



SYSTEMS ENGINEERING NEWSJOURNAL

PPI SyEN 95 – 27 November 2020

Dedicated to inspiring and improving the practice of systems engineering

brought to you by

Project Performance International (PPI)

Systems engineering training and consulting for project success ...

Welcome to the latest edition of PPI SyEN, PPI's monthly publication filled with informative reading for the technical project professional. In this issue, as well as in tons of archived issues available free at www.ppi-int.com, you will find powerful ideas that may help you prosper in the highly competitive world of systems and product development.

Access a mix of news, quick tips, short reads and deep article content that will help you expand your professional skill set, keep up to date on events and activities of interest, and quite possibly unlock levels of project performance you've never before considered possible.

We hope that you find this newsletter to be informative and useful. As the leading provider of systems engineering training worldwide, PPI is passionate about improving project outcomes. Thousands of professionals in aerospace, medical, consumer and other major sectors subscribe to PPI SyEN, as do leading thinkers in academic, government and scientific organizations.

Our editors strive to bring you a diverse range of views and opinions each month, but please know that the views expressed in externally authored articles are those of the author(s), and are not necessarily those of PPI or its professional staff.

Have a topic you would like to learn more about, or possibly ideas you'd like to share with others? We'd love to hear from you! Just email us at syen@ppi-int.info

We would also love it if you shared this edition with others who may benefit, and we encourage you to join our active community on [LinkedIn](https://www.linkedin.com/company/ppi-international). If a colleague sent you this copy, you can easily receive future newsletters directly by signing-up using the form at www.ppi-int.com.

Should you no longer wish to receive PPI SyEN for any reason, simply unsubscribe by clicking the link at the bottom of this email.

IN THIS EDITION

1. Quotations to Open On

[Read More...](#)

2. Feature Articles

- 2.1 A Preview of the Next Generation System Modeling Language (SysML v2) *by Sanford Friedenthal and Ed Seidewitz*
- 2.2 The Greatest Young System Engineers of the Year Challenge *by Ad Sparrius*

[Read More...](#)

3. Additional Articles

- 3.1 INCOSE Program Management and Systems Engineering (PM/SE) Integration Working Group: Bridging the Gap *by John Lomax, Jean-Claude Roussel, Tina Srivastava, Rachel Mouring, Mark Kaufman, and Ralph Young*
- 3.2 The Top Six Myths of PLM *by Lionel Grealou*
- 3.3 INCOSE Colorado USA Front Range Chapter Sponsors Virtual Presentation concerning Resilient Systems

[Read More...](#)

4. Systems Engineering News

- 4.1 A Model Program for Development of Young Systems Engineers is Available for Replication!
- 4.2 Message from the INCOSE President: INCOSE's International Workshop (IW) 2021 will be a Virtual Event
- 4.3 AIAA's Journal of Aerospace Information Systems Announces a Call for Papers on Systems Engineering's Top Space Challenges
- 4.4 INCOSE 2020 Election Results
- 4.5 Don't Just Lead Your People Through Trauma. Help Them Grow
- 4.6 New Research Program for Smart City Solutions in India

- 4.7 PRESENTATION: Model-Based Systems Engineering, a Critical Enabler for Digital Transformation (Part 1 of 3)
- 4.8 Job Opportunity: Senior Director of Systems Engineering at Smiths Medical

[Read More...](#)

5. Featured Organizations

- 5.1 German Aerospace Center (DLR) Institute of Systems Engineering for Future Mobility is Researching New Methods for Traffic Systems
- 5.2 American Society for Engineering Management

[Read More...](#)

6. News on Software Tools Supporting Systems Engineering

- 6.1 Vitech Corporation's GENESYS® and the INCOSE South Africa Graduate Young Systems Engineer of the Year (GYSEOY) 2020 Competition
- 6.2 Eclipse Papyrus™ Modelling Environment
- 6.3 Maplesoft™ Announce the Release of MapleMBSE 2020.2
- 6.4 Maplesoft™ Announces the Release of MapleSim
- 6.5 Gaphor UML/SysML Modeling

[Read More...](#)

7. Systems Engineering Publications

- 7.1 IEEE Systems Journal
- 7.2 On the Architecture of Systemology and the Typology of Its Principles
- 7.3 INCOSE's Guide for the Application of Systems Engineering in Large Infrastructure Projects
- 7.4 Brightline's Ten Guiding Principles for Bridging the Gap between Strategy Design and Delivery
- 7.5 INCOSE INSIGHT Practitioners Magazine Volume 23 Issue 3
- 7.6 Engineering a Safer World: Systems Thinking Applied to Safety

[Read More...](#)

8. Education and Academia

- 8.1 Master of Science in Industrial and Systems Engineering National University of Singapore
- 8.2 Southern Methodist University (USA) Offers Five Master's Degrees in Engineering
- 8.3 ITEA3 Project 'BUMBLE' – Blended Modelling for Enhanced Software and Systems Engineering

[Read More...](#)

9. Some Systems Engineering-Relevant Websites

[Read More...](#)

10. Standards and Guides

- 10.1 Object Management Group is working to Enable Interoperability with Shareability Terminology
- 10.2 Developing Cyber Resilient Systems: A Systems Security Engineering Approach
A NIST Special Publication 800-160 Volume 2
- 10.3 Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems

[Read More...](#)

11. Some Definitions to Close On

[Read More...](#)

12. Conferences and Meetings

- 12.1 IEEE Conference on Decision and Control (CDC) – Virtual, Date: 14 December – 18 December 2020
- 12.2 Another Notable Conference 27th International Conference on Systems Engineering

[Read More...](#)

13. PPI and CTI News

13.1 The PPI Website Gets a Facelift! – Give Us Your Feedback

13.2 Team PPI is Growing Despite Challenging Times

13.3 Become a SyEN Contributor?

13.4 PPI SyEN Gets a New Name

[Read More...](#)

14. PPI and CTI Events

[Read More...](#)

15. PPI Upcoming Participation in Professional Conferences

[Read More...](#)

1. QUOTATIONS TO OPEN ON

Systems engineering will have attained its rightful place when the principles and methods are taught in every MBA Program, not called systems engineering, but represented as the principles and supporting methods of problem definition and problem solving.

Robert John Halligan

The purpose of all systems engineers is to serve humanity.

Ad Sparrius

An objective of systems engineering on any project is to help the team achieve the same understanding. Be the communicator among the disciplines.

David Long

Traumatic moments – whether it’s a personal tragedy or a global crisis – can throw even the most resilient among us off track. But these challenges can also help us grow. Today, as we face a pandemic and its economic consequences, leaders can encourage growth. One way to start? Openly express your values, the principles that matter most to you and your organization, in good times and bad. Then follow up with actions based on those values. Such actions inspire trust and stability, at a time when it’s needed most.

Jamil Zaki

2. FEATURE ARTICLES

2.1 A Preview of the Next Generation System Modeling Language (SysML v2)

by

Sanford Friedenthal, SAF Consulting, safriedenthal@gmail.com and

Ed Seidewitz, Model Driven Solutions, ed-s@modeldriven.com

Version 1.0 September 2020

Abstract

The OMG Systems Modeling Language™ (SysML®) was adopted in 2006 and has been used by many organizations to support their efforts to transition to a model-based systems engineering (MBSE) approach. This paper provides an introduction to SysML v2, the next generation Systems Modelling

Language, which is intended to address many of the limitations of SysML v1. The content of this paper reflects the state of SysML v2 as of the initial submission of the specification to the Object Management Group (OMG) in September 2020. The final submission of the specification is planned for 2021. This paper provides the background and motivation for SysML v2, and an introduction to SysML v2 that highlights some of the differences with SysML v1.

Copyright © 2020 by Sanford Friedenthal and Ed Seidewitz. All rights reserved.

Introduction

This paper provides an introduction to SysML v2 as of the time of the initial submission to the OMG in September, 2020. SysML v2 is intended to enable the application of model-based systems engineering (MBSE). In particular, the emphasis for SysML v2 is to improve the precision, expressiveness, consistency and integration of the language concepts, and the interoperability relative to SysML v1. SysML v2 expresses the core concepts required to precisely specify a system, its elements, and its environment (i.e., the system model). SysML v2 also includes a standardized Application Program Interface (API). The API is intended to further enhance interoperability by specifying standard services to access SysML v2 models.

This paper includes the background and motivation for SysML v2, summarizes the requirements for SysML v2, provides a brief overview of the SysML v2 Submission Team (SST) that is the industry team developing the SysML v2 specifications, introduces the SysML v2 language and API, briefly describes considerations for transitioning to SysML v2, and summarizes the remaining work to be done leading up to the final submission in 2021.

Background and motivation

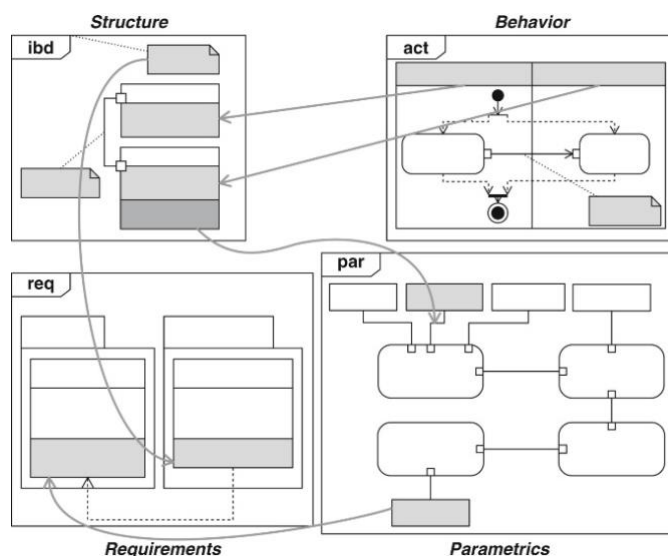
Model-based systems engineering (MBSE) emphasizes the creation and use of model-based artifacts as part of an overall system model to represent the system under development. This contrasts with more traditional document-based methods where information about the system is captured in text documents, spreadsheets, and less formal diagram representations. The MBSE approach can provide artifacts that are more precise, consistent, and traceable versus document-based artifacts, and also can facilitate shared understanding among different discipline engineers and other stakeholders involved in system development.

SysML was developed to provide a standard modeling language to represent systems, and enable a MBSE approach. The requirements for SysML v1 were developed as a joint effort between the OMG, members of the International Council on Systems Engineering (INCOSSE), and the ISO STEP 10303-233 Working Group. The SysML v1 specification was adopted by the OMG in 2006 as a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, people, procedures, and facilities.

SysML is maintained by the SysML Revision Task Force (RTF), which is a body within the OMG consisting of end user and tool vendor representatives. SysML has undergone six revisions, and the

seventh revision (SysML v1.7) is expected to be finalized in 2021. SysML v1.7 is expected to be the last revision of SysML v1 prior to the release of SysML v2.

SysML v1 is a profile of UML (*reference 2*) that uses the standard extension mechanism provided by UML to extend the language to address systems modeling concepts. SysML v1 represents systems in terms of what often are referred to as the four (4) pillars shown in Figure 1, that include structure, behavior, requirements, and parametrics.



Copyright © 2015, 2012, 2009 Elsevier Inc. All rights reserved.

Figure 1. The four pillars of SysML v1

Since its adoption, SysML has been used by a variety of industries and organizations around the world in their pursuits of a model-based systems engineering approach. Multiple tool vendors provide tools that implement SysML. MBSE with SysML has been incorporated into many academic and training curriculum. The Unified Architecture Framework (UAF) is another OMG modeling standard that further extends SysML to provide an enterprise architecture modeling language. Since the introduction of SysML, systems engineers, tool vendors, and people in academia have learned much from their experience, including both the strengths and weaknesses of SysML as a language, and the benefits and challenges of adopting and applying MBSE with SysML.

SysML v2 Requirements

The Systems Engineering Domain Special Interest Group (SE DSIG) is the working group within the OMG that has been responsible for developing the requirements for SysML and for providing the forum for sharing MBSE applications and lessons learned. The SE DSIG developed the requirements for the next generation of SysML to provide improved capabilities over SysML v1 (*reference 1*). In particular, the goals for SysML v2 are to provide a language that is easier to adopt by organizations, and enable more effective application of model-based systems engineering. The specific objectives are to improve:

- Precision and expressiveness of the language.
- Consistency across language concepts.

- Usability for model developers and model consumers.
- Interoperability between systems modeling tools and other model-based engineering tools.
- Extensibility to support modeling of domain-specific concepts.

The OMG process involves developing requirements in the form of a Request for Proposal (RFP). The requirements for SysML v2 were developed and reviewed both internally within the OMG as well as by INCOSE for review and comment in July 2017. The requirements were incorporated into two separate RFP's, one for the language and the other for the API and Services. A RFP for the SysML v2 language was approved and issued by the OMG in December, 2017 (*reference 3*). A second and complementary RFP for standardized SysML v2 API and Services to enable interoperability between SysML modeling tools and other model-based engineering tools was issued by the OMG in June, 2018 (*reference 4*).

The requirements for the SysML v2 language RFP are organized as shown in Table 1, and the requirements for the SysML v2 API & Services RFP are organized as shown in Table 2.

Table 1. SysML v2 Language Requirements

Language Architecture
Metamodel and Profile
Abstract Syntax
Concrete Syntax
Extensibility
Model Interchange
Data Model *
Cross-cutting
Properties, Values, & Expressions
Structure
Interface
Behavior
Requirements
Verification
Analysis
Example Model
Conformance Requirements

*Appendix C of the RFP includes a data model of the concepts in the RFP, and Appendix A.2 provides the glossary of terms.

Table 2. SysML v2 API & Services Requirements.

API & Services Architecture
API & Services Conformance
Mandatory Service Requirements
Service Scope, Conditions, and Response
Model Navigation Service
Model Creation Service
Model Update Service
Model Deletion Service
External Relationship Management Services
Non-mandatory Service Features *
Model Query Services
Advanced Model Construction Services
Model Analysis Services

Model Management Services
Model Transformation Services
Model Query Services
Advanced Model Construction Services

*The submission team can choose whether to address the non-mandatory services in its submission.

SysML v2 Submission Team (SST)

The SysML v2 Submission Team (SST) was formally established at the OMG meeting in December, 2017, following the approval of the SysML v2 RFP. The team objectives were to develop and submit the specification to the OMG in response to both the SysML v2 RFP and the SysML v2 API & Services RFP.

The SST engaged representatives from over 70 organizations to participate, including representatives of system development organizations, tool vendors, academia, and government representatives. The SST team structure consists of the six tracks shown in Table 3, each led by a track lead(s).

Table 3. SST Team Structure

Track	Responsibility
Project Management	Administration, Project planning, Cross track coordination, Submission deliverables
Requirements V&V	Validate requirements address user needs, Verify that the language and API design satisfy the requirements
Profile Development	SysML v2 profile design, SysML v1 to SysML v2 transformation specification
Metamodel Development	SysML v2 metamodel design
API & Services Development	SysML v2 API & services design, API & services pilot implementation
Pilot Implementation	Pilot implementation of language integrated with API & Services. Textual syntax design

The project applies an agile model-based approach to the development of the SysML v2 language and the API and Services specifications. The project is also developing an open-source pilot implementation of both the language and the API. This enables the team to validate the requirements, ensure a high level of specification quality, and provide a kick-start for future tool implementations. The specification and pilot implementation baselines are released monthly as part of the agile process. The general SST development process and associated model-based environment is described in Figure 2.

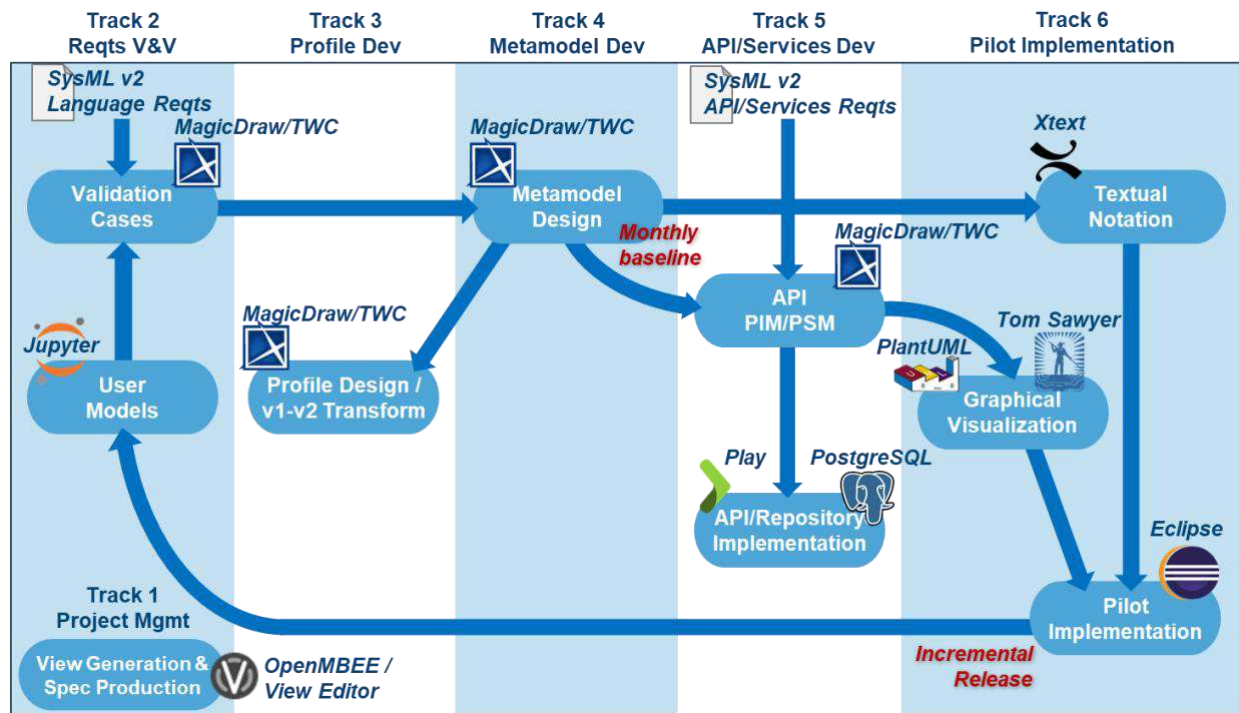


Figure 2. SST Agile Development Process and Model-based Environment

SysML v2 Technical Approach

SysML v2 provides a modeling capability that is a significant improvement over SysML v1. The enhancements include a textual notation and a formal semantic foundation that provides a level of precision that was not achievable previously. This reduces ambiguity, facilitates engineering analysis, and enables the ability to perform complex queries of the model. In addition, SysML v2 is based on a new metamodel that provides a consistent pattern of definition and usage. This should make the language easier to learn, implement, and maintain. The language also provides new levels of expressiveness, such as concepts for modeling variability and explicit modeling of analysis cases. In addition, SysML v2 provides a standardized API that facilitates tool interoperability, and promotes innovative applications that can leverage the system model for visualization, analysis, and other applications. The following sections provide an overview of both the SysML v2 language specification and the SysML v2 API and Services specification, in response to the requirements in the RFPs for the language and API that are summarized above.

SysML v2 Language Specification

Metamodel. The SysML v2 language specification is intended to be implemented in SysML modeling tools that enable users to create models of systems using the SysML v2 language. The language specification is formally represented by a metamodel. The metamodel consists of abstract syntax, concrete syntax, and semantics. The abstract syntax specifies the language concepts and the rules for expressing legal statements in the language, analogous to rules for constructing sentences using verbs and nouns in natural language. The concrete syntax specifies the symbols that can be used to represent the language concepts, either as graphical symbols, as textual symbols, or a combination of the two. The

semantics specify the meaning of the concepts, analogous to using a dictionary definition to give meaning to a word.

The metamodel itself must be expressed in a formal language. The language for expressing the SysML v2 metamodel is the OMG meta-object facility (MOF), which is the same language that is used to specify UML. This MOF includes constructs such as metaclasses, meta-associations, metaproperties, and specialization. These basic constructs are used to specify the metamodel abstract syntax (e.g., language concepts and grammatical rules).

The concrete syntax for the textual notation is specified using the Backus Normal Form (BNF), which is commonly used to specify concrete syntax for programming languages. For SysML v2, the BNF will be further extended to formally represent the graphical symbols in a graph structure.

The foundational semantics are specified using a predicate calculus to provide the formal logic to specify the meaning of the core language concepts. The meaning of other concepts in the language is defined in terms of the meaning of these core concepts, which is analogous to using words in a dictionary to define the meaning of other words.

The metamodel specifies the language concepts, their symbols, and their meaning. However, a user of the language uses these concepts, symbols, and meaning by creating specific instances of the metaclasses in the metamodel. For example, a user can create instances of the PartUsage metaclass from the metamodel that are called parts in the user model, such as a part that represents a battery. This is analogous to using an instance of a noun in the English language to refer to a battery.

Language architecture. The language architecture for SysML v2 is shown in Figure 3. The architecture consists of three foundation layers called the root, core, and kernel, collectively called the Kernel Modeling Language (KerML) metamodel. The SysML metamodel is the layer on top of the KerML metamodel. The layers provide for separation of concerns, where each layer extends its lower layers through specialization.

The root layer includes the basic constructs of element and relationship. The core extends the root to include a small set of concepts that provide the basis for further specialization, and that are grounded in formal semantics. The kernel further extends these core concepts to include more specialized concepts that are useful for specifying domain-specific language concepts. The SysML metamodel further specializes the KerML to include the domain-specific concepts needed to model systems.

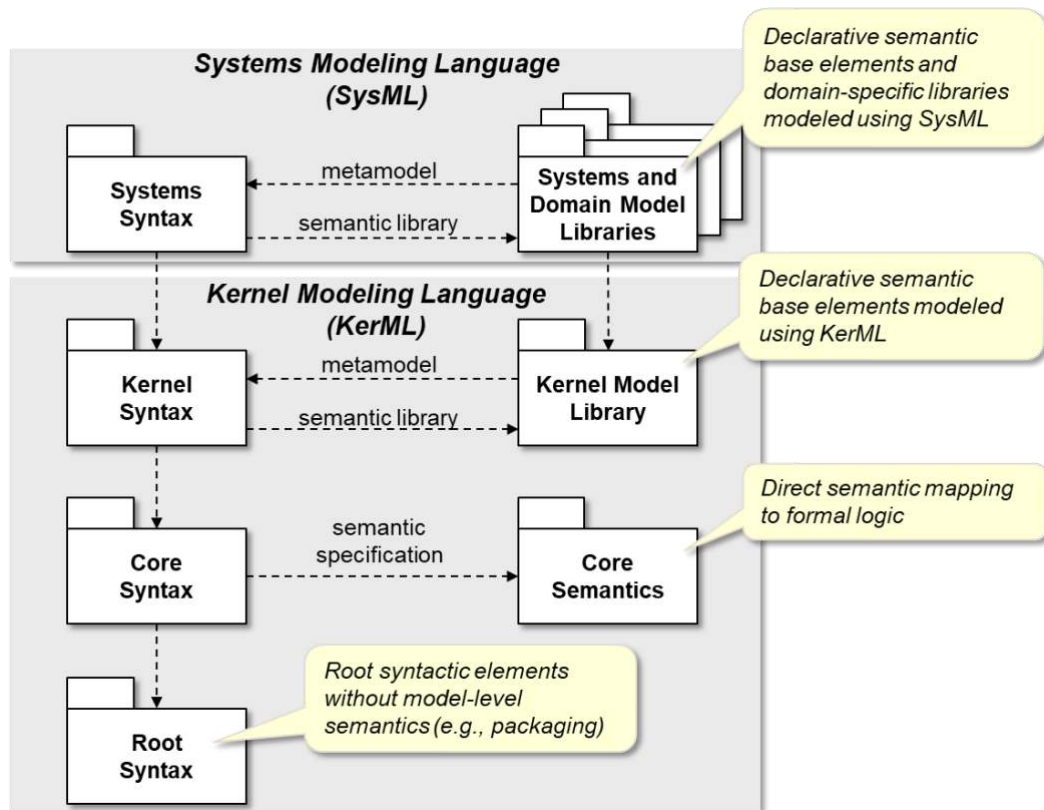


Figure 3. The SysML v2 Language Architecture

SysML v2 Language Design

Kernel Modeling Language (KerML) metamodel design. The root provides basic concepts that include element and relationship. These concepts provide the foundation that are used to define all other kinds of elements in the language. Each element includes a unique id, a name, and any number of aliases. Relationships are a kind of element that relate other elements. A relationship can have multiple ends, and the ends can be ordered. The ordering provides the capability to identify directed relationships where ends can be identified as the source or target end.

