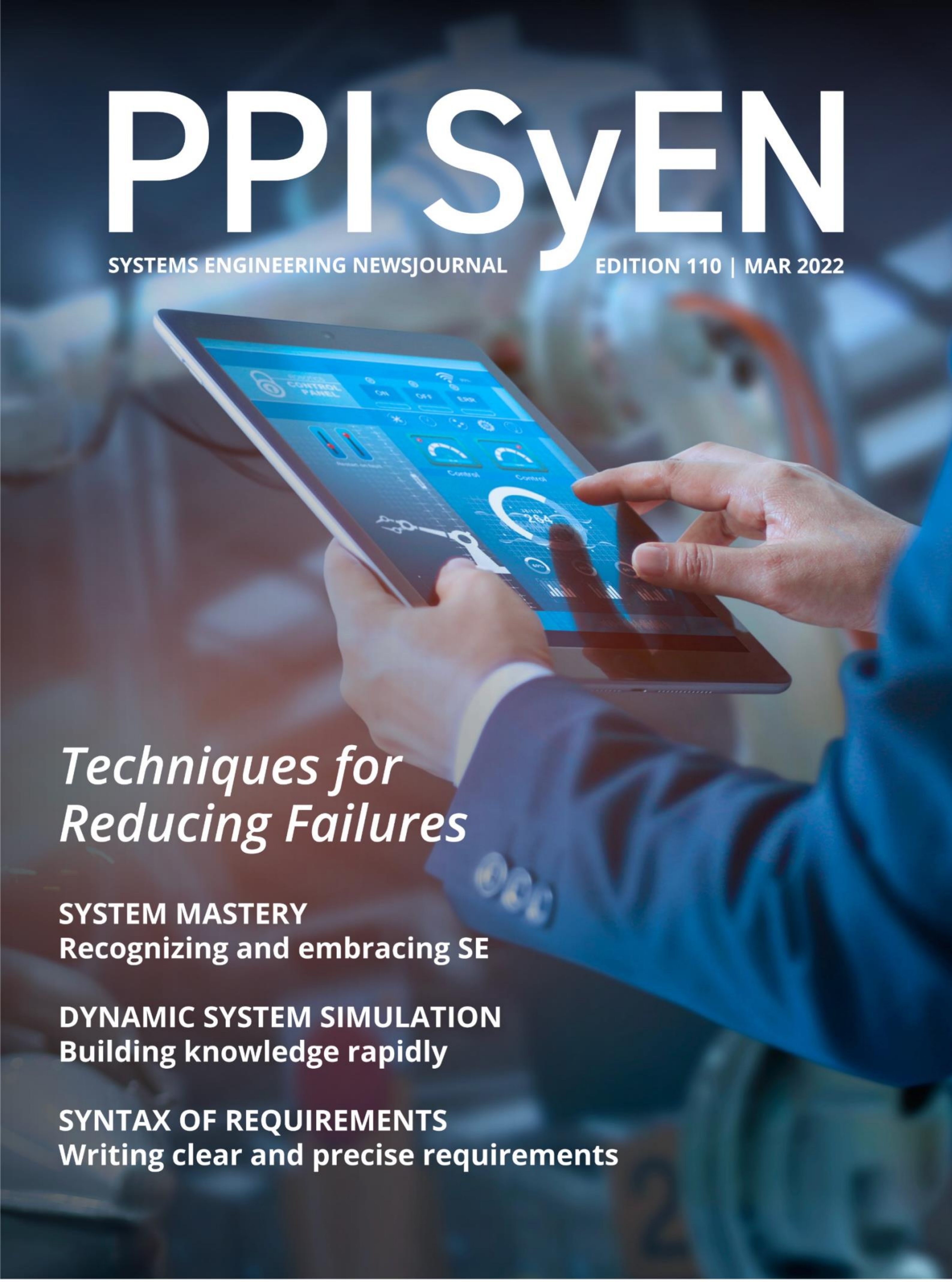
PPI SyEN

SYSTEMS ENGINEERING NEWSJOURNAL

EDITION 110 | MAR 2022



Techniques for Reducing Failures

SYSTEM MASTERY
Recognizing and embracing SE

DYNAMIC SYSTEM SIMULATION
Building knowledge rapidly

SYNTAX OF REQUIREMENTS
Writing clear and precise requirements

A PROJECT PERFORMANCE INTERNATIONAL PUBLICATION | PPI-INT.COM

PPI SyEN

EMAIL: PPISyEN@PPI-Int.com

EDITORIAL STAFF

Editor
John Fitch

Editor-in-Chief
Robert Halligan

Managing Editor
René King

PRODUCTION STAFF

Marketing Manager
Benjamin Bryant

Graphic Designer
Matthew Wong

Marketing Coordinator
Rebeca Carneiro

PUBLISHER

PROJECT PERFORMANCE INTERNATIONAL

Project Performance International
2 Parkgate Drive
Ringwood, Vic 3134 Australia
Tel: +61 3 9876 7345

Tel Brazil: +55 12 9 9780 3490
Tel UK: +44 20 3608 6754
Tel USA: +1 888 772 5174
Tel China: +86 188 5117 2867

www.ppi-int.com

PPI SyEN (PPI Systems Engineering Newsjournal) is published monthly.

Archived editions and subscriptions to future editions are available for free at: https://www.ppi-int.com/syen-newsjournal/

WELCOME

Hello friends of PPI,

Welcome to this March edition of PPI SyEN. This Newsjournal is the 110th edition of the series, and I have to say that this is one of my favorites from the last few years that I have served as Managing Editor. A special thanks to our Editor, John Fitch, who does an incredible job curating content and liaising with authors every month to ensure their ideas are captured best and represented in PPI SyEN.

Right, so what do you have in store for this edition? The theme of this edition is 'Techniques for Reducing Failures.' In this edition, we'll cover the usual ground of providing updates in the systems engineering world, including news on recently established INCOSE Chapters and developments in the space of digital engineering. In terms of conferences, there are some exciting events related to SE in the service industry, cyber resilience, and space engineering.

It's the articles of this edition, in particular, that you won't want to miss. This month we have a fascinating paper by Randall Iliff titled, 'Understanding the Journey to (and from) System Mastery.' Randall Iliff uses the N-2 diagram innovatively to illustrate why we all need to be system literate as complexity compounds exceptionally quickly with just a handful of elements. Next, Erik Åberg takes stock of Technology and Trends in Systems Modeling and Simulation'. Erik explores the potential of building models using Modelica and FMI and how modeling and simulation are helpful throughout the life cycle – not just the development stage. Finally, for the articles, Robert Halligan writes on 'Requirements Writing Patterns– what are the options? Here, Robert describes his Parsing Template to write strong requirements in English, and summarizes alternatives. All three articles bring something very special to the table, be sure to read the knowledge shared by these three authors.

As usual, we conclude with some valuable resources in the systems engineering field. This month we focus on the MBSE special edition of the INCOSE Systems Engineering Journal and videos and tools that you can access to improve your engineering techniques. Finally, we close off this month's edition with insight into how versatile the Functional Flow Block Diagram is by illustrating how the FFBD can be used to model the tuning of a violin. Hopefully, I've made a case for why this edition is one of my favorites.

Regardless of what type or aspect of engineering you focus on in your daily work, I guarantee there is something of interest to you as an engineer in this edition.

Regards,

René

Managing Editor, PPI SyEN

Views expressed in externally authored articles are not necessarily the views of PPI nor of its professional staff.

PPI Systems Engineering Newsjournal (PPI SyEN) seeks:

- To advance the practice and perceived value of systems engineering across a broad range of activities, responsibilities, and job-descriptions
- To influence the field of systems engineering from an independent perspective
- To provide information, tools, techniques, and other value to a wide spectrum of practitioners, from the experienced, to the newcomer, to the curious
- To emphasize that systems engineering exists within the context of (and should be contributory toward) larger social/enterprise systems, not just an end within itself
- To give back to the Systems Engineering community

PPI defines systems engineering as:

an approach to the engineering of systems, based on systems thinking, that aims to transform a need for a solution into an actual solution that meets imperatives and maximizes effectiveness on a whole-of-life basis, in accordance with the values of the stakeholders whom the solution is to serve. Systems engineering embraces both technical and management dimensions of problem definition and problem solving.

SYSTEMS ENGINEERING NEWS

Recent events and updates in the field of systems engineering

INCOSE New Zealand Chapter Charter Approved

The INCOSE Board of Directors (BOD) has approved the charter for a New Zealand chapter. *INCOSE New Zealand Incorporated* is a New Zealand Incorporated Society under the Incorporated Societies Act. INCOSE New Zealand operates as a chartered chapter of the International Council on Systems Engineering (INCOSE) with a goal of supporting and expanding the practice of systems engineering in New Zealand and Oceania.

Congratulations to the interim leadership of INCOSE(NZ) including Christian Parra-John, Jess Tucker, Steve Wallace, Tom McKay and John Welford.

Check out the chapter website here. Subscribe to the mailing list for notifications and updates on the chapter's progress.

PDMA Announces Allan Anderson Ambassador Award

The Product Development & Management Association (PDMA) is proud to announce the creation of the Allan Anderson Ambassador Award (AAAA). This award honors Dr. Allan Anderson, Professor Emeritus of Massey University in New Zealand, Chief Executive of NZ Dairy Research Institute, successful entrepreneur, and lifelong advocate of sustainable product innovation practices. Dr. Anderson served in many leadership roles at PDMA, in particular guiding the fruitful partnership between PDMA and China that has led to the certification of over 10,000 New Product Development Professionals (NPDPs) in recent years.

The announcement of this award coincides with its first presentation to the award's namesake, Allan Anderson.

Nominees for the AAAA are accepted from any member of the PDMA community and are reviewed first by the Executive Committee of PDMA and then later voted on by the Board of Directors of PDMA. To receive the award, the nominee must be shown to have contributed to the advancement or the global expansion of the discipline of Product Innovation in a foundational way.

Additional details here

More News from Zuken Vitech

Zuken (parent company of Vitech) announces the appointment of Enrique Krajmalnik as CEO of US-based Zuken Vitech Inc., formerly, Vitech Corporation, which was acquired by Zuken in 2019. Krajmalnik has more than 25 years of experience in senior management and strategic planning, including serving as Chief Technology Officer for No Magic, Inc.

Speaking about his appointment as CEO of Vitech, Mr. Krajmalnik said, "Vitech, as part of the global family of Zuken companies, is uniquely positioned to bring innovative MBSE solutions to a growing market. Making MBSE more accessible to the broader market will be key to our long-term success."

Zuken, citing the unpredictability of the Covid situation in Europe, also announced that its Zuken Innovation World (ZIW) Europe will be conducted virtually in 2022 as a series of compact and interactive locally-organized web conferences:

- ZIW Digital - German Language Edition: 3-4 May, 2022
- ZIW Digital - Italian Language Edition: 13-14 June, 2022
- ZIW Digital - French Language Edition: 15-16 June, 2022
- ZIW Digital - English Language Edition: 21-22 June, 2022

Zuken Innovation World Americas will be conducted as an in-person conference on 6-9 June in San Antonio, Texas, USA in conjunction with Vitech's Integrate22 Digital Engineering conference. Register here.

IncQuery Group Releases the IncQuery Suite Digital Engineering Platform

The future depends on great engineering. And great engineering depends not only on great tools, but also on novel ways to foster their interworking in a world of increasing project complexity and interdisciplinary thinking.

The IncQuery Group envisions an open engineering paradigm, fostering collaboration, efficiency, high product quality and a holistic perspective.

There are a number of model-based conceptual frameworks, standards and languages (SysML, SysMod, Arcadia, Zachman Framework, DoDAF, TOGAF, ...) available, addressing various facets of digital engineering. Currently, however, most adopters of such frameworks struggle to achieve toolchains with a desirable level of openness and accessibility - most of them ending up in their own, closed customization efforts and their engineering data still being trapped in technological silos.

This is where the IncQuery Suite steps in. It aims at bringing about a paradigm shift in digital engineering - not unlike the fundamental changes that traditional software engineering underwent in the recent past. To that end, IncQuery Suite introduces novel, disruptive technologies in three key areas: quality management, automation and integration.

The IncQuery Group believes that the problems that actually hamper the industry lie within the tool landscape itself. When facing incompatibilities coming from legacy software, the IncQuery Suite comes equipped with a number of customizable connectors. Against the negative consequences of vendor lock-in and the general lack of support for multi-vendor scenarios, the IncQuery Suite creates a unified engineering knowledge graph. For a better integration of the engineering workflow with the existing enterprise infrastructure, the IncQuery Group offers its comprehensive digital engineering dev-ops approach. How to achieve and maintain better quality throughout the product lifecycle, while not increasing, or maybe even reducing engineering costs? The IncQuery Suite answer is, first of all, to provide an accessible, unified data representation - and one that scales both horizontally and vertically.

Its tool-specific and context-specific connectors and bridges offer seamless integration with many of the most widespread tools and technologies, while an OpenAPI-based interface design meets expectations for accessibility and extensibility.

Once users have all their needed data at hand, the real quality management endeavor can begin: IncQuery Suite's unique, customizable analysis facility can be employed on any abstraction level, depending on user needs: from simple conformance checks to complex queries, or even consistency validations, providing a well-founded answer to the central challenges of digital engineering: correctness, completeness and consistency of models and datasets.

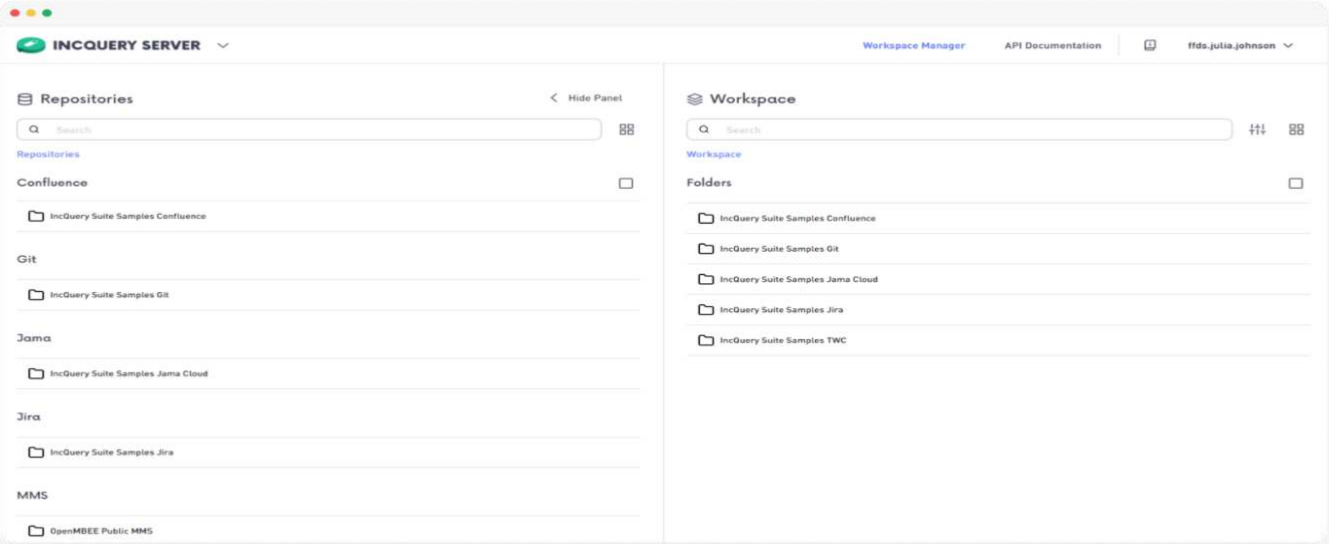
As for automation, the IncQuery Suite is not meant to replace any part of an established toolchain: it is there to complement and expand on what can be achieved with such a toolchain. Essentially, the IncQuery Suite offers a blend of state-of-the-art digital engineering and dev-ops techniques aimed at reducing repetitive manual work, standardizing and automating workflows, fostering predictability and continuous learning.

The aim is for proven techniques and solutions used by software developers to become available in a digital engineering context, be it continuous integration and delivery, test case management, or integration with Git, Jira, Confluence and many more.

The accessibility and integration philosophy of the IncQuery Suite makes digital engineering available to a broader audience than ever seen before: not only established industrial players, but also innovative SMEs, students and educators, hobby practitioners and many other stakeholders who can benefit from interacting with the IncQuery Suite.

The comprehensive, multi-faceted IncQuery Suite consists of 6 distinct, interconnected solutions, each of them contributing to the vision in a particular way.

The Server is the heart of the infrastructure: it helps break the data silos and turn separated data fragments into an integrated engineering knowledge graph - the IncQuery digital thread. Thus, it helps users in breaking the silos and vendor lock-in, and opening up their enterprise to the next level of engineering quality management.



Repositories

Repositories

Confluence

- IncQuery Suite Samples Confluence

Git

- IncQuery Suite Samples Git

Jama

- IncQuery Suite Samples Jama Cloud

Jira

- IncQuery Suite Samples Jira

MMS

- OpenMBEE Public MMS

Workspace

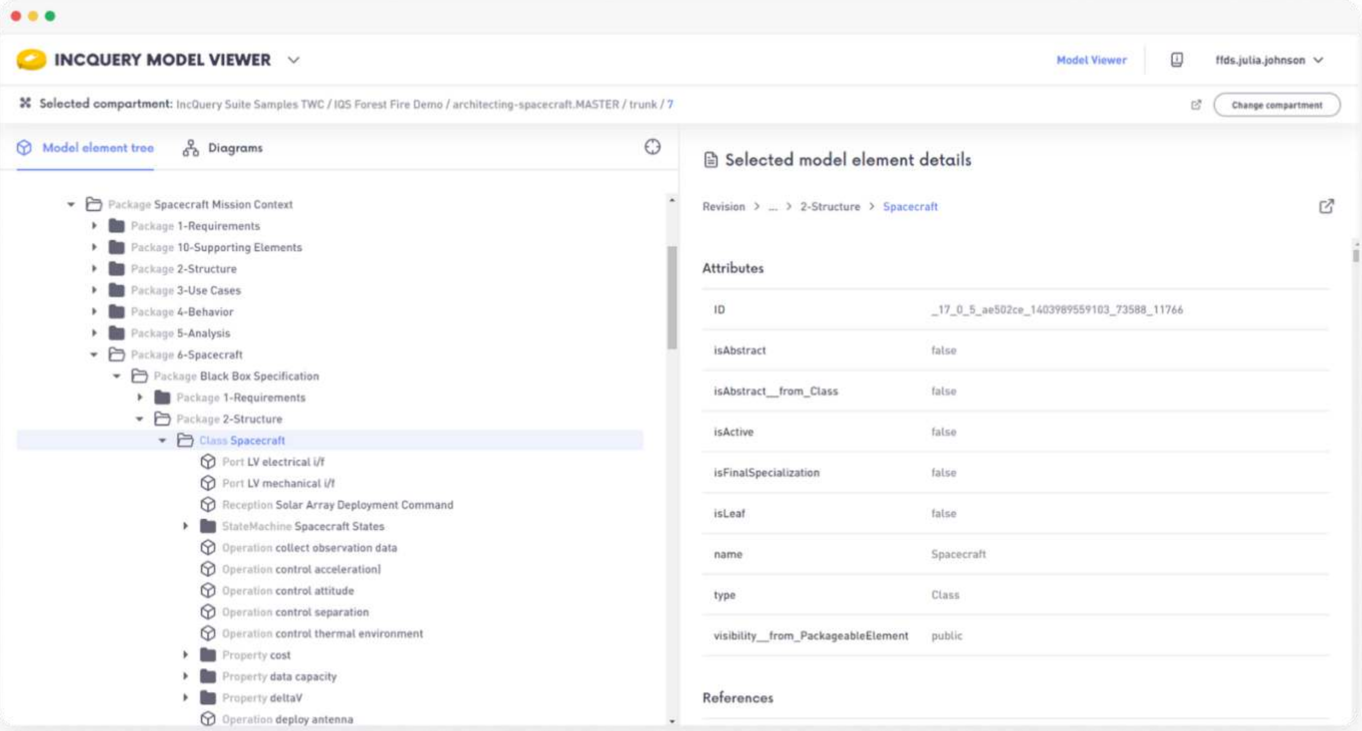
Workspace

Folders

- IncQuery Suite Samples Confluence
- IncQuery Suite Samples Git
- IncQuery Suite Samples Jama Cloud
- IncQuery Suite Samples Jira
- IncQuery Suite Samples TWC

Figure 1: IncQuery Server

The Viewer is an easy-to-use tool for high-level interaction with a digital thread, offering model inspection, change tracking and simple search features directly from a browser. Its design plays a large role in contributing to the increased accessibility of engineering data that is enabled by the IncQuery Suite.



