Welcome to the Webinar! Please note that we have moved to the ZOOM platform. Please join ZOOM audio (Voice over Internet) if you are able to connect. Otherwise, please see the webinar invitation for dial-in phone lines

INCOSE Webinar Series

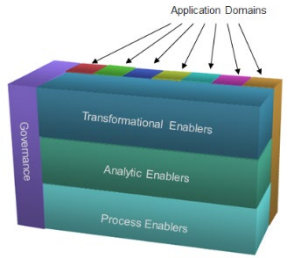
Wednesday 16th August 2023 – Webinar 165

Data-Driven Systems Engineering

Steven Dam

With thanks to our Sponsors for 2023

INCOSE is offering Webinars…



- To provide a forum for experts in the field of Systems Engineering to present information on the “State of the Art”
- To explain how INCOSE works, and how to make the most out of INCOSE membership

INCOSE Systems Engineering Professional PDU Credit

For more information about INCOSE certification renewal, including PDU categories and limits: http://www.incose.org/certification/CertProcess/CertRenew

Claim PDUs

You can claim 1 PDU credit towards your INCOSE Systems Engineering Professional (SEP) renewal by attending this entire webinar.

Eligible Sources To Claim PDU

- Live attendance at the webinar: "Attend non-peer-reviewed Professional Technical Society event."
- Watching a recording of the webinar: "Consume SE-related media, including journal article, book, video, or audio."

Claim PDUs for Other certifications

INCOSE webinars may also apply to the PDU requirements of other organizations, depending on the subject matter.

Choreography

1. Andy Pickard (your host) will introduce the Webinar and the speaker
2. Steve Dam will speak for about 40 to 45 minutes
3. During his talk, participants can write questions using the Zoom Q&A window
4. After Steve completes his talk, he will spend 10 minutes answering questions that Andy selects from those submitted by the audience
5. Andy Pickard will provide information about upcoming Webinars and then end this session
6. This Webinar is being recorded and will be made available on the INCOSE website to members and employees of CAB organizations