The root also includes the concept of a package, which is a kind of container for other elements. A specialized kind of relationship called membership relationship relates the container to its member elements. Another key concept in the root is an annotated element, which is a kind of element, such as a comment, that is used to describe other elements.

The core metaclasses extend the root metaclasses through specialization. The core concepts include type, feature, and generalization. These concepts provide the basis for classifying things in the real world. A “type” is a kind of package that contains features. A feature can be classified by its type. Specialization enables one type to inherit the features of another type. This capability provides the basic foundation to extend the language concepts based on more foundational language concepts.

The kernel metaclasses extend the core metaclasses through specialization. The kernel concepts provide specialized concepts that are generally applicable to creating the domain-specific extensions for SysML, as well as for other domain specific applications. These include concepts such as class and data

type, association and connectors, behaviors, interactions such as item flow and succession, and functions, expressions, and feature values.

These three layers of the KerML provide the foundation for specifying the SysML metamodel, which defines the domain specific concepts for modeling systems.

Systems Modelling Language (SysML) metamodel design. The SysML metaclasses either reuse directly or extend the metaclasses from the KerML. For example, the Package metaclass that is in the root layer of KerML is reused directly in the SysML metamodel, but the Classifier and Feature metaclasses, that are in the kernel layer, are extended to define more specialized concepts related to parts, actions, requirements, and others. The system modeling concepts defined in the SysML metamodel are summarized in the following section.

The system modeling concepts are used to model systems, subsystems, and components in terms of their compositional structure, their interconnection, behavior and interactions, their key performance, physical, and quality characteristics, and their requirements that represent stakeholder-imposed constraints.

Definition and usage pattern. Large investments are often made to develop and validate system models. A key challenge for systems modeling is to facilitate reuse of these models that can be adapted to their context. In order to respond to this challenge, the SysML metamodel introduces the concepts of definition and usage. As the name implies, a definition element defines an element such as a part, action, or requirement. A usage element is a usage of a definition element in a particular context. There can be many different usages of the same definition element in either different contexts, or the same context. This pattern of definition and usage applies to most concepts in the SysML metamodel.

A simple example is a part definition representing a wheel, and two usages of this part definition that represent the left front wheel and right front wheel. These two usages of wheel are part of the same vehicle, which provides a common context for both usages. Another example is the definition of an action to provide electrical power. Different usages of this action can be performed in different contexts. For example, a battery can provide electrical power for one system, and a power supply can provide electrical power for another system. Each of these two actions may be a child of two different higher-level actions that provide the context for each action. This ability to define the concept once and have multiple usages is a key enabler of reuse.

SysML v1 supports the concept of definition and usage, but there are some fundamental differences between SysML v1 and SysML v2. First of all, the pattern of definition and usage is applied uniformly to all SysML v2 concepts. For example, a part that is a usage of a block in SysML v1 is different than an action which is intended to be a usage of an activity. In SysML v2, the pattern of definition and usage for parts and actions are the same. The second fundamental difference is that the usages in SysML v2 can specialize their definition, so that each usage can be adapted to its context. SysML v1 includes property specific types to accomplish this, but this approach has significant limitations. Finally, in SysML v2, the

usage elements can be decomposed into constituent usage elements. For example, a parts tree or action tree can be modeled as a hierarchy of parts or actions respectively. In SysML v1, a block decomposition and activity decomposition can be modeled, but not a part and action decomposition. Furthermore, the activity decomposition in SysML v1 requires adjunct properties to provide meaning to the decomposition. The consistent pattern of definition and usage in SysML v2 greatly simplifies the language, and makes it easier to learn, apply, and implement.

The following is a summary of some key concepts in the SysML metamodel:

- The modeling of *packages* is reused directly from the KerML; it provides a flexible means to logically organize a model into a containment tree.
- The modeling of *dependency* relationships applies to all model elements, as does the modeling of *annotations* that provides additional descriptive information about other model elements.
- The concept of *definition* and *usage* apply to all of the elements described below. In addition, the modeling of *variability* can apply to all definition and usage elements. This includes the definition of *variation* points within a model where choices can be made to select a specific *variant*. A choice at one variation point can also constrain choices at other variation points. A system can be *configured* by making appropriate choices at each variation point consistent with specified constraints.
- The modeling of *structure* to represent how parts are decomposed, interconnected, and classified, which includes:
 - *Parts* that are the foundational units of structure that can be composed and interconnected.
 - *Attributes* that specify characteristics of something that can be defined by simple or compound data types, and dimensional quantities such as mass, length, etc.
 - *Ports* that define connection points on parts that enable interactions between parts.
 - *Connections* and *interfaces* that define how parts are interconnected.
 - *Items* that may flow through a process or system, or be stored by a system.
- The modeling of *individual* items and parts with unique identity can be represented at specific points in their lifetime called snapshots, and over portions of their lifetime called time slices.
- The modeling of *behavior* to represent what a system or component does, and how systems and components interact, which includes:
 - *Actions* performed by a part, including their temporal ordering, and the flows of items between them.
 - *States* exhibited by a part, the allowable *transitions* between states, and the actions enabled within states or by transitions between states.

- The modeling of *calculations* which are parameterized expressions that can be computed to produce specific results.
- The modeling of *constraints*, which specify conditions that a system or part is expected or required to satisfy.
- The modeling of *requirements*, which is a special kind of constraint that a *subject* system or part must satisfy to be a valid solution.
- The modeling of *cases*, which define the steps required to produce a desired result relative to a *subject* to achieve a specific *objective*, including:
 - *Analysis cases*, whose steps are the *analysis actions* necessary to carry out a certain analysis of a subject.
 - *Verification cases*, whose objective is to verify how a certain requirement is satisfied by the subject.
- The modeling of *viewpoints* that specify information of interest by a set of stakeholders, and *views* that are intended to satisfy a particular viewpoint. A specification of a view defines a query of the model to select the model content to be presented, and the specification of how the query results are be rendered.

In a similar way that SysML v2 extends KerML, SysML v2 also provides a language extension capability to enable users to build domain-specific extensions of SysML. This allows SysML to be highly adaptable for specific application domains and user needs, while maintaining a high level of underlying standardization and tool interoperability.

It should be noted that SysML does not contain specific language constructs called system, subsystem, assembly, component, and many other commonly used terms. An entity with structure and behavior in SysML is represented as a part. The language will provide a straightforward mechanism to specify these extensions and others.

SysML v2 Concrete Syntax and Visualization

SysML v1 provides a graphical notation to represent systems. SysML v2 includes a textual notation in addition to the graphical notation. The textual and graphical notation provide equivalent expressions of the same model. Conformant tool implementations may provide the graphical notation only, textual notation only, or both. For tools that provide both textual and graphical notations, a user has the option of using graphical, textual, or both notations to construct and view the model. The two notations are complementary if used together. The graphical notation, is useful for understanding a broader context, as well as complex cross-cutting relationships, and non-sequential processes, while the textual notation is particularly suited to provide efficient expression of detailed self-contained portions of the model.

The textual notation can be thought of as a programming language for systems modeling. It is reasonably intuitive to interpret. A simple example is the following parts tree:

```
part vehicle {  
    part engine {  
        part cylinders [6];  
    }  
    part transmission;  
}
```

In this example, the vehicle contains an engine and a transmission, and the engine contains six cylinders. The vehicle, engine, cylinders, and transmission are all parts.

These parts can be further elaborated with many different features. In the following example, the vehicle parts are elaborated to include a mass; and the actions providePower, generateTorque, and amplifyTorque that the vehicle, engine, and transmission perform respectively:

```
part vehicle {  
    attribute mass;  
    perform providePower;  
    part engine {  
        attribute mass;  
        perform generateTorque;  
        part cylinders [6];  
    }  
    part transmission {  
        attribute mass;  
        perform amplifyTorque;  
    }  
}
```

Each of the usage elements including the parts, attributes, and actions can be defined by definition elements. For example, the part vehicle may be defined by a part definition called Vehicle. The textual notation would include the following:

```
part def Vehicle;
```

```
part vehicle:Vehicle {  
    // the vehicle parts are not shown  
}
```

The part definition Vehicle could include additional features such as its mass, reliability, speed, other actions that the vehicle performs, its states, constraints, and other features. Each part that is defined by Vehicle reuses and potentially modifies these features, and can add other features. As described previously, the pattern of definition and usage applies to other constructs such as actions, states, constraints, and requirements.

The SysML v2 textual notation is defined formally using BNF and mapped to the abstract syntax to provide an unambiguous expression of the underlying model. SysML v2 will provide a graphical notation that maps to the textual notation. The standard SysML v2 notation will be similar to the SysML v1

graphical notation. The parts tree described using the textual syntax above is shown as a graphical representation in Figure 4 (all parts are defined by definition elements in the graphical view).

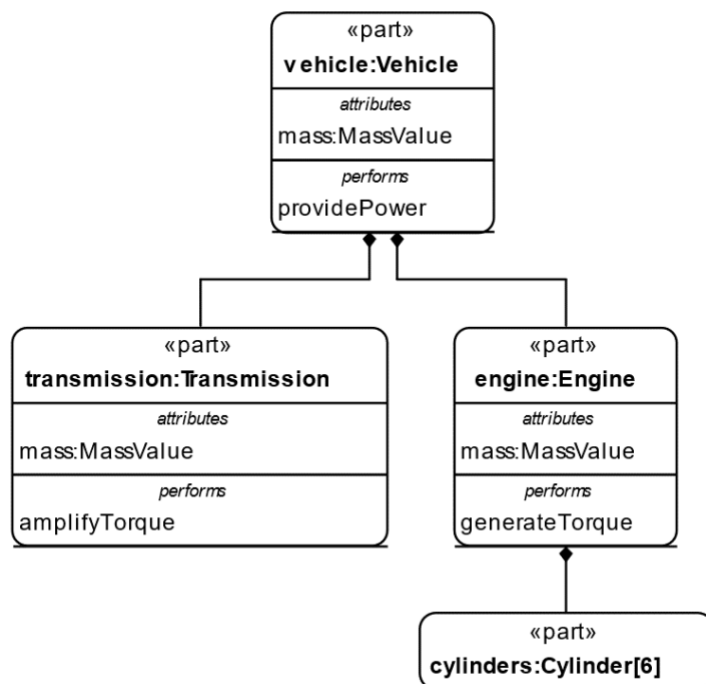


Figure 4. Example of a SysML v2 diagram for the parts tree consistent with the above textual notation

The approach to specify the graphical notation has been defined, but not implemented, as of the initial submission. The ability to formally specify the graphical notation is intended to enable customization of the graphical symbols to support domain specific needs.

SysML v2 will define standard diagram kinds similar to SysML v1 to facilitate the transition from SysML v1 to SysML v2. However, unlike SysML v1 diagrams, SysML v2 will not constrain what can be presented on a particular diagram, as long the model conforms to the SysML v2 metamodel. For example, a SysML v2 parts interconnection diagram is similar to a SysML v1 internal block diagram (IBD). However, the modeler is free to add requirements with satisfy links on a SysML v2 parts interconnection diagram, where-as a requirement graphical symbol cannot be shown on a SysML v1 IBD.

SysML v2 will also provide a capability to specify user defined views of the model information using the view and viewpoint concepts in SysML v2. The view is specified in terms of a model query that specifies the content of the model to be presented, and a rendering specification that specifies how this content should be presented. This capability is intended to enable the creation of user-defined views that are presented in tables and diagrams, and complex hierarchical views that are presented as documents containing text, tables, and diagrams.

SysML v2 API & Services Specification

The SysML v2 Application Programming Interface (API) specification is intended to satisfy the requirements in the SysML v2 API & Services RFP that was issued by the OMG in June, 2018. The API enables other engineering tools and software applications to interact with SysML models stored in a

repository using standard service calls. These service calls provide a way for other tools and applications to operate on the SysML model, such as to navigate and query the model, to create model elements, to update the model, and to establish versions of the model and model elements to support configuration management. There are other requirements for services in the RFP that may be incorporated into the SysML v2 API specification, including view and viewpoint services, model analysis services, and transformation services.

The API specification was developed using an approach that enables the API to be implemented by SysML tool vendor implementations using different API Technologies (e.g., HTTP, Java, C#) and different data repository technologies (e.g., MySQL, Graph databases). The service calls are specified using a platform independent model (PIM) that is a logical abstraction independent of any underlying software implementation technology. A platform specific model (PSM) is then developed for each technology option, and mapped to the PIM. The use of the PIM and PSM is illustrated in Figure 5.

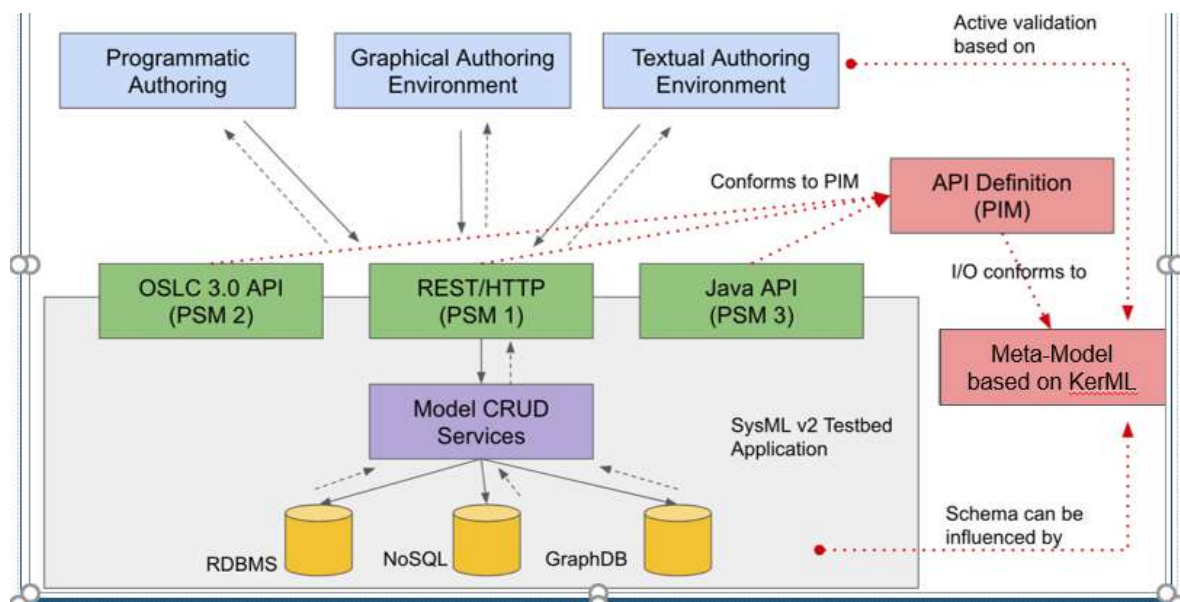


Figure 5. The use of PIM and PSM to specify the API & Services¹

The initial API technology option that is specified for SysML v2 is the REST/HTTP API that uses HTTP requests to Get, Put, Post, and Delete data. The REST/HTTP API technology is used broadly by many web applications. A second technology option that is specified for SysML v2 is the Open Services for Lifecycle Collaboration (OSLC) implementation, which builds on the REST API. This API option provides a standard interface that supports linked data interfaces, where other tools link to data in the SysML model without extracting the data from the SysML repository. This type of interface promotes loose coupling between other software tools and applications, and the SysML modeling tool. Other technology options may be specified in the future.

¹ CRUD stands for "Create, Read, Update, and Delete," which are the four basic database operations. Many HTTP services also model CRUD operations through REST or REST-like APIs. ... For example, to get the product whose ID is 28, the client sends a GET request for <http://hostname/api/products/28>. See [Enabling CRUD Operations in ASP.NET Web API 1](#).

Transition to SysML v2

The transition from SysML v1 to SysML v2 will require careful planning (as is the case for any transition). Some organizations will be new to SysML, and will begin with SysML v2 without any heritage SysML v1 infrastructure. Other organizations that have been practicing with SysML v1 may have a hybrid SysML v1 and SysML v2 environment for some time. A possible approach to transition is to introduce SysML v2 on new programs.

Organizations will need to assess the impacts on their modeling practices, methods, tools, and training. Organizations should pilot SysML v2 within their organization to fully understand the impacts, and also understand how to obtain increased value from the new capabilities provided by SysML v2. SysML v2 should also enable significant reuse within organizations and across industry. Organizations, industry standards bodies, and professional societies can begin to formulate strategies for developing reusable assets, and for leveraging these assets across programs and industry domains.

Remaining Work for SysML v2 Final Submission

The initial submission was submitted to the OMG in late August, 2020 and presented to the OMG Analysis and Design Task Force (ADTF) on September 16, 2020. The initial submission includes individual specifications for the Kernel Modeling Language, Systems Modeling Language, and the Systems Modeling API & Services. This version of the specifications will be reviewed by the OMG, and with the broader systems modeling community through a stakeholder review that will include INCOSE representatives. In addition, an open source version of the pilot implementation will be made available for community evaluation and feedback.

The submission addresses many of the requirements of the SysML v2 RFP and the SysML v2 API & Services RFP. However, there is considerable work to be performed for the final submission, which is planned for mid to late 2021. Work that is planned as of September 2020 includes the following:

- Formalization of the graphical syntax. There are some early graphical visualization prototypes, but the syntax has not been formally specified.
- Design and implementation of the language extension concept (*reference 8*)
- Additional language functionality. Much of the language functionality has been addressed by the initial submission, but several specific requirements, such as support for trade studies, still need to be addressed.
- The profile for SysML v2 that maps to a subset of the SysML v2 metamodel. A first effort has been made. The current approach is that the SysML v2 profile will provide the same modeling capability as SysML v1 but interoperable with other SysML v2 models.
- The specification of model interchange. Although the API facilitates the dynamic interaction with SysML modeling tools, there is a requirement to provide a standard format for model interchange and long-term model retention. A preliminary approach has been defined, but not implemented.

- The SysML v1 to SysML v2 transformation needs to be fully specified to enable vendors and users to transform their legacy SysML v1 models to a SysML v2 format, and leverage their SysML v1 model investments. This has been partially defined as of the initial submission.
- Selected non-mandatory API services, subject to prioritization.
- The conformance test suite for the SysML v2 language and the SysML v2 API & Services.

Summary

SysML v1 was adopted in 2006, and has been used across industry as part of their organizational efforts to adopt an MBSE approach. Much has been learned from these applications in terms of both the strengths and weaknesses of SysML v1 and MBSE.

SysML v2 is intended to address many of the limitations of SysML v1 in terms of its precision, expressiveness, consistency, usability, interoperability, and support for reuse. SysML v2 introduces a new, simplified but sophisticated metamodel that is grounded in formal semantics. It provides a precise textual notation along with the graphical notation. It also provides a standard API to enable tool interoperability. SysML v2 also will provide a solid foundation for specifying domain specific extensions. Together, these capabilities collectively make SysML v2 a true next-generation systems modeling language that can be applied more broadly and more effectively than SysML v1.

The initial submission was presented to the OMG in September 2020. This version of the specification is available for review together with an open source implementation. The current plan is to submit the final specification in 2021 for adoption by the OMG.

List of Acronyms Used in this Paper

<u>Acronym</u>	<u>Explanation</u>
ADTF	Analysis and Design Task Force
Alf	Action Language for Foundational UML
API	Application Program Interface
BNF	Backus Normal Form
CRUD	Create, Read, Update, and Delete
fUML	Foundational Subset for Executable UML Models
IBD	internal block diagram
HTTP	Hypertext Transfer Protocol
INCOSE	International Council on Systems Engineering
KerML	Kernel Modeling Language
MBSE	Model-based Systems Engineering
MOF	Meta Object Facility
OMG	Object Management Group
OSLC	Open Services for Lifecycle Collaboration
PIM	Platform Independent Model
PSM	Platform Specific Model
RFP	Request for Proposal
RTF	Revision Task Force
SE DSIG	Systems Engineering Domain Special Interest Group
SoaML	Service Oriented Architecture Modeling Language
SST	SysML v2 Submission Team
STEP	Standard for the Exchange of Product Model Data
SysML	Systems Modelling Language

Acknowledgements

The authors acknowledge the contributions of the SysML v2 Submission Team (SST) members. There are over 70 organizations that identified representatives to participate in the SST, and they are listed in the acknowledgement sections of the SysML v2 initial submission (references 5, 6, and 7). Many representatives have contributed to this effort; they are too numerous to mention. A special acknowledgement goes to the other current and previous SST track leads that provide the day-to-day leadership for this effort. These individuals include Manas Bajaj, Yves Bernard, Bjorn Cole, Charles Galey, Karen Ryan, and Tim Weilkiens.

References

1. *Friedenthal, Sanford. Requirements for the Next Generation Systems Modeling Language (SysML® v2), INCOSE INSIGHT March 2018, Volume 21, Issue 1, pages 21-25.*
2. *Object Management Group Systems Modeling Language (OMG SysML™) Version 1.6 OMG Document: formal/19-11-01 dated November 2019.*
3. *Object Management Group Systems Modeling Language (SysML®) v2 Request for Proposal (RFP) OMG Document: ad/2017-12-02 dated 09 December, 2017.*
4. *Object Management Group Systems Modeling Language (SysML®) v2 API and Services Request for Proposal (RFP) OMG Document: ad/2018-06-03 dated 21 May 2018.*
5. *Object Management Group Kernel Modeling Language Version 1.0 Initial Submission OMG Document: ad/2020-08-02 dated August 2020.*
6. *Object Management Group Systems Modeling Language (SysML®) Version 2.0 Initial Submission OMG Document: ad/2020-08-03 dated August 2020.*
7. *Object Management Group Systems Modeling Application Programming Interface (API) and Services Version 1.0 Initial Submission OMG Document: ad/2020-08-04 dated August 2020.*
8. *Seidewitz, Ed, On a Metasemantic Protocol for Modeling Language Extension, MODELSWARD 2020, Valletta, Malta, February 2020.*

About the Authors



Sanford Friedenthal. Mr. Friedenthal is an industry leader and independent consultant in model-based systems engineering (MBSE). He has over 30 years of experience as a system engineering practitioner, department manager, and leader of organizational initiatives for Lockheed Martin. Mr. Friedenthal also has been a leader of the industry standards effort through the Object Management Group (OMG) and INCOSE to develop the Systems Modeling Language (OMG SysML®) that was adopted in 2006. He is

co-author of “A Practical Guide to SysML” and “Architecting Spacecraft with SysML” and is now leading the SysML v2 Submission team to develop the next generation of SysML v2.