Selected compartment: IncQuery Suite Samples TWC / IQS Forest Fire Demo / architecting-spacecraft.MASTER / trunk / 7

- Package Spacecraft Mission Context
  - Package 1-Requirements
  - Package 10-Supporting Elements
  - Package 2-Structure
  - Package 3-Use Cases
  - Package 4-Behavior
  - Package 5-Analysis
  - Package 6-Spacecraft
    - Package Black Box Specification
      - Package 1-Requirements
      - Package 2-Structure
        - Class Spacecraft
          - Port LV electrical i/f
          - Port LV mechanical i/f
          - Reception Solar Array Deployment Command
          - StateMachine Spacecraft States
          - Operation collect observation data
          - Operation control acceleration]
          - Operation control attitude
          - Operation control separation
          - Operation control thermal environment
          - Property cost
          - Property data capacity
          - Property deltaV
          - Operation deploy antenna

Selected model element details

Revision > ... > 2-Structure > Spacecraft

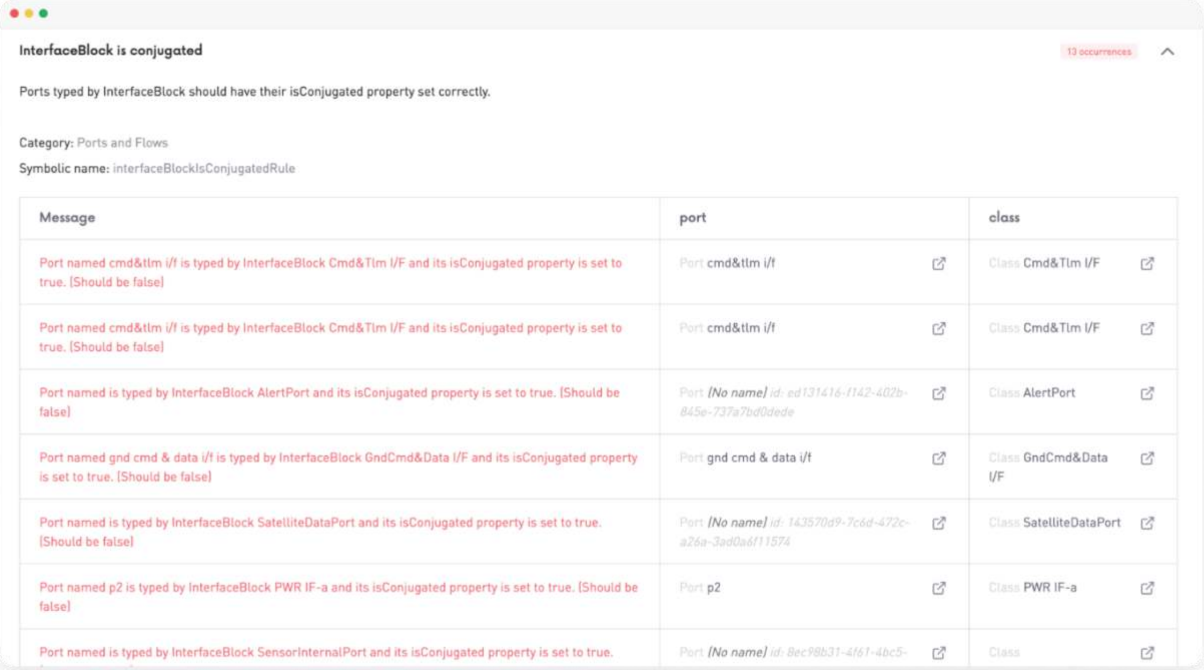
Attributes

ID	_17_0_5_ae502ce_1403989559103_73588_11766
isAbstract	false
isAbstract__from_Class	false
isActive	false
isFinalSpecialization	false
isLeaf	false
name	Spacecraft
type	Class
visibility__from_PackageableElement	public

References

Figure 2: IncQuery Viewer

The Validator is a key ingredient for validation workflows: it comes with built-in, automated validation and conformance verification features for SysML and its dialects, AUTOSAR and many more; as well as a smooth integration into a dev-ops infrastructure, allowing for engineers to get direct and comprehensible feedback on the quality of what they do and how they are doing it.



InterfaceBlock is conjugated

13 occurrences

Ports typed by InterfaceBlock should have their isConjugated property set correctly.

Category: Ports and Flows

Symbolic name: interfaceBlockIsConjugatedRule

Message	port	class
Port named cmd&tlm i/f is typed by InterfaceBlock Cmd&Tlm I/F and its isConjugated property is set to true. [Should be false]	Port cmd&tlm i/f	Class Cmd&Tlm I/F
Port named cmd&tlm i/f is typed by InterfaceBlock Cmd&Tlm I/F and its isConjugated property is set to true. [Should be false]	Port cmd&tlm i/f	Class Cmd&Tlm I/F
Port named is typed by InterfaceBlock AlertPort and its isConjugated property is set to true. [Should be false]	Port *[No name]* id: ed131416-f142-402b-845e-737a7bd0dede	Class AlertPort
Port named gnd cmd & data i/f is typed by InterfaceBlock GndCmd&Data I/F and its isConjugated property is set to true. [Should be false]	Port gnd cmd & data i/f	Class GndCmd&Data I/F
Port named is typed by InterfaceBlock SatelliteDataPort and its isConjugated property is set to true. [Should be false]	Port *[No name]* id: 143570d9-7c6d-472c-a26a-3ad0a6f11574	Class SatelliteDataPort
Port named p2 is typed by InterfaceBlock PWR IF-a and its isConjugated property is set to true. [Should be false]	Port p2	Class PWR IF-a
Port named is typed by InterfaceBlock SensorInternalPort and its isConjugated property is set to true.	Port *[No name]* id: 8ec98b31-4f61-4bc5-	Class

Figure 3: IncQuery Validator

The Desktop is a bridge between established desktop engineering tools and IncQuery Suite's validation features: the same features become available through add-ins for the most popular modeling tools. The Desktop component highlights the complementary nature of IncQuery Suite to existing, established toolchains.

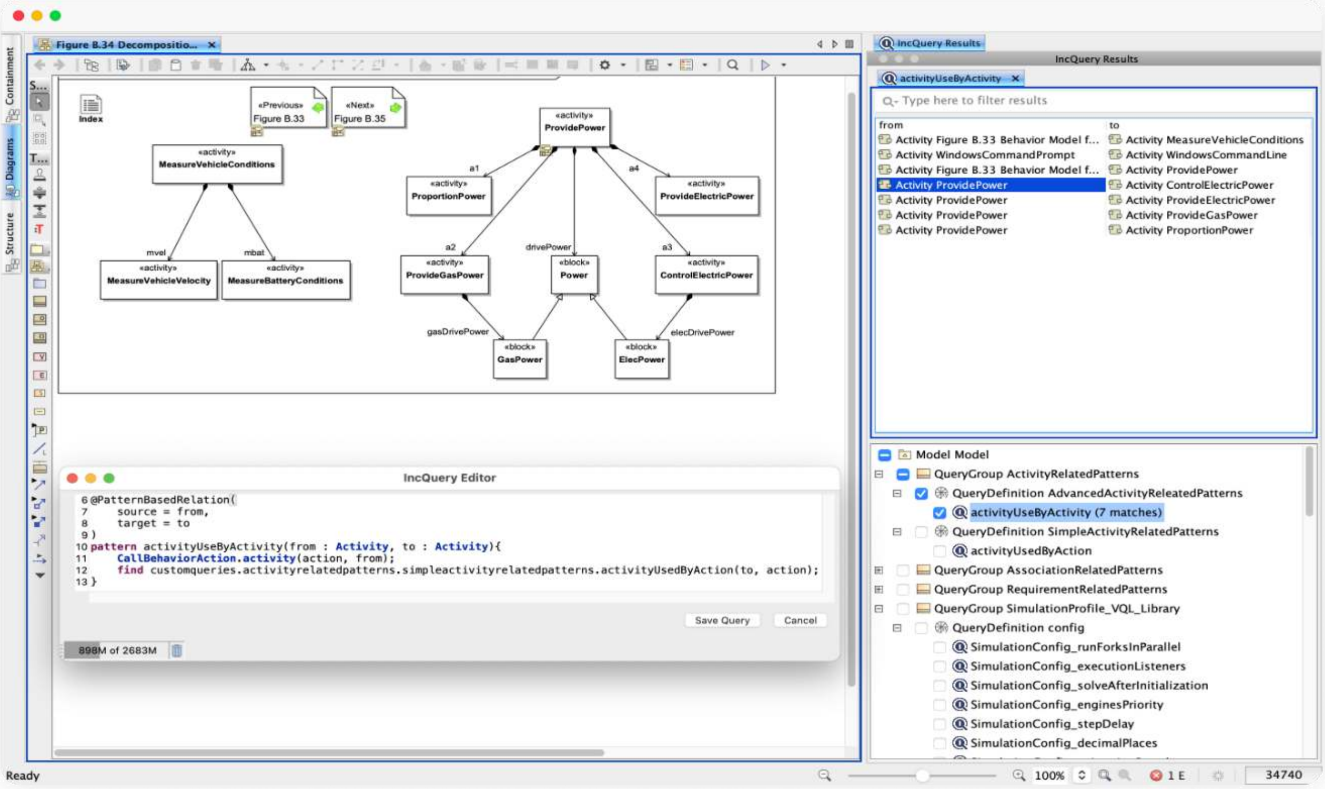


Figure B.34 Decompositio...
Containment
Diagrams
Structure
Index
«Previous» Figure B.33
«Next» Figure B.35
«activity» ProvidePower
«activity» MeasureVehicleConditions
a1
a4
«activity» ProportionPower
«activity» ProvideElectricPower
mvel
mbat
a2
drivePower
a3
«activity» MeasureVehicleVelocity
«activity» MeasureBatteryConditions
«activity» ProvideGasPower
«block» Power
«activity» ControlElectricPower
gasDrivePower
elecDrivePower
«block» GasPower
«block» ElecPower
IncQuery Results
activityUseByActivity
Type here to filter results
Ready
100%
1 E
34740

IncQuery Editor

```
@PatternBasedRelation(
    source = from,
    target = to
)
pattern activityUseByActivity(from : Activity, to : Activity){
    CallBehaviorAction.activity(action, from);
    find customqueries.activityrelatedpatterns.simpleactivityrelatedpatterns.activityUsedByAction(to, action);
}
```

Save Query Cancel

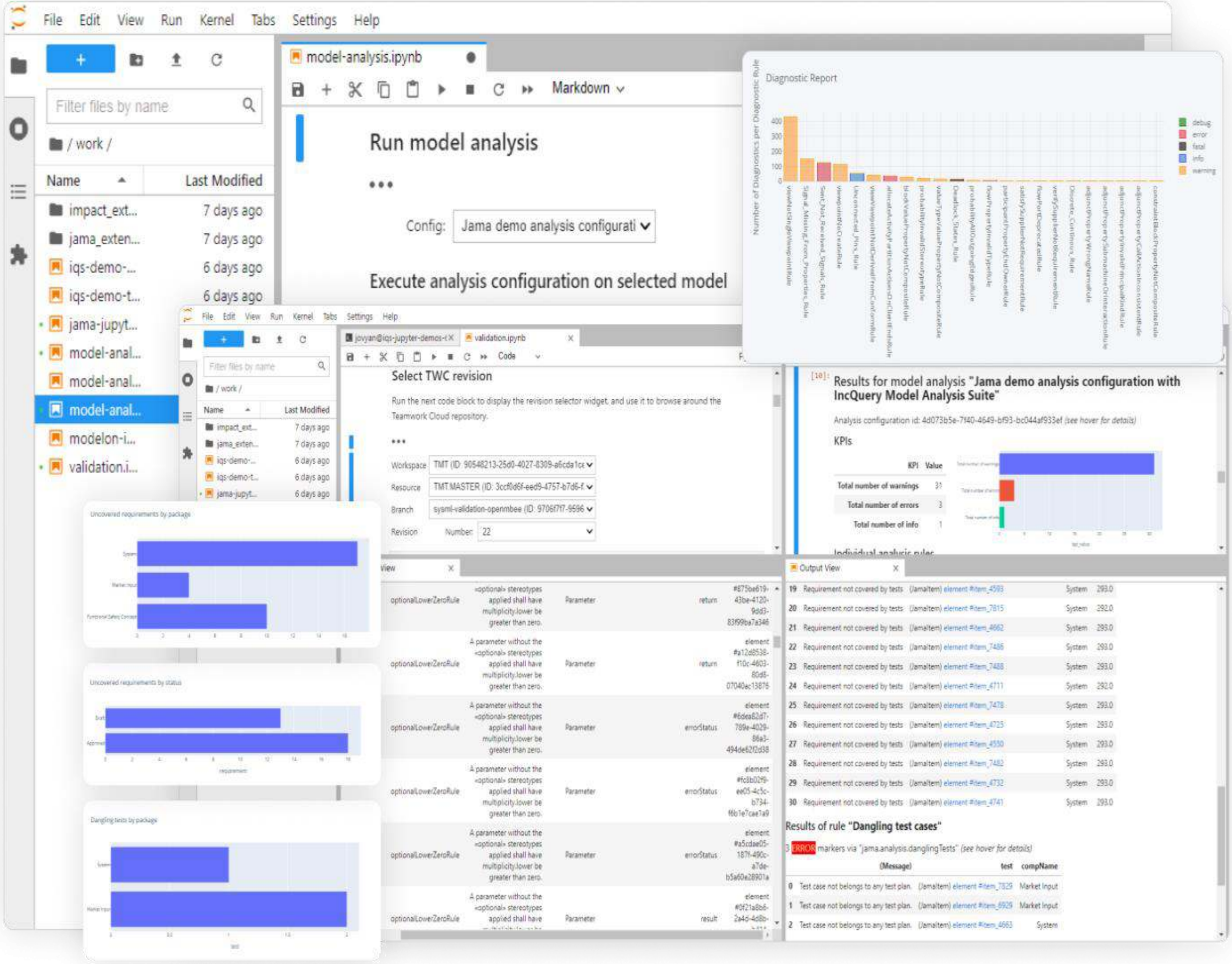
898M of 2683M

from	to
Activity Figure B.33 Behavior Model f...	Activity MeasureVehicleConditions
Activity WindowsCommandPrompt	Activity WindowsCommandLine
Activity Figure B.33 Behavior Model f...	Activity ProvidePower
Activity ProvidePower	Activity ControlElectricPower
Activity ProvidePower	Activity ProvideElectricPower
Activity ProvidePower	Activity ProvideGasPower
Activity ProvidePower	Activity ProportionPower

Model Model
- QueryGroup ActivityRelatedPatterns
  - ☑ QueryDefinition AdvancedActivityReleatedPatterns
    - ☑ activityUseByActivity (7 matches)
  - □ QueryDefinition SimpleActivityRelatedPatterns
    - □ activityUsedByAction
- □ QueryGroup AssociationRelatedPatterns
- □ QueryGroup RequirementRelatedPatterns
- □ QueryGroup SimulationProfile_VQL_Library
  - □ QueryDefinition config
    - □ SimulationConfig_runForksInParallel
    - □ SimulationConfig_executionListeners
    - □ SimulationConfig_solveAfterInitialization
    - □ SimulationConfig_enginesPriority
    - □ SimulationConfig_stepDelay
    - □ SimulationConfig_decimalPlaces

Figure 4: IncQuery Desktop

The Analyzer is a flexible and customizable extension of the Validator, allowing users to build their own analysis scenarios, ranging from simple validation goals to multi-domain consistency checks and complex data retrieval queries, based on the established and easy-to-use Jupyter environment. In whatever flavor you are facing the three main analysis challenges - correctness, completeness, consistency - the Analyzer.



Name	Last Modified
impact_ext...	7 days ago
jama_exten...	7 days ago
iqs-demo-...	6 days ago
iqs-demo-t...	6 days ago
jama-jupyt...	
model-anal...	
model-anal...	
model-anal...	
modelon-i...	
validation.i...	

Run model analysis

Config: Jama demo analysis configurati

Execute analysis configuration on selected model

Select TWC revision

Run the next code block to display the revision selector widget, and use it to browse around the Teamwork Cloud repository.

Workspace: TMT (ID: 90548213-25d0-4027-8309-a6cda1ce

Resource: TMT.MASTER (ID: 3ccf0d6f-eed9-4757-b7d6-f

Branch: sysml-validation-openmbee (ID: 9706f7f7-9596

Revision Number: 22

optionalLowerZeroRule	«optional» stereotypes applied shall have multiplicity.lower be greater than zero.	Parameter		return	#875be619-43be-4120-9dd3-83f99ba7a346
optionalLowerZeroRule	A parameter without the «optional» stereotypes applied shall have multiplicity.lower be greater than zero.	Parameter		return	element #a12d8538-f10c-4603-80d8-07040ac13876
optionalLowerZeroRule	A parameter without the «optional» stereotypes applied shall have multiplicity.lower be greater than zero.	Parameter		errorStatus	element #6dea82d7-789e-4029-86a3-494de62f2d38
optionalLowerZeroRule	A parameter without the «optional» stereotypes applied shall have multiplicity.lower be greater than zero.	Parameter		errorStatus	element #fc8b02f9-ee05-4c5c-b734-f6b1e7cae1a9
optionalLowerZeroRule	A parameter without the «optional» stereotypes applied shall have multiplicity.lower be greater than zero.	Parameter		errorStatus	element #a5cdae05-187f-490c-a7de-b5a60e28901a
optionalLowerZeroRule	A parameter without the «optional» stereotypes applied shall have	Parameter		result	element #0f21a8b6-2a4d-4d8b-

[10]: Results for model analysis "**Jama demo analysis configuration with IncQuery Model Analysis Suite**"

Analysis configuration id: 4d073b5e-7140-4649-bf93-bc044af933ef *(see hover for details)*

KPIs

KPI	Value
Total number of warnings	31
Total number of errors	3
Total number of info	1

Individual analysis rules

19	Requirement not covered by tests	(JamaItem) element #item_4593	System	293.0
20	Requirement not covered by tests	(JamaItem) element #item_7815	System	292.0
21	Requirement not covered by tests	(JamaItem) element #item_4662	System	293.0
22	Requirement not covered by tests	(JamaItem) element #item_7486	System	293.0
23	Requirement not covered by tests	(JamaItem) element #item_7488	System	293.0
24	Requirement not covered by tests	(JamaItem) element #item_4711	System	292.0
25	Requirement not covered by tests	(JamaItem) element #item_7478	System	293.0
26	Requirement not covered by tests	(JamaItem) element #item_4725	System	293.0
27	Requirement not covered by tests	(JamaItem) element #item_4550	System	293.0
28	Requirement not covered by tests	(JamaItem) element #item_7482	System	293.0
29	Requirement not covered by tests	(JamaItem) element #item_4732	System	293.0
30	Requirement not covered by tests	(JamaItem) element #item_4741	System	293.0

Results of rule "**Dangling test cases**"

3 ERROR markers via "jama.analysis.danglingTests" *(see hover for details)*

	(Message)	test	compName
0	Test case not belongs to any test plan.	(JamaItem) element #item_7829	Market Input
1	Test case not belongs to any test plan.	(JamaItem) element #item_6929	Market Input
2	Test case not belongs to any test plan.	(JamaItem) element #item_4663	System

Figure 5: IncQuery Analyzer

The Integrator enables efficiently transfer data between all the domains and contexts involved in a digital thread. To that end, it comes with a number of essential bridges (e.g., between SysML and AUTOSAR or SysML and GENESYS) and extension points for scenarios. Many engineers know how tedious and error-prone inter-domain data transfer can be; the bridges offer a scalable, customizable and traceable alternative, relying on a combination of existing bridge implementations and additional services to tailor the Integrator to the use case.