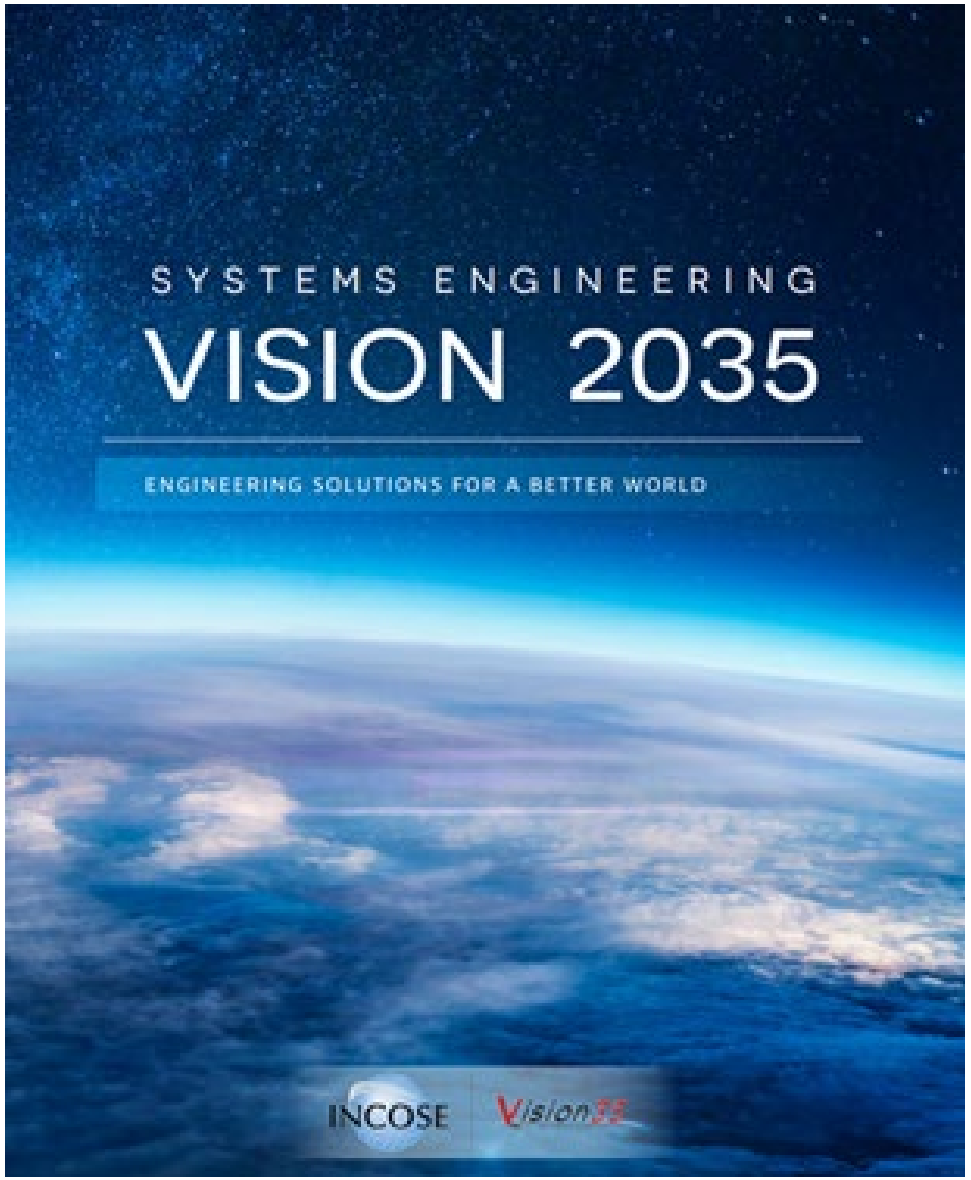
SPEC INNOVATIONS

Data-Driven Systems Engineering

Steven H. Dam, Ph.D., ESEP
President, SPEC Innovations
571-485-7805
steven.dam@specinnovations.com

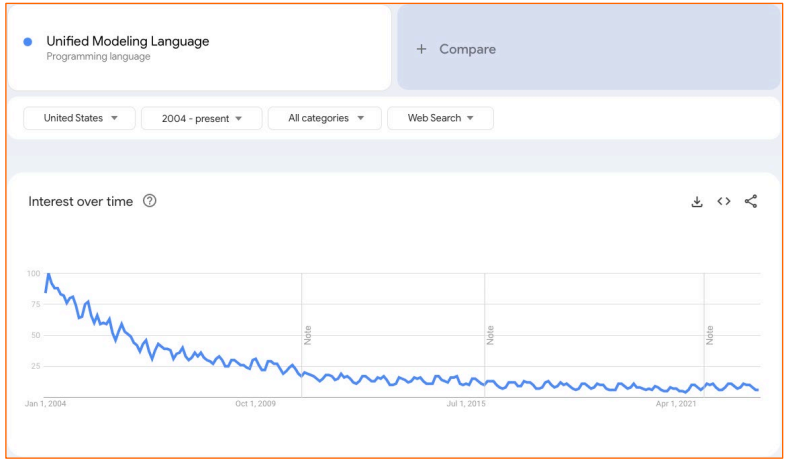
INCOSE Vision 2035

- In this document, they state:
  - “The future of Systems Engineering is Model Based, leveraging next generation modeling, simulation and visualization environments powered by the global digital transformation, to specify, analyze, design, and verify systems.” Vision 2035 p.30
- Someone asked me at the 2022 International Workshop what I thought.
  - I answered, “One thing I know for sure is we won’t be calling it MBSE!”
- But that begged the questions: What next? What’s wrong with MBSE? How do we need systems engineering to evolve beyond MBSE?
- To answer these questions, let’s look at the current state of MBSE.