Ed Seidewitz. Ed Seidewitz is Chief Technology Officer at Model Driven Solutions, Inc., a long-time provider of enterprise and systems architecture services using model-based methods. Mr. Seidewitz has over 30 years of professional experience with the modeling, architecture, and development of systems spanning diverse domains including aerospace, finance, acquisition, and health care. Mr. Seidewitz has been active with the Object Management Group (OMG) for 20 years, including involvement in every UML Revision Task Force and with the Service Oriented Architecture Modeling Language (SoaML) and System Engineering Modeling Language (SysML) specifications. He was primary author of the Foundational

Subset for Executable UML Models (fUML) and Action Language for Foundational UML (Alf) specifications and maintains open source reference implementations for both. He is currently chair of the OMG Model Interchange Special Interest Group and of Revision Task Forces for fUML, Alf, Precise Semantics of UML Composite Structures, and Precise Semantics of State Machines. He is also co-leader of the SysML v2 Submission Team and leader of the Precise Semantics of Time for fUML Submission Team.

2.2 The Greatest Young System Engineers of the Year Challenge²



30th Annual INCOSE
international symposium

Cape Town, South Africa
July 18 - 23, 2020

by

Ad Sparrius

Graduate School of Technology Management, University of Pretoria

Ad Sparrius System Engineering and Management (Proprietary) Limited

ad_sparr@iafrica.com

Copyright © 2020 by Ad Sparrius. Permission granted to INCOSE and PPI to publish and use.

² This article provides an update of an INCOSE International Symposium (IS) 2020 paper. It adds the results of the GYSEYO 2019/20 Challenge and provides a detailed plan for GYSEYO 2020/21. It provides a flowchart indicating the activities involved in creating the program.

Abstract

In 2015 the “Greatest Young Systems Engineers of the Year Challenge” (the “Challenge”) was launched as part of the INCOSE South Africa (SA) annual conference. This annual challenge has occurred five times from 2015 through 2019/20 with a total participation of 69 contestants from nine employer companies. This article summarizes the Challenge’s objectives, results, and lessons learned. In addition, following a presentation made by the author at the 30th Annual INCOSE International Symposium, the suggestion was made that “the Challenge” should be replicated by other INCOSE Chapters throughout the world. This is considered by many a “golden opportunity” to further strengthen and improve the education, training, and professionalism of young systems engineers everywhere, and also the performance and effectiveness of the systems engineering profession. Accordingly, this article provides information about the Program, feedback from several of the participants, and guidance to enable replication and implementation of similar programs.

Introduction

The objective of the Greatest Young Systems Engineers of the Year Challenge³ (GYSEOY) (pronounced as “g-eye-soy”) is to foster a deep interest in system engineering in young graduate engineers by means of a challenge to solve a business problem using advanced system engineering principles, including model-based techniques. The underlying rationale is to foster system engineering leadership for the next generation, by focusing on the development of technical skills, personal skills, and professionalism.

GYSEOY is a wonderful way to enrich an employer’s in-house engineer-in-training training process. It is a once-in-a-career opportunity for a young engineer. Any young graduate engineer fresh out of university, say not more than three years after their previous degree, may become a contestant. Teams of three or four engineers (each team preferably from a single employer), may participate in the challenge - not individuals. Explicit support from the employer is a precondition for the acceptance of a team. Each contestant needs to commit to investing substantive effort, with experience showing that about 200 person-hours are required per contestant⁴. This effort consists partly of working time, and the employer must be willing to accommodate that commitment. The duration of the challenge is roughly one year.

The GYSEOY Challenge is launched with a “Get-to-Know-You Afternoon”, at which final arrangements are made and the contestants are introduced to various individuals who will play a role. The following documents are sent out to all participants prior to this meeting:

- The GYSEOY scenario.
- The GYSEOY master schedule, defining all stages and their events⁵ and dates.

³ This article uses the terms “challenge” and “contest” interchangeably - both terms mean a competitive situation to select a winner [Oxford Dictionary of English, 2019]. GYSEOY is a *challenge* and its participants are *contestants*.

⁴ Nine contestants from the GYSEOY winning teams of 2017, 2018, and 2019/20 invested an average of 190 person-hours, with a standard deviation of 20.4 person-hours. The maximum time spent was 224 person-hours and the minimum time spent was 169 person-hours.

⁵ All events for GYSEOY 2020/21 will be online due to the Covid-19 pandemic.

- A list of all of the contestants and their contacts information, together with a passport-type photograph of each participant.
- Information concerning how to download the one-year license for the university edition of GENESYS™.

Each contestant receives a one-year license for the university edition of GENESYS™, a model-based system engineering software tool, with compliments from Vitech Corporation⁶. A total of six training days spread throughout the GYSEOY duration are presented “just-in-time” (that is, the training is provided immediately prior to the time when it is required for use), covering an introduction to the systems engineering process and hands-on training on GENESYS™.

The formal evaluation of teams occurs at a “Requirements Review” and a “System Design Review”. At the System Design Review, each team has a two-hour time slot to provide a presentation. The final deliverable from each team is the GENESYS™ model, a system description document generated from the GENESYS™ design repository, and a presentation. An evaluation panel of six senior system engineers performs these reviews; the members of the panel provide constructive feedback to each team based on the strengths and weaknesses of each team’s solution. The evaluation process is provided primarily as a learning exercise; the winning team is selected and identified.

Periodic “Question and Answer Sessions” assist the participants with any problems they encounter throughout the Challenge. If appropriate, tutorials concerning specific topics are arranged. Teams are encouraged to solicit advice from their colleagues, and over time, this evolves into a mentor-mentee relationship. To provide independent mentorship, each team is also assigned an external mentor from outside the employer company.

The challenge culminates at the INCOSE SA annual conference, with the System Design Review scheduled for the days leading up to that conference. All contestants receive complimentary registration to the conference. The INCOSE SA Central Management Committee hosts a special GYSEOY cocktail reception for all contestants. A special morning track is devoted to GYSEOY at the conference. Each team receives an opportunity to present its solution in the form of a conference presentation, as part of the GYSEOY learning process. Unfortunately, limited time prevents teams from preparing a formal conference paper. The succeeding GYSEOY challenge and its scenario is also revealed. The winning team receives a floating trophy, (also known as a rolling trophy, since the winners do not retain it, but holds it until the next challenge when it rolls over to the new winners) as well as a cash prize sponsored by INCOSE SA. A floating trophy for the Sharpest Young System Engineer of the Year sometimes known as SYSEOY is also awarded. The point is repeatedly made that although one team receives the floating trophy, all contestants are winners—there are no losers!

⁶ Why Vitech? Because it’s local agent proposed it. Why Genesys™? For GYSEOY purposes, it is representative of current MBSE tools. From 2015 through 2018 CORE™, a predecessor of GENESYS™, was used.

The most important result from participating in the GYSEYOY challenge is not the additional systems engineering skills mastered—most contestants would learn these skills during their first few years after graduation.

Uniquely, GYSEYOY provides an accelerated learning curve lasting a few months. The most important result is strengthened self-confidence. The GYSEYOY scenario is developed purposely to be beyond the experience of most contestants, so as not to unfairly advantage some team. It is also beyond the experience of most practicing system engineers. That forces everybody to start from first principles, the fundamental concepts or assumptions on which a theory or method is based. Experience shows that in uncertain situations such as the Challenge, the advice of seasoned practitioners is at times not much better than that of the young contestants, creating self-confidence that is crucial in the early stages of an engineer's career!

The GYSEYOY Scenario

The selection of an appropriate scenario is crucial to the success of GYSEYOY. The purpose of all engineering is to serve humankind; hence, all of the requirements in the scenario should be focused on human needs. The scenario is therefore designed in the context of a socio-economic problem. The scenario describes a particular individual, in a specific location, and in a specified environment, who experiences particular symptoms of a vague problem. Appendix 1 identifies the scenarios for all five of the GYSEYOY challenges held to date. A GYSEYOY scenario is not a problem statement, but rather a story. Each team needs to define a concise statement (in a sentence or two) of the root cause problem, with associated measures of effectiveness (MOE) that describe what the world would look like if the problem were successfully solved.

Appendix 2 contains the scenario from the GYSEYOY 2018 challenge, which is representative of other GYSEYOY scenarios. The root problem is ill-defined and open-ended, with no clear boundary and incomplete, contradictory, and changing requirements that are usually difficult to recognize and elicit. Solutions to ill-defined problems are not right-or-wrong, but better or worse—it makes no sense to talk about “optimal solutions”. These are often called “wicked” problems, not because they are evil, but after the formulation of a solution [[Wikipedia](#), Wicked problem, 2019]. Some people feel that the GYSEYOY scenario should be more engineering-oriented. However, the solution to each GYSEYOY scenario always requires engineering elements, with the nature and number of those elements dependent on each team's particular solution. The amount of engineering in each solution depends on the solution developed by the team. But let there be no doubt—the problems humanity faces are super messy, for example, climate change. The crucial insights obtained from battling with the requirements of a socio-economic system will be very useful. Even though GYSEYOY does not aim to be a learning vehicle for handling wicked problems, young system engineers should be prepared for that kind of problem.

"I think the vagueness of the problem forced my team to really understand the system engineering theory, forcing us to apply it. A clear-cut problem would not have had the same effect."

Respondent to GYSEYOY questionnaire, November 2019

"My team expended considerable effort to understand the scenario, since the requirements were very vague. We dissected the problem and used tools such as causal loop diagrams to structure this nebulous, multi-stakeholder, complex social problem into something that could be objectively analyzed. Only then were we able to define a solution system that would have a real impact, instead of simply moving the problem elsewhere."

Respondent to GYSEYOY questionnaire, November 2019

"Social problems are really difficult to iron out, to determine what is actually needed and what is achievable. Maybe that makes GYSEYOY so intriguing."

Respondent to GYSEYOY questionnaire, November 2019

"The GYSEYOY scenario for me represents a real-life problem. Changing it to an engineering-type problem would result in us missing out on critical thinking skills and unlocking new solutions to our problems."

Respondent to GYSEYOY questionnaire, November 2019

The GYSEYOY learning outcomes are as follows:

- | | |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Outcome 1</i> | Develop a concise statement of the root problem with appropriate measures of effectiveness based on the symptoms from the GYSEYOY Scenario. |
| <i>Outcome 2</i> | Identify relevant stakeholders and elicit their requirements. |
| <i>Outcome 3</i> | Specify all functional requirements and their performance requirements needed to solve the root problem. |
| <i>Outcome 4</i> | Identify candidate solutions and select the most appropriate solution. |
| <i>Outcome 5</i> | Specify all functional requirements and their performance requirements for all the solution elements. |
| <i>Outcome 6</i> | Specify how all requirements from Outcomes 3 and 5 will be validated or verified. |
| <i>Outcome 7</i> | Prove bidirectional traceability for all requirements from Outcomes 3 and 5. |
| <i>Outcome 8</i> | Demonstrate basic project management principles applied to the GYSEYOY project. |

The deliverables required from each team at the System Design Review must provide evidence that these learning outcomes have been achieved. The evaluation rubric from Appendix 3 defines the assessment criteria for these learning outcomes.

The Stages of the GYSEYOY Challenge

GYSEYOY has been divided into five distinct stages, each lasting approximately eight weeks.

Stage 1—Problem Statement

Two training days will be used to launch Stage 1. The deliverables of Stage 1 are: Definition of the root problem, definition of key performance indicators (measures of effectiveness) and specification of as-is and to-be values of those key performance indicators, including the rationale for each key performance

indicator. A context diagram with external co-functioning systems and the specification of all external interfaces must be provided.

Two Question and Answer sessions will occur.

The close-out of Stage 1 will be a formal Requirement Review by the Evaluation Panel, based on deliverable documentation and a short oral presentation.

Stage 2—System Requirements

Two training days will be used to launch Stage 2. The deliverables of Stage 2 are: A validation or verification requirement for each key performance indicator (measure of effectiveness), and a validation or verification requirement for each external interface. A list of all stakeholders. The definition of relevant solution concepts, including the operations concept, support concept, personnel concept, et cetera. A use case diagram, and the performance requirements for each use case, including the rationale for each performance requirement. Validation or verification requirements for each use case and its performance requirements.

Two Question and Answer sessions will occur.

The close-out of Stage 2 will be an informal review by the Evaluation Panel.

Stage 3—Functional Architecture

Two training days will be used to launch Stage 3. The deliverables of Stage 3 are: An activity diagram for each use case. The specification of the performance requirements for each activity (function) in each activity diagram, including the rationale for each. Validation or verification requirements for the performance requirements of each activity in each activity diagram.

Two Question and Answer sessions will occur.

The close-out of Stage 3 will be an informal review by the Evaluation Panel.

Stage 4—Physical Architecture

Two training days will be used to launch Stage 4. The deliverables of Stage 4 are: The allocation of each function and its performance requirements to a system element. A schematic block diagram showing all system elements and all external and internal interfaces. Specification of each internal interface. Verification requirements for each internal interface.

Two Question and Answer sessions will occur.

The close-out of Stage 4 will be an informal review by the Evaluation Panel.

Stage 5—System Specification

The deliverables of Stage 5 are: A system specification (minimally, the substantive core of a system specification). An individual GYSEYOY 2020/21 diary for each team member where she/he records: Date, topic discussed, duration, and location.

The close-out of Stage 5 will be a formal System Design Review by the Evaluation Panel, based on the complete GENESYS™ model and an oral presentation.

As noted previously, each of Stages 1 through 4 starts with two training days, and ends with a review by the Evaluation Panel. The evaluation panel will evaluate the deliverables from Stages 1 through 4 based exclusively on the provided documents, except that there will also be an oral presentation for the Requirement Review. Reviews are part of the learning process. The same oral feedback will be provided to all teams, to prevent any possibility that one team is provided specific unintended feedback that unfairly advantages it. Since system engineering is an iterative process, each team may update their interim deliverables, based on the feedback provided in the review and further insight obtained during the GYSEYOY process. Document version control will need to be strictly performed. The final review at the conclusion of Stage 5 is the formal System Design Review based on each team's final deliverables and an oral presentation. The results from the System Design Review will determine the winning team. Each of the first four stages includes two Question and Answer sessions.

Mentorship

“You cannot create experience—you need to undergo it.
Albert Camus, *Notebooks*”

“Nothing ever becomes real until it has been experienced. Even a proverb
is not a proverb until your own life has illustrated it to you.”
John Keats, 1795—1821”

“A lot of people have gone further than they thought
because someone else thought they could.”
Zig Ziglar

During the first GYSEYOY challenge it became clear that contestants needed to ask more experienced colleagues for technical advice, for instance: “Should I use an activity diagram or an enhanced functional flow diagram to elaborate each use case?” or “How do we select the most appropriate solution from amongst a range of alternatives?” Equally important were personal development questions, for instance: “I think I will need to interview the Chief Engineer of another division. May I just ask her? And how should I prepare for that interview?” or “Should I rather pursue earning an MBA or an M.Eng?”

Mentorship is an intense developmental relationship in which a more experienced or more knowledgeable person guides a less experienced or less knowledgeable person. The mentor has a certain area of expertise that is informally conveyed to the mentee, usually face to face over a sustained period. It is a learning and development partnership between someone with lots of knowledge, experience, wisdom, and lessons-learned, and someone who wants to learn. Mentorship emphasizes not only the self-discovery process of the mentee, but also the mutual learning and development of both the mentee and the mentor. Coaching is a related technique that focuses on short-term mechanistic

matters, for instance: “How is an activity diagram developed? How does it differ from an enhanced functional flow diagram? Where would you use the one and where the other?” Coaching is quite specific, by insisting on first step 1, then step 2, et cetera. It is often proactive before the problem actually appears. Mentorship is more reactive by focusing on learning and self-development. It is self-evident that GYSEYOY needs a balance between coaching and mentoring, since aspects of both will be required. Technical issues are easily resolved by mentoring from internal mentors, as well as at the question and answer sessions. GYSEYOY has no influence on which internal mentors are appointed, and how many there are.

However, in consideration of personal development issues, confidentiality is important, and hence it turns out that external mentors will usually be more appropriate. An external mentor is one who is not employed by the same employer as the mentee, and thus is ignorant of internal politics or project priorities, or related make-or-break decisions that are urgent and crucial. The formula is simple—The GYSEYOY manager selects some experienced system engineers to act as external mentors and allocates them to the various teams; one external mentor per team. The first external mentors were appointed for GYSEYOY 2018, and the first formal evaluation of external mentorship occurred at the conclusion of GYSEYOY 2019/20. Based on that feedback, mentorship training for both mentor and mentee will be considered. Lessons learned during the various WiSEMOMOY awards were implemented [Sparrius, The Wisest System Engineering Mentor and Mentee Award of the Year, 2019].

“In my experience some mentors were ill-prepared for mentorship, were unaware of the time commitment needed, and thus did not provide the required effort. As young engineers, the resultant confrontations were awkward, given that some mentors were also line managers. Tight collaboration between mentor and mentee(s) should strongly be encouraged. Perhaps the evaluation panel should also assess mentors.”

Respondent to GYSEYOY questionnaire, November 2019

Evaluation

Even though the term “evaluation” is used, it is in fact a learning process just like the rest of GYSEYOY. A panel of six or so highly-experienced practicing system engineers evaluates each team’s deliverables. To maintain its independence and objectivity, the evaluation panel is not involved with the GYSEYOY training sessions nor the question and answer sessions. No evaluator should be a manager or a colleague of the team. Each evaluator should preferably be experienced in leading post-graduate students in their research, and have inter-personal skills and leadership skills. The Requirement Review is mainly used to steer the teams in the correct direction; it focuses on the methods used. Each team submits their GENESYS model data, and the evaluation panel examines the evidence provided by these deliverables. Assumptions are questioned. A free-flowing discussion follows. To prevent creating bias between teams and to keep the contest alive, any one team is not allowed to attend another team’s presentation and discussion. The System Design Review is the final evaluation and is used for scoring each team’s GENESYS™ model, presentation, and answers to questions posed by the evaluation panel.

The scoring rubric used for System Design Review evaluation is shown in Appendix 3. Evaluation focuses on the system engineering processes used by each team, the project management processes, the actual

proposed solution, and the presentation at the System Design Review. Technical development issues are evaluated as well as personal development issues, such as teamwork and the resolution of any team problems that might have occurred. The winning team is the one with the highest total weighted score, summed over all evaluators. The evaluation scoring rubric is regularly refined and updated.

The evaluation panel also decides on the individual who acted the most professionally and intelligently of all the GYSEOY contestants. That person becomes the Sharpest System Engineer of the Year and is awarded the floating trophy. The evaluation is as follows: Mentors from each team jointly select the individual from the team they think should receive the award. (In exceptional cases they may nominate two persons if they cannot agree on a single nominee.) Team members, independently from the mentors, select the individual they think qualifies for the award on a majority vote basis. The evaluation team considers the nominees from the mentors and the teams to make their final decision. Their decision will also be based on a majority vote of who, in their opinion, best fulfills the criteria defined above.

“The evaluation process should focus more on the merit of deconstructing and understanding the problem, and the methods used to achieve that.”

Respondent to GYSEOY questionnaire, November 2019

“More detailed feedback on the CORE model itself would be helpful. My team had many discussions concerning how to model some aspects of the system, and I am not convinced we did everything correctly. It would have been valuable if a more experienced perspective were provided by the panel.”

Respondent to GYSEOY questionnaire, November 2019

“The selection of the Sharpest System Engineer of the Year should be more transparent. Which criteria were used? Surely it cannot be based just on the presentations to the panel?”

Respondent to GYSEOY questionnaire, November 2019

Lessons Learned

GYSEOY started on a learn-by-doing approach. Rather than spending many hours debating whether the concept was useful, and how it should be handled, a plan of action was created and implemented. Experience is a great master, and the program has evolved substantially. Huge trees start from small seeds. The current version of the flowchart for GYSEOY is shown in Figure 1.

GYSEOY has been well received by individual engineers and by their employers. An example is that contestants can now meaningfully reason with the most experienced system engineers.

GYSEOY encourages cultural diversity. To loosely quote the UNESCO Universal Declaration on Cultural Diversity: Culture is the set of distinctive spiritual, material, intellectual, and emotional features of society or a social group, that encompasses lifestyles, ways of living together, value systems, traditions and beliefs. Individuals have a plural identity, and communities are themselves also plural. Cultural diversity is an adaptive process for expression, creation, and innovation. It widens the range of options, and is one of the roots of development, not simply in terms of economic growth, but also as a means to achieve

a more satisfactory intellectual, emotional, moral and spiritual existence [UNESCO, 2001]. A summary of the cultural diversity of the participants in the GYSEYOY Program is provided in Table 1, below.

Table 1. Cultural diversity of GYSEYOY participants⁷

Female	Male	Black	SA Indian	Colored	White	Total
19	50	27	21	6	15	69

At least two employer companies are, at least informally, using GYSEYOY as an integral part of their in-house professional development program for their young engineers.

The future contribution of South African systems engineering lies to a considerable extent in the hands of the GYSEYOY contestants! GYSEYOY has also been a major contributor to the INCOSE SA conference, not only by increasing the number of attendees, but even more importantly, it has further strengthened the value of the Conference. Many of those who have successfully completed the GYSEYOY Program have formed an informal group with many personal and professional interconnections.

A few teams have dropped out during the challenge, for instance, in 2017 three teams dropped out at various stages and for various reasons. Some contestants underestimated the effort required; also some teams were depleted when contestants resigned from their employer. Other contestants were engaged in post-graduate studies as well as in GYSEYOY, and decided to focus on their studies. A deliberate effort has been made during the participant selection process to explain the effort that is required from each individual and each team, in order to prevent any misconceptions.

Training is crucial, since the selection criteria for contestants implies that they have limited system engineering experience, and few have experienced formal system engineering training. As discussed previously, two days of online training are provided in support of each GYSEYOY stage. The scheduled question and answer sessions are also used to present tutorials on selected system engineering topics relevant to GYSEYOY, for instance on use cases, activity diagrams, and decision-making models. Some argued that mentors should perform such training, but that would undercut the focus of mentoring on personal development.

Replication of the GYSEYOY Program in other Locations

It has been suggested that some INCOSE chapters might want to implement a similar program. To facilitate accomplishing this, Figure 1 below provides a flowchart of the activities involved.

There has been some criticism that the training focused too much on GENESYS™ at the expense of system engineering. However, it appears that the future will be dominated by model-based system

⁷ These numbers are different from those in the previous version of this article. The definition of a contestant has been tightened—only those who have *fully completed* all GYSEYOY requirements are included here. For instance, during GYSEYOY 2019/20 some teams discontinued before the GYSEYOY completion, and thus are not counted.

engineering; GENESYS™ is an appropriate introduction. This also forces the contestants to follow the logic of system engineering as embedded within GENESYS™. For instance, verification requirements and traceability are compulsory and not negotiable.