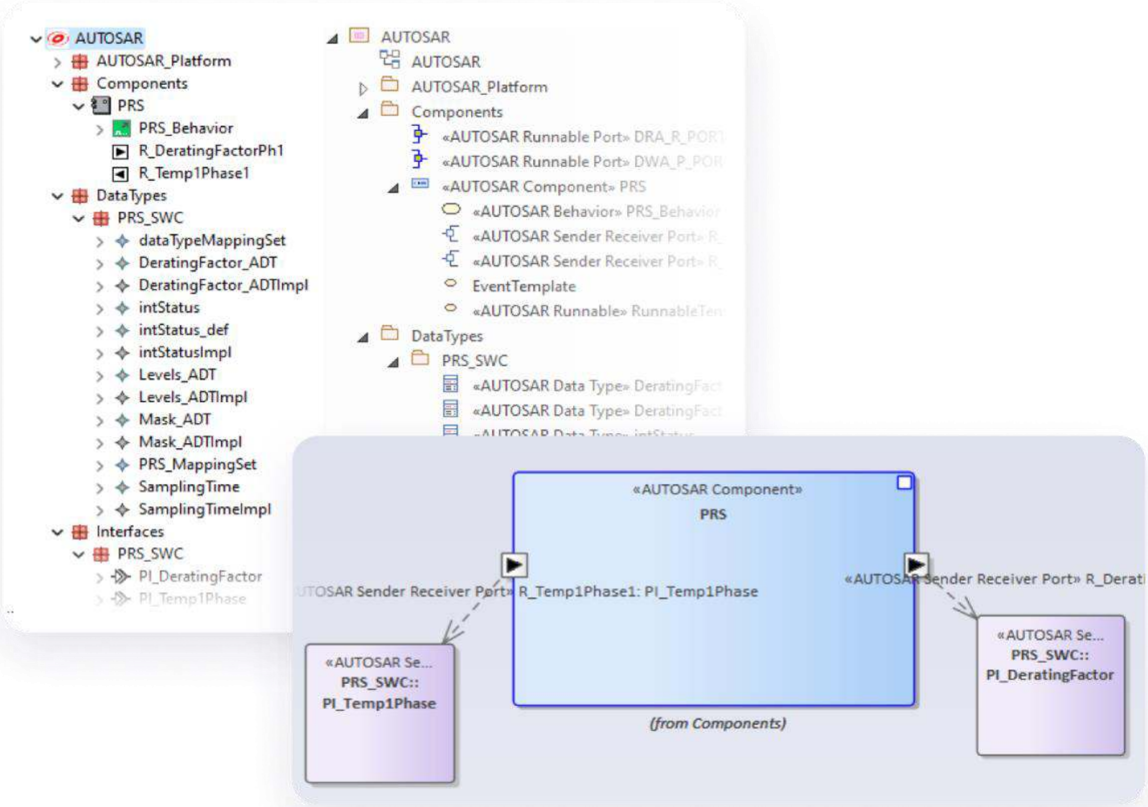


Figure 6: IncQuery Integrator

Follow these links for more details on the IncQuery Suite digital engineering platform.

CONFERENCES, MEETINGS & WEBINARS

IEEE SysCon 2022 Goes Virtual

The 16th Annual IEEE International Systems Conference (SysCon 2022), scheduled for 25-28 April has switched to a virtual conference format because of COVID-19 impacts on international travel. Attendees will receive access to all of the presentations for the duration of the conference through the Virtual Conference Platform.

Registration discounts are available for IEEE and INCOSE members. Register here.

Call for Papers: IEEE Service-Oriented System Engineering Conference

Service-Oriented Computing exploits services as the fundamental elements of computer-based systems. It has been applied to various areas and promotes fundamental changes to the way software systems are being analyzed, architected, designed, implemented, tested, evaluated, delivered, consumed, maintained and evolved. Consequently, a new paradigm of software and system engineering discipline, i.e., Service-Oriented Engineering, has emerged and advanced rapidly in the past decade.

The 16th Annual IEEE International Conference on Service-Oriented System Engineering (IEEE SOSE 2022) is scheduled for 15-18 August in the San Francisco Bay, California, USA area. SOSE 2022, part of the IEEE International Congress On Intelligent And Service-Oriented Systems Engineering, covers all aspects of Service-Oriented Engineering from architectures, techniques, tools and languages to methodologies.

Papers are sought in multiple tracks:

- Advanced Models for Service Engineering
- Computing Environments and Virtualization
- Methods, Languages & Tools for process-based SOC systems
- Service-based Cyber-Physical Systems (CPS)
- Intelligent Services for a Smart World
- Security Engineering for Service Systems
- Practical Experiences

Regular research papers, no more than 10 pages in length, should report on new, unpublished research. Work-in-progress and Vision papers associated with emerging and rapidly changing issues, paradigms, technologies or applications should be limited to 5 pages. Papers that report on industrial practice, case studies or empirical research should be betweeen 6-10 pages in length. All papers will be presented live at the conference.

Deadlines and important dates include:

- Full paper submission: 8 May
- Paper acceptance notification: 8 June
- Author registration: 1 July

See paper submission details.

Call for Papers: IFAC Human-Machine Systems Conference

IFAC
INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL
IFAC HMS 2022 September 12-15 San Jose, USA

The International Federation of Automatic Control (IFAC) invites the submission of papers, posters and session proposals to the the 15th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFAC HMS 2022), to be held at San José State University, September 12-15, 2022.

The objective of the symposium is to provide an international forum for the latest scientific and technological achievements in Human-Machine Systems research across application domains such as intelligent and autonomous systems, autonomous driving, urban air mobility, robotics, virtual reality, and healthcare. The symposium is co-sponsored by:

- International Federation for Information Processing (IFIP)
- International Federation of Operational Research Societies (IFORS)
- International Energy Agency (IEA)

Topic of interest include:

- Design, Analysis, and Evaluation of Human-Machine Systems
- Human Interaction with Artificial-Intelligence Systems
- Human-Machine Teaming
- Human-Machine Interfaces
- Extended Reality
- Human Performance Aspects of Human-Machine Systems
- Ethics of Human-Machine Systems
- Application Domains of Human-Machine Systems
- Human-Machine System Aspects of Computer Systems, Networks and Cybersecurity

Deadlines and important dates include:

- Draft Paper, Posters, and session submission deadline: 25 April
- Invited-session papers submission: 9 May
- Notification of acceptance: 20 May
- Camera-ready papers deadline: 27 June
- Early registration deadline: 3 July

See submission guidelines for details. Submit papers here.

CONFERENCES, MEETINGS & WEBINARS

2022 International System Dynamics Conference: Early-bird Registration and Plenaries

The 2022 International System Dynamics Conference (ISDC2022), scheduled for 18-22 July, has announced its lineup of topics and speakers for plenary sessions and also early-bird registration at reduced rates.

The hybrid conference, with the opportunity for in-person attendance in Frankfurt, Germany, will provide access to the full conference program with either a Frankfurt or Virtual ticket. Register here before 9 June to obtained early-bird pricing. Society members enjoy an additional conference discount. Join here.

Plenary sessions at ISDC2022 are organized into two threads:

Plenary: A 360° View on Energy Transitions
The energy transition is among the most pressing societal challenges. In this plenary, we explore this transition from diverse perspectives.

Merla Kubli, Assistant Professor for Managing Climate Solutions at the University of St. Gallen, Switzerland, and program manager of the novel master certificate Managing Climate Solutions (MaCS-HSG) will speak on ***Co-creating Energy Solutions***.

Josephine Kaviti Musango, Professor at the School of Public Leadership, Stellenbosch University, South Africa will share her insights on ***The Value of Qualitative System Dynamics for Co-learning About Energy Transitions and Gender Mainstreaming***.

Santiago Arango-Aramburo, Professor at Universidad Nacional de Colombia and Fulbright Visiting Professor at MIT will address ***Renewable Energy Sources: Diversity of Impacts, Perspectives and Challenges***.

Plenary: Diversity
This plenary addresses how we can socialize the concept of diversity and make people aware of the path dependency of privilege via teaching. It also addresses how we can create more equitable structures via including perspectives of those who are strongly affected but often least likely to be engaged.

John Sterman, Jay W. Forrester Professor of Management at the MIT Sloan School of Management, director of the MIT System Dynamics Group, and co-director of the MIT Sloan Sustainability Initiative will speak on ***The Dynamics of Privilege***.

Saras Chung is Executive Director at SKIP DesignEd and an Adjunct Faculty for Washington University in St. Louis' Brown School of Social Work and Public Health will share her insights on ***How To Reduce Inequality Using System Dynamics***.

CONFERENCES, MEETINGS & WEBINARS

Webcast: The Design & Innovation Strategy Behind Hyundai's Walking Car Concepts

The Product Development Management Association (PDMA) will host a webinar on 21 April, 2022 that explores "*The Design & Innovation Strategy Behind Hyundai's Walking Car Concepts*." David Byron of Sundberg-Ferar Product Innovation Studio will discuss the innovation techniques and design processes that were used on this Hyundai project.

The webinar will attempt to answer the following questions:

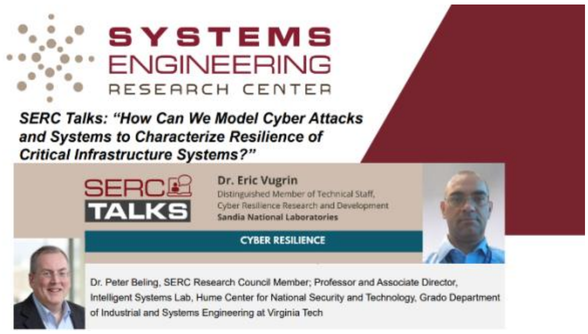
- Where do you start to come up with out-of-the-box ideas?
- What makes a walking car so different and compelling?
- Why does anyone need a car that can walk?
- How strategic research led to the development of compelling use cases?
- How you can use these same methods to create ideal experiences and boost your product portfolio, whether you make walking cars or water purifiers?

This workshop is free for PDMA members; $10 for non-members. See details. Register here (PDMA membership or guest account required). Join PDMA.

SERCTALKS - Cyber Resilience - Modeling Cyber Attacks

The Systems Engineering Research Center (SERC) hosted the first of its Spring 2022 SERC TALKS focusing on the topic of Cyber Resilience. Moderated by Dr. Peter Beling of Virginia Tech, the series hopes to stimulate an ongoing and more collaborative dialogue between academia, government and industry sectors on this important topic.

On 23 February, Dr. Eric Vugrin of Sandia National Laboratories delivered the first talk, titled "***How Can We Model Cyber Attacks and Systems to Characterize Resilience of Critical Infrastructure Systems?***"



SYSTEMS ENGINEERING RESEARCH CENTER

SERC Talks: "How Can We Model Cyber Attacks and Systems to Characterize Resilience of Critical Infrastructure Systems?"

SERC TALKS

Dr. Eric Vugrin
Distinguished Member of Technical Staff,
Cyber Resilience Research and Development
Sandia National Laboratories

CYBER RESILIENCE

Dr. Peter Beling, SERC Research Council Member; Professor and Associate Director, Intelligent Systems Lab, Hume Center for National Security and Technology, Grado Department of Industrial and Systems Engineering at Virginia Tech

Abstract: Recent high profile cyber attacks on critical infrastructures have raised awareness about the severe and widespread impacts that these attacks can have on everyday life. This awareness has increased the demand for cyber resilient critical infrastructure and for proof that proposed mitigations actually increase resilience. In this presentation, we will discuss an approach to cyber modeling and experimentation that we are using at Sandia National Laboratories to analyze the resilience of industrial control systems. The approach leverages virtual cyber testbeds, threat emulation, and resilience metrics to provide quantitative resilience evaluations. Demonstration of the platform and analysis process are illustrated through a use case involving the control system for a pressurized water reactor.

Download slides:

Recordings are available for this talk on the SERC YouTube Channel.

Planned SERCTALKS in the Cyber Resilience series include:

- 6 April: ***Cyber Resilience: a Technical Concept or Vague Desiderata?*** By Dr. Alexander Kott, U.S. Army Research Lab
- 15 June: ***Cyber Resilience*** with Ms. Melinda K. Reed, Director Resilient Systems, Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E).

PPI RESOURCES

PPI offers a multitude of resources available to all of our clients, associates and friends! Click on any of the links below to access these resources today.

Systems Engineering FAQ: https://www.ppi-int.com/resources/systems-engineering-faq
Industry-related questions answered by PPI Founder and Managing Director Robert Halligan.

Key downloads: https://www.ppi-int.com/keydownloads/
Free downloadable presentations, short papers, specifications and other helpful downloads related to requirements and the field of Systems Engineering.

Conferences: https://www.ppi-int.com/resources/conferences-and-meetings/
Keep track of systems engineering-relevant conferences and meeting dates throughout the year.

Systems Engineering Goldmine: https://www.ppi-int.com/se-goldmine/
A free resources with over 4GB of downloadable information relevant to the Engineering of systems and a searchable database of 7,800+ defined terms. You can expect the content of the SE Goldmine to continue to increase over time.

Systems Engineering Tools Database (requires SEG account to log in from the Systems Engineering Goldmine): https://www.systemsengineeringtools.com/
A resource jointly developed and operated by Project Performance International (PPI) and the International Council on Systems Engineering (INCOSE). The SETDB helps you find appropriate software tools and cloud services that support your systems engineering-related activities. As a PPI SEG account holder, you have ongoing free access to the SETDB.

PPI SyEN Newsjournal (actually a substantial monthly SE publication): https://www.ppi-int.com/systems-engineering-newsjournal/
You're already reading our monthly newsjournal! However click on the link to access the history of 100+ monthly newsjournals containing excellent articles, news and other interesting topics summarizing developments in the field of systems engineering.

FEATURED ARTICLES

Iliff's Hierarchy of SE – Understanding the Journey to (and from) System Mastery

by Randall C. Iliff, PPI Presenter and Principal Consultant

Email: riliff@ppi-int.com

Abstract

After 40 years spent helping the world embrace the benefits of SE, it's clear that leaders outside of traditional SE application areas still struggle to understand and value our gift. As a profession we have as yet failed to motivate the general market.

I find this particularly strange given that the ability to confidently envision and deliver system solutions has never been more valuable, nor more desperately needed. The business math clearly favors exploring and implementing any method that improves quality and overall results. The natural question then becomes "Ok, what force is more powerful than profit and thus blocking leaders from recognizing and embracing SE?"

In the field of Psychology, Maslow's Hierarchy of Needs has become a classic model for explaining motivation and is widely used to explore why people don't do what others expect. Having applied that logic for years in general management, it seemed reasonable to see what it might offer as a pattern for exploring SE implementation challenges.

In this article I'll respectfully borrow Maslow's concept of hierarchy and introduce a system development view consisting of five discrete levels, resulting in twenty possible transition paths among those levels. This view makes it possible to envision the entire journey towards system mastery, reveals profound insights as to the danger of awareness without corresponding motivation, and also offers a guide to understanding why institutional regression is more common than progress.

Introduction

It hardly requires a background in SE to notice that the world struggles to anticipate and satisfy complex system needs. For over forty years I've watched the world struggle with systemic problems, and despite the best efforts of INCOSE, IEEE and many others, continue to deliberately ignore a body of methods ideally suited to understanding and managing those challenges. This is especially true outside of what are often called traditional SE application areas such as Aerospace.

For a significant period my job required that I deliver the benefits of SE to commercial sector clients without requiring that they understand or participate in the miracle. We sold the *function* of SE with ease, but any time we tried to sell the *logic* and *mechanics* of SE our market rebelled. Our clients did not understand the origin of benefit, thus they focused only on the final outcome.

As a business manager I simply adjusted our market strategy accordingly and fulfilled the contracts. As an engineer I recognized the enormous disconnect and began searching for the root cause. Like so many interface issues, the problem is neither the sender nor the receiver but the ability to connect the two in practice.

In 2002 I published parallel papers that explored why Manufacturing cultures struggle to understand the logic behind SE and Project Management, and I've also written extensively about the fundamental difference between repetitive tasks and those that involve creative effort of some type. Honoring the nature of the development task itself certainly holds part of the answer, but the balance of root cause is found only in the mind of decision makers.

I eventually realized that you cannot motivate an individual towards the benefits of SE until he or she is in a position to recognize and respond to your arguments. I also realized that even well-intentioned attempts to prematurely expose people to the message were at best wasteful and at worst offensive. Thus it seemed that the answer to why people don't widely embrace SE falls in the realm of human psychology rather than SE itself.

Why Introduce Yet Another Model?

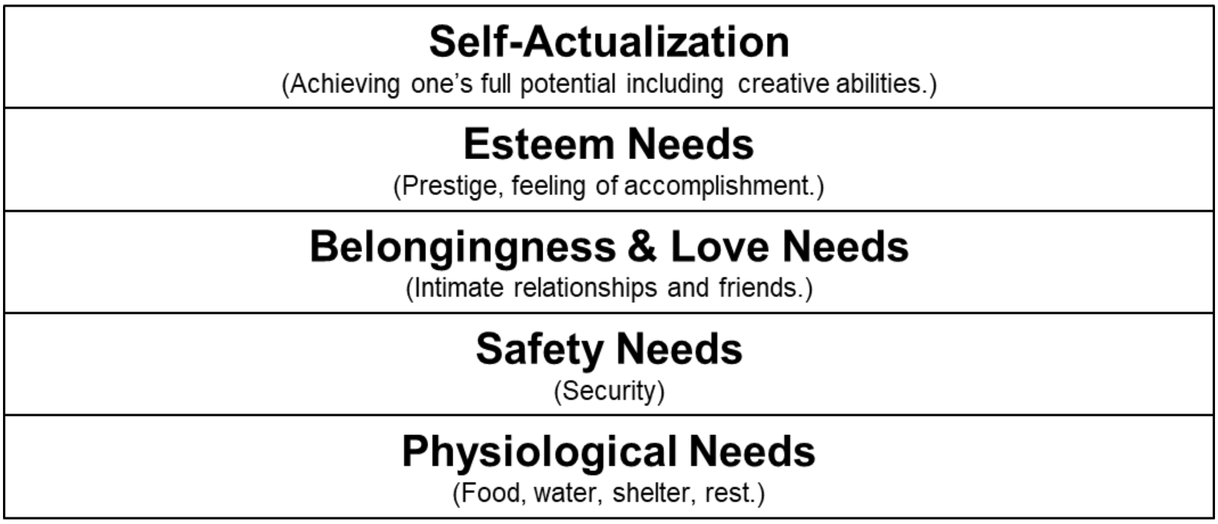
The goal is simply to provide a structure that facilitates useful conversation. The logic is important, the terms are merely placeholders for ideas that have many names. I don't offer it as truth, claim it be superior to other models, or suggest that it is a replacement for any other view.

My wish is that you embrace it as just another tool in your professional toolbox, useful for whatever benefit it offers in your situation.

Maslow's Hierarchy of Needs

First, let me be clear that I am an engineer rather than a psychologist, and I make no claim to unique insight regarding Maslow's work. Like all models there are strengths and weaknesses, but thinking of motivation as "states and modes" offers a valuable way to understand and predict human behavior.

For our purposes the simple five-layer view shown in Figure 1 is sufficient:



Self-Actualization
(Achieving one's full potential including creative abilities.)

Esteem Needs
(Prestige, feeling of accomplishment.)

Belongingness & Love Needs
(Intimate relationships and friends.)