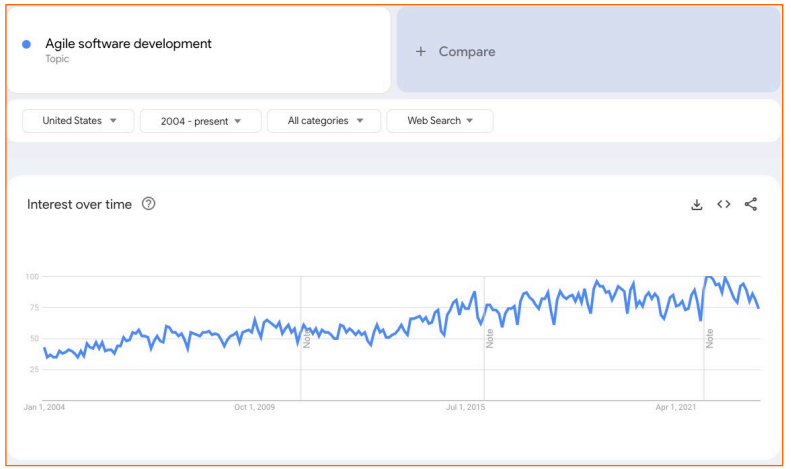


Current State of Systems Engineering

- Since 2007, focus has been on moving from document-based systems engineering to model-based systems engineering.
  - INCOSE's Definition of MBSE: *"Model-based systems engineering (MBSE) is the **formalized** application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases."*
- Many practitioners in our community moved to an object-oriented approach for systems engineering (SysML), which is based on a software engineering technique: Unified Modeling Language (UML).
  - Part of the justification for this was that we needed to establish better communications with software developers.
- However, the software community has moved away from UML since (at least) 2004. They have adopted Agile Software Development techniques that rely on functional requirements provided by systems engineering, not diagrams/models.