A suggestion was that additional emphasis on systems thinking should be provided. However, despite the increased interest in systems thinking, it seems that few systems engineers are incorporating that approach in systems engineering practice. At the risk of oversimplification, the main contribution of systems thinking to systems engineering is the focus on the web of interrelations between system elements that results in emergent behavior that is often dynamic. The dynamic behavior of systems is created by positive and negative feedback loops, as explored by causal loop diagrams and stock and flow diagram that lead to system archetypes. System thinking provides powerful tools and mental models [Sterman, Business Dynamics: Systems Thinking and Modeling for a Complex World, 2000]. However, it is not evident that GYSEOY would benefit from incorporating those tools, and system thinking would inevitably complicate the learning needed to complete the challenge. In a similar vein, during the first few GYSEOYs the dynamic behavior of a system was handled by means of state machine diagrams. Those diagrams required an afternoon of training, but turned out to not be needed in order to provide a reasonable solution to the scenario. State machine diagrams were subsequently removed from the training.

There are two keys to GYSEOY success: Firstly, the responsibilities for undertaking and managing the GYSEOY Program must be shared among many individuals - "Many hands make light work." In this case, "Many hands make it possible to provide the Program". Providing a program such as GYSEOY requires a team effort. Many skills are needed, including mentoring, leadership, training, developing scenarios, and others - no one person (or even a few people) can do everything. Secondly, a network is needed of experienced system engineers who are willing to participate in the professional and personal development of young system engineers. Recently retired individuals have been the main source of these collaborators. People in the prime of their career, say in their forties or fifties, are often too busy to perform the required roles in assisting GYSEOY.

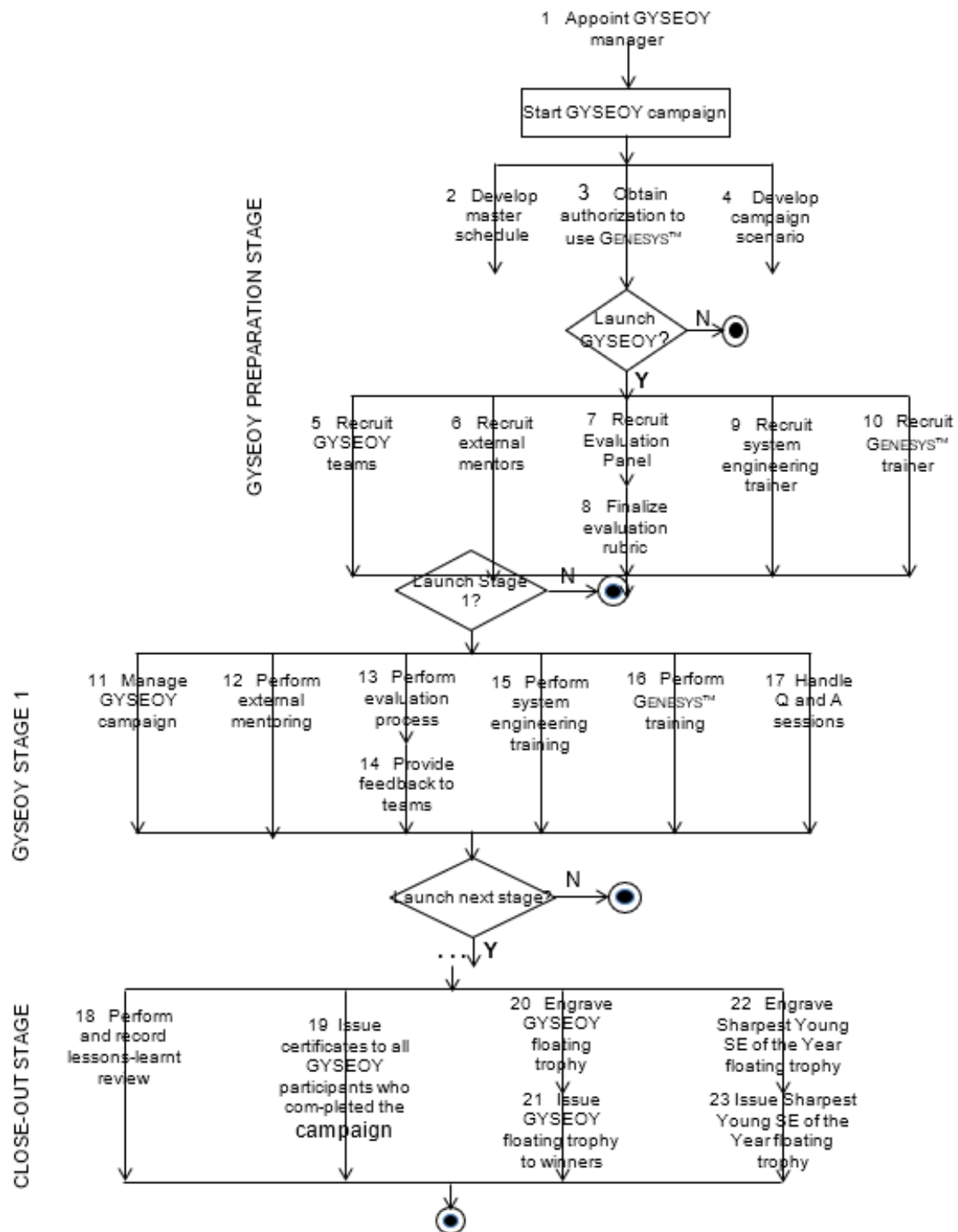


Figure 1. Activities to Instantiate a Similar Program

Unresolved Issues

Some issues have been regularly debated but no consensus has yet emerged. Since GYSEYOY is a learn-by-doing process, it will take some time before these issues will be resolved⁸. Arguably the most problematic issue of all is the nature of the scenario and the resultant problem statement. Some argue that a socio-economic problem is a poor place to start a systems engineering career since it is too complicated and abstract, and outside the comfort zone of a young system engineer. Substantial up-front

⁸ Some reviewers of this article desired answers to these issues, but that will need a few more GYSEYOY challenges to resolve.

contextual and conceptual effort is needed before the system engineering process can be launched. In my personal experience, the importance of fulfilling stakeholder requirements is crucial to engineering, but I did not learn that insight from system engineering. A marketing course in MBA studies revealed that to me. GYSEYOY has been designed to forcibly bring that insight home to contestants from day one—stakeholders are difficult to identify, and eliciting their requirements is difficult. Proxies need to be identified for generic users. That lesson should be learned at the beginning of a system engineer's career—day one is almost a day too late. A socio-economic problem is ideal to master that insight. In my opinion, the best way to learn those skills is to be thrown into the deep end in an unfamiliar area where one is forced back to first principles you didn't even know existed.

Another unresolved issue is scaling—would it be possible to scale up GYSEYOY? The maximum number of contestant teams in any year has been six. Could that be scaled up to say twelve teams? Each design review used for evaluation would have to double its length to four days. Twelve external mentors would be needed. The online training seminars and online Question and Answer sessions would easily be expandable. The effort of the GYSEYOY manager would also need to double. Only experience will resolve the scaling issue.

"The overall GYSEYOY experience was valuable, not only in a systems engineering context, but also the exposure to and nurturing of skills related to project management, conflict management, communication with stakeholders, and building confidence in presenting and defending concepts to an evaluation panel. GYSEYOY was worth the effort."

Respondent to GYSEYOY questionnaire, November 2019

"GYSEYOY was useful to the advancement of my own skill set, which made me more adaptable and knowledgeable in areas that I otherwise may not have been exposed to at this stage of my career. The Requirements Review was the first real occasion where I had to present my views to an evaluation panel—I had previously only been involved in smaller, informal reviews. This experience helped me at a Critical Design Review meeting later that year, and subsequently at various other reviews of my designs since then. Additionally, the exposure to the principles and application of deriving useful requirements from abstract, informal conversations with stakeholders is an extremely important skill to have nurtured."

Respondent to GYSEYOY questionnaire, November 2019

"I enjoyed the interaction with stakeholders, eliciting stakeholder needs, and the process of subsequently deriving them into measurable and verifiable requirements. This was a useful exercise that helped me to apply systems thinking, which is beneficial to any problem. This experience should be further enhanced by placing more emphasis on this stage of GYSEYOY, and by also incorporating it into the evaluation process."

Respondent to GYSEYOY questionnaire, November 2019

"The best aspect of GYSEYOY is exploratory learning in a team context."

Respondent to GYSEYOY questionnaire, November 2019

As a complementary component of the training provided to the participants, it would be useful if each GYSEYOY team developed and presented a formal conference paper concerning its solution and then presented it at the INCOSE SA conference. A constraint is that the time pressure leading up to the System Design Review to successfully complete GYSEYOY would preclude the development of any

formal paper. It is hoped that an approach will evolve that will enable this component of the training to be provided.

Acknowledgements

GYSEYOY has been a marvelous example of the truth of the old proverb “Teamwork divides the task and multiplies the success.” Many people contributed their skills and knowledge to create and sustain GYSEYOY. Cobus Scannell deserves special thanks because he managed the 2015 through 2018 versions of GYSEYOY and also provided a solid foundation for its future evolution.

Acronyms Used in this Article

<u>Acronym</u>	<u>Explanation</u>
GENESYS™	Vitech Corporation’s Model-based Systems Engineering Software
GYSEYOY	Greatest Young Systems Engineers of the Year Challenge
M.Eng	Master of Engineering (university degree)
MBA	Master of Business Administration (university degree)
MOE	Measure of Effectiveness
INCOSE	International Council on Systems Engineering
IS	International Symposium
SA	South Africa
UNESCO	United Nations Educational, Scientific and Cultural Organization
WiSEMMOY	The Wisest System Engineering Mentor and Mentee Award of the Year

References

Sparrus, A 2019, ‘The Wisest System Engineering Mentor and Mentee Award of the Year’, Unpublished manuscript 95 prepared for IS 2020.

Churchman, 1967, ‘Wicked Problems’, *Management Science*, vol 14, no 4, B-141–B-146, <https://doi.org/10.1287/mnsc.14.4.B141>

Oxford University Press, 2020, Oxford Dictionary of English, MobiSystems Dictionary Viewer, Version 12.4.191. UNESCO (United Nations Educational, Scientific and Cultural Organization), 2001, UNESCO.

Universal Declaration on Cultural Diversity. Retrieved 11 March 2020.

Wikipedia contributors, accessed 12 Oct 2019, ‘Wicked problem’, Wikipedia—The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Wicked_problem&oldid=920500451

Sterman, Business Dynamics: Systems Thinking and Modeling for a Complex World, McGraw-Hill Education, 2000.

Biography of Ad Sparrius



Ad Sparrius has been awarded four degrees—Bachelor of Science, Bachelor of Engineering (University of Stellenbosch), Master of Science in Electrical Engineering (University of California, Berkeley), and Master of Business Leadership (University of South Africa), and is professor extraordinarius at UNISA's Graduate School for Business Leadership. Ad got involved in system engineering during the late 1970s, and has been passionate about that discipline ever since. He presents various post-graduate courses as a consultant, and at the

Graduate School of Technology Management at the University of Pretoria and at the Graduate School for Business Leadership at UNISA. Ad has been the Technical Chair of INCOSE South Africa's 2012, 2013, 2015, 2016, 2017 and 2018 conferences, and has been instrumental in elevating that conference to a new level of distinction. As part of the INCOSE SA conference, he launched the Greatest Young System Engineers of the Year 2015, 2016, 2017 and 2018 challenges, as well as the Wisest System Engineering Mentor and Mentee of the Year 2016, 2017 and 2018 contests. Most of South Africa's system engineers had their initial system engineering training with Ad Sparrius. He is a Fellow of INCOSE. His interests include astronomy, especially high-redshift galaxies.

Appendix 1 - GYSEOY Scenarios

Year 2015

Title Colorectal Cancer Detection.

Short description: By the time colorectal cancer is detected, it may already be too late—the tumor rapidly grows and may have metastasized. Currently the only sure way of diagnosing colorectal cancer is with a biopsy. The goal is not to eliminate this, but rather to find a way to test for the possible presence of colorectal cancer that will then justify a biopsy. The detection system should preferably be non-invasive, make economic sense, should be feasible, and based on mature technology.

Year 2016

Title Musa's Story.

Short description: At an early age, while studying at the School of Music, it became evident that Musa was a musical prodigy with most instruments. Although his greatest love was the clarinet, Musa also took up conducting. The conductor's role of leading and serving musicians in an orchestra was natural for Musa, resulting in a life-long bond between Musa and his World Unified Philharmonic Orchestra. When conducting, Musa mouthed instructions to certain musicians, with perfect timing, that resulted in a single note emphasizing the subtleties of the composition. Musa also used facial expressions, leaning in or not, and stepping forward and backward to indicate the intensity needed from each instrument. At the peak of his career, Musa lost both his hands and a leg just below the knee in a disastrous accident. Develop and evaluate concepts that will enable Musa and the World Unified Philharmonic Orchestra to continue their global performances.

Year 2017

Title Better Meals for Everybody.

Short description: Aamilah Slamet lived in Macassar. After completing school, but not having sufficient money to enter university, she was investigating her career options. Based on the people living around her, and inspired by modern food technology, she decided on this objective: Better meals for everybody—Deliver great-tasting, balanced and healthy nutrition to low and middle income people in urban areas without detriment to the environment. She hoped to develop a nutrition value chain using modern technology that could rival the economies of scale of large existing agricultural production systems and their supply chains. However, she first needed to develop a business case sufficiently robust to solicit finance from an angel investor.

Year 2018

Title Teaching Science.

Short description: Parvati Mara had recently retired as school inspector of science in Gauteng. She had dedicated herself to the emotional and mental development of children by imparting her passion for a scientific evidence-based understanding of the world and the practical application of science in everyday life. She had focused on students who were born into economical and cultural circumstances where book learning and theoretical thinking were regarded as an unnecessary luxury. At the age of 68, Parvati was not yet ready to step away from education. She had encountered the Langa Education Assistance Program (LEAP) Science and Math schools that provided free education to students from high-need communities. The six LEAP schools had mathematics, physical science, and English as mandatory subjects. Parvati became obsessed with how science teaching for the LEAP schools should be adapted to both foster and exploit information and communication technology. Did best practices for science teaching exist—surely the wheel did not have to be reinvented. Why was a school itself not a learning system? She was starting to view a school as a complex adaptive system within a knowledge-driven ecosystem. But what should be the requirements for a science teaching knowledge management system for the LEAP school in Diepsloot? What should its architecture be?

Year 2019/20

Title Preparing for Ampie's Endgame.

Short description: Ampie Steyn, his wife Katryn and son Hansie lived in Jamestown in a small old house with no mortgage bond. It had two bedrooms, its structure was still fine, but the electric wiring was getting somewhat dodgy. Ampie had been a carpenter with his own business, but had re-tired a few years ago after his wife had died. Ampie was now 75 old and showed his age. Hansie was 37 years old and was an architect. Hansie was worried about his father's future, since Ampie had recently sustained a nasty fall in the bathroom. What was more, Hansie would certainly not be able to look after his father on a full-time basis. Ampie was still strong, and was classified as able to live safely on his own. But for how much longer? Both Ampie and Hansie did not consider an old age home as a viable option for Ampie. Hansie

wondered what Ampie's requirements would be for the life that was left to him. Research showed that as people become aware of the finitude of their life, they did not ask for much. They did not seek more riches. They did not seek more power. They asked only to be permitted, insofar as practical, to keep shaping the story of their life—to make choices and sustain connections to others according to their own priorities. Although Ampie was a member of a medical scheme that provided reasonable medical care, frail care was unfortunately classified as resulting from advanced age, and was thus handled as social care, not medical care. Would assisted living be an option?

Appendix 2 - GYSEYO 2018 Scenario Teaching Science

Parvati Mara has recently retired from her post as school inspector of science in Gauteng. She has dedicated her professional life to the emotional and mental development of a few generations of South African children by imparting her passion for a scientific evidence-based understanding of the world and the practical application of science in everyday life. She was particularly driven to work with students who were born into economical and cultural circumstances where book learning and theoretical thinking were usually regarded as an unnecessary luxury. At the age of 68, Parvati was not yet ready to step away from education altogether, and as any good teacher was eager to learn more herself. On the advice of friend she had investigated and then contacted the LEAP Schools for Science and Mathematics, and had volunteered her services. She had been invited to spend a day at the LEAP 4 School in Diepsloot and had been overwhelmed by the challenge and the opportunities. She especially liked the LEAP Schools' point of departure:⁹

"Every child in South Africa can and will exceed expectations if provided a real opportunity to learn and liberate in a school-learning environment that is caring, challenging, personal and lives out relentless high expectations!"

But Parvati wondered how she could help the LEAP 4 School. What key contributions could she make to further improve its performance?

The six Langa Education Assistance Program (LEAP) Science and Math schools provide free education to students from high-need communities, and has mathematics, physical science and English as mandatory subjects. A school schedule includes compulsory Saturday classes and formal holiday programs. The school day lasts nine hours, from 8h15 to 17h15. Based in a converted warehouse in Diepsloot, the LEAP 4 School faces the infrastructure challenges so prevalent in informal communities. The warehouse is very noisy and is shared with a pre-primary school, hence the conceptualization and planning for a new school building has started. The students at Diepsloot are a mix of many cultures, predominantly Sepedi. The school offers isiZulu and Sepedi as home languages, along with English. Every LEAP School is partnered with a more privileged school, Dainfern College for LEAP 4, as well as a township school in the community the school serves. This three-way collaboration creates an

⁹ See <http://leapschool.org.za/>.

opportunity to share excellence in all spheres. The founding supporter for the LEAP 4 School is the Aveng Group.

The 2016 results of the LEAP 4 Diepsloot School were impressive—28 out of 29 students passed Grade 12, with all 28 gaining access to tertiary education. The overall LEAP pass rate in 2016 was 93%, with 78% qualifying for study at an academic or a technical university. What is more, all LEAP students write mathematics and science, compared to only 42% who wrote math and 33% who wrote physical science nationally in 2014. Historically 72% of LEAP learners pursue graduate studies, with 32% in accounting, 14% in engineering and 12 in education.

Digital literacy—the skills of searching for, discerning, generating and managing information—is key to higher-order thinking skills, and is crucial to participating in the national and global economy. For instance, surely students should be able to use their own smartphones for learning during class time? Surely e-readers should form the basis for all textbooks? However, many teachers are digitally illiterate. Teachers will need considerable professional development to change their teaching methods. Schools will have to digitize their education management, governance and administration. How should that happen?

Parvati thought long and hard about these issues. How should science teaching for the LEAP schools be adapted to both foster and exploit information and communication technology? Why could the tacit knowledge of master teachers not be unlocked, systematized and then replicated in the next-generation teachers? Why did the wheel have to be reinvented—surely there were science teaching best practices? These would obviously need to be adapted to Diepsloot. The science teaching outcomes clearly depended on the student, the teacher, their relationship, the curriculum, the school environment, the student's home environment, and the presence, or the absence, of a culture of learning. But why was a school itself not a learning system? She recalled the old science saying—to measure is to know. Why were science-teaching outcomes not routinely measured as a key part of on-going data-driven controlled experiments to improve those outcomes? Should LEAP not develop evidence-based science teaching knowledge management? That would clearly demand the most-appropriate technology, life-long learning of the teacher, and incisive cultural interventions, and changes to the school's organization. Parvati realized that she was starting to view the school as a complex adaptive system within a knowledge-driven ecosystem.

Team4Tech would be a part of the solution. It comprises a group of volunteers from different parts of the world who help LEAP schools enhance teacher and student competencies in information and communication technology. One of their focuses is the greater use of technology in the classroom. Team4Tech consists of two core competency groups: The Technology group works towards improving LEAP hardware and software resources while the Teacher Training group works towards enhancing teachers' skills in order to implement technology as a tool in the classroom. For instance, the Technology group had implemented the free web-based Book Source Classroom Organizer application for better library management. Parvati was just concerned that Team4Tech considered technology as the solution,

whereas she was focusing on a science teaching knowledge management system. Those were very different objectives!

What should the requirements be for Diepsloot science teaching knowledge management? What should its architecture be?

Parvati started blogging as a means to clarify her own thoughts on the matter. A team of young systems engineers happened upon this blog and decided to volunteer their skills. They were eager to put their systems engineering knowledge and skills to the test to tackle this complex problem. They would apply requirements elicitation and develop a solution proposal that would address these requirements.

Appendix 3 - Scoring Rubric

	Weight
<i>I. The System Engineering process</i>	65%
1.1 Stakeholder requirements	20%
Were stakeholders and their requirements identified from analyzing the scenario?	
Was expert knowledge obtained via research and/or interviews?	
Was a context diagram constructed and were external interfaces specified?	
1.2 System requirements	15%
Were use cases defined for all stages in the life cycle, including installation cases, maintenance cases, repair cases, disposal cases, et cetera?	
Were performance requirements specified for each use case?	
Was a verification case defined for each performance requirement?	
Were non-functional requirements defined, for instance environmental requirements, physical requirements, -ility requirements?	
1.3 System element requirements	15%
Was an activity diagram developed for each use case, in other words were element functions (sub-functions) defined?	
Were performance requirements specified for each element function (allocated from system performance requirements)?	
Was a verification case defined for each performance requirement?	
Were all functional and physical requirements tree-up and tree-down traceable?	
1.4 Architectural design	15%
Was a functional architecture defined? Were alternative architectural architectures defined?	
Was a physical architecture designed (block diagram that also specifies internal interfaces)? Were alternative physical architectures defined?	
If alternative architectures were defined, was an explicit value model used to select the best architecture?	
Were non-functional requirements allocated to elements?	

2. *The Proposed Solution*

20%

Has a concise problem statement with appropriate measures of effectiveness been defined?

Was an explicit root cause analysis performed?

Does the proposed solution as defined in the system design description genuinely solve the stated root problem?

If you owned the root problem and suffered from its symptoms, would you invest your personal money in this solution?

3. *Project Management*

10%

Was an integrated project plan developed, including a work breakdown structure, deliverables for each activity, a schedule for each activity, and a person-hour budget for each activity?

Were appropriate ISO 15288 system engineering processes selected and tailored to this project by means of the person-hour budget?

Was the actual versus planned person-hour budget and actual versus planned schedule regularly measured? Was corrective action taken when necessary?

Was a closeout report included in the System Design Review that included lessons learnt, recommendations for future GYSEOY challenges, as well as hard measured data on each individual's effort?

Were internal design reviews conducted in preparation for the formal GYSEOY design reviews?

Were risks identified, analyzed, prioritized and entered into a risk register?

4. *System Design Review Presentation and Interview*

5%

Was the presentation enthusiastic and inspiring?

Was the GYSEOY content appropriately presented? Did the entire team participate?

Was the response to questions asked and issues raised?

3. ADDITIONAL ARTICLES

3.1 INCOSE Program Management and Systems Engineering (PM/SE) Integration Working Group: Bridging the Gap

by

John Lomax, Jean-Claude Roussel, Tina Srivastava, Rachel Mouring,

Mark Kaufman, and Ralph Young

INCOSE PM/SE Integration Working Group

October 12th, 2020

Email addresses of the authors: john.lomax@airbus.com, jc.roussel6231@gmail.com, tina.srivastava@incose.org, Rachel.mouring@gmail.com, mkaufman@mitre.org, ryoung@ppi-int.com

Abstract

The PM/SE Integration Working Group (WG) of INCOSE was created in July 2016 at the International Symposium of INCOSE in Edinburgh (UK). This was done in conjunction with several long-term actions initiated by INCOSE, PMI, and MIT; the need for the working group surfaced several years prior to its instantiation. The idea was born from a strategic alliance of INCOSE with the Project Management Institute (PMI). This alliance was first established through a Memorandum of Understanding (MOU) between INCOSE and PMI. A joint white paper conceptualizing the potential union was penned by the respective President of INCOSE and CEO of PMI (Samantha Robitaille/John Thomas and Mark A. Langley), and was published in *INSIGHT* and *PM Network Magazines* in 2011.