Safety Needs
(Security)

Physiological Needs
(Food, water, shelter, rest.)

Figure 1 – Maslow's Hierarchy of Needs

To illustrate the logic:

- A person walking away from the smoldering wreckage of their aircraft has immediate Physiological needs that must be addressed. Until food, water and protection from the elements are in hand, nothing else really matters.

- Once immediate needs are met, that person moves to a larger system perspective of Safety, creating as physically secure an environment as they can.
- Once these basic needs have been met, the next level of need is that of Belonging. Loneliness now matters, whereas before it was irrelevant until food and shelter had been secured.
- Upon rescue and return to civilization, all needs for food, shelter and safety are met and no longer act as motivators. Instead, our pilot is motivated by the recognition and notoriety that comes with public esteem.
- The adulation of others eventually becomes routine, and once all of the basic needs and all of the externally-connected social needs have been met, only Self Actualization remains. Here the motivation is to become as rich a version of *yourself* as possible – a mode of beneficially applied selfishness that is impossible to imagine or sustain at lower levels.

There are several important ideas here to point out:

- At any given moment, an individual operates within the rules and motivations of the level they are in.
- The other levels are either fulfilled (those below) or aspirational (those above) and thus have little impact on behavior or motivation.
- Maslow's original work assumed a journey from bottom towards the top, his later work and that of others includes that path but considers the transition between levels to be much more fluid and allows for regression as well as progress.
- We live in a dangerous and unpredictable world, even Self-Actualized people can suddenly find themselves at the bottom of the pyramid desperately struggling to meet basic needs.

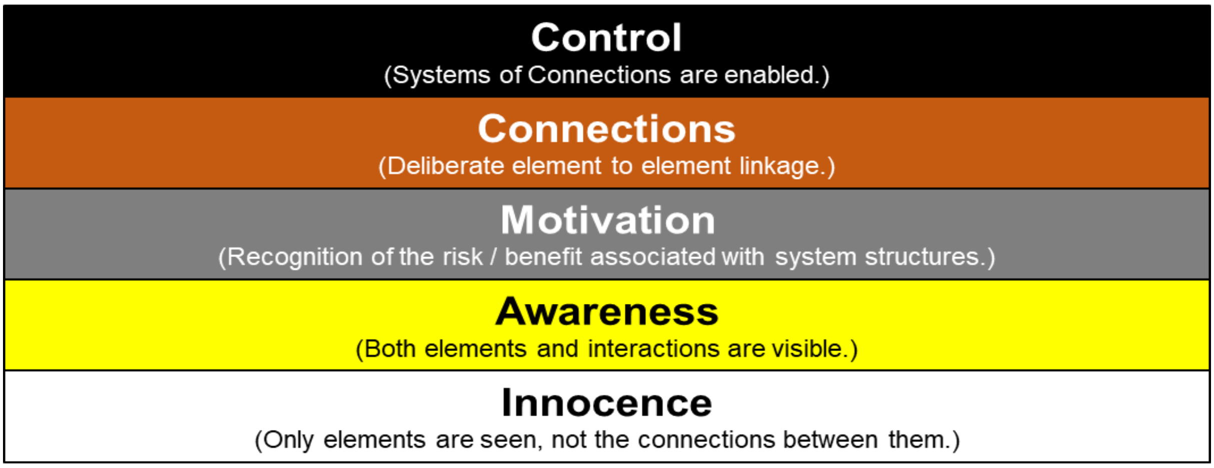
Together these ideas are used to assess where people are currently, identify the needs and motivations that matter to them in that context, and to manage the transitions between levels. The ability of the model to explain past behavior and predict future behavior makes it useful.

Iliff's Hierarchy of SE

The focus of Maslow's model is an entire individual, operating within some social context. The focus of the model introduced here is much narrower, and refers to an individual's capacity to recognize and respond to system structures they encounter.

As with Maslow's work, the model consists of a set of levels, each related to those above and below, that create unique operating states as well as important transition considerations.

We'll start by briefly introducing the five levels shown in Figure 2, working our way from the bottom of the stack towards the top.



Control
(Systems of Connections are enabled.)

Connections
(Deliberate element to element linkage.)

Motivation
(Recognition of the risk / benefit associated with system structures.)

Awareness
(Both elements and interactions are visible.)

Innocence
(Only elements are seen, not the connections between them.)

Figure 2 – Iliff's Hierarchy of SE

Innocence

At the lowest level, *Innocence*, system elements are seen only in isolation and in one moment at a time. The concept of "system" is missing completely. Each thing they encounter stands alone, zero energy is spent examining or exploring the relationships that exist between them. Life activities that are dependent on individual "things" go smoothly and efficiently, the origin of inevitable problems in other areas is largely unknown and attributed to luck, fate, or superstition.

It's tempting to describe *Innocence* as a weak or undesirable level, particularly from the view of a seasoned SE practitioner, but that is a fundamental error with enormous impact. Humans revere the state of *Innocence*, we grieve its loss, and we spend our lives seeking to reacquire bits of that peace wherever we can.

Awareness

The next level, *Awareness*, consists of seeing both elements and interactions. This is the threshold of system *Awareness*, and can be immensely intimidating. Each element you used to interact with comfortably and naturally is still there, but now each comes with a degree of fear and uncertainty about what the newly visible relationships might mean.

At the level of *Awareness*, life gets harder rather than easier. You are now aware of possible problems you don't know how to manage, and there is a pronounced longing to return to the era when you didn't know about any of that stuff. Being the messenger of system awareness is a profoundly dangerous job.

Awareness by itself is terrifying, and without a solid reason to keep going the natural response is to run as quickly as possible back to Innocence. If you've ever wondered why really bright people in senior leadership positions so often lie to themselves, this is your answer.

Motivation

Only at the next level, *Motivation*, do we recognize the potential to manage interactions and thus gain some degree of access to benefit or protection from harm. *Motivation* offers a reason to sustain and build *Awareness* rather than suppress it and hide in Innocence.

Awareness must always be coupled with *Motivation* in practice. Too often as messengers of SE we assume that our audience will understand the value of the complexity we bring, yet without a clear story of why they should care all we do is terrify them and encourage them to cover their eyes and ears.

Perhaps you have experienced this dynamic in a class setting, either as instructor or student. Unless students are "pulling information" based on motivation, everything the instructor "pushes" their direction just makes everyone's day more painful.

Connections

At the level of *Connections*, we begin for the first time to take active control over the relationships between elements around us. At this early stage of system awareness, these will largely be in the form of isolated element to element linkages, but can be very effective at opening specific opportunities and blocking discrete threats.

While these isolated but beneficial linkages may seem primitive from an SE perspective, they are sufficient to address the overwhelming majority of individual and organizational needs throughout history. What prevents even large numbers of *Connections* from achieving *Control* is the lack of connection between connections.

Connection offers lines and the potential to postulate crude mesh-frameworks, in *Control* we aspire to represent the entire enclosed volume of problem and solution spaces. In both cases validated, reusable patterns are highly sought after since they reduce the quantity of complexity we must manage. Examples include standards, procedures, policies, interface protocols, even Patents and the rule of Law.

Control

At the peak level, *Control*, both elements and interfaces receive simultaneous attention and we are now creating systems of *Connections*. *Control* can be thought of as the inverse of *Innocence*: Tons of work, zero fun.

This is the domain of all structured development models, including but by no means limited to SE as we know it today. At peak effectiveness, *Control* can bring into existence anything human minds can imagine.

Playing god comes with a ton of overhead though, and is maddeningly difficult to sustain in practice. From a human *Motivation* standpoint, there has to be a great reason to keep plugging away and constant monitoring to prevent a drift back towards simple *Connections* or even a complete bail-out back to *Innocence*. How many of us have seen an SE group thrive for decades rather than just the duration of a strong leader's presence?

Because *Control* comes at such a high cost, it is in our best interest to impose only the minimum level of *Control* needed to meet our objectives. Overcomplicated *Control* models fail to generate benefit proportionate to expense, but even more critically they create enormous pressure to abandon *Control* completely. How many of us have seen an organization rebel against an overly restrictive policy?

The function of tailoring in SE is intended to serve this role, but in practice we often end up applying "standard solutions" to unique situations. To an audience looking for excuses to do less, these negative examples are a gift we need to stop giving.

Twenty Possible Transition Relationships

We've touched on some of the level to level transitions, but it is in our interest to examine them comprehensively. Figure 3 shows the five levels in the form of a modified n-squared diagram, with arrows showing the twenty internal relationships that arise.

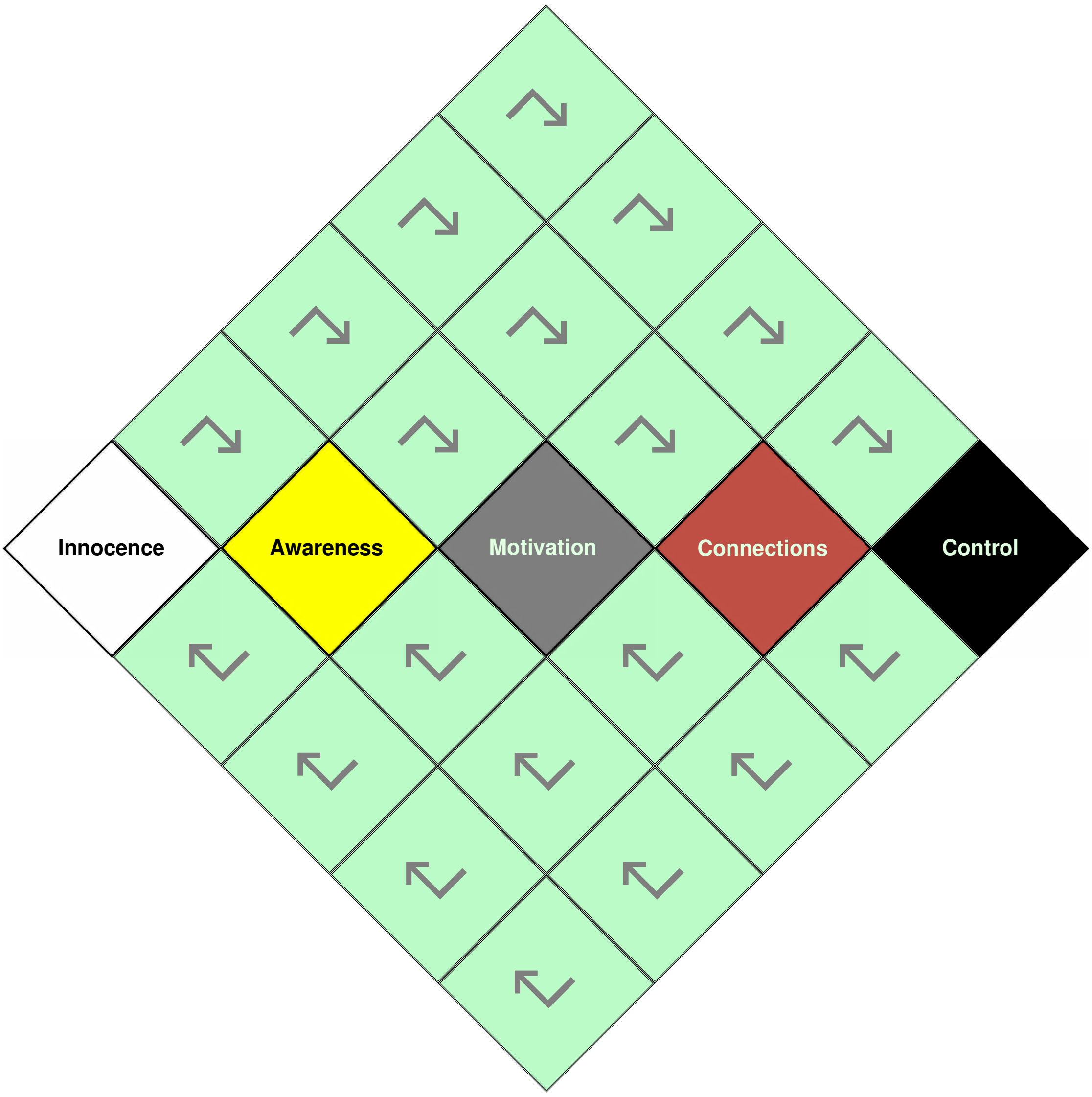


Figure 3 – Overview of All Possible Transitions

This perspective is useful in that it shows all of the possible transitions in a single view, but we'll break them down a bit further into sets of relationships.

Direct Transition Between Levels

The easiest relationships to imagine are those that directly link the level to the one above or below. Progress from *Innocence* towards *Control* simplistically follows the path shown in Figure 4, and involves four possible transitions between levels:

- *Innocence* to *Awareness*
- *Awareness* to *Motivation*
- *Motivation* to *Connections*
- *Connections* to *Control*

These are easy to imagine and have been discussed to some extent already. The problem child in this path is *Awareness*, as noted earlier *Awareness* without *Motivation* results in frustration for all concerned.

The forward path requires the investment of energy, often very significant amounts over long periods of time, which are ideally captured and stored in the form of retained knowledge.

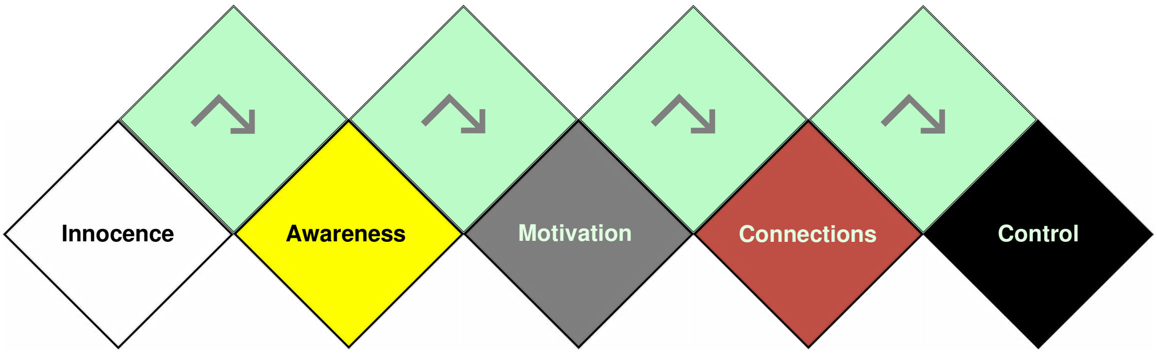


Figure 4 – Direct Forward Transitions

We also have the possibility of regression from Control towards Innocence, which simplistically follows the path shown in Figure 5 and involves four possible transitions between levels:

- *Control* to *Connections*
- *Connections* to *Motivation*
- *Motivation* to *Awareness*
- *Awareness* to *Innocence*

The first point we need to make here is that regression sounds bad but may be the correct response to changes in circumstance. Regression is only bad when it is at odds with the nature of the task, and is absolutely the right management response whenever possible. Unless there is sound reason to do otherwise, humans deserve the chance to operate in a mode of *Innocence* and play as much of the time as possible.

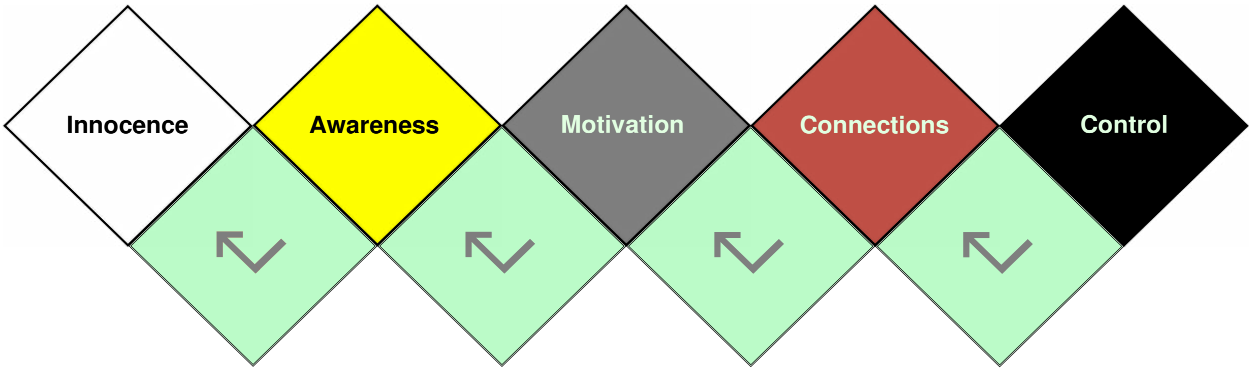


Figure 5 – Direct Backward Transitions

We've all seen the transition at the end of a major project from a period of intense *Control* to one characterized by diligent management of a few defined interfaces. This is the classic "healthy" example of transition from *Control* to *Connections*.

Unhealthy transitions occur as the result of incompatible leadership / motivation patterns and give up *Control* in return for what feels like an easier path to follow. The classic example of this is prematurely cutting off the requirements process and jumping into design unprepared.

Connections are lost gradually, usually as a result of diminished *Motivation*. The fire that happened yesterday is an urgent motivator to keep matches in a safe place, whereas the fire that only a few people remember has no motivating ability and it seems like no big deal to keep the matches on the stove where they are handy.

Without *Motivation* there is no reason to sustain *Awareness*, and the fall through to *Innocence* is almost inevitable.

Transitions that Bypass a Level

The direct linkages dominate most of the practical world, but there are important paths that skip past a level. These are shown in Figure 6, and consist of:

- *Innocence* to *Motivation*
- *Awareness* to *Connections*
- *Motivation* to *Control*
- *Control* to *Motivation*
- *Connections* to *Awareness*
- *Motivation* to *Innocence*

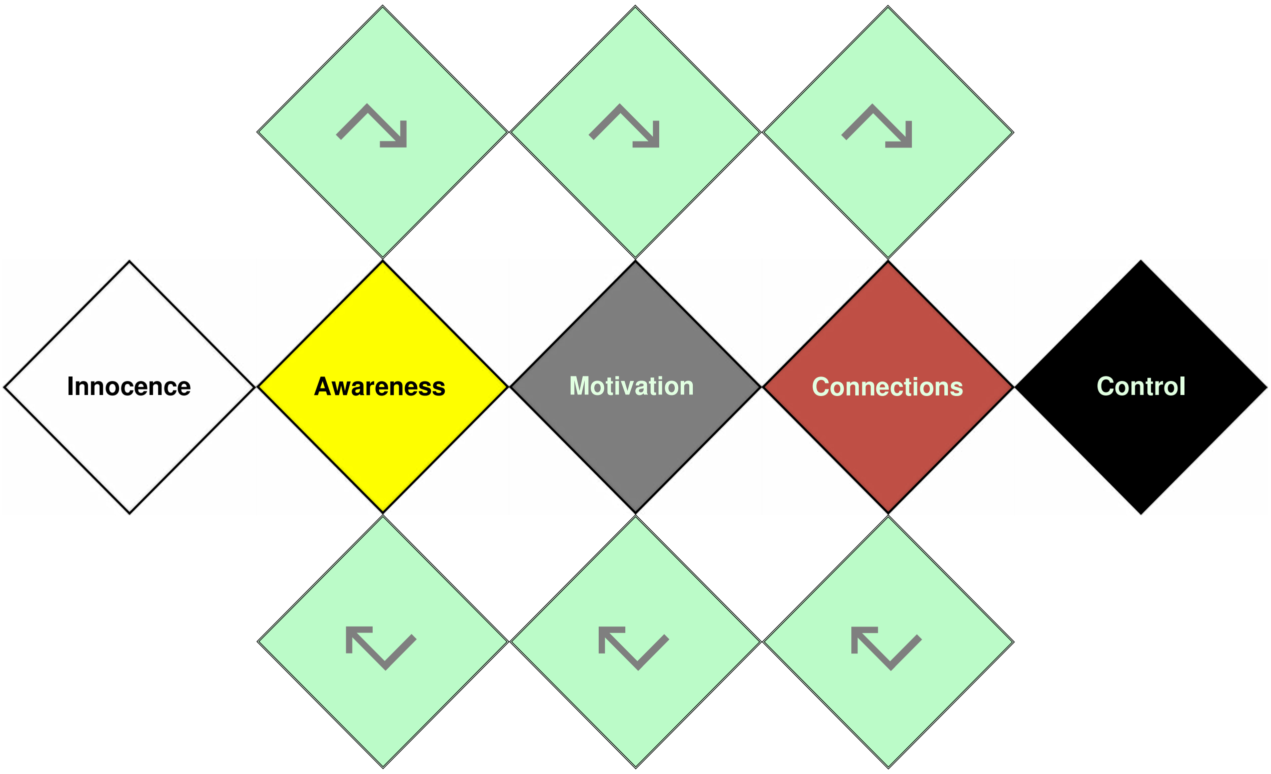


Figure 6 – Transitions that Bypass a Level

The path from *Innocence* to *Motivation* is typically driven by comparison to another situation in which awareness already exists. It is possible to want to fly with no awareness of what it takes to do so. This is the realm of fantasy and aspiration rather than meaningful progress, and must always be accompanied by remedial *Awareness*.