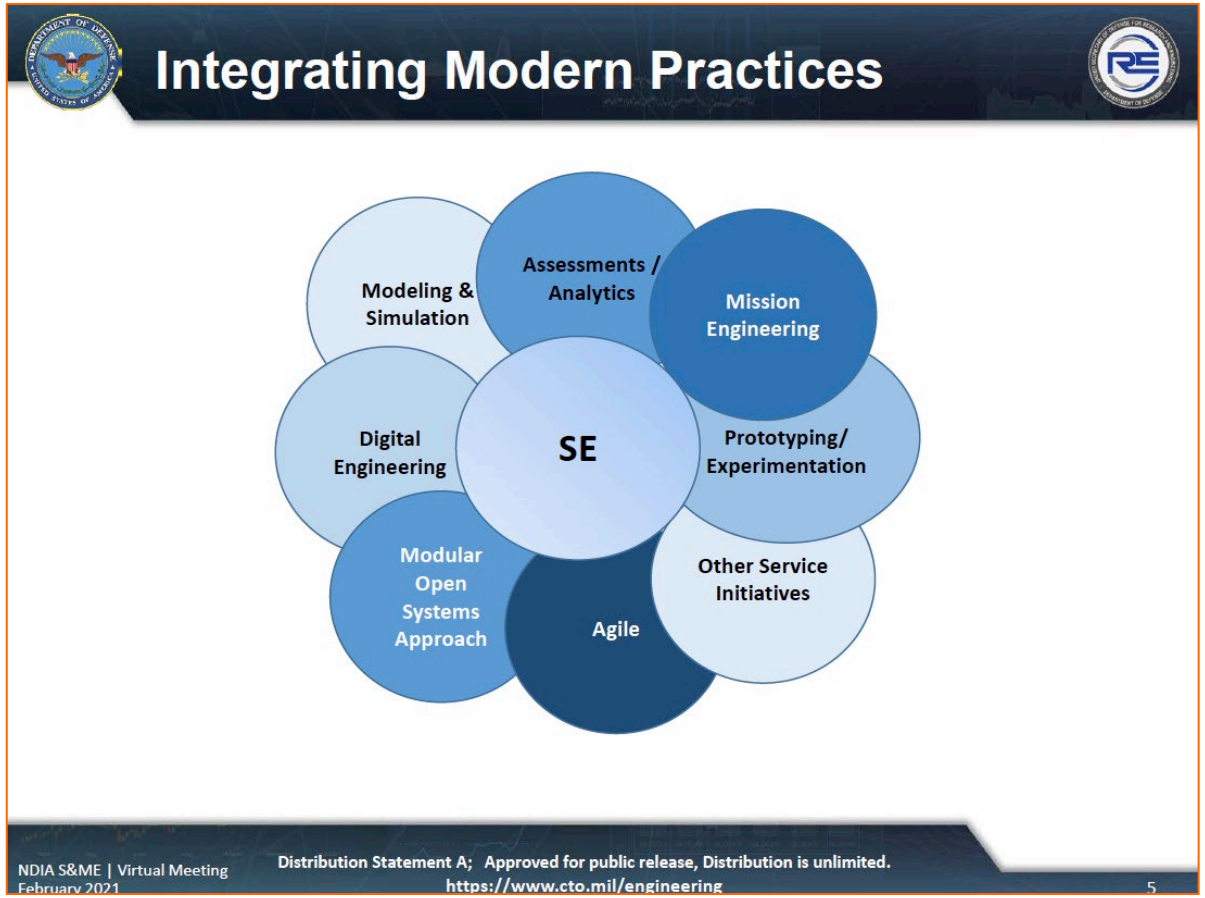
Google Trends. Accessed 8/4/2023



INNOSLATE®

OSD's Initiative: Systems Engineering Modernization

- The Department of Defense (DoD) Office of the Under Secretary of Defense (USD) for Research and Engineering has identified a number of "systems engineering practices" that require "integration."
- The objective is to "modernize systems engineering to support the delivery of capability to meet mission needs."
- Congress also has provided legislative direction in four of these practice areas:
  - Digital Engineering
  - MOSA
  - Mission Engineering
  - Agile Development



Root Cause of this problem: Lack of tool interoperability between disciplines

SE Modernization – SERC 's View

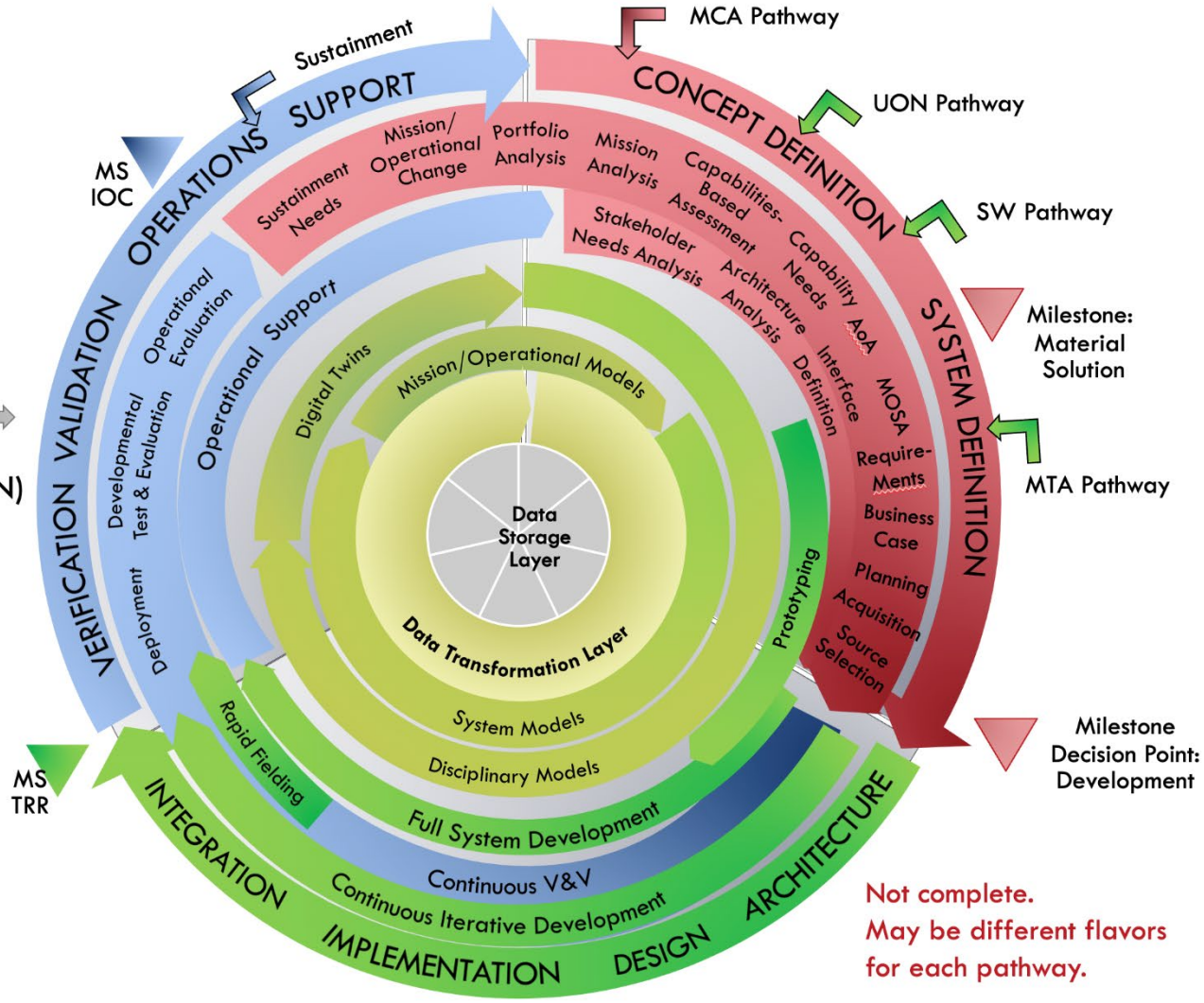
From https://sercuarc.org/systems-engineering-modernization/?utm_source=newsletter&utm_medium=email&utm_content=Systems%20Engineering%20%28SE%29%20Modernization&utm_campaign=SERC%20UPDATES%20JUN%202022 accessed 6/22/2022