A joint study between INCOSE, PMI and MIT was conducted in 2013 and 2014 that included a large survey among several hundred companies worldwide. This study has been published by INCOSE (International Symposium “IS” 2013) [REF 1]. A foundational and groundbreaking book on this topic was initiated in 2015 and published in 2017 under the title *Integrating Project Management and Systems Engineering* [REF 2].

The PM/SE Integration WG was established with the intention of identifying and promoting opportunities associated with effective integration of the Systems Engineering and Project/Program Management disciplines. This WG aims to facilitate collaboration between SE and PM communities, produce useful deliverables that support effective integration, and provide thought leadership on integration challenges. The WG is co-chaired by Tina Srivastava (MIT), Jean-Claude Roussel (Airbus, France - Retired), and John Lomax (Airbus Defense & Space, UK).

The WG has undertaken a number of initiatives, which are topically organized and serve to address the WG scope. These initiatives include Strengthening Program Management and Systems Engineering

Integration, Comparison of the *PMBok* and *SE Handbook*, Strategic Technical Planning, Project Breakdown Structures, Digital Transformation, Competencies, and Education. The intent of this article is to provide an overview and status of the WG Objectives and Initiatives and to provide an on ramp for those who may wish to become involved.

Copyright © 2020 by John Lomax, Jean-Claude Roussel, Tina Srivastava, Rachel Mouring, Mark Kaufman, and Ralph Young. All rights reserved.

Introduction

The scope of the PM/SE Integration WG encompasses activities related to defining, capturing, evolving, and communicating PM/SE integration best practices. This includes the development and/or proliferation of training material, guidelines material, recommendations for industry best practices and standards, and shared output with industry working groups from other organizations. Additionally, we seek to partner with other INCOSE working groups, including the Requirements, Risk, Lean SE, and Agile SE working groups, among others, as appropriate. Through these partnerships, we endeavor to ensure that subject matter expertise is integrated into various aspects of the systems engineering process, and we seek to explore common problems and/or practices.

The purpose of this WG is to identify and promote opportunities associated with the effective integration of the Systems Engineering and Project/Program Management disciplines. To accomplish this mission, we explore the linkages necessary to create effective integration and collaboration between systems engineers and program managers. We also serve as the intersection point where systems engineers and program/project managers collaborate and integrate their efforts, as shown in Figure 1, below.

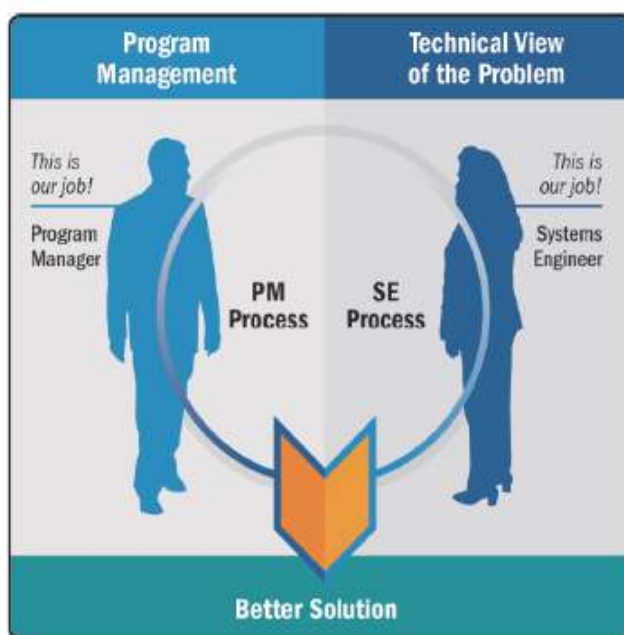


Figure 1 – Integration of Program Management and Systems Engineering

Core Objectives

The core objectives as defined in the WG charter are as follows:

- Facilitate collaboration between the systems engineering and program management communities.
- Demonstrate the value of integrating systems engineering and program management to develop better solutions that drive strategic business results and outcomes.
- Produce useful deliverables that support effective integration and practice of collaborative systems engineering and program management.
- Provide thought leadership concerning open integration challenges involving program management and systems engineering.
- Bring external thinking into the systems engineering and program management communities to facilitate thinking “outside the box”.
- Represent a think tank for free thinking and engagement concerning critical issues associated with program management and systems engineering.
- Draft guidelines and/or influence existing guidelines (e.g., the Program Management Book of Knowledge (PMBok), the Systems Engineering Handbook (SEH), and many others.) based on experience and exchanges concerning PM/SE integration and collaboration.

PM/SE Integration WG Initiatives

The PM/SE Integration WG leverages INCOSE member volunteer efforts to advocate the WG’s initiatives, and formulates ways of working collaboratively, based on agreed upon rules of engagement. The WG normally meets face to face at the INCOSE International Workshop (IW) and at the INCOSE International Symposium (IS), each of which is held annually; the working group provides status, presentations, and a workshop environment to communicate the status and activities of the WG’s initiatives and related activities. The WG has evolved a set of ‘rules of engagement’ to guide these initiatives and activities

1. Initiatives are suggested to the Co-Chairs who approve or not; and if accepted, the volunteers progress at their own discretion.
2. It is up to the leaders of each initiative to organize and coordinate the topics of virtual meetings, the frequency of meetings, the nature of specific activities to be pursued, reporting of progress and results, and so forth, and also to distill and consolidate the information gathered from WG Meetings into specific initiatives, to set objectives, and to produce deliverables.
3. The WG Co-chairs organize separate Virtual WG Meetings during each year; Initiative Leaders are requested to provide reports of progress as part of the agenda of the WG Co-chairs meetings.
4. Due to the current COVID pandemic, we are currently utilizing virtual meetings to perform our efforts.

Current Initiatives

Integration of Program Managers' and Chief Systems Engineers' Mindsets

This initiative originated very recently and is led by Dr. Ralph R. Young.

As noted previously, REF 2 (referred to below as “Rebentisch et al 2017” and “the book”) provided updated research that was performed by several key players in the systems engineering and project and program management communities. Major conclusions of the book include:

- The disciplines of systems engineering and project and program management have experienced different evolutions based in part on divergent tools, practices, and standards. Today, elements of this divergent evolution appear to be impacting the ability of the two disciplines to effectively align their work practices and collaborate (Rebentisch et al 2017, p. 11).
- In addition to potentially competitive issues between the two disciplines, organizational systems (or the lack thereof) also inhibit effective engineering program management and performance. Often, the lack of aligned practices is blamed when program disruptions occur (Rebentisch et al 2017, p. 11).
- Effective practices are critical for integrating efforts to deliver results in program environments. Engineering program environments require good planning approaches, proactive risk management, stakeholder engagement, and other similar capabilities (Rebentisch et al 2017, pp. 11-12).
- New research indicates that integration and collaboration find challenges in both the public and private sectors (Rebentisch et al 2017, p. 12).
- The book highlights successful engineering programs along with key integration elements that played a role in that success. Learning from failure is absolutely critical to advancing engineering program performance (Rebentisch et al 2017, p. 12).
- The book blazes a new path by focusing on approaches for better enabling collaborative work between program managers and systems engineers. While there is plenty of published material focused on enhancing the performance of each individual discipline, very little published matter spotlights how the two disciplines align their efforts and work collaboratively. The book reports on how the two disciplines can align their efforts to deliver results (Rebentisch et al 2017, p. 12).
- The book shines a light on enabling factors that support engineering programs. It presents new research and a framework for integration to help program managers, systems engineers, and their executive leaders enhance joint effort, joined thinking, and common language. This examination yields insights into factors that either enable collaboration or create barriers to integrated approaches (Rebentisch et al 2017, p. 13)

- The book provides further research to advance understanding of dynamics of interdisciplinary collaboration (Rebentisch et al 2017, p. 13).

The key takeaway from the book is that changes in the mindsets (mental inclinations, beliefs, and attitudes) of chief systems engineers and project and program managers of complex projects are required in order for other improvement initiatives to have a substantial impact. Implementation of the current practice of system engineering is not sustainable unless systems engineering and program management are more fully integrated (Rebentisch et al 2017, Chapter 16, pp. 343-363).

Comparison of the PMBoK and SE Handbook

This initiative originated during the PM/SE Integration WG formation; it is now led by Mark S. Kaufman. The team performing this effort started with a review of existing works that provided comparisons between *PMBoK* [REF 3] and the *INCOSE SE Handbook* [REF 4]. This included review of Eileen Arnold's "Call for an Effective Alignment of Program Management and Systems Engineering Risk Management Practices" from *INCOSE IS 2013*. [REF 6]. There is recognition that the lexicon used by each discipline may have different meanings; for example, what program managers mean by "risk" can be different from what systems engineers mean by "risk." An excerpt of this analysis (prepared by Kenneth Zemrowski) is shown below:

Term	PMBOK Guide Definition [REF 3]	SE Handbook [REF 4] and/or SEBoK Definition [REF 5]
Acceptance criteria	A set of conditions that is required to be met before deliverables are accepted (<i>A Guide to the Project Management Body of Knowledge (PMBOK® Guide)</i> . -- <i>Sixth Edition</i>) [REF 3]. See also: <i>requirement, test criteria</i> .	(1) Criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity (<i>ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary</i>) [REF 7]. (2) The procurement specification, in the context of the overall agreement, should clearly state the criteria by which the acquirer will accept delivery from the supplier. A <i>verification matrix</i> can be used to clarify these criteria (<i>SE Handbook</i>) [REF 4].
Accuracy	Within the quality management system, accuracy is an assessment of correctness (<i>A Guide to the Project Management Body of Knowledge (PMBOK® Guide)</i> . -- <i>Sixth Edition</i>). See also: <i>precision</i> .	(1) Qualitative assessment of correctness, or freedom from error (<i>ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary</i>) [REF 7]. (2) Quantitative measure of the magnitude of error (<i>ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary</i>) [REF 7].

Term	PMBOK Guide Definition [REF 3]	SE Handbook [REF 4] and/or SEBoK Definition [REF 5]
Acquisition	Obtaining human and material resources necessary to perform project activities. Acquisition implies a cost of resources, and is not necessarily financial (<i>A Guide to the Project Management Body of Knowledge (PMBOK® Guide)</i> -- Sixth Edition).	Process of obtaining a system, product, or service (<i>ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.2</i>) [REF 8]; (<i>ISO/IEC/IEEE 15288:2015 Systems and software engineering--System life cycle processes, 4.1.2</i>) [REF 9]; (<i>ISO/IEC/IEEE 24748-1:2018 Systems and software engineering--Life cycle management--Part 1: Guidelines for life cycle management, 3.2</i>) [REF 10].

Understanding how these terms map to one another can facilitate PM/SE integration. This initiative also includes developing a model of a sample project, explaining the relationship between program management and systems engineering.

Strategic Technical Planning

The Strategic Technical Planning Initiative was initiated as part of the PM/SE Integration Working Group in 2017, during the International Workshop (IW), and is currently led by John Lomax. This Initiative is built on some of the findings of the *Integrating Program Management and Systems Engineering book* [REF 2]. The main objective is to establish and agree on the overall problem statement(s) related to Strategic Technical Planning and to identify solutions, and then formulate material guidance that can be published as updates to existing materials.

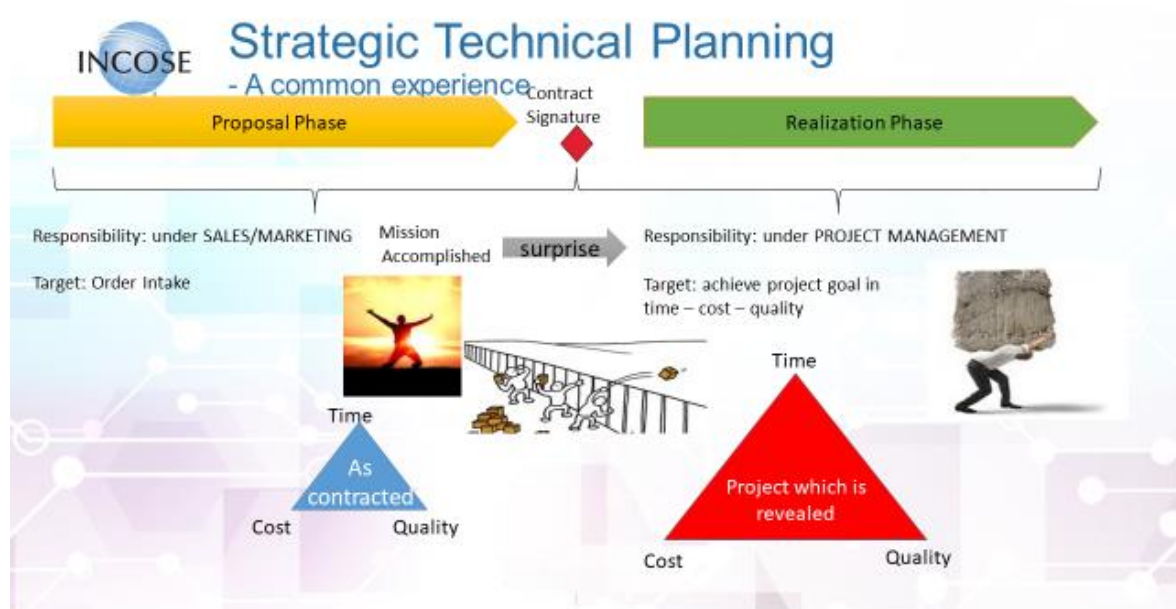


Figure 2 – A Common Experience

The main problem statements reflected in Figure 2 as currently understood, are strategic in nature and describe the main issues:

- The Contractor-Supplier ‘acquisition game’ of perceptions, influence, persuasion, and potential conflicts of interest, leading to an unsatisfactory outcome for both sides [REF 11].
- The ‘surprise’ of the Contract award, expected to be delivered by Program Management and Systems Engineering, when the ‘seeds of failure have already been sown’.
- ‘Unproductive Tension’ caused by Program Managers and Systems Engineers silos.
- Inability of system engineers to execute Performance-Based Earned Value [REF 19].
- The subsequent effects of weak planning (see Figure 3).

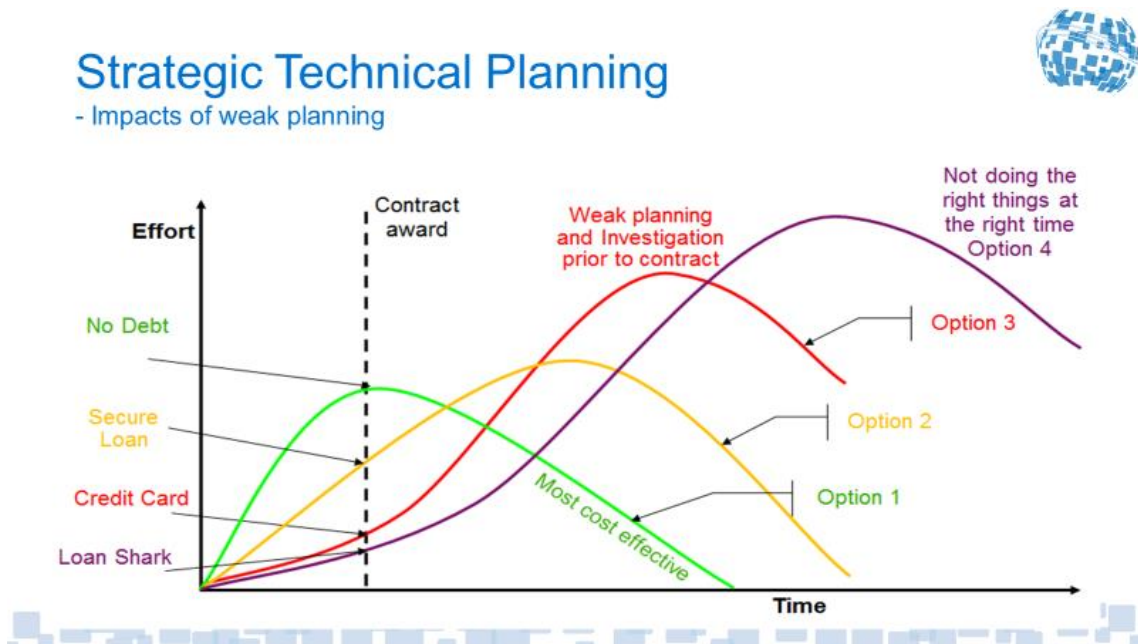


Figure 3 – The Impacts of Weak Planning

The Initiative’s approach is to:

- Understand and define the strategic approach that addresses the above issues and technical maturity requirements across the lifecycle.
- Recognize the importance of the technical management process as part of Strategic Technical Planning.
- Understand the coupling between Program Management (Plan Do Check Act [PDCA]) and Technical Management (Observe Orient Decide Act [OODA]) approaches.
- Act as a home for proposed related concepts and custom-made solutions and concepts for further exploration, e.g. PhD research activities. [REF 12]

This initiative continues to explore the boundary of these issues, with the objective of providing work products that will drive needed change.

Project Breakdown Structures

The Project Breakdown Structure (PjBS) initiative was introduced at the IW 2017 event, during which a dedicated breakout session was provided. Jean-Claude Roussel has been the leader of this initiative since that time.

The purpose of a PjBS is to have a common and consistent view of the different breakdown structures (Functional Breakdown Structure, Work Breakdown Structure, Product Breakdown Structure, Organization Breakdown Structure, and Cost Breakdown Structure) all along the project life cycle from a project management point of view as well as from a systems engineering point of view. The most current Project Breakdown Structures considered are:

- The Functional Breakdown Structure (FBS) defining the main functionalities of the product: the WHY.
- The Product Breakdown Structure (PBS) describing the product components: the WHAT.
- The Work Breakdown Structure (WBS) enumerating the tasks to realize: the HOW.
- The Organization Breakdown Structure (OBS) summarizing the responsibilities of the tasks: the WHO.
- The Cost Breakdown Structure (CBS) providing the budget/cost allocated to each task: the HOW MUCH.

These Project Breakdown Structures are the key elements and the pre-requisites for effective Integrated Planning, which is an early source of tension between Project Managers and Systems Engineers, as discussed in *the Joint MIT/PMI/INCOSE survey – White paper presented at the 23rd INCOSE Annual International Symposium, Philadelphia, June 2013* [REF 1].

Project Breakdown Structures are usually described in the Project Plan in the project management world or in the Systems Engineering Management Plan (SEMP) in the systems engineering world, and most often they are created separately and asynchronously by different staff belonging to either the PM or the SE team, with no collaborative exchange required or undertaken. The consequences of this lack of collaboration are inconsistencies between these breakdowns (mainly between the PBS and the WBS), jeopardizing successful project outcomes.

The WBS is well known and documented in the project management world; the PBS is more familiar to systems engineers. The WBS, PBS, and other breakdown structures are generally poorly documented, not well understood, and the maintenance of them is not well supported.

The relationship between these breakdowns and the respective view from PM and SE perspectives is described in Figure 4 below:

Project Breakdown Structures The SE and PM Views

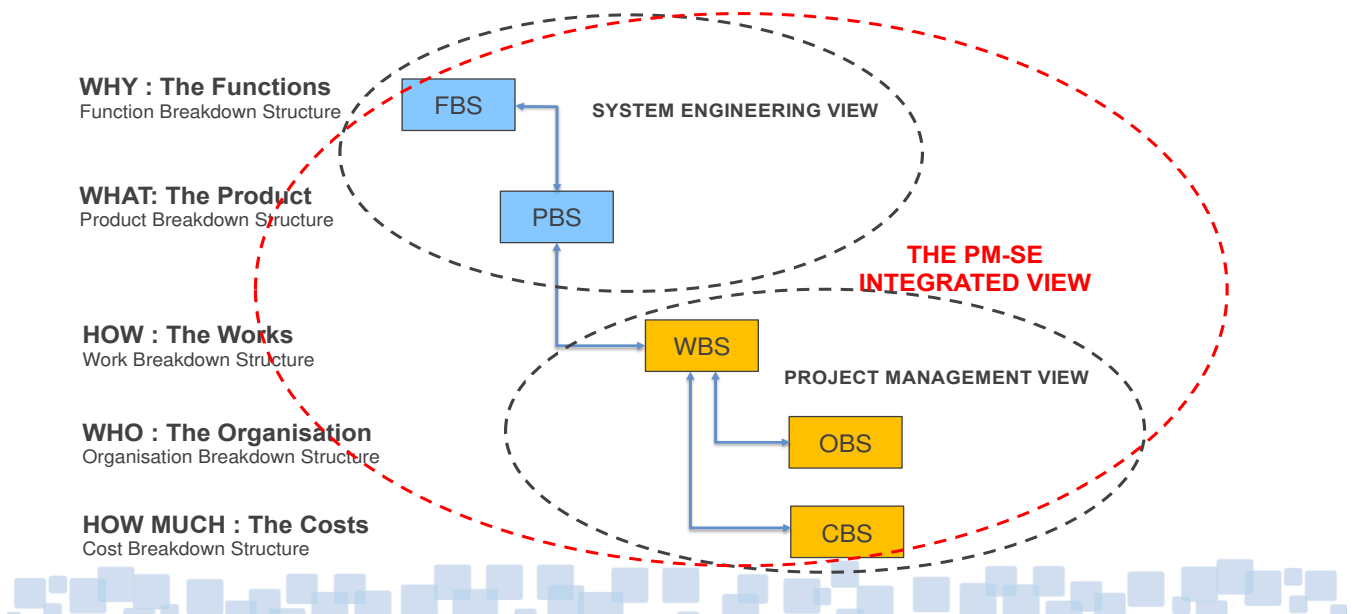


Figure 4 – The Project Breakdown Structures (PM/SE WG, IW 2020, JC Roussel)

One of the goals of the PjBS initiative is to harmonize the methods of the different breakdown elaboration from PM and SE points of view by ensuring a consistency of their respective reference standards; for program management, see the *PMBok* [REF 3] and for the systems engineer, see the *INCOSE Handbook* [REF 4] and *SEBoK* [REF 5]. The existing ISO documents ISO 21511 [REF 13] concerning the WBS and ISO 27026 [REF 14] concerning Project Management Structure breakdowns should be updated accordingly.

Education Initiative

This initiative was developed during a WG breakout session at the 2019 International Symposium (IS). The current leader is Jean-François Veron (jean-francois.veron@enac.fr).

An increasing number of universities and engineering schools are realizing the impact of PM or SE Certification when students are seeking their first job. INCOSE ASEP Certification is now recognized as one of the best certificates if one desires to work in systems engineering; similarly, the Certified Associate in Project Management (CAPM), the Project Management (PM) Certificate from PMI, is a suitable target for students who want to be involved in project management. Some schools, however, have experienced interest in combining learning sessions to help students obtain certifications in both systems engineering and project management. As emphasized earlier in this article, systems engineering and project management are closely linked in the conception, development, and delivery of products.