The path from *Awareness* to *Connections* is barren without *Motivation*.

From *Motivation* to *Control* however is a very different story. Here we have the opportunity to translate our *Awareness* and *Motivation* into a decision to jump directly to *Control*. This is a natural and healthy transition we need to encourage whenever the underlying circumstances demand *Control* rather than a few *Connections*. We are able to then avoid a "growing pains" period of transition through inadequate methods.

There is also a natural and common transition from *Control* back to the level of *Motivation*. For many this occurs on completion of a major project and finding oneself back in the proposal team looking for the next billable charge number. We are at a state of readiness, but no longer actively applying *Control*.

From *Connections* to *Awareness* occurs when the *Motivation* to act is no longer present. Note the caution that *Awareness* without *Motivation* is unstable and will decay quickly back into *Innocence*.

Outliers

The n-squared view forces us to consider the possibility of six other types of possible level to level transition. These are shown in Figure 7, and consist of:

- *Innocence* to *Connections*
- *Innocence* to *Control*
- *Awareness* to *Control*
- *Control* to *Awareness*
- *Control* to *Innocence*
- *Connections* to *Innocence*

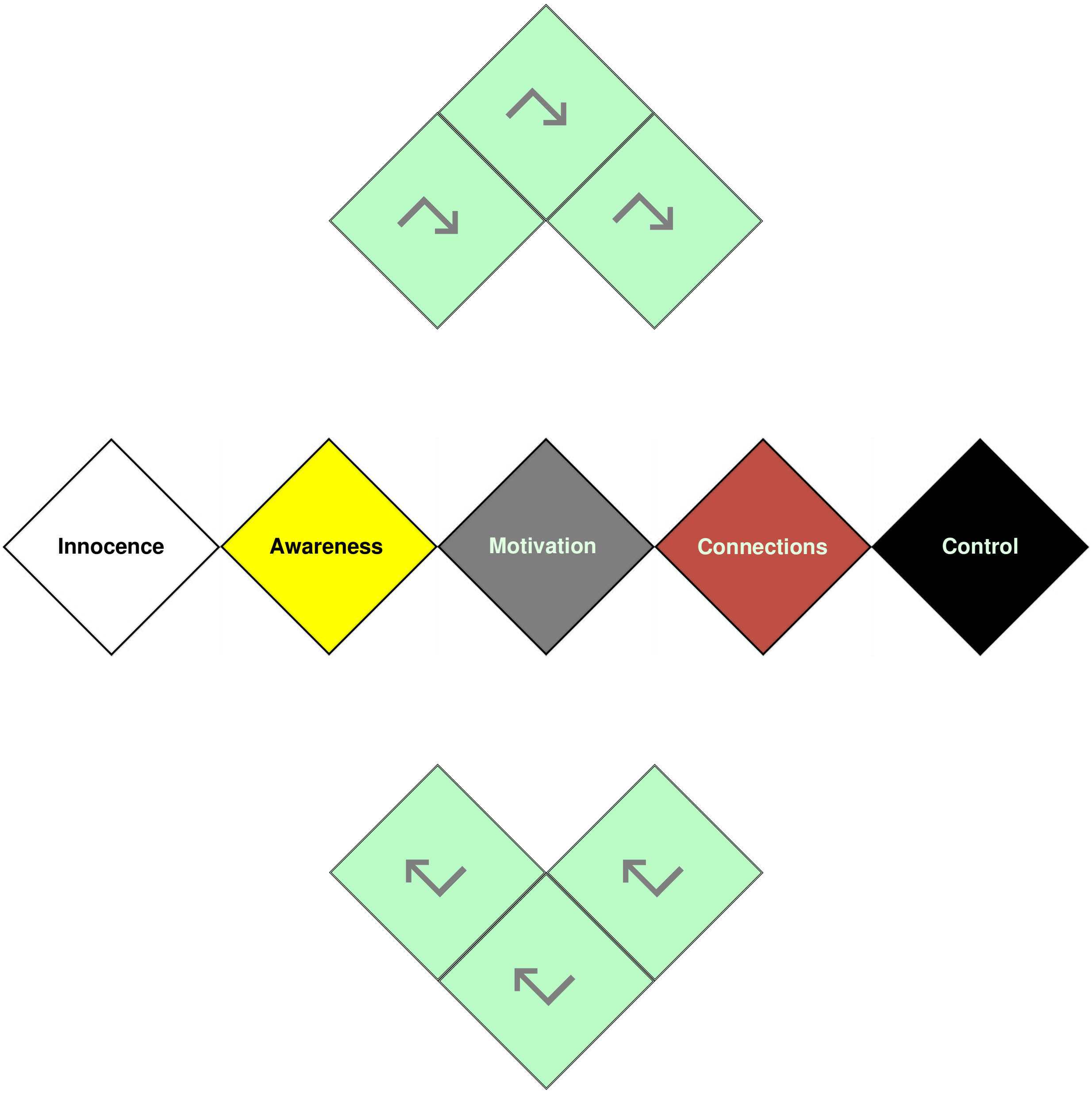


Figure 7 – "Outlier" Transitions Between Levels

The path from *Innocence* to *Connections*, and from Innocence to *Control*, are most commonly encountered when rules are imposed from outside. Laws require that you obey traffic lights even if you don't understand why they are there. Since we lack the benefit of *Awareness* and *Motivation*, compliance is directly proportional to enforcement, and thus inherently unstable over time.

From *Control* to *Awareness* is also common, and in an ideal situation allows the individual to retain *Awareness* even without *Motivation*. In practice *Awareness* decays quickly so the eventual effect is a transition to *Innocence*.

Transition from *Control* to *Innocence* (as well as *Connections* to *Innocence*) is not only possible, it is something that all of us long for as we schedule our vacation time. This transition is bad only when the need for *Connections* or *Control* is still present or the situation can be expected to reoccur.

Implications:

Important Observations for Teams and Groups

It is very common for people to operate at different levels for different topics or at different points in their life. We are dealing with complex response surfaces rather than a simple point to point transition. This consideration is even more important when we extrapolate the model to work groups or overall organizations.

Training and process can help reduce the range of levels that must be managed at any given time, and to help guide the transitions. If the difference in levels is too great, however, the transition will take place via the outlier paths versus the more manageable level to level paths we've described.

Beware the Leaky Stack

For anyone seeking to implement *Control*, the natural home of SE, it is very important to establish a comprehensive foundation of *Awareness* and *Motivations*, and for there to be recognition of the value of *Connections*. Without these enabling conditions it will be impossible to create or sustain a realistic *Control* level.

The greater the degree of *Control* you seek, the stronger the pressure vessel of *Awareness* and *Motivation* you must provide, and the more frequently you'll need to top off the tank.

Imagine filling an Olympic Swimming pool with water, and then being responsible to keep the pool full over time. At the top you'll lose water to evaporation, splashing, testing samples and so on. This loss is predictable, visible and easily offset. In our daily world of *Connection* and *Control* "splashing" takes the form of arguments about tools, terms, budgets and countless other details, and "evaporation" is a metaphor for loss arising through attrition or role-changes.

The more significant danger lurks at the bottom of the pool, where pressure is greatest and ongoing attention the least. Cracks in the bottom of the pool not only result in additional water loss that must be replenished, the leakage will erode the very structure of the containment vessel and lead to sudden catastrophic failure.

Without the protective firewall of discriminating intelligence and education, *Awareness* can easily be hijacked by external players whose *Motivations* are at odds with yours. When the intent is *marketing*, the worst that happens is you end up buying something you don't need. When the intent is *conquest*, propaganda and lies have the ability to disarm an opponent before the battle is joined, thus are more powerful weapons of all.

Constant monitoring for leaks and wet spots at all levels is strongly encouraged, as is applying the logic of cybersecurity to protect your system from external influence.

Energy must be Added to Raise Levels

Innocence is in a sense absolute zero, and *Control* the highest temperature your process oven can create. Between those extremes fall the other three levels.

It takes energy to raise the level of interaction, but it is very important to note that the response will not be linear. Much like phase-change in physical materials you can expect to pour energy in without apparent impact until a characteristic threshold is reached. The close coupling of *Awareness* and *Motivation* magnifies this effect, more akin to sublimation than transition through melting, which explains in part why the SE community has had such difficulty altering overall development community behavior.

Indecision is a largely avoidable waste of energy and demoralizing for all involved. The "bid / no-bid" decision that wastes half of the response period, dithering over tool investments, asking for "just a little more information" before signing a contract, are all common examples of indecision in practice. Indecision always imposes an efficiency penalty, and far too often leads to complete loss of the opportunity or ability to defend against risk.

An inability to commit is nearly always the result of vacillation between *Innocence*, *Awareness* and *Motivation*. Indecision results in traveling the same path over and over, incurring hysteresis loss each cycle that must be travelled. Meanwhile the layers of *Connection* and *Control* are left rudderless, creating further costs and inefficiency.

Energy is Required to Sustain any Level Above Innocence

The further up the levels you go the greater the energy required to sustain that level. *Control* is orders of magnitude more difficult to maintain than *Connections* and the recurring cost of sustaining *Control* is higher than many individuals or organizations can support for other than brief periods.

Energy can be Extracted by Simplification

Just as it takes energy to raise the level, it is possible to free up energy as you simplify. Like regenerative braking in an electric vehicle, it is extremely desirable to capture and redirect this energy rather than waste it.

An effective lessons learned process can retain the essential combination of *Awareness* and *Motivation* long after the need for active *Control* has passed. The memory of overall system nature gained in pursuit of *Control* can be used to defend a handful of essential *Connections*.

People Hate Awareness Without Motivation

Perhaps the most important take-away is the need to consistently couple *Motivation* with *Awareness*. Revealing the scary truth about the web of systems we live in is only tolerated when there is a strong reason that knowledge is important. The pull of *Innocence* is very powerful, and people will rationalize or distort hard facts to avoid taking actions they fear.

Stated another way, *Awareness* brings work without benefit and simply adds to the stress level. Coupling *Awareness* and *Motivation* offers a reason to accept the stress in return for a greater benefit obtained through *Connections* or *Control*.

Since different people are motivated by different things, creating a consistent level of *Awareness* in an organizational context requires communicating not one, but an entire family of associated *Motivations*. If engineering is aware of complexity and therefore motivated to pursue SE that's a great start, but engineering complexity may have zero motivating effect on finance or sales.

Without an equivalent benefit recognizable in their language they will seek to suppress organizational *Awareness* rather than support it.

How to Apply this Hierarchy of SE Model:

Examine Yourself

Impartial self-assessment can lead you to discover areas where *Awareness* without *Motivation* is creating stress. An engineer longing to try a shot at sales or a leadership position feels one type of this stress, the engineer who was promoted to management and now hates their job is another.

Do you relax to a state of *Innocence* when you step away from work, or does *Awareness* keep you trapped in constant mental anguish? When I first transitioned from employment to a life of consulting, I allowed myself an 8 week break – longer than any period I had ever had off before – and it took nearly the entire time to finally let go of the nagging feeling that I was missing something that urgently needed to be done.

It only counts as rest if you are able to let go of *Awareness* and experience a period of *Innocence*, let that realization guide you to take better care of yourself and those around you.

Is your first instinct to impose *Connection* or *Control*, perhaps before others have *Motivation* or *Awareness* of what you are trying to accomplish?

Good intentions are far more rewarding when the recipient is prepared to appreciate the benefit. Taking your message where it is welcomed is easy. Sharing a message with everyone else requires that you invest the time to create a suitable environment first. Doing so will both reduce your stress and multiply the benefit of your effort.

Are you enamored with the idea of *Control*, even when *Connection* is sufficient to address all relevant needs? There's nothing wrong with proudly owning a beautiful set of precision machine-tools, but a sharp axe and dry matches are more useful on a camping trip. Let your goal be to keep things as simple as possible but no simpler.

Examine Your Current Environment

It is very useful to identify both the peak and minimum levels people around you are operating in. The peaks represent opportunity, the minimums represent barriers to effective system effort. Be especially alert to the need to couple *Awareness* and *Motivation*.

This insight will enable you to predict challenges, prioritize needs, and target key individuals with the messages that are most effective based on the level they are currently in.

Characterize any Environment that you Seek to Help

Before you show up as the messenger of SE intent on saving someone, it never hurts to know a little about whether they want to be saved or not. Perhaps even more important is knowing whether they need to be saved. If a collection of *Connections* are meeting everyone's informed expectations, there may not be a need to impose an order of magnitude greater complexity in the form of *Control*.

More likely you'll find that gaps in *Awareness* and *Motivation* are the reason they have not already elected to seek *Control*, and that any attempt to encourage them without that foundation will be futile.

Become a Champion of Innocence

If you are visibly enamored with the prospect of making things as simple as possible whenever you can, people will give you much greater buy-in when you must eventually point out aspects that do require a bit more attention. Don't point out complexity unless / until there is reason that they will understand as motivating action of some form.

It's also important to become a champion of *Innocence* for yourself and those around you. Create an environment in which everyone can responsibly and comfortably decouple from *Awareness* as much as possible. Recharged people are far more effective than those with a permanent shorting bar across their terminals.

Don't Jump to Control when Connections are Sufficient

Control is an enormously expensive and difficult to sustain level. It requires the expenditure of time, money, and large commitments of skilled talent. The opportunity cost is very high, make sure that the opportunity is worth it.

There are tasks that absolutely demand this degree of attention and investment (such as weapon systems, complex medical devices, and so on) for which a less robust set of *Connections* would be impractical or irresponsible.

For the vast majority of human activity, however, a fraction of the investment needed for *Control* can provide a perfectly adequate degree of *Connections*. Not only will you save time and money, you'll free scarce resources to tackle the tasks that truly need them.

List of Acronyms Used in this Paper

Acronym	Explanation
SE	Systems Engineering
PM	Project / Program Management
IEEE	Institute of Electrical and Electronics Engineers
PDMA	Product Development Management Association

References

Iliff, Randall C. (1992) Mapping Systems Engineering Principles & Practices to the Commercial Sector (INCOSE International Symposium 1992)

Iliff, Randall C. (2002) A Question of Context - Why Manufacturing Cultures Don't Understand Project Management (Project Management Institute 2002)

Iliff, Randall C. (2002) A Question of Context - Why Manufacturing Cultures Don't Understand Systems Engineering (INCOSE International Symposium 2002)

Iliff, Randall C. (2010) Introducing Systems Engineering Principles to the Commercial Product Development Cycle (INCOSE Technical Operations Webinar, May 27, 2010)

Iliff, Randall C. (2010) Systems Engineering Principles for Commercial Product Development (PDMA Presentation, Chicago, September 30, 2010)

Iliff, Randall C. (2012) "Aligning Cultures – The Secret to Successfully Managing Product Development Projects" (INCOSE Presentation, Chicago, July 19, 2012)

Maslow, A. H. (1943) A Theory of Human Motivation. Psychological Review. 50 (4) 370-396

Maslow, A. (1954) Motivation and Personality. New York, NY: Harper.

Technology and trends in systems modeling and simulation

by Erik Åberg, Eradity AB

Email: erik.aberg@eradity.com

Introduction and background

The purpose of dynamic system simulation is to replicate the dynamic behavior of a product or process. There are many reasons why this is important. The most important reason is that it provides valuable insights into the system at hand. Insights may be hard and costly to gain using traditional methods such as physical testing. With simulation, it's possible to build this knowledge rapidly and continuously during the development process, promoting an iterative and agile way of working. This also enables early evaluation of new concepts and "What if"-scenarios and optimization of existing designs and processes.

The idea of using simulation in development processes isn't new. It is a well-proven method that has been around for more than 60 years. The original version of Fortran, the first high-level programming language, was released by IBM in the late 1950s [1]. Well-suited for scientific computing, it was soon adopted by pioneers in computational modeling and simulation. About the same time, new methodologies for systems modeling started to show up, most notably Bond Graph theory, which made it possible to represent dynamic physical systems graphically [2]. Both Fortran code and Bond Graphs have played a major role in the evolution of system simulation technology and are used to this day by companies across all industries.

While simulation has been around for decades, simulation technology has taken significant leaps forward in recent years. Modern simulation technology offers convenience, flexibility, and openness to its users. In combination with advances in computational power, cloud computing, and artificial intelligence, this technology now redefines the role of simulation in product development as well as in the other stages of the product lifecycle.

This article explores some of these modern simulation technologies, specifically Modelica and the Functional Mock-up Interface (FMI), and the value of using system simulation for applications throughout the product lifecycle.

Overview of Modelica

As mentioned in the introduction, the purpose of system simulation is to gain insights into the behavior of a system. There are a number of simulation tools on the market that can achieve this, and many of them are based on the Modelica modeling language. Modelica is an open, equation-based, and object-oriented modeling language for dynamic modeling and simulation of physical systems. The language has been growing rapidly in popularity in recent years and is supported by a large variety of simulation tools, both open-source and commercial.

The open nature of Modelica promotes great flexibility and the creation of scalable models that can be reused and employed for different purposes throughout the development process. Models are represented both graphically and in code and are easy to adapt to fit specific needs. The multi-domain capabilities of the language make it possible to capture the dynamic behavior of any physical domain, including fluid dynamics, mechanics, electrical, and thermal. Ready-made component models for the different physical domains are available in the Modelica Standard Library, preinstalled with most Modelica tools. It is a freely available library of well-proven base components, ready to use in new modeling projects.

The graphical view of a Modelica model of a pendulum is shown in Figure 1. This view is used for building models graphically by dragging-and-dropping component models from a component library, in this case, the Modelica Standard Library, and connecting them. The graphical view resembles the system layout in the same way as a traditional system diagram or a simple sketch, making the model easy to understand even for people unfamiliar with the system.