"…this view emphasizes the DE transformation using a layered model with data storage and transformation at the core…"

NOTIONAL VIEW: FULL SE MODERNIZATION LIFE CYCLE

- Cyclic nature of modern SE
- Still milestone-based
- SE core principles in every Acq pathway
- Flexible system life cycle entry points:
  Learn-Build-Measure (MCA)
  Build-Measure-Learn (Mid-Tier, SW, UON)
  Measure-Learn-Build (Sustainment)
- Continuous Iterative Development processes (around the circle)
- Continuous Data Management and Transformation processes (at the core)

Learn	Needs analysis & Planning
Build	Implementation
Measure	Test, Evaluation, Support
Data & Models	



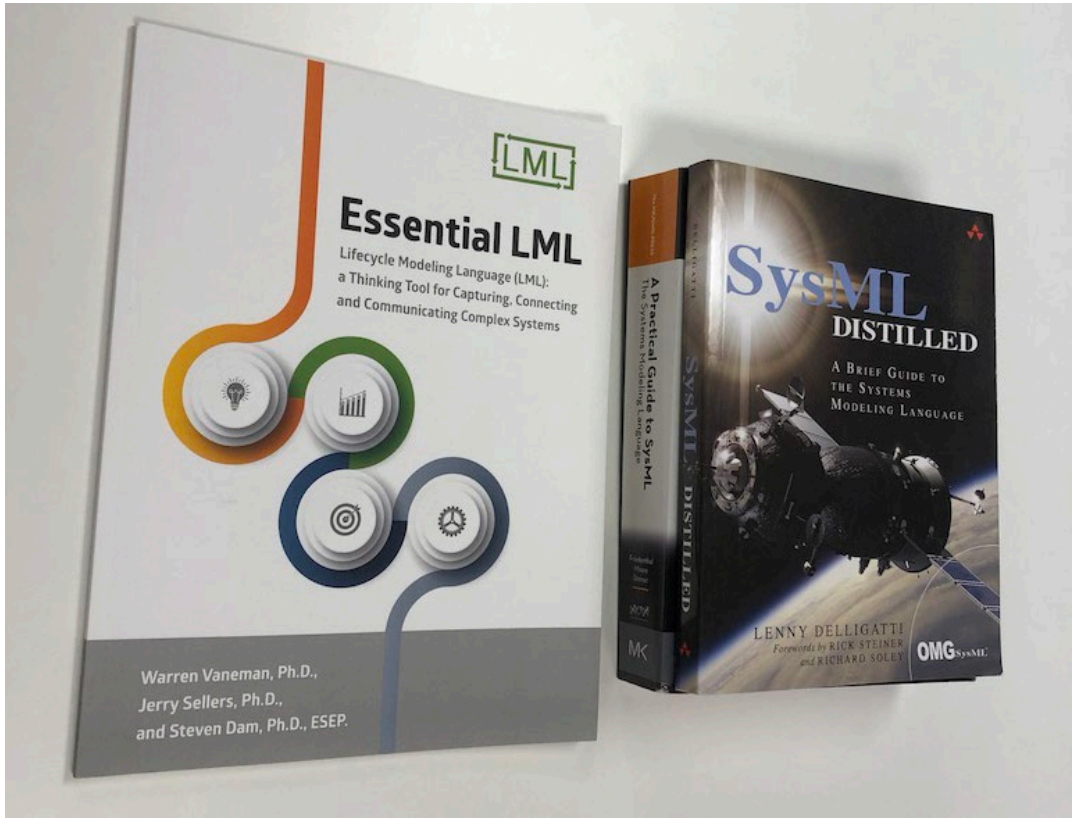
To implement this approach, we need to be data-driven in systems engineering.

What's Data-Driven Systems Engineering (DDSE)?

- To integrate the "modern engineering practices," we need a common language, not just a set of models
- By using a language-driven approach, we can focus on the data and less on the form ("model").
- This data-driven approach needs a new way to think about systems engineering (DDSE).
- The Lifecycle Modeling Organization (LMO) defines DDSE as:
  - *"the formalized development, exchange, correlation, validation and verification, and management of data, derived from multiple sources, processes, and tools to support system design and analysis, throughout all phases of the system lifecycle."*
- This definition refocuses us on the underlying basis for systems engineering.
  - Design engineers get clear, easy-to-understand requirements (data).
  - Decision makers get the information (data) they need to make good decisions.
- Fortunately, a language already exists to meet this need: the Lifecycle Modeling Language (LML).

Lifecycle Modeling Language (LML)

- LML was developed by a group of systems engineers who realized that SysML was not meeting the needs of the systems engineering and program management communities.
- LML is currently managed by the not-for-profit, Lifecycle Modeling Organization (LMO).
- The Steering Committee is led by Dr. Warren Vaneman, USN CAPT (retired) and Professor of Practice at the Naval Postgraduate School (NPS).
- LML is taught in over 200 universities around the world, including MIT, George Mason University, Stevens Institute of Technology, West Point, NPS, and Air Force Academy.
- LML is easy to learn, use, and extend.
- Visit http://www.lifecyclemodeling.org.