The Education Initiative aims at explaining these links in a pedagogical way by using material of the other's initiatives and through interactive teaching tools (for example, Quiz, e-learning, Massive Online open Courses (MOOC) (courses delivered online and accessible to all for free), and others). An initial effort, planned for 2021, is to build a database of 250 questions with detailed explanations of the answers and specific references to the *PMBOK* and to the *INCOSE SE Handbook*. The resulting quizzes would

be available on the INCOSE and PMI websites. We are currently looking for individuals who are interested in helping to develop and hone these questions and explanations.

Other Initiatives and Activities

There are many initiatives within the scope of the PM/SE Integration WG. The scope and number of initiatives are constantly changing, due to volunteer availability and progress. Digital Transformation¹⁰ and Competencies¹¹ among other initiatives are in need of new leaders. Members of the WG continually evaluate new Initiative ideas, for example, incorporating agile project management into initial project planning and execution [REF 15]; independent papers such as *Integrating Program/Project Management and Systems Engineering in Practice* [REF 16], and many others. If you have Initiative ideas please contact one of the WG Members, we will be pleased to receive and evaluate your suggestion.

Summary

The purpose of the PM/SE Integration WG is to identify and promote opportunities associated with the effective integration of the systems engineering and project/program management disciplines. The scope of the PM/SE Integration WG encompasses activities related to defining, capturing, evolving, and communicating PM/SE integration best practices. The PM/SE Integration WG leverages INCOSE member volunteer efforts to advocate the WG's initiatives, and formulates ways of working collaboratively, based on agreed upon rules of engagement. It is up to the leaders of each initiative to organize and coordinate the topics of virtual meetings, the frequency of meetings, the nature of specific activities to be pursued, reporting of progress and results, and so forth, and also to distill and consolidate the information gathered from WG Meetings into specific initiatives, to set objectives, and to produce deliverables.

The Working Group provides thought leadership and engagement concerning critical issues associated with program management and systems engineering. The WG develops materials that demonstrate the value of integration to help influence change and that support effective integration.

Five initiatives were described in this article: Integration of Program Managers' and Chief Systems Engineers' Mindsets; Comparison of the PMBoK and SE Handbook; Strategic Technical Planning; Project Breakdown Structures; and an Education Initiative. Two additional initiatives are planned: Digital Transformation and Competencies.

Recent research performed by Rebentisch and many others concluded that changing the mindsets of program managers of complex projects and chief systems engineers is prerequisite to achieving needed change. Implementation of the current practice of system engineering is not sustainable unless systems engineering and program management are more fully integrated. Therefore, the purpose, goals, scope,

¹⁰ The aims of the Digital Transformation Initiative are to address opportunities for future digitalization across the systems engineering and project management disciplines. This Initiative looks at various data-structure constraints and toolsets.

¹¹ The purpose of the Competencies Initiative is to identify the commonalities and overlaps between SE and PM roles and responsibilities, including complementary profile (collaboration) to ensure consistency.

and initiatives of the members of the INCOSE Program Management and Systems Engineering Integration Working Group are critical to the capability of the systems engineering profession being able to manage complex projects and to meet the needs of the World. INCOSE can continue to be a leader in pursuing a multi-pronged approach that includes targeted communications and activities to the various stakeholder groups as well as different levels of stakeholders. This effort will also require broad industry support and it will require collaboration. Perhaps the biggest challenge of driving change is changing the culture. This will require getting individuals and organizations to embrace needed change.

The members of the PM/SE Integration WG encourage that proactive efforts are made and will be delighted to receive ideas and suggestions. We encourage anyone who is willing to join in or provide input to contact any of the authors of this article – email addresses of all of the authors are provided at the beginning of the article.

List of Acronyms Used in this Paper

<u>Acronym</u>	<u>Explanation</u>
ACM	Association for Computing Machinery
ASEP	Associate Systems Engineering Professional
BKCASE	Body of Knowledge and Curriculum to Advance Systems Engineering
CAG	Certification Advisory Group
CAPM	Certified Associate in Project Management
CBS	Cost Breakdown Structure
ESEP	Expert Systems Engineering Professional
FBS	Functional Breakdown Structure
INCOSE	International Council of Systems Engineering
IS	International Symposium
IW	International Workshop
LSM	Lean Startup Method
MIT	Massachusetts Institute of Technology (USA)
MOOC	Massive Open Online Courses
OBS	Organizational Breakdown Structure
OODA	Observe Orient Decide Act
PBS	Product Breakdown Structure
PDCA	Plan Do Check Act
PjBS	Project Breakdown Structure
PM	Project Management
PMI	Project Management Institute
PMP	Project Management Professional
PM/SE	Program Management and Systems Engineering
PMBok	Program Management Book of Knowledge
SEBoK	Systems Engineering Book of Knowledge
STP	Strategic Technical Planning
WBS	Work Breakdown Structure
WG	Working Group

Acknowledgements

Education Initiative description - Jean Francois Veron (ENAC France)

Figures 2 & 3 - Gary R Smith & Chris Boenisch (Airbus Defense & Space)

References

1. Joint MIT/PMI/INCOSE Survey Report: *Improving Integration of Program Management and System Engineering* (Results of a Joint Survey by PMI and INCOSE), Whitepaper presented at the 23rd INCOSE Annual International Symposium, Philadelphia, June 2013.
2. *Integrating Project Management and Systems Engineering*, Eric Rebentisch et al, Wiley, PMI, INCOSE, and the Consortium for Engineering Program Excellence (CEPE) at MIT, (2017). Available [here](#).
3. *The Project Management Institute (PMI) Body of Knowledge (PMBok)*, Sixth Edition, 2017, www.pmi.org. Available [here](#).
4. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes*, Wiley, 4th Edition, 2015. Available [here](#).
5. *Guide to the Systems Engineering Body of Knowledge (SEBoK)* www.sebokwiki.org
6. “Call for an Effective Alignment of Program Management and Systems Engineering Risk Management Practices” by Eileen Arnold (INCOSE IS 2013).
7. ISO/IEC/IEEE 24765:2017: *Systems and Software Engineering-Vocabulary*.
8. ISO/IEC/IEEE 12207:2017: *Systems and software engineering--Software life cycle processes*.
9. ISO/IEC/IEEE 15288:2015: *Systems and software engineering--System life cycle processes*.
10. ISO/IEC/IEEE 24748-1:2018: *Systems and software engineering--Life cycle management*.
11. Wasson, Charles et al, “Changing the Acquisition Game- Alleviating Unreasonable PM/SE Constraint Risks” (28th INCOSE International Symposium (IS), 2018). Available at <https://onlinelibrary.wiley.com/doi/epdf/10.1002/j.2334-5837.2018.00476.x>. See Abstract, below.
12. *A Model(s)-Based Approach to PM/SE Integration*. PhD Study by Kamran Shahroudi (kami@colostate.edu) and Raymond Jonkers (ray.jonkers@merlantec.ca).
13. ISO 21511: *Work breakdown structures for project and program management*.
14. ISO 27026 Space Systems: – *Program Management – Breakdown of project management structures*.
15. Lean Startup Method (LSM) and Agile (Agile SE/SWE and Agile PjM) for Initial Project Planning (IPP) - Mike Pafford (mepafford@verizon.net).
16. “Integrating Program/Project Management and Systems Engineering in Practice” IS 2019 – Hahn and Hodges. Available [here](#).
17. Young, Ralph R. Article series addressing program managers and chief systems engineers integration. Victoria, Australia: *Project Performance International Systems Engineering Newsletter* (PPI SyEN), Issues 56-73, August 2017 through January 2019. Available at <https://www.ppi-int.com/syen-newsletter/>.
18. INCOSE Webinar: *Three Years Later- Building the Mindset for PM/SE Integration*. Randall C. Iliff, Eric S. Rebentisch, and Stephen Townsend. June 15, 2020. Available at <https://connect.incose.org/Library/Webinars/Pages/INCOSE-Webinars.aspx>.
19. Solomon, Paul J. and Ralph R. Young, *Performance-Based Earned Value*. IEEE Computer Society and John Wiley & Sons, 2007. Available [here](#).
20. *The Project Management Institute (PMI) Body of Knowledge (PMBok)*, Seventh Edition, 2021, www.pmi.org. This new and completely re-focused edition will become available [here](#) on March 31, 2021.

About the Authors

Mr. John Lomax



John Lomax is a Program Systems Engineer currently at Airbus Defense & Space, UK, who has over 30 years of experience gained mainly across the Aerospace, Defense, and Air-traffic Management domains. He has previously worked for BAESYSTEMS, Internationally at Lockheed Martin, and at the SESAR Joint Undertaking (SJU) European Commission Agency. He is an INCOSE CSEP, CAR & CAG Member and is presently a Co-Chair of the INCOSE WG on PM/SE Integration since July 2019.

Mr. Jean-Claude Roussel



Jean-Claude is a Systems Engineering Senior Expert with 37 years of experience in the Airbus Company (Toulouse, France). He is now a consultant in systems engineering and project management, following his retirement from Airbus in June of 2018. His last appointment at Airbus was in the Corporate Technical Office for leading the Systems Engineering Steering Committee, which defines the strategy for systems engineering for the Airbus Company. He has been in charge of Configuration Management, Project Management, and Systems Engineering successively for most of the aircraft developments (A320/A319, A330/A340, A380, A400M and A350 programs), and also invested several years working on space programs for ESA within Airbus Defense and Space division. He was President of the French Chapter of INCOSE in 2007/2008, Technical Director of INCOSE in 2011 through 2013 and Director of the EMEA Sector of INCOSE in 2014/2018. He was a member and co-author of the BKCASE project and is presently Co-Chairman of the INCOSE WG on PM/SE Integration since July 2016. He is INCOSE ESEP.

Dr. Tina Srivastava



Dr. Tina P. Srivastava is an innovator, entrepreneur, and technology expert. She is the author of *Innovating in a Secret World: The Future of National Security and Global Leadership*. Her experience spans roles as chief engineer of electronic warfare programs at Raytheon to cofounder of a venture-backed security startup. Dr. Srivastava recently served on INCOSE's Board of Directors as Secretary, and she received the INCOSE Inaugural David Wright Leadership Award for technical and interpersonal competencies in the practice of system engineering as a means for solving the great challenges of our planet. Dr. Srivastava is co-chair of the PM/SE Integration Working Group and is one of the editors of the book *Integrating Program Management and Systems Engineering*. She is an FAA-certified pilot and instructor of MIT's Pilot Ground School course. She is a lecturer at MIT in the areas of aerodynamics, complex systems, technology road-mapping and selection, and aviation. Dr. Srivastava earned her PhD in Strategy, Innovation, and Engineering, a Master's in System Design and Management, and a Bachelors in Aeronautics and Astronautics, all from the Massachusetts Institute of Technology (MIT) (USA).

Ms. Rachel Mouring



Rachel Mouring is a Systems Engineer at Centauri, where she has worked for two years. She is Security+ and ASEP certified, and has a background in Cybersecurity Engineering. She is currently pursuing a Master's Degree from the University of Maryland Global Campus in Information Technology Systems Engineering.

Mr. Mark Kaufman



Mr. Mark Kaufman is a certified Project Management Professional (PMP), who works for the MITRE Corporation, with more than 40 years' experience in the development and management of IT systems. His expertise within MITRE is recognized as the Department Head for Enterprise Program and Risk Management. Prior to joining MITRE, Mr. Kaufman served as the Vice President for Program Management at Affiliated Computer Services, a Fortune 500 company. In this role, he was responsible for the implementation of project management best practices, as well as for the direct oversight of 70 technology projects. Prior to this, Mr. Kaufman was the Project Manager for Lockheed Martin's Earth Observing System project and the Data Archive and Distribution Service for the Hubble Space Telescope. Mr. Kaufman holds an MS degree in Electrical Engineering from the University of Maryland, and a BE degree from Stevens Institute of Technology. In addition to being a certified PMP, he has a Lean Six Sigma Green Belt, and is a certified Scaled Agilist.

Dr. Ralph Young



Dr. Ralph R. Young holds a Bachelor of Arts from the University of New Hampshire (USA), and a Master's Degree and Doctorate from The George Washington University in Washington D.C. (USA). He earned the CSEP certification. His career focused on the development of systems, including managing development efforts, teaching, and consulting. He has written five books including *Effective Requirements Practices*; *The Requirements Engineering Handbook*; *Project Requirements: A Guide to Best Practices*; *How to Save a Failing Project*; and (with Paul Solomon) *Performance-Based Earned Value*. He has served as Editor of the *Project Performance International Systems Engineering Newsletter* (PPI SyEN) since 2011. In recent issues of *PPI SyEN*, he provided eighteen articles concerning Integration of Program Management and Systems Engineering; one article addressing each Chapter of 'The Book', *Integration of Program Management and Systems Engineering*, Eric Rebentisch, et al, 2017. Ralph

enjoys working with authors of *PPI SyEN* articles, learning more concerning aspects of systems engineering, reading, writing, the outdoors, boating, and being a grandfather of six.

3.2 The Top Six Myths of PLM

by

[Lionel Grealou](#)

Article source: [Engineering.com](#) Published in *PPI SyEN* with permission

There are many interpretations—misinterpretations—of what Product Lifecycle Management (PLM) is about: how to define it, how to implement the tools supporting it, who it is for, who needs to understand it, and most importantly, how to realize value from it. Others argue about how to sell it and how it relates—or doesn't relate—to the underlying tools and technologies used to implement it. This post explores the six most commonly encountered myths about PLM.

Myth 1: PLM is a Tool (or to narrow it down further: an “engineering” tool)

This is the recurring debate that, at times, can get very emotional. The debate has deep roots which go back to product data management (PDM), as well as basic engineering collaboration tools. Most “PLM platforms” used in the manufacturing sector evolved from the world of 3D and CAD data management and expanded to manage the relational data of the whole enterprise. Simply put, PLM covers the “*process to create*” products, including the creative design, and now digital twins. It encompasses every activity and function involved directly and indirectly in the product creation process: designers, engineers, product engineers, manufacturing engineers, shop-floor assembly engineers, service engineers, product attribute managers, sales and marketing managers, etc. The “toolset” or system element is the means to the end: to enable data traceability and automation. PLM started as a tool and has evolved into a discipline. It took 20 years. Now it's here and it will continue to reshape and evolve.

PLM is not only about engineering, it is also about converting ideas and concepts into virtual simulations and digital models, ultimately leading to physical products and associated services. Arguably, a significant part of the product creation process is rooted in engineering. However, we as engineers, are also a very conservative people.

Many functions beyond engineering are intrinsically involved in the product creation process: marketing, sales, manufacturing, supply chain integration, finance, procurement, program management, quality, compliance, etc. They don't call it “PLM” but they carry several overlapping disciplines.

Myth 2: There is no Value from “PLM”; No need for a Business Case to Define Expected Benefits

Value from PLM is hard to measure as it is not only about resource or process efficiency. It is also difficult to measure how effective people are at collaborating or creating great products (before they actually

create them). The fact that it is complex does not mean that it has no value or should not be looked at properly. On the contrary, it actually needs to be explained over and over in context of improving operations and business change.

Typical business benefits from implementing PLM solutions focus on better data and change traceability, more effective issue identification and early resolution, improved supplier delivery performance, reduced product creation cost, reduced rework and data duplication, improved quality product, robust / on-time virtual build, reduced number of physical prototypes, improved concurrent engineering (aka collaboration), improved right first time product manufacturing and assembly, etc.

The PLM business case typically starts “**top-down**” to get stakeholder buy-in and converge on the required investments and approvals, based on a combination of:

- **New business capability introduction**—including visualization, simulation, and cross-functional data integration, vertical and horizontal process alignment.
- **Existing capability improvement**—building on new and existing processes and data, based on change management and automation opportunities.

As organizations embark on PLM implementations, initially through the realization of demonstrators and proof-of-concept, the business case can be validated (and adjusted) against a more “**bottom-up**” approach to benefit calculation. This contributes to making real value from the business change, putting in place new operational efficiency measures and scalability targets. Educating business leaders exchange with Subject Matter Experts (SMEs) about this iterative process is critical to avoid misunderstanding.

Failure to realize value from PLM is often linked to failure to measure operational efficiency, value from automation and capability enablement. This can also be amplified by continuous changes of business strategy, requirements or even executive sponsorship before benefits are officially realized. This reinforces the need for an all-encompassing big picture approach and ongoing “business change” and stakeholder management—continuously (re-)aligning top-down and bottom-up expectations. Creating the illusion that PLM will transform everything is can be equally dangerous as reality misses on surreal expectations.

Myth 3: Any Organization can simply "buy" PLM, and use it Out-of-the-Box

Most organizations will not look at simply buying a tool or platform to enable PLM related processes. They expect operational efficiency and implementation “best practices” to help them become more effective and competitive: i.e. “do more with the same” resources and be able to scale their activities while reducing cost. Whereas buying apps or tools is quite straightforward via a license model, making good use of an integrated and streamlined working practice across multiple functions is not always obvious. This can be quite complex based on product or process requirements which can be contradictory at different product maturity stages.

In addition, most enterprise digital platforms now cover a lot more than what former “PLM system” used to deliver. Buying a PLM platform is not a simple decision as it typically involves medium to long term commitment to a vendor, and short/medium term engagement with a strategic implementation partner.

Even if they cover more scope, no single platform or tool can stand “out-of-the-box” (OoB) on its own unless only covering a very narrow self-contained scope; implementation complexity rises with product, data and business model complexity, in addition to multiple legacy ERP and MES integration points and legacy data migration requirements. PLM processes need to be contextualized for any “brownfield” organization wanting to get value from it. Similarly, for start-up or “greenfield” organizations, PLM processes need to be tailored to the business maturity—aka customized in a controlled manner, balancing short- and long-term requirements.

Myth 4: PLM “Best Practices” are Universal

Transferring “best practices” from one organization to another is clearly a myth: only the learning can be shared but there is no guarantee that what worked elsewhere will be as effective somewhere else. Some practices will certainly work in small pockets of scope. However, working practices cannot be fully reproduced out of context, even if the system implementation can be copied. Every organization is unique, has its own legacy of challenges, from a data and process perspective. This concerns both start-ups and “brown field” companies which will be able to implement business change at different pace and levels of success based on their expertise and context.

It is not uncommon to hear about “industry-ready” bundles. These often come at the price of process and integration compromises. They combine with the uncomfortable truth that there is no such thing as one-size-fits-all solution. Whatever capability is in the box, a number of core principles underpin any PLM initiative; it is important to look at strategic alignment, short- and medium-term compatibility when selecting any technology vendor (*and their implementation partner*) to minimize deviation from the OoB standards where possible.

Failing to understand these principles implies lengthy PLM implementation and future maintenance nightmares.

Myth 5: PLM Platform Configuration is Good, whereas Customization is Bad

There are probably 10 or more definitions of system configuration and customization for enterprise digital platforms. So, let’s not go there. To be effective, PLM processes must be tailored to the enterprise what will be using them.

Every digital solution requires some sort of tailoring design, including adaptation and integration with the rest of the enterprise—because they are “contextual” and (for the most) do not consist on simple transactional processes. No single platform or solution will cover it all, neither will such platform be used in complete isolation of any other solution, especially for advanced product engineering and manufacturing. There are multiple schools of thought in terms of adopting PLM processes:

- The minimalist approach which involves keeping the level of adaptation to the strict minimum and adapting working practices and methods to what's in the box (as close as possible to the out of the box, or OOTB, process)—with sale slogans such as “changing the process, not the tool.” This rarely applies to everything and is therefore either temporary, of limited scope, or not intended to scale (see Myth 4 above).
- The tailored approach which implies adapting the toolset to meet specific custom process or data requirements; most platforms offer capabilities to personalize the solutions to the business model that they serve... this can actually be perceived as “thinking-outside-of-the-box”.

Some vendors openly claim that their platform cannot be implemented without customization, whereas others claim that it can be used OOTB—the usual dilemma. Some capabilities are likely to be more customized than other, and differently to for every organization. Like everything with PLM, there are no common ways to assess levels of customization and integration (or lack of). It remains important to consider future maintenance of such alterations and continuously assess potential cost and knowledge implications.

Myth 6: Cloud-based SaaS Platforms will solve all PLM Implementation Challenges

Infrastructure and other IT related benefits are becoming part of the new normal; it is fair to recognized that PLM is finally “catching up with the cloud” which to-date has mainly been the land of ERP, CRM, etc.

PLM in the cloud can mean many things in terms of infrastructure hosting. This includes, although not limited to, two types of cloud-based software-as-a-service offerings:

1. **PLM SaaS for single tenant:** pay for what you use, when you use it—embedding your own processes and enterprise hybrid-cloud integration.
2. **PLM SaaS with multi-tenants:** with added economies of scale when sharing a commonly administrated platform across independent organizations.

SaaS platforms bring the promise to reducing “entry barriers” to PLM adoption, focusing on low cost and future scalability and flexibility. Multi-tenant platforms also have the potential to change the landscape of small and medium enterprises. Start-up and other large enterprises typically require tailored solutions as they aim to scale and optimize operations as they grow or diversify.

Clearly the need for PLM-on-premise is to be challenged; SaaS platforms offer new ways to experiment early and swiftly validate process designs before (or even without) making long-term costly commitments. The need for business change, integration at both process and system level, education, etc. still remain. Other considerations of data migration, integration bottlenecks, and questions of future “transferability” are also to be explored.

3.3 INCOSE Colorado USA Front Range Chapter Sponsors Virtual Presentation Concerning Resilient Systems

Editor's Note: This article is based on notes taken by PPI SyEN Editor Ralph Young concerning a presentation made for INCOSE's America West Chapter by John S. Brtis, P.E., PMP, CSEP

Co-chair, INCOSE's Resilient Systems Working Group

Lead, Loss-Driven Systems Engineering (LDSE) INCOSE Initiative

"A capability delivered without resilience is one not delivered."

Systems must be resilient to adversity. There is challenge associated with delivering systems that are resilient.

Some historical references:

- Latin "resilio" – to bounce
- 2006: "resilience engineering" – Hollnagel
- 2008: "resilience of systems" – Haimes
- 2010: USA White House Directive - National Space Policy
- One study claims to have identified 119 unique definitions of resiliency in the U.S. Department of Defense. Two particularly useful definitions:
 - Resilience is the ability of a system to continue providing required capability in the face of system failures, environmental changes, or adversarial actions. [[U.S. Air Force Space Command](#) (AFSPC), "Resiliency and Disaggregated Space Architectures" (2013)].
 - Resilience is the ability to provide required capability in the face of adversity. [INCOSE Systems Engineering Body of Knowledge (SEBoK) and [U.S. National Institute of Standards and Technology](#) (NIST) 800-160].
- In this presentation, we are talking about the resilience of engineered systems.
- Fundamental objectives of resilience are:
 - Avoid,
 - Withstand, and
 - Recover.