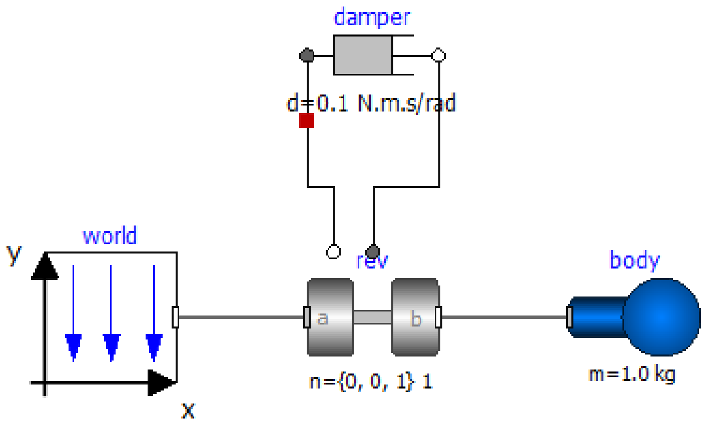
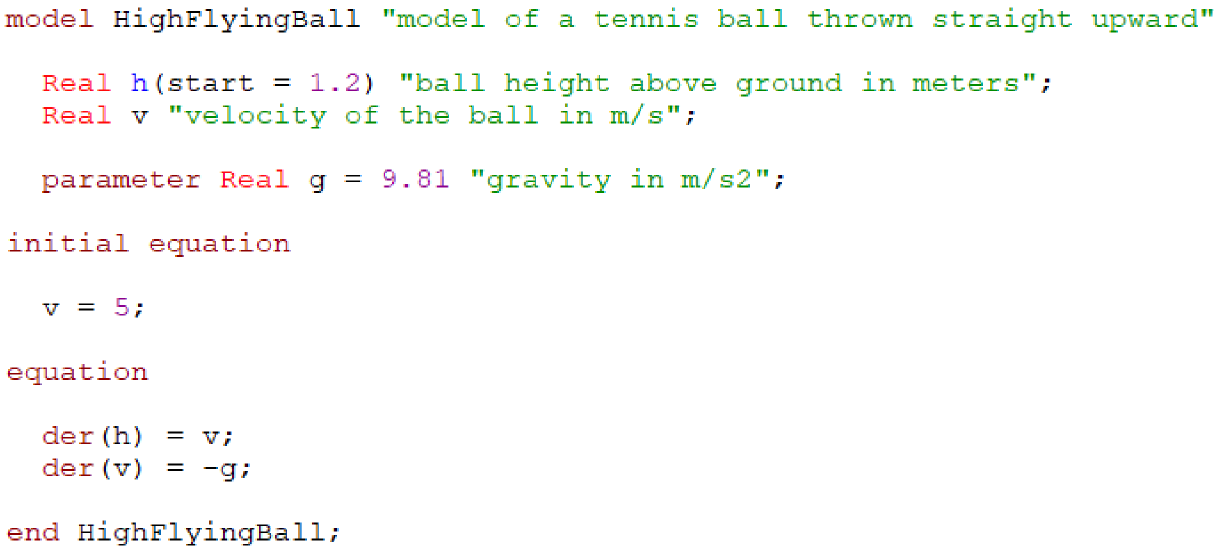


Figure 1: Graphical view of a Modelica model of a pendulum

The connections between the different components are acausal and represent actual physical couplings. In a real-world system, a change to one component not only affects the component itself and components downstream. Components upstream will also be affected. These reaction forces can be captured in models based on acausal connections, allowing components to interact just like in reality. The acausality of Modelica is a major difference compared to signal-based simulation software, like Simulink, or simulation software based on Bonds Graphs, where the causality needs to be predefined. Besides being closer to reality, the acausal connectors simplify the modeling process by making it possible to build models through drag-and-drop modeling without thinking about the direction in which information flows.

Another advantage, compared to most other system simulation software, is the possibility to easily access, modify, and write code. While most other tools offer some functionality for creating new models from scripts or C code, Modelica-based tools come with a comprehensive modeling language. The Modelica code is equation-based and describes relations between variables rather than assignments. The equations can be written in any order and any form, similarly to writing equations on paper. This intuitive way of working makes it easy to create new component models in Modelica.

An example is shown in Figure 2. It shows the code of a simple model of a free-falling body. Each component of the pendulum model of Figure 1 consists of a similar set of equations that describe the physical behavior of that specific component. By combining component models in different ways, different system behavior is obtained.



```
model HighFlyingBall "model of a tennis ball thrown straight upward"

  Real h(start = 1.2) "ball height above ground in meters";
  Real v "velocity of the ball in m/s";

  parameter Real g = 9.81 "gravity in m/s2";

initial equation

  v = 5;

equation

  der(h) = v;
  der(v) = -g;

end HighFlyingBall;
```

Figure 2: Modelica code of a free-falling body

Simulating a system model takes just a few seconds, even for more complex systems. The results, showing the pendulum dynamics in this case, can be displayed either as a 3D animation or by plotting the individual variables of the model.

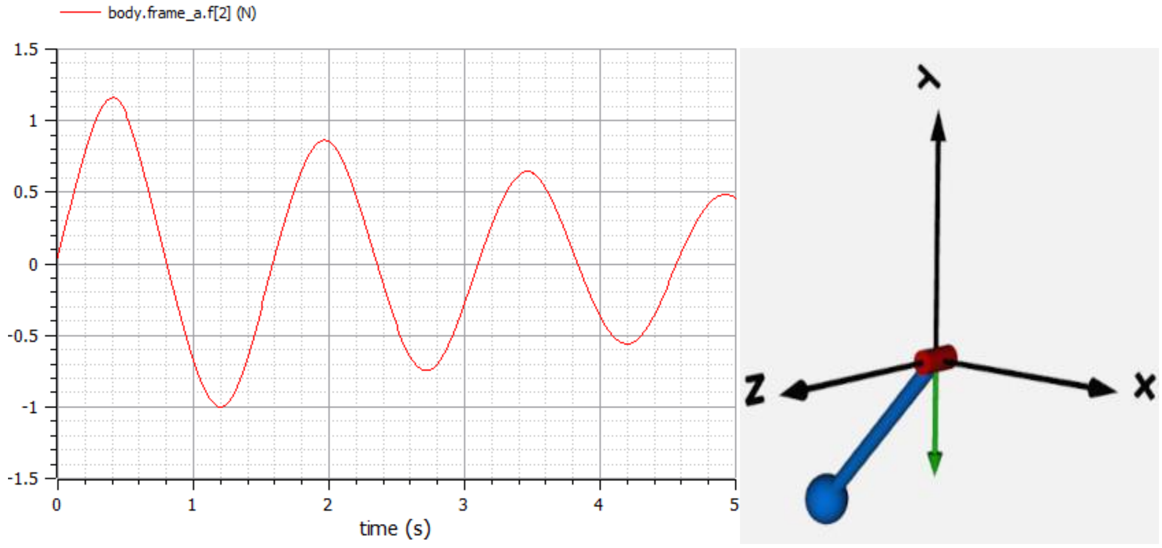


Figure 3: 3D animation and plot of the pendulum dynamics

Benefits for the industry

For the industry, the use of Modelica brings lots of benefits. It is a capable language with very few limitations for its users. Applications stretch from dynamic modeling of 0D and 1D systems to 2D and 3D, as shown earlier in the pendulum example. Access to the Modelica Standard Library and other application-specific component libraries simplifies the modeling process, while the openness and simplicity of the language also guarantee that new models can be created if necessary.

The openness and the large number of Modelica tools to choose from are also beneficial from a commercial point of view. Since the tools use a common modeling language, a model developed in one software can easily be exported and used in another, reducing the dependency on a specific tool and preventing vendor lock-in. The openness of Modelica is further strengthened by the addition of the Functional Mock-up Interface, FMI. It is an open standard for model exchange and co-simulation that allows users to export simulation models and simulate them within another software. Although maintained by the Modelica Association, FMI is a Modelica independent interface, supported by numerous simulation software and scripting environments such as Python. By combining the modeling capabilities of Modelica with deployment via FMI, companies can create ecosystems of models that can be employed for different purposes throughout the product lifecycle.

The simulation-based development process

The advantages of Modelica and FMI fit well with the requirements of the ongoing virtualization of the industry. The traditional hardware development process relies on physical prototyping and testing for verification and validation purposes. It is a costly process, both concerning time and money. By introducing simulation, these costs can be cut. It is achieved by letting one or several simulation models act as virtual prototypes. These virtual prototypes resemble the system dynamics and can be used to predict challenges and provide unique insights into the system. This enables iterative workflows, where incremental changes and rapid evaluation of new concepts and scenarios are possible. This way of working is desirable as it makes it possible to deal with increasingly complex products and processes consisting of both hardware and software.

Even with a simulation-based development process, there may still be a need for physical prototypes. However, as much of the analysis is done using simulation, the testing can focus on validating the performance of the final design. There is no conflict between simulation and testing. On the contrary. Utilizing both methods strengthens the knowledge about the product, making it possible to track and solve complex issues that are too hard to solve with testing or simulation alone.

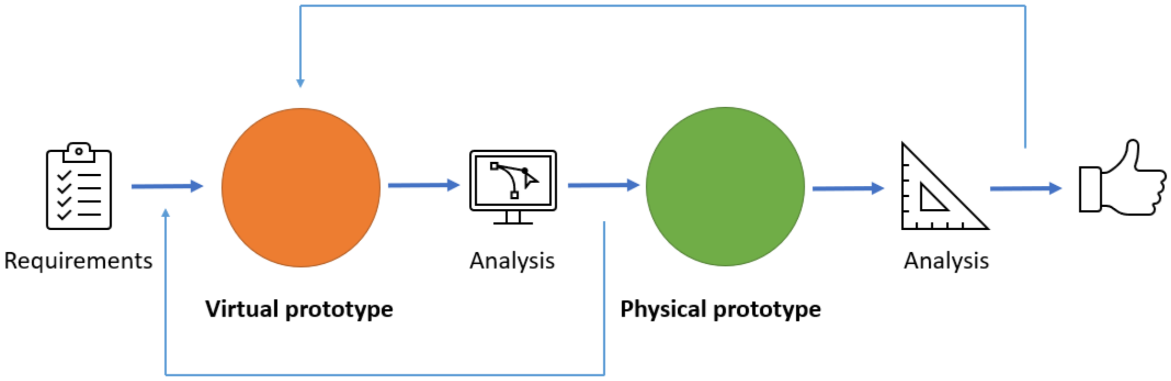


Figure 4: Overview of the simulation-based development process

The virtualization and adoption of systems modeling and simulation are well on the way in many industries. Using the Modelica and FMI technologies, companies have been successful in building ecosystems of scalable models that can be reused for different purposes. The list of use cases from the industry is extensive: scenario evaluation [3], control design [4], hardware-in-the-loop testing [5], drive cycle analysis [6], plant optimization [7], and driving simulators [8], to mention just a few. New applications are added to the list daily as system simulation grows in popularity. The drivers behind that growth are the challenges that companies face. Sustainability, machine autonomy, and energy efficiency are some of the areas where companies find simulation essential for uncovering new innovative solutions.

Trends in system simulation software

The adoption of simulation in the industry is further encouraged by the software vendors who continuously update tool functionalities and workflows. A strong trend in recent years has been on democratizing simulation tools and processes to increase the accessibility of models, results, and analyses in organizations. By lowering the thresholds, more users can benefit from system simulation. The improved user experience also has other advantages. With simplified modeling workflows, users can focus more on value-creating activities, such as reviewing results, rather than on solving modeling issues. Better post-processing capabilities, enhanced model libraries, and automatic generation of simulation models from CAD drawings are examples of recent advances in system simulation tools that improve the user experience.

Another way for software vendors to increase accessibility is to move their simulation platforms to the cloud. Apart from increasing the availability of the platform itself, transitioning to the cloud adds access to cloud-native capabilities and interfaces. These capabilities enable customizations and further flexibility. One example is web applications based on simulation models that can be shared with others [9]. Building and sharing web apps make predictive analysis available to more people and help remove information islands in organizations. Another advantage of moving to the cloud is that it simplifies using simulation models with cloud-native methods, such as machine learning. By combining simulation and machine learning, companies can create operational digital twins. If fed with real-time process data, these digital twins can provide valuable insights about the process, detect anomalies, and predict future outcomes and challenges.

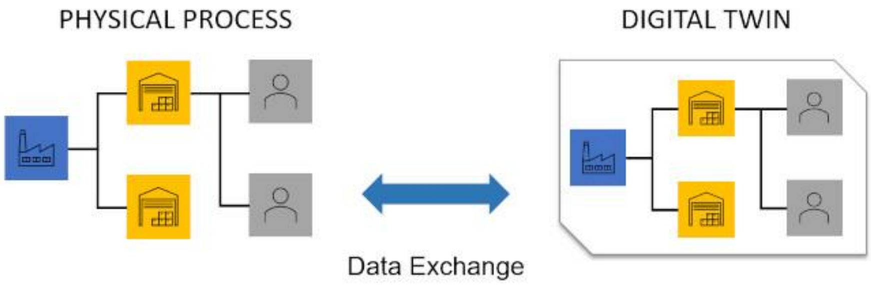


Figure 5: Interaction between a physical process and a Digital Twin of the same process

Future outlook

Looking at the recent trends in system simulation, it becomes clear that simulation is moving from being a pure development and analysis method to something that can bring value also for the other stages of the product lifecycle. The possibility to build ecosystems of models based on Modelica and FMI, democratization, web applications, and deployment of digital twins, are contributing factors.

While system simulation will continue to bring value and strengthen product development processes, there are plenty of other areas within a product's lifecycle where the use of dynamic modeling is yet to be explored. There are likely many applications within manufacturing and logistics where dynamic system simulation, combined with existing solutions and machine learning, would bring a unique value.

Convenience, flexibility, and openness are the key drivers of the current trend, enabling synergies between applications and methods. Systems modeling and simulation are moving into a new exciting era. Thanks to modern simulation technology and recent advances in tools, the role of simulation is about to be redefined.

Further reading: "Modelica Series I – Introduction to Modelica", https://www.eradity.com/blog/29-modelica-series-introduction-to-modelica

References:

[1] John Backus, "The History of FORTRAN I, II and III". (1979)

[2] Wolfgang Borutzky, "Bond Graph Modelling of Engineering Systems Theory, Applications and Software Support". (2011)

[3] Senkel, A., Carsten, B., Schmitz, G., "Evaluating the Resilience of Energy Supply Systems at the Example of a Single Family Dwelling Heating System". (2019)

[4] Rahman, M., Avelin, A., Kyprianidis, K., "An Approach for Feedforward Model Predictive Control of Continuous Pulp Digesters". (2019)

[5] Fröjd, K., Axelsson, K., Torstensson, I., Åberg, E., Osvaldsson, E., Dolanc, G., Pregelj, B., Eborn, J., Pålsson, J. "Development of a Real -Time Fuel Processor Model for HIL Simulation". (2014)

[6] Sigfridsson, S., Li, L., Runvik, H., Gohl, J. ,Joly, A., Soltesz, K., "Modeling of Fuel Cell Hybrid Vehicle in Modelica: Architecture and Drive Cycle Simulation". (2018)

[7] Lind, A., Sällberg, E., Velut, S., Gallardo Yances, S., Åkesson, J., Link, K., "Start-up Optimization of a Combined Cycle Power Plant". (2012)

[8] Obialero, E., "A Refined Vehicle Dynamic Model for Driving Simulators". (2013)

[9] Coïc, C., Andreasson, .J, Pitchaikani, A., Åkesson, J., Sattenapalli, H, "Collaborative Development and Simulation of an Aircraft Hydraulic Actuator Model". (2020)

> "
>
> *Objectives drive principles. Principles drive process. Process drive software tools.*
>
> **Robert Halligan**

Requirements Writing Patterns – What are the Options?

by Robert J. Halligan FIE Aust CPEng IntPE(Aus)

Project Performance International

Email: rhalligan@ppi-int.com

Abstract

This paper deals with syntax of requirements, firstly describing a pattern developed and recommended by the author, referred to as the "Parsing Template", or commercially, the "PPI Parsing Template^{TM}". The Parsing Template contains a possible nine defined information elements. The organization of these elements to give a structural template for writing strong requirements in English is described. Use of the template for other languages is then discussed and illustrated. The paper concludes by surveying alternative requirements writing patterns.

Introduction

Writing requirements well in natural language challenges most people, based on the experience of the author in delivering requirements training to over 11,600 professionals and having performed numerous requirements consultancies. And yet, writing requirements well reduces what in the experience of the author has been the most common source of project problems – developing the wrong thing. Two main factors contribute to good requirements. The first is the information content, which comes about from the correctness and completeness with which requirements are created or captured. The second is the expression of the information content, which itself has two dimensions. Expression involves, for natural language, both word choice and syntax. This paper deals with syntax, defining nine information elements, two of which must be present in a requirement and seven of which may (or may not) need to be present. These pieces may be organized to give a structural template for writing requirements in English, a template or pattern developed by the author and referred to as the "PPI Parsing Template^{TM}" or for short in this paper, the Parsing Template.

Many other requirements patterns have appeared in the literature. These are summarized towards the end of this paper.

Natural language is only one of a few ways of expressing requirements. Often requirements are best expressed by means other than natural language, such as modeling languages (for example, SysML 2, Harel Statecharts) or mathematically-based requirements languages (for example, Z, OBJ), or even a sealed, reference physical sample. This paper does not evaluate the merits of the use of different language types. Rather, the content of this paper is predicated on a decision having been made to express requirements in natural language.

The Nine Information Elements in Context

The nine information elements potentially involved in the expression of a requirement are shown in Figure 1, organized to form the structural template for the English language. Figure 2 provides an example that incorporates all nine elements. Each of the elements is subsequently defined and explained. Note that any given requirement incorporates only the needed elements, the practical minimum, based on experience, appearing to be three despite the theoretical minimum of two (for example, *The product shall display*). The criterion for "being needed" is that, in a given set of circumstances, any solution that meets the requirement as written will, in those circumstances, satisfy the need that the specified requirement purports to communicate.

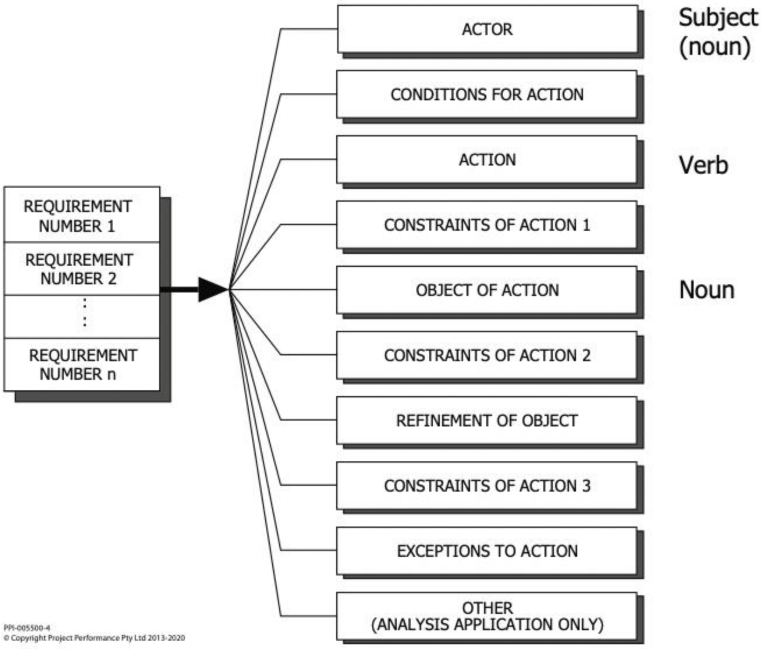
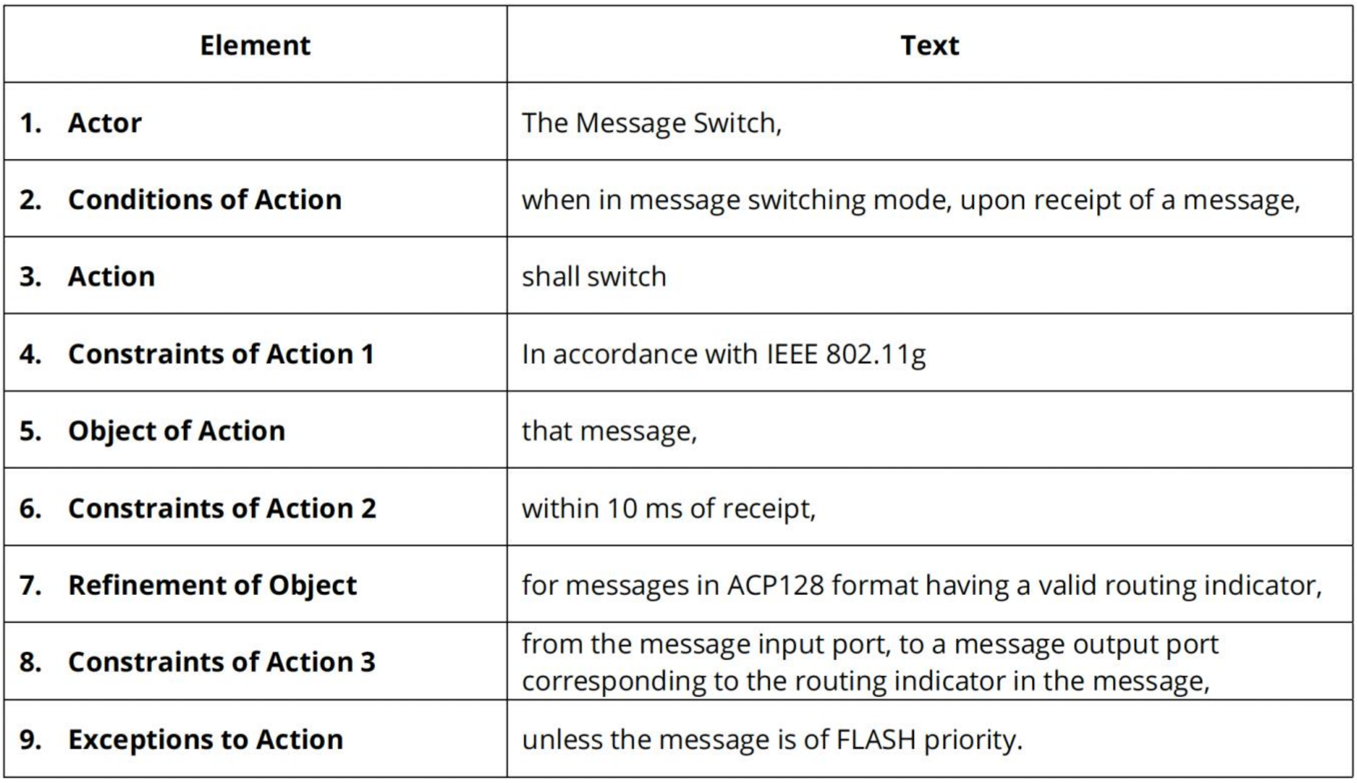


Figure 1: Illustration of the PPI Parsing Template™



Element	Text
1. Actor	The Message Switch,
2. Conditions of Action	when in message switching mode, upon receipt of a message,
3. Action	shall switch
4. Constraints of Action 1	In accordance with IEEE 802.11g
5. Object of Action	that message,
6. Constraints of Action 2	within 10 ms of receipt,
7. Refinement of Object	for messages in ACP128 format having a valid routing indicator,
8. Constraints of Action 3	from the message input port, to a message output port corresponding to the routing indicator in the message,
9. Exceptions to Action	unless the message is of FLASH priority.