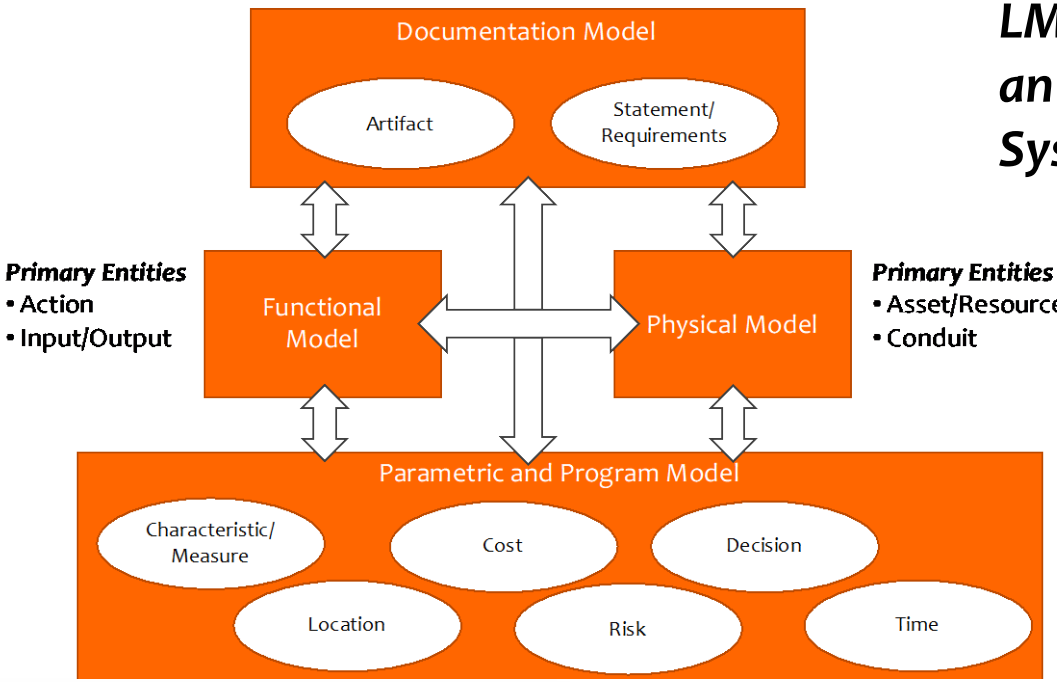


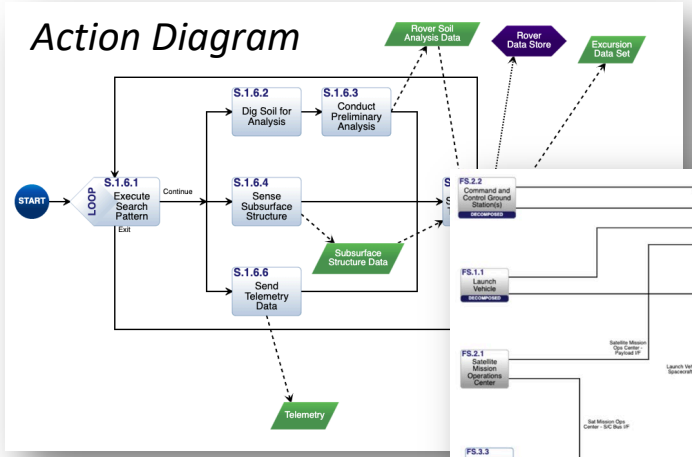
LML has proven to provide a strong ontology for systems engineering and program management and forms the basis for Data-Driven Systems Engineering (DDSE).