There are many metrics associated with resilience of systems, including:

- Expected availability of Required Capability
- Cost-benefit
- Fault tolerance
- Mean Time Between Failure

- Operational availability
- Operational reliability
- Maintainability
- Best practices
- And many others
- Resilience does not necessarily mean “returning to the original state”.
- INCOSE has done an assessment and is currently addressing several processes that need to be augmented to provide additions that address resilience. Work is underway to provide additions to the INCOSE SEH to properly address resilience.
- Another current INCOSE activity is to develop formal patterns for Resilience Requirements, for example, develop a pattern based on SysML/DoDAF extensions – the plan is to provide patterns in the next update of the SEH.
- INCOSE’s Loss-Driven Systems Engineering (LDSE) Initiative is addressed in the October 2020 issue of INCOSE INSIGHT. Loss-Driven Specialty Areas include:
 - Availability
 - Environmental Impact
 - Maintainability
 - Resilience
 - Risk Management
 - Survivability
 - System Safety
 - System Security
 - Quality

[More Information](#)

4. SYSTEMS ENGINEERING NEWS

4.1 A Model Program for Development of Young Systems Engineers is Available for Replication!

by

Dr. Ralph Young

Editor, PPI SyEN

Perhaps the most critical need within the Systems Engineering community is the urgent requirement to provide leaders. The most compelling aspect of this need is the professional development of young systems engineers. Thanks to the vision, expertise, and passion of one of INCOSE's own, a superb Program for the development of young systems engineers, already five years in place, is available for replication and implementation by INCOSE chapters and other systems engineering organizations.

That Program, the *Greatest Young Systems Engineers of the Year Challenge*, is described in detail in Feature Article 2.2 in this issue of the *Project Performance International Systems Engineering Newsletter* (PPI SyEN). The article provides information about the Program, feedback from several of the participants, and guidance to create similar programs. *It is our fervent hope that leaders throughout the world will leverage this golden opportunity to replicate the Program throughout the World!*

The Team that developed this Program was led by Ad Sparrius of the Graduate School of Technology Management, University of Pretoria, an INCOSE Fellow.

The objective of the Greatest Young Systems Engineers of the Year Challenge (GYSEOY) (pronounced as "g-eye-soy") is to foster a deep interest in system engineering in young graduate engineers by means of a challenge to solve a business problem using advanced system engineering principles, including model-based techniques. The underlying rationale is to foster system engineering leadership for the next generation, by focusing on the development of technical skills, personal skills, and professionalism.

The most important result from participating in the GYSEOY challenge is not the additional systems engineering skills mastered, although they are substantial. The most important result is strengthened self-confidence of aspiring leaders.

Another Type of Challenge:

To Individuals, INCOSE Chapters, and other Systems Engineering Organizations

Hopefully you are convinced of the value of this Program and also of the merit of replicating it and implementing it in many locations around the World. It is hoped that individuals, INCOSE chapters, and other systems engineering organizations will form teams of committed leaders to invest themselves in young systems engineers and the systems engineering profession. We've taken a special effort to

provide details about the Program – how it came to life, the process used to provide the program, feedback from participants, etc. in order to enable others to go forth and make things happen.

The INCOSE Program Management/Systems Integration (PM/SE) Working Group

Another article in this issue is “The INCOSE Program Management and Systems Engineering (PM/SE) Integration Working Group (WG): Bridging the Gap”, by John Lomax, Jean-Claude Roussel, Tina Srivastava, Rachel Mouring, Mark Kaufman, and Ralph Young. The PM-SE Integration WG was set up with the intention of identifying and promoting opportunities associated with effective integration of the Systems Engineering and Project/Program Management disciplines. This working group aims to facilitate collaboration between SE and PM communities, produce useful deliverables that support effective integration, and provide thought leadership on integration challenges. The WG is co-chaired by Tina Srivastava (MIT), Jean-Claude Roussel (Airbus, France - Retired) and John Lomax (Airbus Defense & Space, UK).

The members of the PM/SE WG are interested to learn of initiatives being taken to further strengthen and improve PM/SE integration. Subscribers of PPI SyEN are encouraged to provide information and news to the any of the three co-chairs of the PM/SE Integration Working Group. Their email addresses are provided at the beginning of their article.

4.2 Message from the INCOSE President: INCOSE’s International Workshop (IW) 2021 will be a Virtual Event

Due to the uncertainties related to COVID-19-driven travel restrictions expected over the next four months, we have made the decision to move forward with the International Workshop, IW 2021 as a fully virtual event. This was a particularly hard decision to make as IW 2021 was our first time planning to conduct the workshop outside of the USA, in the great location of Seville, Spain. However, we need to continue to keep adapting our operations under the constraints imposed by the pandemic to provide our members with high quality INCOSE services, benefits, and opportunities. This is just another example of our resilience and evolution as a technical organization. The health and safety of our members, contractors and alliances continues to remain our highest priority.

You will receive details over the coming weeks on how the International Workshop will be conducted through the various INCOSE news channels and on our IW website. Within the IW timeframe we are still planning to hold an opening plenary, carry out the installation of the 2021 Board of Directors, run town halls as appropriate, and conduct a virtual “Working Group Market Place” at the conclusion. Yet now, under a virtual program, we have the added advantage of scheduling workshops for many groups and communities without the restriction of needing to allocate physical rooms. With this greater flexibility a group may be able to leverage the virtual format for IW 2021 by having more than one meeting, or could schedule a progressive workshop across various time zones, to name just a few possibilities.

To all our members and contributors to IW 2021, we understand the workshop is an important event to plan and progress the objectives of working groups and communities. Moving to a virtual program will

continue to provide a strong collaborative environment to support the workshop, including greater outreach to participants. By building on the success of IS 2020 and our investment in its virtual platform, we are confident that IW 2021 will also deliver the value our members expect from the International Workshop.

If you have any general questions relating to this message, please do not hesitate to contact us at helpdesk@incose.org.

Lastly, I would like to again thank all our INCOSE volunteers and members who continue to contribute to the success and wellbeing of INCOSE. I wish the best to you and your families.

And, I look forward to catching up with you at IW 2021, the virtual International Workshop.

Keep well, keep safe.



Kerry Lunney

INCOSE President

4.3 AIAA's Journal of Aerospace Information Systems Announces a Call for Papers on Systems Engineering's Top Space Challenges

Systems engineering has permeated the engineering ethos in the same way information systems previously redefined the very approach to aerospace science. As Systems Engineering has grown, numerous core challenges have become paramount in the space domain. This special issue in the *Journal of Aerospace Information Systems* addresses four pressing, space-related systems engineering challenges that demand cross-disciplinary solutions. This special issue's purpose is to collect and publish a selection of high-quality journal articles that describe and advance the latest systems engineering approaches in the aerospace information-related fields related to these systems engineering space challenges. This special issue seeks to elicit submissions from researchers and practitioners from a wide range of backgrounds who are interested in the enigmatic questions posed by AIAA's Systems Engineering Technical Committee.

Systems Engineering Space Challenges:

1. Should mass still be a driver for most space missions?
2. Are existing required design margins in handbooks and standards adequate for modern space systems?

3. Should a systems engineering glossary/definitions/ontology be enforced to support the development of a space system?
4. Do space engineers need to learn Model-Based Systems Engineering to successfully adopt Digital Engineering?

Authors need not respond directly to one of the Systems Engineering challenges if the manuscript relates to the challenges or is in the spirit of solving complex Systems Engineering problems.

Deadline for Submitted Manuscripts: 1 March 2021

Anticipated Publication Date: 1 December 2021

Guest Editors:

Jeff Newcamp (jnewcamp@gmail.com)

Alejandro Salado (asalado@vt.edu)

Alessandro Golkar (alessandro.golkar@gmail.com)

Mike Miller (mikez.miller@gmail.com)

Wenjong Gu (wq262@cornell.edu)

[More Information](#)

4.4 INCOSE 2020 Election Results

On the 29th of September 2020, the INCOSE Nominations & Election Committee is pleased to announce the results of the 2020 election. These individuals will join the INCOSE Board of Directors on Friday, 29 January 2021, when they are installed during the opening plenary of the Virtual 2021 International Workshop.

Position (Term of Office)

- **Asia-Oceania (Sector III) Director (3 years):** Serge Landry
- **Chief Information Officer (3 years):** Barclay Brown
- **Director for Outreach (3 years):** - Julia Taylor
- **Secretary (2 Years):** Kyle Lewis

Congratulations to our new officers and directors!

4.5 Don't Just Lead Your People Through Trauma. Help Them Grow

by
Jamil Zaki

Provided in the September 11th 2020 Issue of *Harvard Business Review's*

Leadership and Managing People Series



The last several months have stacked painful experiences on top of each other: a global pandemic, economic collapse, and new reminders of perennial racial injustice and police violence. In July, rates of depression and anxiety in the USA were more than triple those of early 2019. The simple question, “How are you?” has turned into an emotional minefield.

Workplaces are saturated with trauma, too, and leaders are agonizing over how to keep their teams healthy as everyone works remotely and juggles any number of stressors. The science of trauma offers some insight about this moment, and some surprising hope: Instead of asking how we will recover from these painful times, we should ask how we will be changed by them. In many cases, we have an opportunity to change for the better.

[Read This Article](#)

4.6 New Research Program for Smart City Solutions in India

The National University of Singapore (NUS) and ST Engineering are collaborating on a S\$9 million, multi-year advanced digital technologies research program to further their common goals of building a people-centric, smart future for Singapore and beyond.

Research efforts of this new program will focus on technologies related to Smart City as well as Smart Maintenance, Repairs, and Overhaul (MRO), covering five areas: resource optimization and scheduling; prescriptive analytics; decision and sense-making; reasoning engine and machine learning; as well as digital twin. These research areas support ST Engineering's focus on developing differentiated and people-centric, smart city solutions that meet the present and future needs of cities around the world. The interdisciplinary research areas are also aligned with NUS' endeavors as a driving force behind smart city innovations, leveraging its deep expertise that spans multiple domains and faculties.

4.7 Presentation: Model-Based Systems Engineering, a Critical Enabler for Digital Transformation (Part 1 of 3)

Trident Isofol Launched a Webinar on Model-Based Systems Engineering on 8 November 2020. The webinar covers:

1. Model Based Systems Engineering: What, Why and How?
2. Pillars of MBSE
3. Overview of SysML and the need for common semantics
4. MBSE methods and best practices
5. Tenets that enable digital engineering

View the 60-minute webinar here:

https://www.youtube.com/watch?v=G_EBJGOkAtc&feature=share

4.8 Job Opportunity: Senior Director of Systems Engineering at Smiths Medical

Job Description

Smiths Medical is currently hiring a **Senior Director, Systems Engineering** in Minneapolis, MN.

The **Senior Director, Systems Engineering** will direct business critical Systems Engineering R&D function through functional management levels where the scope and complexity of responsibilities require the integration of multiple disciplines and departments. This role will set strategy and direction for the Systems Engineering R&D function in alignment with strategic plans established by top R&D management and CTO. Decisions and recommendations made in this role will have a significant long-term impact on the corporation and affect the financial or employee position of the company.

Diversity & Inclusion

We believe that different perspectives and backgrounds are what make a company flourish. All qualified applicants will receive equal consideration for employment regardless of race, color, religion, sex, sexual orientation, gender identity, national origin, economic status, disability, age, or any other legally protected characteristics. We are proud to be an inclusive company with values grounded in equality and ethics, where we celebrate, support, and embrace diversity.

The Individual

- Scientific University Degree (BS degree) required.
- 4 year degree in engineering or similar.

- 15+ years of experience or 13+ years with an advanced degree.
- Applies extensive technical expertise in the engineering field and has full knowledge of other related disciplines (GPM, RAQA, Sales and Ops).

Apply via [LinkedIn](#)

5. FEATURED ORGANIZATIONS

5.1 German Aerospace Center (DLR) Institute of Systems Engineering for Future Mobility is Researching New Methods for Traffic Systems

The DLR Institute of Systems Engineering for Future Mobility located in Oldenburg, Germany is researching methods for developing and assessment of automated and autonomous traffic systems.

The focus is on the development of new efficient methods and tools of systems engineering for the proof of functionality (verification) and practicability (validation) as well as the further development of trustworthy systems. This is a major challenge for future generations of automated and autonomous transport systems.

In its three departments - Systems Theory and Design, System Evolution and Operation, and Application and Evaluation - address central questions concerning the technical trustworthiness of integrated control systems up to complete transport systems in the fields of automotive, marine, and railway.

[More Information](#)

5.2 American Society for Engineering Management



The American Society for Engineering Management (ASEM) is an international professional society that is focused on promoting and advancing the field of Engineering Management. The subject of Engineering Management is concerned with the management of people and projects in a technological or engineering systems context.

ASEM is responsible for a number of technical publications associated with the field of Engineering Management. This includes an academic journal (Engineering Management Journal) and practitioner

focused publication (Practice Periodical) as well as the Guide to the Engineering Management Body of Knowledge (EMBoK) and Engineering Management Handbook.

Source: [Wikipedia](#)

6. NEWS ON SOFTWARE TOOLS SUPPORTING SYSTEMS ENGINEERING

6.1 Vitech Corporation's GENESYS® and the INCOSE South Africa Graduate Young Systems Engineer of the Year (GYSEOY) 2020 Competition

https://www.vitechcorp.com/?page_id=32354

Those who attended the INCOSE 2020 International Symposium (IS) that took place online in July earlier this year may remember the paper delivered by Ad Sparrius concerning the Greatest Young Systems Engineers of the Year (GYSEOY) Challenge. This challenge encourages employers to sign up their young graduate engineers in teams to compete in the execution of a system conceptual design challenge as defined by the organizing committee. This is the fifth year of the challenge; it has always had a Model-Based Systems Engineering (MBSE) orientation. Teams are required to analyze a scenario description, formulate a problem statement, capture requirements, and produce two or more conceptual designs that can be traded off against each other to result in a final recommended solution concept. The teams capture all of the above work activities in an MBSE tool prescribed for use by the organizing committee. At the beginning of the GYSEOY challenge, Vitech Corporation, through their local representative Letter27, generously offered all the participants student licenses as part of their [CORE University Program](#). This year the challenge upgraded participants' student licenses to the newer Vitech GENESYS MBSE tool.

For someone like me who has used CORE extensively in the past in academic work as well as part of the GYSEOY challenge assessment, the transition to GENESYS was virtually seamless. Once you learn the new user interface, it is business as usual.

To quote from the Vitech website: “**GENESYS** combines a proven, model-centric approach to systems engineering with an enterprise-ready architecture, giving you the ability to deliver model-based systems engineering (MBSE) seamlessly and consistently across your project team. GENESYS takes the guesswork out of implementation and delivers on context-driven modeling for complex systems engineering problems.

The collaborative nature of GENESYS provides tools for all phases of model-based systems engineering to:

- Collect, import, and manage requirements.
- Examine use cases, define system behavior, and perform functional analysis.

- Create and modify architectures.
- Test models with on-board simulation enabling virtual prototyping from the earliest stages of system definition.
- Use pre-populated templates to generate industry standard documentation or create your own reports with drag-drop simplicity.

The [GENESYS Web Page](#) offers extensive additional information concerning the tool capabilities. As a past user of CORE and a new adopter of GENESYS, I can highly recommend this tool for all your MBSE needs.

Alwyn Smit

Pr Eng CSEP

PPI Principal Consultant and Course Presenter

6.2 Eclipse Papyrus™ Modelling Environment

<https://www.eclipse.org/papyrus/>

Eclipse Papyrus is an open source Model-Based Engineering tool for industrial and academic applications. Eclipse Papyrus has become a PolarSys (the Industrial Working Group of Eclipse) Solution.

In order to federate the industrial needs and efforts on MBE, a Papyrus Industry Consortium has been setup.

Eclipse Papyrus provides for the following technologies:

UML 2.5.0

Eclipse Papyrus is a graphical editing tool for [UML 2](#) as defined by [OMG](#). It targets to implement 100% of the OMG specification!

Eclipse Papyrus provides editors for all the following UML diagrams:

- | | |
|-------------------------|--------------------------------|
| • Class Diagram | • Object Diagram |
| • Package Diagram | • Composite Structure Diagram |
| • Component Diagram | • Deployment Diagram |
| • Profile Diagram | • Use case Diagram |
| • Activity Diagram | • State machine Diagram |
| • Communication Diagram | • Sequence Diagram |
| • Timing Diagram | • Interaction overview Diagram |

SysML 1.1 and 1.4

Eclipse Papyrus also provides complete support to [SysML](#) in order to enable model-based system engineering. Specific tabular and graphical editors required for SysML are also provided:

- Block Definition Diagram
- Internal Block Diagram
- Requirement Diagram
- Parametric Diagram
- Requirement table
- Allocation table

Model execution

Thanks to [Moka](#), Eclipse Papyrus can execute models using a rich and extensible animation and simulation framework. Also, as graphical modelling is not always the best way for specifying the behavior of executable models, Eclipse Papyrus provides textual notation edition with syntax highlight, completion and content assist. It is also a customizable feature of Eclipse Papyrus.

Fully customizable environment

All the modelling features of Eclipse Papyrus are designed to be customizable and to maximize reuse. You can adapt the standard configuration for a specific domain, notation, modelling practice or use the powerful customization mechanisms of Eclipse Papyrus to adapt the modelling environment to suit your needs. With many configurations in Eclipse Papyrus being model-based, the customization can be done live.

- Define your own graphical, textual or tabular notation.
- Filter existing palettes or define your own ones with a model-based configuration.
- Define dedicated properties views to present just the characteristics that are important to you.
- Read your model with dedicated model explorer structuring and rendering.
- Reuse standard languages or define your own modelling language thanks to the UML profile editor.

Eclipse Papyrus relatives

Many technologies complement, extend or use Papyrus. Following are key examples:

- [Papyrus RT](#) for Real-time Systems modelling.
- [Papyrus for Robotics](#): a Papyrus-based modelling environment dedicated to robotics.
- [Papyrus PolarSys Solution](#): Eclipse Papyrus packaged as a PolarSys Solution.
- [Eclipse UML Profiles Repository](#).
- [Papyrus Tools](#): a collection of tools that complement Eclipse Papyrus.
- [Eclipse Safety Framework](#): a set of tools for integrating safety techniques within a model driven engineering process.

6.3 Maplesoft™ Announces the Release of MapleMBSE 2020.2

MapleMBSE 2019.0 is an Excel-based tool that enables companies to employ a model-based systems engineering (MBSE) approach to their design projects without requiring every engineer on the project to be an expert in complex MBSE tools.

Maplesoft™ announced a new release of MapleMBSE, MapleMBSE 2020.2, that now supports the MBSE platform Capella, in addition to improved performance and usability.

6.4 Maplesoft™ Announces Release of MapleSim™

In July 2020, Maplesoft™ announced the release of MapleSim™ Insight, a new software product from Maplesoft that gives machine builders simulation-based debugging and 3-D visualization capabilities that directly connect to their automation tools. As a result, engineers can perform simulation-based testing of their controller easily and efficiently.

Read the full article at: <https://news.thomasnet.com/fullstory/new-software-connects-automation-tool-and-displays-visual-results-40038131>

6.5 Gaphor UML/SysML Modeling

<https://gaphor.org/#about>

Gaphor is a UML and SysML modeling application written in Python. It is claimed to be easy to use while still being powerful. Gaphor implements a fully-compliant UML 2 data model, so it is much more than a picture drawing tool. You can use Gaphor to quickly visualize different aspects of a system as well as create complete, highly complex models.

Some of the features of Gaphor are:

Gaphor works on Windows, MacOS, and Linux.

- It is 100% open source, available under an Apache 2 license. Gaphor provides for both the casual modeler documenting a project or a Model Driven Development expert.
- Gaphor is extensible by plugging in a code generator or exporting your diagrams for documentation. You can also create custom extensions and access them through the Graphical User Interface (GUI) or Command Line Interface (CLI).

- Further extensions planned include a Safety and Reliability Profile and the C4 model (Context, Containers, Components, and Code) for visualizing software architectures.
- Customize the diagrams you create with the built-in styling engine.
- The tree view of the tool allows you to navigate all elements of your model quickly.
- Gaphor also includes Dark Mode.

The software is downloadable at <https://gaphor.org/download.html>.

Alwyn Smit

Pr Eng CSEP

PPI Principal Consultant and Course Presenter

7. SYSTEMS ENGINEERING PUBLICATIONS

7.1 IEEE Systems Journal

The IEEE Systems Journal (ISJ) is the technical journal of the IEEE Systems Council, and is published quarterly. ISJ provides a systems-level, focused forum for application-oriented manuscripts that address complex systems and system-of-systems of national and global significance. It intends to encourage and facilitate cooperation and interaction among IEEE Societies with systems-level and systems engineering interest, and to attract non-IEEE contributors and readers from around the globe. Our IEEE Systems Council job is to address issues in new ways that are not solvable in the domains of the existing IEEE or other societies or global organizations. These problems do not fit within traditional hierarchical boundaries. For example, disaster response such as that triggered by hurricanes, tsunamis, or current volcanic eruptions is not solvable by pure engineering solutions. We need to think about changing and enlarging the paradigm to include systems issues.

[More Information](#)

7.2 On the Architecture of Systemology and the Typology of Its Principles

by

David Rousseau



Systems engineering is increasingly challenged by the rising complexity of projects undertaken, resulting in increases in costs, failure rates, and negative unintended consequences. This has resulted in calls for more scientific principles to underpin the methods of systems engineering. In this paper, it is argued that our ability to improve the methods used in systems engineering depends on making the principles of systemology, of which systems engineering is a part, more diverse and more scientific. An architecture for systemology is introduced, which shows how the principles of systemology arise from interdependent processes spanning multiple disciplinary fields, and a typology is introduced, which can be used to classify systems principles and systems methods. This framework, consisting of an architecture and a typology, can be used to survey and classify the principles and methods currently in use in systemology, map vocabularies referring to them, identify key gaps, and expose opportunities for further development. It may, thus, serve as a tool for coordinating collaborative work towards advancing the scope and depth of systemology.

[Access the Article](#) (Figure 11 in this article provides the architecture of systemology)

[Rosseau: A Framework for Understanding Systems Principles and Methods](#)

[Rosseau: A Systematic Framework for Exploring Worldviews and Its Generalization as a Multi-Purpose Inquiry Framework](#)

7.3 INCOSE's Guide for the Application of Systems Engineering in Large Infrastructure Projects

This Guide covers the application of Systems Engineering (SE) practices to Large Infrastructure Projects (LIPs). Such projects include the construction of infrastructure (e.g., highways, railways, electricity generation and distribution, water collection, storage, and distribution, and waste water collection and transfer), and the construction of major industrial plants, such as oil & gas platforms, refineries, mines, smelters, water and wastewater treatment and steel works.

These projects may include a design stage, if this has not been completed prior to going to construction, but the emphasis of this Guide is on how to use SE practices to better perform the construction stage of a project. The focus is on the realization of the designed (or engineered) solution during construction and the transition into service of the resulting built product, and as a consequence, the application of SE practices is concentrated more on the construction process than on the design of the product [1] or on the continuing operation and maintenance stage.

The purpose of the Guide is to reposition traditional systems engineering practices that have been successfully developed and applied in the defense, aerospace, manufacturing and telecommunications industries, into the context of the construction industry and thereby provide professionals engaged on LIPs a convenient and comprehensive access to the relevant parts of the system engineer's toolkit.

This Guide is not an introduction to, or textbook on, systems engineering, and it is assumed that the reader will have either some understanding of good engineering practices or take the time to access the references highlighted throughout the Guide. However, for completeness, Appendix C gives a brief introduction to systems engineering.

To achieve this purpose, the Guide presents the case for applying SE practices to LIPs, and, particularly, to the planning and management of the construction process (Section 2) and then describes a LIP from a systems viewpoint to establish concepts (Section 3) that are then used to explain how the application of SE practices can be beneficially used to execute the construction process (Section 4). The Guide concludes with Section 5 that summarizes the recommendations for applying SE practices to LIPS. The appendices provide additional reading material for those interested in gaining further background and contextual information.