Anonymous 2021. Details omitted for double-blind reviewing.

Figure 2 Illustration of a populated Parsing Template

The presence of three separate “Constraints of Action” fields will be observed. Also, similarity of the “Conditions for Action” and “Exceptions to Action” fields is apparent. The reasons for this apparent redundancy are explained below.

The Parsing Template lends itself to integration with model-based approaches implemented with contemporary Model-Based Systems Engineering (MBSE) tools, because of its precise and fixed syntax.

We now explain each of the elements of the Parsing Template.

Actor

The Actor is grammatically the subject of the sentence. As such, the Actor is a noun or a gerund (noun-verb combination). The Actor includes any indefinite article (such as the, a, an, …), together with any associated qualifying adjective or phrase. Examples of potential actors (but not necessarily *good* actors) are:

a. The system
b. Applicants who have been vaccinated
c. Computing
d. Security that is subject to Regulation 106-14
e. The product
f. The Mains Power Interface
g. The Very Fast Train
h. Users
i. Water.

Using the Parsing Template, the sentence always starts with the Actor. This is because the Actor is the primary discriminant for the applicability or relevance of the requirement, so starting the sentence with the Actor minimizes reading time and contributes to work efficiency. A sentence that is irrelevant because of a wrong Actor can be immediately passed over by the reader. For well-constructed requirements, the Actor will be the item that is the subject of the set of requirements, or a required interface of that item, with rare exceptions.

Conditions for Action

As the name implies, the Condition for Action is one or more conditions under which the requirement applies. Conditions for Action do not change the Action (see later). Conditions for Action are of two types:

a. Conditions during which the requirement applies, and/or
b. Triggering or initiating and terminating conditions,

in any mix of types and quantity. A requirement with multiple Conditions for Action is regarded as a singular (atomic) requirement because of the possibility of relationships between Conditions for Action. Another reason for this criterion for singularity is avoidance of an explosion in the number of requirements and the consequential impact on engineering work effort, by way of requirements specification, requirements traceability, development of verification requirements, design verification, development of item verification procedures, and verification of the item as meeting its requirements. This criterion, in practice, has a very large impact on “work avoided”, in the experience of the author.

Examples of potential Conditions for Action (but not necessarily *good* Conditions for Action) are:

a. if Inputs A and B are simultaneously present
b. having identified an error in the entered password

c. upon receipt of an On input at the User Control Interface
d. when in Executing State
e. from receipt of an On input to receipt of an Off input at the User Interface.

Conditions for Action always immediately follows the Actor. This is because the Conditions for Action is the next discriminator as to the applicability or relevance of the requirement, so following the Actor with Conditions for Action minimizes reading time, which contributes to work efficiency. The sentence that does not apply because the Condition for Action is not satisfied can be immediately passed over by the reader of the requirement.

Action

The Action is the verb as it relates to the Actor, together with any modal auxiliary verb (the complete set is *can*, *could*, *may*, *might*, *must*, *shall*, *should*, *will*, and *would*, although only two of these, *must* and *shall* are suitable for use in constructing requirements, based on dictionary meanings). The Action also includes any adverb(s). An adverb is a word or phrase that modifies or qualifies an adjective, verb, or other adverb or a word group, expressing a relation of place, time, circumstance, manner, cause, degree, etc. [1] (for example, *quickly*, *quite*, *then*, *there*, *not*).

The verb may be a passive verb such as *be*, *do*, or *have*, or an active verb such as *display*, *heat*, *raise* or *compute*.

Examples of potential Actions (but not necessarily *good* Actions) are:

a. shall be
b. shall display
c. shall have been developed
d. shall not display
e. shall not have been published
f. shall not have been published before.

The Action follows the Conditions for Action, to maximize ease of reading and understanding of the sentence.

Constraints of Action

Unlike Conditions of Action, Constraints of Action *do* qualify, reduce the scope of, the Action. A constraint of Action can be easily identified by placing the words immediately after the Action and answering the question “does this limit the action” (not the thing on which the Action is performed, if any). Constraints of Action are purposefully limited to one or more phrases, not usually individual words, to maximize the readability of the sentence. Individual words such as adjectives and adverbs are included as a part of the entity that they qualify, for example, “shall display clearly” is all Action, but “in a clear way” is a Constraint of Action with respect to the Action “shall display”.

Inspection of the Parsing Template shows three Constraints of Action fields. Having available the three fields placed differently with respect to Object of Action and Refinement of Object (see later) accommodates an enormously wide range of potential information content in a singular requirement. The approach also facilitates the avoidance of ambiguity. This latter point is illustrated at Table 1, where for the same set of elements, only the sequence defined by the Parsing Template, indicated in bold, is defect-free.

Table1: Effect of Variation of Placement of Phrases for an Example Requirement

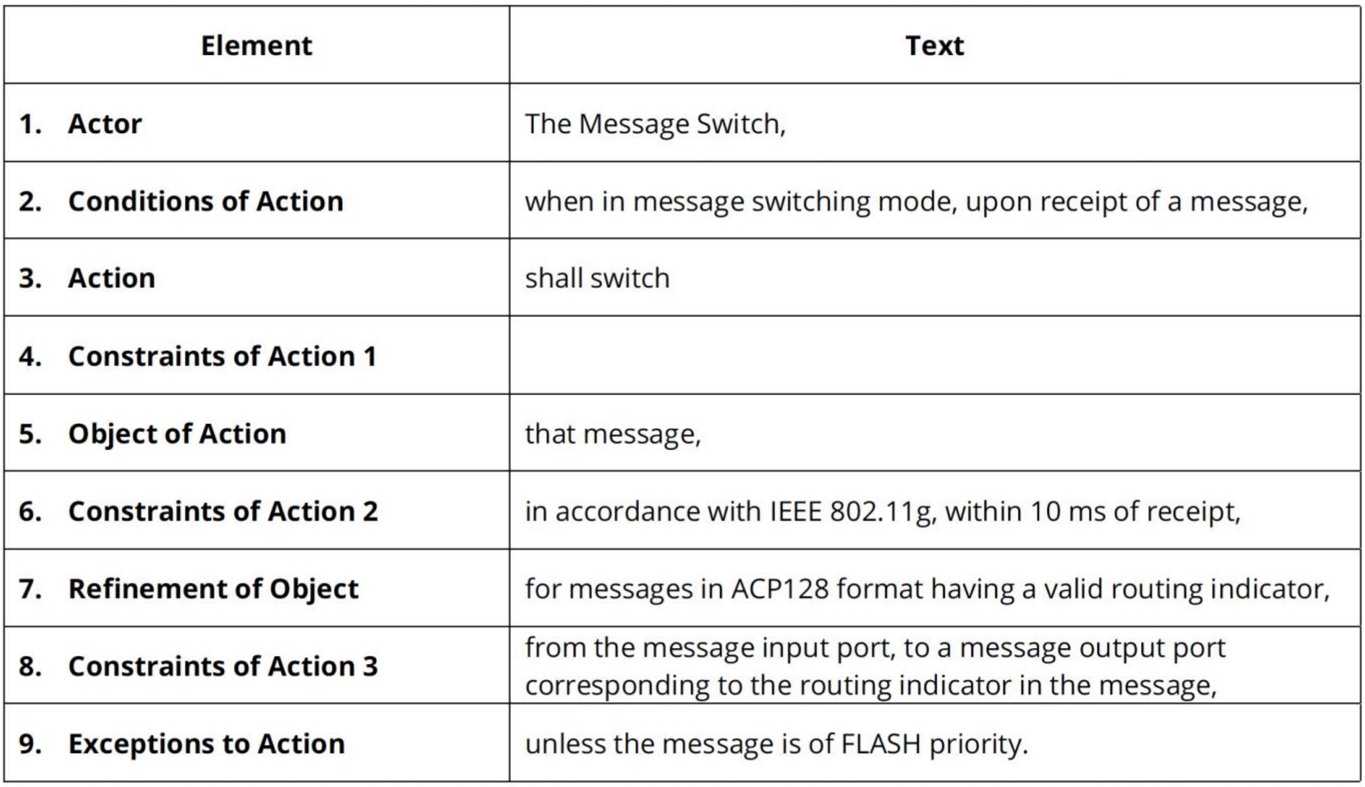
The car shall continuously display, when in reverse motion mode, an image at the center console display interface, indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%, unless the view of the obstacle is obstructed. *Issue: Is the displaying to be done at the centre console display interface, or is the image at the center console display interface to be displayed?*
The car, when in reverse motion mode, shall display, continuously, an image at the center console display interface, indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%, unless the view of the obstacle is obstructed. *Issue: Is the displaying to be done at the centre console display interface, or is the image at the center console display interface to be displayed?*
The car, when in reverse motion mode, shall continuously display an image at the center console display interface, indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%, unless the view of the obstacle is obstructed. *Issue: Is the displaying to be done at the centre console display interface, or is the image at the center console display interface to be displayed?*
The car, when in reverse motion mode, shall continuously display an image indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%, at the center console display interface, unless the view of the obstacle is obstructed. *Issue: Implies that the resolution and accuracy are to be at the center console display interface.*
The car, when in reverse motion mode, shall continuously display an image at the center console display interface, indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%, unless the view of the obstacle is obstructed. *Issue: Is the displaying to be done at the centre console display interface, or is the image at the center console display interface to be displayed?*
The car, when in reverse motion mode, shall continuously display an image at the center console display interface, indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%, unless the view of the obstacle is obstructed. *Issue: Is the displaying to be done at the centre console display interface, or is the image at the center console display interface to be displayed?*

The car, when in reverse motion mode, unless the view of the obstacle is obstructed, shall continuously display an image at the center console display interface, indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%.

Issue: Is the displaying to be done at the centre console display interface, or is the image at the center console display interface to be displayed?

The car, when in reverse motion mode, shall continuously display at the center console display interface, an image indicating to the driver the distance in metres between the car and the nearest obstacle behind the car, to a resolution of 0.01m and an accuracy of +/-2%, unless the view of the obstacle is obstructed.

Another example of the effect of variation of placement of a Constraint of Action is at Figure 3. The Figure 3 version offers improved readability in comparison with the Figure 2 version.



Element	Text
1. Actor	The Message Switch,
2. Conditions of Action	when in message switching mode, upon receipt of a message,
3. Action	shall switch
4. Constraints of Action 1	
5. Object of Action	that message,
6. Constraints of Action 2	in accordance with IEEE 802.11g, within 10 ms of receipt,
7. Refinement of Object	for messages in ACP128 format having a valid routing indicator,
8. Constraints of Action 3	from the message input port, to a message output port corresponding to the routing indicator in the message,
9. Exceptions to Action	unless the message is of FLASH priority.

Anonymous 2021. Details omitted for double-blind reviewing.

Figure 3 Improved placement of Constraints of Action

Examples of potential Constraints of Actions (but not necessarily *good* Constraints of Actions) are:

a. as specified herein
b. in accordance with IEEE 802.11g
c. such that contact does not occur
d. within 10ms.

As can be seen from the examples, Constraints of Action include performance (how well a function is to be performed), and can also take many other forms.

Object of Action

Simplistically, the Object of Action is the thing acted upon by the Actor in taking the Action. This criterion is fine for action verbs. The criterion is less convincing but still satisfactory for passive verbs. Examples of Action/Object of Action combinations are:

a.	Action: shall be	Object of Action: a fuel cell
b.	Action: shall display	Object of Action: current time
c.	Action: shall have	Object of Action: a Lifting Interface
d.	Action: shall not be above	Object of Action: 5%.

Refinement of Object

A Refinement of Object is one or more phrases that qualifies/that qualify the Object of Action. Examples of Object of Action/Refinement of Object combinations are:

a.	Object of Action: 5%	Refinement of Object: measured at a height of 1m
b.	Object of Action: a fuel cell	Refinement of Object: approved by the FAA
c.	Object of Action: a Lifting Interface	Refinement of Object: located as shown at Fig. 4.5.7-1
d.	Object of Action: current time	Refinement of Object: in hh-mm-ss format

Exception to Action

Exceptions to Action are similar to Conditions for Action except that they are exclusive, as contrasted to inclusive for Conditions of Action. Exceptions to Action are placed as the last element in the template, because clarity is improved without a significant increase in average reading time (given the small percentage of requirement usage in which an Exception to Action appears and the exception applies).

Examples of Exceptions to Action are:

a. except in Off State
b. unless of Flash priority.

Examples of Requirements Written in the Parsing Template Format

The efficacy of the Parsing Template may be judged by reviewing example requirements and comparison with other writing strategies. Examples follow.

The Child, while in an awake state, shall maintain perfect silence, unless given permission to speak.

The Waiter, in waiting mode, upon arrival of each guest, shall deliver to that guest a current-day menu in both English and Spanish, within 30 seconds of that arrival, unless the arrival of the guest has occurred within 30 minutes of posted kitchen closing time.

Usability with Other Languages

The author, having demonstrated the efficacy of the English language version of the template by extensive usage within PPI's client community, has investigated application to the Chinese language. Initial results have been promising, the Chinese language template taking the form:

1 Actor 主语
2 Conditions for Action 前提条件
3 Constraints of Action 1 状语/补语 1
4 Constraints of Action 2 状语/补语 2
5 Action 谓语

6 Refinement of Object 定语（修饰宾语)
7 Object of Action 宾语
8 Constraints of Action 3 状语/补语 3
9 Exceptions to Action 例外情况

The contributions to this work of PPI's Victoria Huang and Caitlyn Liu are gratefully acknowledged. Future work is planned to validate the Chinese version of the template by trial use in engineering environments in which Chinese is the first language.

The nature of the Parsing Template, based on information elements that do not assume any particular requirements subject matter, also suggests applicability of the information elements to model-based approaches. Investigation of this aspect using SysML 2.0 is planned.

Alternative Requirements Patterns

Table 2 lists other requirements patterns known to the author.

Line-oriented languages such as Gherkin [13], YAML [14] and Python [15] are also relevant to expression of requirements using fragments of natural language. Gherkin is a natural language parser central to a behavior-driven development supported by a software tool called Cucumber. YAML is a human-readable data-serialization language. Python is a high-level, general-purpose programming language. Such languages, however, are outside the scope of this paper.

Table 2 Other requirements patterns known to the author

Name of the Pattern and Reference	Comments
R. Fuji, 1989 [2]	A 7-element pattern promoted for finding defects in requirements. Has three elements identical to the PPI Parsing Template.
R. Halligan (1993) [3]	A 7-element pattern used for measuring requirements quality, extending the work of Fuji [2]
U.K. Future Surface Combatant (FSC) Project requirement patterns, 1998, reported by [4]	A set of about 120 patterns constructed from a standard set of pattern elements, such as operational condition, user, quantity, time unit, system function, Action, entity, event, and many others. The reference concluded that the approach was clearly beneficial.
UK BAA Heathrow Terminal 5 Project requirement patterns, 2002, reported by [4]	A 7-element "capability template", comprising pattern elements of Actor, Act, Action Subject, Qualification, Qualification Subject, Dispatch Act, Dispatch Object. Of the patterns known to the author, this is the only pattern similar in concept to the Parsing Template, beyond the original Fuji pattern [2].
Planguage, T. Gilb, 2001 [5]	A language with a rich set of keywords that permits precise statements of individual quality attributes and other project goals, using one line per information element. Weak in opinion of the author in expressing the relationships between the information elements of a requirement. Contains much more than the actual requirement.

Name of the Pattern and Reference	Comments
Scott, Kasser & Tran, 2006 [6]	A 9-element pattern expressed in Entity Relationship Attribute format, the elements comprising System, Action, Capability, Attribute, Subcomponent, Constraint, Target, Performance, and Condition.
EARS – Easy Approach to Requirements Syntax 2009 [7]	A set of four templates of limited expressiveness, in the author's assessment.
ISO/IEC/IEEE 29148:2011 [8]	A 5-element pattern involving Condition, Subject, Action, Object, and Constraint. Assessed by the author to be logically sound, but with issues that limit its successful practical application. Omitted from ISO/IEC/IEEE 29148:2018.
PABRE 2011 [9]	PABRE-Man provides a number of topic-specific patterns for software and IT-related requirements. The author assesses these patterns to be individually problematic in relation to sound requirements engineering practice.
Hull et al. 2012 [10]	Pages 44-71, provides some guidance on requirement structure.
Boeing - Carson 2015 [11]	A set of four, 5-element patterns involving Agent, Shall, What, How well, Under what conditions. Assessed by the author to be logically sound, but with issues that limit their consistent, practical application.
RAT – Authoring Tools (The Reuse Company) 2021 [12]	The tool provides a set of user-selectable patterns. All are assessed by the author to be useful. None addresses in full the issues addressed by the Parsing Template.

Conclusion

The PPI Parsing Template has been exposed as providing a highly effective framework for writing clear, precise, efficient-to-read requirements in English, the efficacy of doing so having been both argued and illustrated in this paper with examples. The benefits of clear, precise and efficient-to-read requirements are self-evident. Beyond the main focus of this paper, English language expression of requirements, the Parsing Template has potential for successful use with natural languages other than English, and with model-based expression of requirements. PPI is planning to undertake future investigation in these directions.

The Parsing Template is recommended for adoption by practicing engineers and enterprises who to any extent write requirements in English. The concept of the Parsing Template is recommended for consideration by engineers who express requirements in languages other than English, natural and otherwise.

References:

[1] *Oxford Languages* 2021, <https://languages.oup.com/google-dictionary-en/> Accessed 31 August 2021

[2] Fuji, R 1989, *training course notes on Independent Verification and Validation*.