[Download this Guide from the INCOSE Store](#)

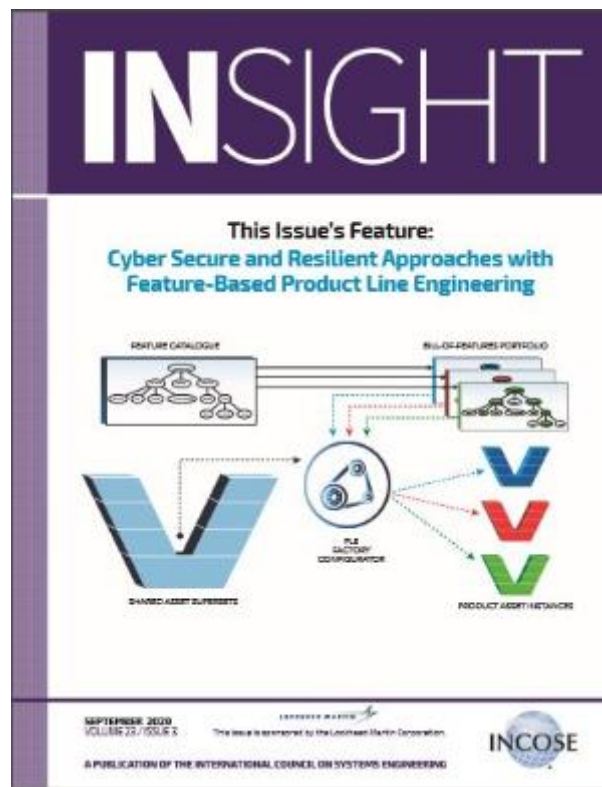
7.4 Brightline's Ten Guiding Principles for Bridging the Gap between Strategy Design and Delivery

Practices can change, business models are disrupted, technology evolves, but principles do not change. They are the soul of strategy design and delivery. These principles are for us both a moral rule and a basic truth. They were developed by a group of experts, practitioners, and researchers.

[Download the Guiding Principles](#)

7.5 INCOSE INSIGHT Practitioners

Magazine Volume 23 Issue 3



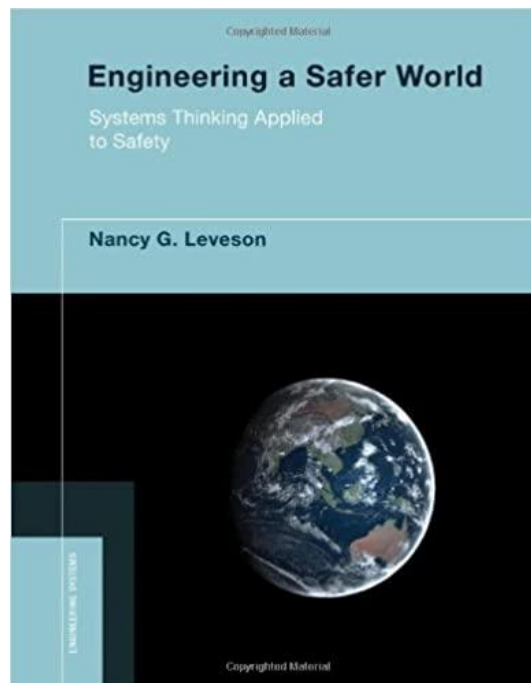
INSIGHT's mission is providing informative articles advancing the systems engineering practice state. The intent is accelerating knowledge dissemination closing the gap between the practice state and the research state as Systems Engineering, the Journal of INCOSE, also Wiley published, captures.

The INSIGHT August 2020 issue's theme is a joint INCOSE Systems Security Engineering (SSE) Working Group and Product Line Engineering (PLE) Working Group project to bring systems security into product line design.

The SSE Working Group's mission is providing systems engineers and systems engineering effective sustainable system functionality means and methods under advanced adversarial attack. Their objectives are instilling systems engineering responsibility for sustainable systems functionality facing intelligent, determined, and highly competent system adversaries; facilitating responsibility assimilation and dispatch; and instigating self-sustaining cross- community involvement between systems engineers, security engineers, and system security standards.

The PLE Working Group's mission is promoting PLE and related systems engineering best practices and to coordinate activities around PLE at the INCOSE level and share results. The working group's objectives are helping our members acquire knowledge comparing to the state-of-art, share concerns, experiences, good practices, and traps to avoid while providing guidelines to set up and evolve organization PLE.

7.6 Engineering a Safer World: Systems Thinking Applied to Safety



From Amazon:

Engineering has experienced a technological revolution, but the basic engineering techniques applied in safety and reliability engineering, created in a simpler, analog world, have changed very little over the years. In this groundbreaking book, Nancy Leveson proposes a new approach to safety -- more suited to today's complex, sociotechnical, software-intensive world -- based on modern systems thinking and systems theory. Revisiting and updating ideas pioneered by 1950s aerospace engineers in their System Safety concept, and testing her new model extensively on real-world examples, Leveson has created a new approach to safety that is more effective, less expensive, and easier to use than current techniques.

Arguing that traditional models of causality are inadequate, Leveson presents a new, extended model of causation (Systems-Theoretic Accident Model and Processes, or STAMP), then shows how the new model can be used to create techniques for system safety engineering, including accident analysis, hazard analysis, system design, safety in operations, and management of safety-critical systems. She applies the new techniques to real-world events including the friendly-fire loss of a U.S. Blackhawk helicopter in the first Gulf War; the Vioxx recall; the U.S. Navy SUBSAFE program; and the bacterial contamination of a public water supply in a Canadian town. Leveson's approach is relevant even beyond safety engineering, offering techniques for "reengineering" any large sociotechnical system to improve safety and manage risk.

ISBN-13: 978-0262016629

ISBN-10: 0262016621

PPI-007070-1

78 of 89

8. EDUCATION AND ACADEMIA

8.1 Master of Science in Industrial and Systems Engineering

National University of Singapore



The Master of Science (Industrial & Systems Engineering) program is designed to prepare individuals for a life-long career addressing critical engineering and managerial decision making in various industrial sectors, such as manufacturing and service sectors. It is a program dedicated to helping students manage problems from a systems perspective or to deal with problems that arise from systems adapting to constant change.

This program will equip students with skills and techniques in solving problems and optimizing processes encountered in the industry, especially in the context of systems. It offers specializations in niche areas such as project management and operations research. Students will have the chance to access latest and real-world industrial examples through faculty members affiliated with research centers and institutes.

[Find out more](#)

8.2 SOUTHERN METHODIST UNIVERSITY (USA) OFFERS FIVE MASTER'S DEGREES IN ENGINEERING

The Lyle School of Engineering at Southern Methodist University (SMU) Department of Engineering Management, Information, & Systems (EMIS) aims to prepare students to meet the demand for advanced analytics expertise with a solid education in engineering and technical management.

Students are offered curriculum content in optimization, stochastics, data science fundamentals and industrial/systems engineering methodologies with applications in business, industry, government and the nonprofit sector.

The EMIS Department offers five master's programs:

- M.S. in Systems Engineering
- M.S. in Engineering Entrepreneurship
- M.S.E.M. in Engineering Management

- M.S.I.E.M. in Information Engineering and Management
- M.S. in Operations Research

[More Information](#)

8.3 ITEA3 Project ‘BUMBLE’ – Blended Modelling for Enhanced Software and Systems Engineering

The [ITEA3 BUMBLE project](#) (funded in Sweden by Vinnova and run in cooperation with 20+ academic and industrial partners from 4 countries, is recruiting a postdoctoral researcher at [Mälardalen University](#), Sweden.

BUMBLE aims at providing an innovative system and software development framework based on blended modelling notations/languages (e.g. textual and graphical). The framework is expected to provide automatic generation and management of fully-fledged blended modelling environments from arbitrary domain-specific modelling languages. Blended modelling environments are expected to greatly boost the development of complex multi-domain systems by enabling seamless textual and graphical collaborative modelling.

The postdoc will focus on a variety of challenges related to blended modelling, including: definition of (meta)models for the description of blended languages, design and implementation of model transformations to ensure seamless synchronization across notations, consistency management and change propagation for collaborative modelling/design.

The postdoc will work with the project team in BUMBLE, but is expected to manage her/his duties independently and to contribute with, e.g., co-supervision to PhD students’ research.

The position is a temporary employment of 2 years.

More info and application page at: <https://bit.ly/postdocMDH>

Contact person:

Federico Ciccozzi

Associate Professor

+4621101736

federico.ciccozzi@mdh.se

9. SOME SYSTEMS ENGINEERING-RELEVANT WEBSITES

NASA Engineering and Safety Center

A link to the NASA Engineering and Safety Centre (NESC) Academy web page on Systems Engineering related Webcasts:

<https://nescacademy.nasa.gov/catalogs/58f99dd43fa0470c8184841b445f16a84d>

What is Product Line Engineering?

Have you ever wished you had a nice definitive definition of exactly what Product Line Engineering is and how it is different from "normal" product engineering?

The [Product Line Engineering website](#) does just that. It defines PLE as "a way to engineer a portfolio of related products in an efficient manner, taking full advantage of the products' similarities while respecting and managing their differences". By the use of the term "engineer", all of the activities involved in planning, producing, delivering, deploying, sustaining, and retiring products is included.

The PLE approach considers a portfolio of related product as a single entity to be managed, as opposed to managing a multitude of separate products individually. This approach enables organizations to achieve order-of-magnitude improvements in engineering cost, time to market, productivity, product line scalability, and product quality.

This website provides an fantastic overview and a collection of informational resources regarding the increasingly important field of Systems and Software Product Line Engineering (PLE).

Brightline

Brightline is a [Project Management Institute \(PMI\)](#) initiative together with leading global organizations dedicated to bridge the gap between strategy design and delivery.

[Website](#)

International Council for Research and Innovation in Building and Construction

The International Council for Research and Innovation in Building and Construction (CIB) was established in 1953 as an Association whose objectives were to stimulate and facilitate international cooperation and information exchange between governmental research institutes in the building and construction sector, with an emphasis on those institutes engaged in technical fields of research.

CIB has since developed into a worldwide network of over 3,000 experts from about 500 member organizations with a research, university, industry or government background, who collectively are active in all aspects of research and innovation for building and construction.

CIB is the acronym of the abbreviated French (former) name: "Conseil International du Bâtiment" (in English: International Council for Building). In 1998 the abbreviation was kept but the full name changed to International Council for Research and Innovation in Building and Construction.

[Website](#)

10. STANDARDS AND GUIDES

10.1 Object Management Group is working to Enable Interoperability with Shareability Terminology

For the past several decades, OMG has been busy creating standards for ontologies for business and industry: APIs for Knowledge Platforms (AP4KP), MOF to RDF Structural Mapping in Support of Linked Open Data (MOF2RDF), Ontology Definition Metamodel (ODM), Distributed Ontology, Model and Specification Language (DOL), FIBO (Financial Industry Business Ontology), FIGI (Financial Instrument Global Identifier), SBRM (Standard Business Report Model) IEF, (Information Exchange Framework), and ontologies for the retail industry and the robotics industry.

10.2 Developing Cyber Resilient Systems: A Systems Security Engineering Approach

A NIST Special Publication 800-160 Volume 2

Abstract

This publication is used in conjunction with ISO/IEC/IEEE 15288:2015, Systems and software engineering—Systems life cycle processes, NIST Special Publication 800-160, Volume 1, Systems Security Engineering—Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, and NIST Special Publication 800-37, Risk Management Framework for Information Systems and Organizations—A System Life Cycle Approach for Security and Privacy. It can be viewed as a handbook for achieving the identified cyber resiliency outcomes based on a systems engineering perspective on system life cycle processes in conjunction with risk management processes, allowing the experience and expertise of the organization to help determine what is correct for its purpose. Organizations can select, adapt, and use some or all of the cyber resiliency constructs (i.e., objectives, techniques, approaches, and design principles) described in this publication and apply the constructs to the technical, operational, and threat environments for which systems need to be engineered. The system life cycle processes and cyber resiliency constructs can be used for new systems, system upgrades, or repurposed systems; can be employed at any stage of the system life cycle; and can take advantage of any system or software development methodology including, for example, waterfall, spiral, or agile. The processes and associated cyber resiliency constructs can also be applied recursively, iteratively, concurrently, sequentially, or in parallel and to any system regardless of its size, complexity, purpose, scope, environment of operation, or special nature. The full extent of the application of the content in this publication is guided and informed by stakeholder protection needs, mission assurance needs, and concerns with cost, schedule, and performance. The tailorable nature of the engineering activities and tasks and the system life cycle processes ensure that systems resulting from the application of the security and cyber resiliency design principles, among others, have the level of trustworthiness deemed sufficient to protect stakeholders from suffering unacceptable losses of their assets and

associated consequences. Trustworthiness is made possible, in part, by the rigorous application of the security and cyber resiliency design principles, constructs, and concepts within a structured set of systems life cycle processes that provides the necessary traceability of requirements, transparency, and evidence to support risk-informed decision-making and trades.

Download the publication [here](#)

10.3 Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems

This publication contains systems security engineering considerations for ISO/IEC/IEEE 15288:2015, Systems and software engineering — System life cycle processes. It provides security-related implementation guidance for the standard and should be used in conjunction with and as a complement to the standard.

With the continuing frequency, intensity, and adverse consequences of cyber-attacks, disruptions, hazards, and other threats to federal, state, and local governments, the military, businesses, and the critical infrastructure, the need for trustworthy secure systems has never been more important to the long-term economic and national security interests of the United States. Engineering-based solutions are essential to managing the growing complexity, dynamicity, and interconnectedness of today's systems, as exemplified by cyber-physical systems and systems-of-systems, including the Internet of Things. This publication addresses the engineering-driven perspective and actions necessary to develop more defensible and survivable systems, inclusive of the machine, physical, and human components that compose the systems and the capabilities and services delivered by those systems. It starts with and builds upon a set of well-established International Standards for systems and software engineering published by the International Organization for Standardization (ISO), the International Electrotechnical Commission (IEC), and the Institute of Electrical and Electronics Engineers (IEEE) and infuses systems security engineering methods, practices, and techniques into those systems and software engineering activities. The objective is to address security issues from a stakeholder protection needs, concerns, and requirements perspective and to use established engineering processes to ensure that such needs, concerns, and requirements are addressed with appropriate fidelity and rigor, early and in a sustainable manner throughout the life cycle of the system.

[PDF Version](#)

11. SOME DEFINITIONS TO CLOSE ON

CATWOE Analysis

CATWOE is an acronym that stands for Customers – Actors – Transformation process – World view – Owners – Environmental constraints. It's a simple checklist to find solutions to problems. It offers surprising solutions and stimulates multiple approaches. The CATWOE Analysis makes it possible to identify problem areas, look at what a company wants to achieve, and which solutions can influence the stakeholders. The analysis uses thought solutions from multiple perspectives.

Source: [Tools Hero](#)

Cognitive Bias

A cognitive bias is a systematic pattern of deviation from norm or rationality in judgment. Individuals create their own "subjective reality" from their perception of the input. An individual's construction of reality, not the [objective](#) input, may dictate their [behavior](#) in the world. Thus, cognitive biases may sometimes lead to perceptual distortion, inaccurate judgment, illogical interpretation, or what is broadly called [irrationality](#).

Source: [Wikipedia](#)

Measurement of Uncertainty

A set of possible states or outcomes where probabilities are assigned to each possible state or outcome – this also includes the application of a probability density function to continuous variables\

Source: [IEEE Neural Network-Based Uncertainty Quantification: A Survey of Methodologies and Applications](#)

Worldview

A worldview or world-view is the fundamental cognitive orientation of an individual or society encompassing the whole of the individuals or society's knowledge and point of view. A worldview can include natural philosophy; fundamental, existential, and normative postulates; or themes, values, emotions, and ethics.

Source: [Wikipedia](#)

12. CONFERENCES AND MEETINGS

For more information on systems engineering related conferences and meetings, please proceed to [our website](#).

12.1 Featured Conference

2020 IEEE Conference on Decision and Control (CDC) – Virtual

Date: 14 December – 18 December 2020

Location: Virtual

The CDC is recognized as the premier scientific and engineering conference dedicated to the advancement of the theory and practice of systems and control. The CDC annually brings together an international community of researchers and practitioners in the field of automatic control to discuss new research results, perspectives on future developments, and innovative applications relevant to decision making, automatic control, and related areas.

The 59th CDC will feature contributed and invited papers, as well as workshops and tutorial sessions.

The IEEE CDC is hosted by the IEEE Control Systems Society (CSS) in cooperation with the Society for Industrial and Applied Mathematics (SIAM), and the Japanese Society for Instrument and Control Engineers (SICE).

[Visit the website](#)

[Register](#)

12.2 Another Notable Conference

27th INTERNATIONAL CONFERENCE ON SYSTEMS ENGINEERING

Date: 14 December – 16 December 2020

Location: Virtual

The **International Conference on Systems Engineering (ICSEng)** is the series of International Conferences, jointly organized on a rotational basis among three institutions:

- University of Nevada Las Vegas, United States - International Conference on Systems Engineering (ICSEng)
- Wrocław University of Technology, Poland - International Conference on Systems Science (ICSS)
- Coventry University, United Kingdom - International Conference on Systems Engineering (ICSE)

Conference covers the Systems Engineering with the focus on applications and was first held in 1974 in Wrocław (Poland) as 1st ICSS.

The Conference will cover the general area of Systems Engineering, with particular emphasis being placed on applications. It is expected to include sessions on the following themes:

- Aero-Space Systems (Avionics, Unmanned Aerial Vehicles, Aviation Control)
- General Control Systems (Control Theory, System Identification and Adaptive Control, Nonlinear Controls, Applications)
- Power Systems (Environmental Systems, Energy Systems, Renewable Energy, Nuclear Energy)
- Uncertain Systems
- Machine Learning and Analytics (Deep Learning)
- Industrial Automation and Robotics
- Information and Communication Systems (Information and Communication Theory, Geographic Information Systems, Global Position Systems, Applications)
- Distributed Computer and Computer Networks Systems (Modeling and Analysis, Distributed and Wireless Systems, Distributed Servers, Parallel and Distributed Systems, Networks)
- Middleware and Cloud Computing
- Security (Networks, IoT, Distributed Systems, Embedded Systems, Cloud Computing)
- Bigdata (Data Mining, Data Warehouses, Sensor Networks, Data Classification, Regression)
- Internet of Things (Smart Cities, Precision Agriculture, Industrial IoT)
- Analog and Digital Hardware Systems (Real-time Systems, Embedded Systems, Hybrid Embedded Systems, Mixed Signal Designs, Multi-media Systems)
- Intelligent Systems (Expert Systems, Artificial Intelligence, Neural Network, Fuzzy, Optimization Techniques, Hybrid Systems, Applications)
- Gaming and Entertainment Systems (Technology, Security, Design, Tools)
- Bio-metrics System (sensors, integration, data analysis, verification techniques)
- Bio-surveillance Systems (heterogeneous data sources, monitoring techniques, signal detection algorithms, privacy protection)
- Systems Engineering Standards, Modeling, Paradigms, Metrics, Testing, Management, Optimization, Simulation, Scheduling, Reliability and Fault Tolerant
- Systems Engineering Education
- Computer Assisted Medical Diagnostic Systems (single and multiple modality medical data analysis, expert systems, prompting systems, databases, performance evaluation)
- Environmental Systems

13. PPI AND CTI NEWS

13.1 The PPI Website Gets a Facelift! – Give Us Your Feedback

With the assistance of a small but dedicated team, PPI is proud to announce that our website is better than ever. Users can expect a more attractive UI, a more intuitive layout and navigation and faster page load times. Check out the website here:

<https://www.ppi-int.com/>

Let us know what you think about the website by providing feedback via website@ppi-int.com

13.2 Team PPI is Growing Despite Challenging Times

2020 has certainly been a year to remember, for better or for worse. We at PPI and CTI, as with many companies, have faced our own set of challenges, resulting in many long days and nights spent by our global team members, hard at work for our clients. Given these trying months, we are particularly proud and blessed to be stronger than ever, so much so that our team continues to grow in both professional and support staff. Latest additions to the team are Billy Lockyer and Geordan Trakman, both of whom have come on board to strengthen our IT operations in a virtual world. Without a doubt, Team PPI will continue to grow in the coming months, as we continue to pursue our mission to make the world a better place through the power of systems engineering. We are particularly looking for additional world-leading professionals in systems and requirements engineering, individuals who have the capability to deliver on the vision of systems engineering as an enabler for a better world. Please contact PPI Managing Director Robert Halligan if you would like to initiate a discussion on a potential match to PPI's needs for additional consulting and training capacity.

An additional opportunity is available, based physically in Brasil during 2021 and beyond, for on-site requirements engineering consulting. Gostaríamos muito de ouvir de você.

13.3 Become a SyEN Contributor?

Many hands make light work! And can produce a better product! We are looking for sub-editors in the SyEN content areas of SE Academia, general SE News, SE Publications, and SE Websites, The role involves researching and contributing each month news in the content area that is consistent with Editor-provided guidelines for that content area. A small (very small!) honorarium is payable.

The Editor of SyEN is also looking for correspondents to report on systems engineering news from particular organizations or sectors. Readers will see the benefits of this initiative in future editions of SyEN.

13.4 PPI SyEN Gets a New Name

The name of PPI SyEN is changing! We are changing the name of PPI SyEN starting with the December 2020 Edition to PPI Systems Engineering Newsjournal, no longer Newsletter, to better reflect the nature and content of our publication. We will continue to bring to you substantial articles on aspects of systems engineering, in combination with news on tools, publications, societies, academic activities, useful websites and general news in the field of systems engineering.

14. PPI AND CTI EVENTS

<https://www.ppi-int.com/ppi-live-online/>

For a full public PPI training course schedule, please visit <https://www.ppi-int.com/course-schedule/>

For a full public CTI Live-Online™ INCOSE SEP Exam Preparation course schedule, please visit <https://certificationtraining-int.com/incose-sep-exam-prep-course/>

To enquire about CTI Live-Online™ INCOSE SEP Exam Preparation Training for your organization, please visit <https://certificationtraining-int.com/on-site-training/>

15. UPCOMING PPI PARTICIPATION IN PROFESSIONAL CONFERENCES

PPI will be participating physically in the following upcoming events. We support the events that we are sponsoring, and look forward to meeting old friends and making new friends at the events at which we will be exhibiting.

The INCOSE International Workshop 2021

Date: 29 – 31 January, 2021

Location: Virtual

The INCOSE International Conference 2021

Date: 17 – 22 July, 2021

Location: Honolulu, USA

Kind regards from the PPI SyEN team:

Robert Halligan, Editor-in-Chief, email: rhalligan@ppi-int.com

Dr. Ralph Young, Editor, email: syen@ppi-int.com

René King, Managing Editor, email: rking@ppi-int.com

Project Performance International

2 Parkgate Drive, Ringwood, Vic 3134 Australia

Tel: +61 3 9876 7345

Tel Brasil: +55 12 9 9780 3490 (BrenoBacci)

Tel UK: +44 20 3608 6754

Tel USA: +1 888 772 5174

Tel China: +86 188 5117 2867 (Victoria Huang)

Web: www.ppi-int.com

Email: contact@ppi-int.com

Copyright 2012-2020 Project Performance (Australia) Pty Ltd, trading as
Project Performance International

Tell us what you think of PPI SyEN. Email us at syen@ppi-int.info.