[3] Halligan, R, 2015. Project Performance International internal documentation and training courseware

[4] Dick, J and Llorens, J 2012, Using statement-level templates to improve the quality of requirements, October 2012, *24th International Conference on Software & Systems Engineering and their Applications*, Paris.

[5] Gilb, T 2001, *A Handbook for Systems and Software Engineering using Planguage*, Addison Wesley.

[6] Scott, W, Kasser, J and Xuan-Linh Tran 2006. Improving the Structure and Content of the Requirement Statement, INCOSE.

[7] EARS – Easy Approach to Requirements Syntax, <http://ieeexplore.ieee.org/document/5636542/>, 2009.

[8] ISO/IEC/IEEE 29148:2011 *Systems and software engineering - Life cycle processes - Requirements engineering*.

[9] *The PABRE Catalog*, 2011: <http://www.upc.edu/gessi/PABRE/index.html>.

[10] Hull et al 2012, *Requirements Engineering*, Springer.

[11] Carson, R 2015. “Implementing Structured Requirements to Improve Requirements Quality.” *Proceedings of the 25th Annual INCOSE International Symposium.*

[12] RAT – Authoring Tools (The Reuse Company), 2021, Accessed 21 August 2021 https://www.reusecompany.com/rat-authoring-tools

[13] Gherkin Syntax, https://cucumber.io/docs/gherkin/, accessed 18 March 2022.

[14] https://yaml.org/, accessed 18 March 2022.

[15] https://www.python.org/, accessed 18 March 2022.

> “
>
> *High achievement always takes place in the framework of high expectation.*
>
> **Charles F. Kettering**

SYSTEMS ENGINEERING RESOURCES

Useful artifacts to improve your SE effectiveness

Systems Engineering Journal - MBSE Special Edition

SYSTEMS ENGINEERING
The Journal of The International Council on Systems Engineering
INCOSE

INCOSE, through the Wiley online library, has published a Special MBSE Edition of the Systems Engineering Journal titled "Model-Based Systems Engineering and Design Approaches". This December, 2021 release includes six articles; SyEN has included abstracts to guide our readers to which of these mind-stretching topics best fit their interests.

Unified design approach for systems engineering by integrating model-based systems design with axiomatic design
Authors: Haoqi Wang, Hao Li, Chengtong Tang, Xu Zhang, Xiaoyu Wen
Abstract: Model-Based Systems Design (MBSD) formalizes the application of system modeling. However, it is still difficult for designers to implement MBSD in practice. One of the challenges is to make the field of design more unified, as different MBSD methodologies focus on specific areas, and various designers from different areas use ambiguous terms.

Moreover, the failure of some products originates in undiscovered couplings of functions in the design stage. Axiomatic Design (AD) provides a scientific foundation for system design. In order to reduce ambiguity and alleviate functional couplings, this paper integrates AD with the MBSD and proposes a unified systems design approach, called Axiomatic Model-Based Systems Design (A-MBSD). A-MBSD uses four pillars: (a) the fundamental framework of AD; (b) the Independence Axiom; (c) a new behavior domain (BD); and (d) an A-MBSD modeling profile.

Finally, the design of a Forest Fire Satellite Monitoring and Control System demonstrates that the proposed approach can improve unity and alleviate the coupling of functions in a system design.

Model-based design of project systems, modes, and states
Authors: Ali Shafaat, C. Robert Kenley
Abstract: The field of project management has been trying to find generalizable sets of rules, and this article applies approaches and methods from systems engineering to develop such rules that support flexible project design. It presents an approach for modeling projects that divides project systems into three interacting subsystems (planned, executed, and interface systems) and proposes a framework for designing projects that integrates project implementation, verification, and validation.

To build a generalizable model, this article revisits the notation of modal systems developed by Wymore and combines it with more recent developments in systems engineering, project design, and management. The temporary and changing environments of projects are addressed by adjusting the Wymorian definitions and related mathematical models to represent projects as a sequence of transient modes rather than activities. This approach enables better design and control of projects by reducing uncertainty and its consequences through a systematic framework for planning, monitoring, and controlling projects.

Exploring and managing the complexity of large infrastructure projects with network theory and model-based systems engineering - The example of radioactive waste disposal

Author: Andreas Poller

Abstract: Given that model-based systems engineering (MBSE) captures the structure and behavior of an engineered system in an overarching system model, MBSE appears to be a promising approach to managing large infrastructure projects (LIPs). However, it is not apparent how to most appropriately organize the associated system model - and hence the infrastructure project itself. Furthermore, MBSE may today not be readily accepted by the civil engineering industry.

In this research, a hypothetical project for the geological disposal of radioactive waste is taken as an example of an LIP and initial system models of the entire disposal project are created. Furthermore, a network representation of the project is generated and examined with network theory. Based on the results, different project organizations are synthesized and evaluated. Eventually, the initial system models are updated to accommodate the most suitable organization according to the network analysis results. In addition, the perception of, and attitude toward MBSE is assessed by means of a cross-sectional survey in a civil engineering company.

The generation of system models of LIPs is found to be straightforward. Network theory is able to unveil the complex structure of LIPs in order to identify the most suitable way to organize them and the associated system models. The survey results suggest that MBSE may find broad acceptance in the civil engineering industry.

Operationalizing digital twins through model-based systems engineering methods

Authors: Jason Bickford, Douglas L. Van Bossuyt, Paul Beery, Anthony Pollman

Abstract: In recent years there has been increased demand for readiness and availability metrics across many industries and especially in national defense to enable data-driven decision making at all levels of planning, maintenance, and operations, and in leveraging integrated models that inform stakeholders of current operational system health and performance metrics.

The digital twin (DT) has been identified as a promising approach for deploying these models to fielded systems although several challenges exist in wide adoption and implementation. Two challenges examined in this article are that the nature of DT development is a system-specific endeavor, and the development is usually an additional effort that begins after initial system fielding. A fundamental challenge with DT development, which sets it apart from traditional models, is the DT itself is treated as a separate system, and therefore the physical asset/DT construct becomes a system-of-systems problem.

This article explores how objectives in DT development align with those of model-based systems engineering (MBSE), and how the MBSE process can answer questions necessary to define the DT. The key benefits to the approach are leveraging work already being performed during system synthesis and DT development is pushed earlier in a system's lifecycle. This article contributes to the definition and development processes for DTs by proposing a DT development model and path, a method for scoping and defining requirements for a DT, and an approach to integrate DT and system development. An example case study of a Naval unmanned system is presented to illustrate the contributions.

Toward a better integration of requirements and model-based specifications

Authors: Benoît Lebeaupin, Antoine Rauzy

Abstract: As of today, most specifications of technical systems still rely on requirements written in natural language. However, this approach is known to be problem-prone, due to the inherent ambiguity of natural languages. On the other hand, fully formal or model-based approaches seem to be out of reach in many practical cases, especially in early design phases of systems.

In this article, we study how to combine in a pragmatic way natural language requirements with models. We propose to keep both formats and to link pieces of text in requirements with elements of models. In other words, corpuses of requirements are managed as hypertexts with links to models. For this approach to be fully efficient, the text of requirements is not free, but relies on controlled natural language techniques leading to a partial structuring of the text. We show that this makes it possible to design (semi)automatic verifications on requirements and models, which would be impossible with unconstrained natural language. We illustrate here our approach on a small illustrative example and we report results obtained on a full size industrial application.

Synergizing model-based systems engineering, modularity, and software container concepts to manage obsolescence
Authors: Markeeva Morgan,Thomas Holzer, Timothy Eveleigh
Abstract: Costs to replace obsolete parts are estimated to exceed $10 billion annually for military expenditures. Other industries experience high costs related to obsolete parts as well. Synergies between Model-Based Systems Engineering (MBSE) and modular design concepts should be leveraged to reduce those costs early in the design process.

To date, obsolescence management, MBSE, and modularity have been studied, advanced, and presented in a mostly disjunct manner. Consequently, work to synergize these areas specifically to mitigate obsolescence impacts is lacking. The research presented herein illuminates that the untapped synergy at the nexus of obsolescence management, MBSE, and modular design can be leveraged to enhance strategic management of obsolescence.

Borrowing from the modular software engineering "container" concept, the MBSE container modeling approach presented provides systems engineers, architects, and designers with a novel method to influence the system design by minimizing the number of system modules affected by obsolescence. By applying this model-based strategic obsolescence management approach early in the system architecting and design stages, system engineering teams can reduce the associated complexity, lifecycle costs, and risk of obsolete parts replacement.

INCOSE members in good standing may access all Systems Engineering Journal content through their INCOSE Connect login. Non-members may subscribe to the journal or purchase access to individual articles at the URLs associated with the article titles, above.

Resilience Engineering Association Newsletter - Compounding Disasters

The Resilience Engineering Association (REA) addresses the subject of compounding disasters in its latest newsletter. Compounding disasters are concurrent events or cascading strenuous incidents that call for prolonged resilience; all of which are becoming increasingly prevalent in our social, business, and political systems.

Topics include:

- Unique challenges of recent compounding disasters and the role of resilience engineering
- Nurses' utilization and extension of adaptive capacity in sustaining hospital care during the Covid-19 pandemic
- Outmaneuvering complexity in worlds of surprise
- Want to better understand resilience? Study homeostasis!

Access the REA newsletter.

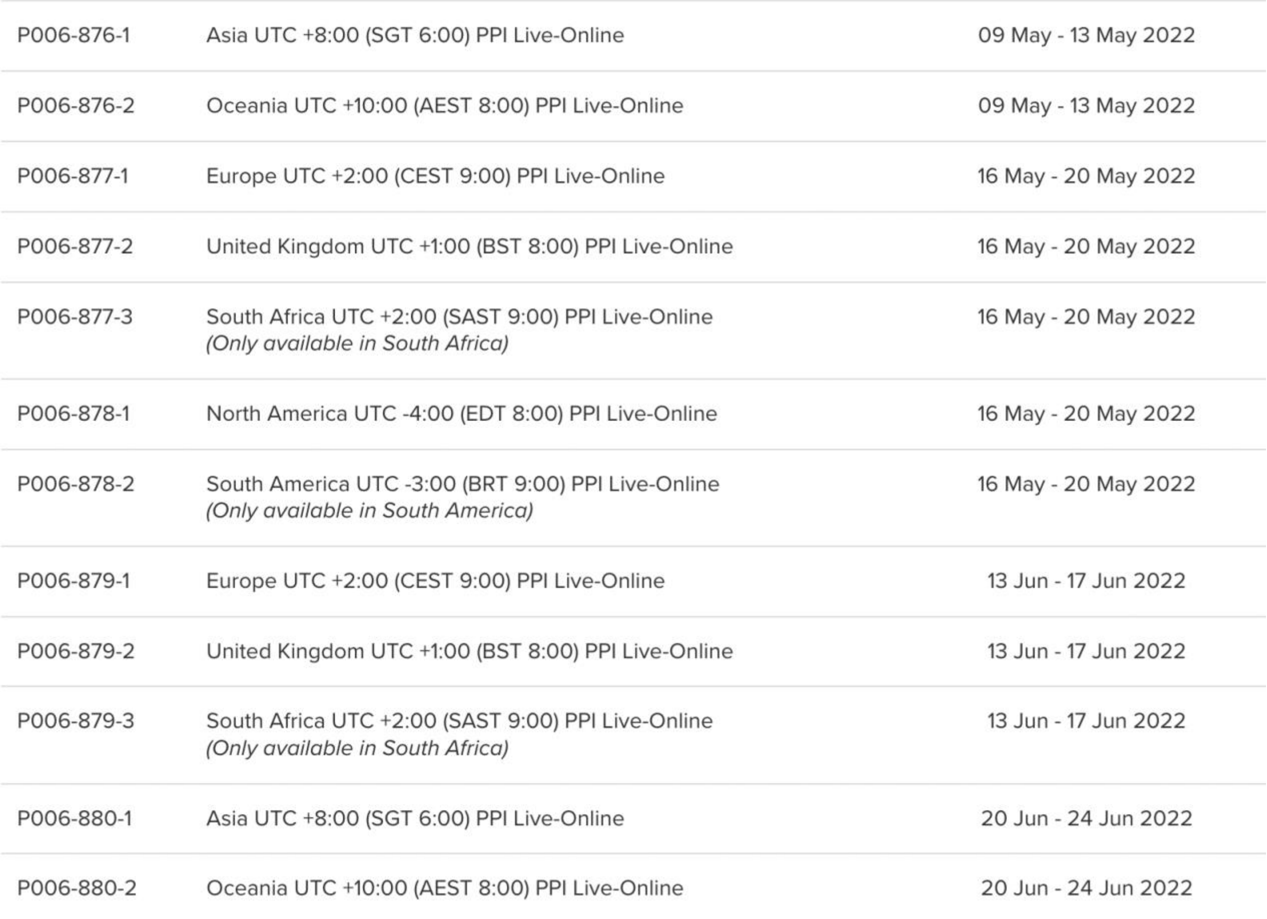
System of Systems Modeling with Capella

OBEO has made available the video and slides associated with a recent webinar on System of Systems Modeling with Capella. Presented by Tony Komar, Siemens PLM MBSE Evangelist, the webinar addresses the scalability of Capella to meet system of system development challenges. The webinar provides seven specific "How-to" guidelines for System of Systems (and IOT) modeling using Capella and the ARCADIA method.

Download the slides. View the webinar.

Upcoming PPI Live-Online ™ Systems Engineering Five Day Courses

Click here to view the full schedule or register for an upcoming course.



P006-876-1	Asia UTC +8:00 (SGT 6:00) PPI Live-Online	09 May - 13 May 2022
P006-876-2	Oceania UTC +10:00 (AEST 8:00) PPI Live-Online	09 May - 13 May 2022
P006-877-1	Europe UTC +2:00 (CEST 9:00) PPI Live-Online	16 May - 20 May 2022
P006-877-2	United Kingdom UTC +1:00 (BST 8:00) PPI Live-Online	16 May - 20 May 2022
P006-877-3	South Africa UTC +2:00 (SAST 9:00) PPI Live-Online *(Only available in South Africa)*	16 May - 20 May 2022
P006-878-1	North America UTC -4:00 (EDT 8:00) PPI Live-Online	16 May - 20 May 2022
P006-878-2	South America UTC -3:00 (BRT 9:00) PPI Live-Online *(Only available in South America)*	16 May - 20 May 2022
P006-879-1	Europe UTC +2:00 (CEST 9:00) PPI Live-Online	13 Jun - 17 Jun 2022
P006-879-2	United Kingdom UTC +1:00 (BST 8:00) PPI Live-Online	13 Jun - 17 Jun 2022
P006-879-3	South Africa UTC +2:00 (SAST 9:00) PPI Live-Online *(Only available in South Africa)*	13 Jun - 17 Jun 2022
P006-880-1	Asia UTC +8:00 (SGT 6:00) PPI Live-Online	20 Jun - 24 Jun 2022
P006-880-2	Oceania UTC +10:00 (AEST 8:00) PPI Live-Online	20 Jun - 24 Jun 2022

FINAL THOUGHTS FROM SYENNA

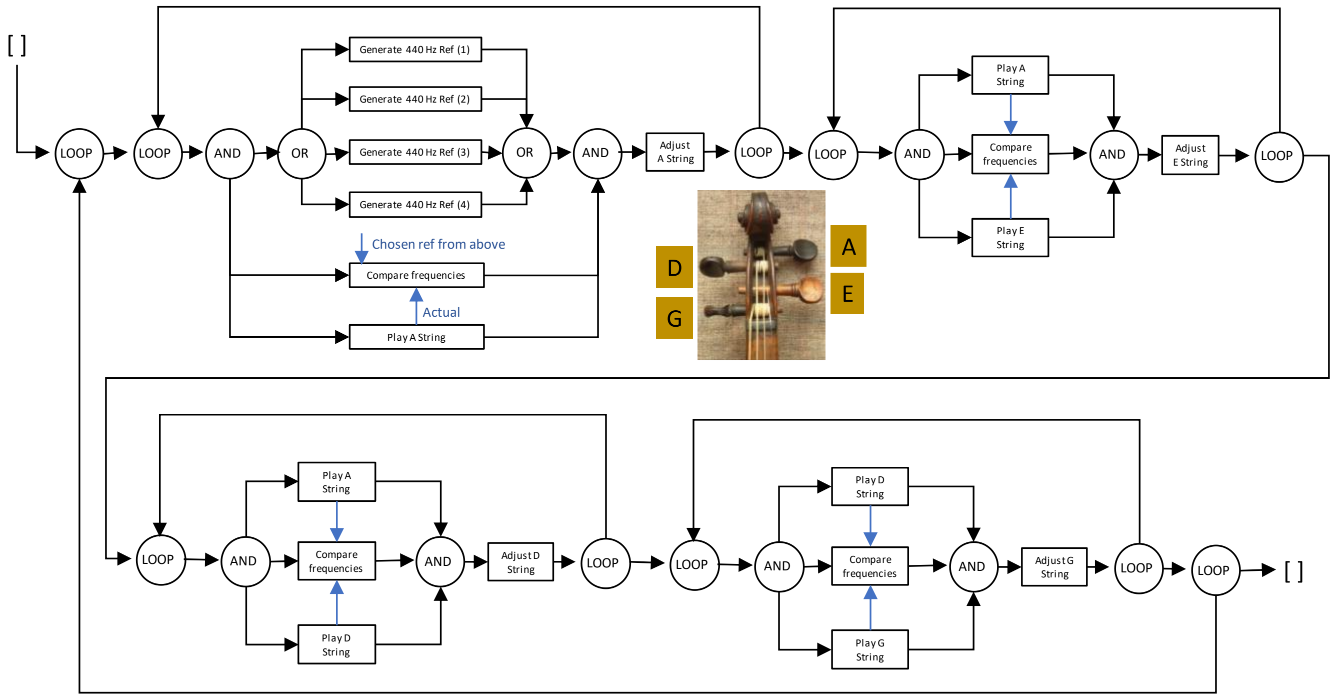
Dear Reader,

I sometimes like to try out Systems Engineering methods outside of the workplace. This month I wanted to share what happened when I applied Function Flow Block Diagrams (FFBDs) to tuning my violin – see diagram below.

A diagram is intended to save a thousand words, but a few points may be of interest:

- We normally start by tuning the A string to "concert" pitch (440 Hz in modern times, but it used to be lower). Various sources can be used for the reference frequency. Typically, a concert orchestra will use the oboe. In other situations, a tuning fork, piano, organ, or other instrument may be used. Some people even have a capability known as "absolute pitch" (or "perfect pitch"), which means that they can hear or sing a given note without any external reference;
- We then tune pairs of strings to achieve perfect fifths between them. The frequency ratio should be close to 3:2, which can be detected by listening for beating effects;
- The violin bow can normally only play two adjacent strings simultaneously, and hence it isn't possible to compare the G to the A. This illustrates that you need a knowledge of a candidate physical architecture when devising the FFBD, and is something you might miss if you didn't engage somebody with domain knowledge in the process;
- Having tried out the string pairs, it's normal to go back to the start and make finer adjustments. For one thing, the violin may flex as the pegs are adjusted, which may disturb earlier tunings. There is also the question of the "tempering" of the musical scale, which is a subject in its own right.

I hope this is of interest to you and that you might like to try it on your own hobbies. Send in your results to ppisyen@ppi-int.com for potential inclusion in a future edition!



[ ]
LOOP
LOOP
AND
OR
Generate 440 Hz Ref (1)
Generate 440 Hz Ref (2)
Generate 440 Hz Ref (3)
Generate 440 Hz Ref (4)
OR
AND
Chosen ref from above
Compare frequencies
Actual
Play A String
Adjust A String
LOOP
LOOP
AND
Play A String
Compare frequencies
Play E String
AND
Adjust E String
LOOP
D
G
A
E
LOOP
AND
Play A String
Compare frequencies
Play D String
AND
Adjust D String
LOOP
LOOP
AND
Play D String
Compare frequencies
Play G String
AND
Adjust G String
LOOP
LOOP
[ ]