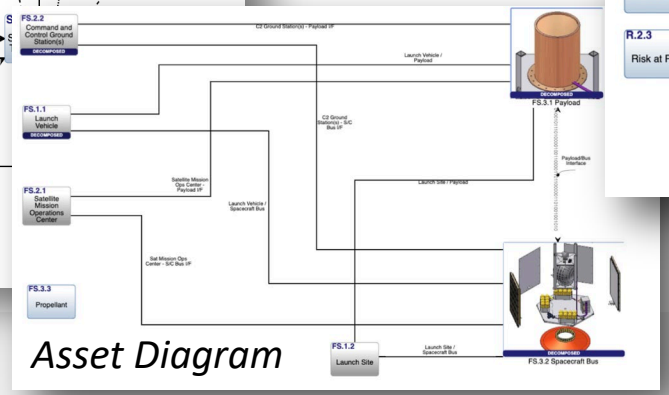
LML Overview

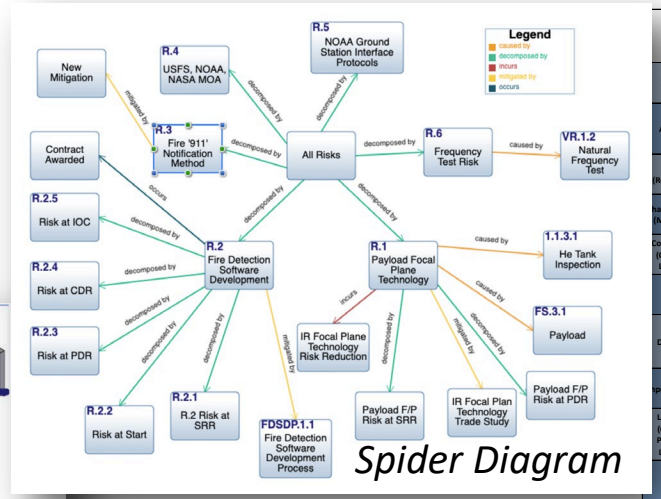
- A simple, easy to use set of classes (nouns)
- Maps to DoDAF, UAF, SysML, BPMN, etc.
- Connected by two-way relationships (verbs)
- Easily extended
- Contains data elements for all the SE Modernization practices

LML provided an ontology for SysML in 2015!









	Action	Artifact	Asset (Resource)	Characteristic (Measure)	Connection (Conduit, Logical)	Cost	Decision	Input/Output	Location (Orbital, Physical, Virtual)	Risk	Statement (Requirement)	Time
Action	decomposed by* related to*	references	(consumes) performed by (produces) (seizes)	specified by	-	incurs	enables results in	generates receives	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Artifact	referenced by	decomposed by* related to*	referenced by	referenced by specified by	defines protocol for referenced by	incurs referenced by	enables referenced by results in	referenced by	located at	causes mitigates referenced by resolves	referenced by (satisfies) source of traced from (verifies)	occurs
Asset (Resource)	(consumed by) performs (produced by) (seized by)	references	decomposed by* orbited by* related to*	specified by	connected by	incurs	enables made responds to results in	-	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Characteristic (Measure)	specifies	references specifies	specifies	decomposed by* related to* specified by*	specifies	incurs specifies	enables results in specifies	specifies	located at specifies	causes mitigates resolves specifies	(satisfies) specifies traced from (verifies)	occurs specifies
Connection (Conduit, Logical)	-	defined protocol by references	connects to	specified by	decomposed by* joined by* related to*	incurs	enables results in	transfers	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Cost	incurred by	incurred by references	incurred by	incurred by specified by	incurred by	decomposed by* related to*	enables incurred by results in	incurred by	located at	causes incurred by mitigates resolves	incurred by (satisfies) traced from (verifies)	occurs
Decision	enabled by result of	enabled by references result of	enabled by made by responded by result of	enabled by result of specified by	enabled by result of	enabled by incurs result of	decomposed by* related to*	enabled by result of	located at	causes enabled by mitigated by result of resolves	alternative enabled by traced from result of	date resolved by decision due occurs
Input/Output	generated by received by	references	-	specified by	transferred by	incurs	enables results in	decomposed by* related to*	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Location (Orbital, Physical, Logical)	locates	locates	locates	locates specified by	locates	locates	locates	locates	decomposed by* related to*	locates mitigates	locates (satisfies) traced from (verifies)	occurs
Risk	caused by mitigated by resolved by	caused by mitigated by references resolved by	caused by mitigated by resolved by	caused by mitigated by resolved by specified by	caused by mitigated by resolved by	caused by incurs mitigated by resolved by	caused by enables mitigated by results in resolved by	caused by mitigated by resolved by	located at mitigated by	caused by* decomposed by* related to* resolved by*	caused by mitigated by resolved by	occurs mitigated by
Statement (Requirement)	(satisfied by) traced to (verified by)	references (satisfied by) sourced by traced to (verified by)	(satisfied by) traced to (verified by)	(satisfied by) specified by traced to (verified by)	(satisfied by) traced to (verified by)	incurs (satisfied by) traced to (verified by)	alternative of enables traced to results in	(satisfied by) traced to (verified by)	located at (satisfied by) traced to (verified by)	causes mitigates resolves	decomposed by* traced to* related to*	occurs (satisfied by) (verified by)
Time	occurred by	occurred by	occurred by	occurred by specified by	occurred by	occurred by	date resolves decided by occurred by	occurred by	occurred by	occurred by mitigates	occurred by (satisfies) (verifies)	decomposed by* related to*

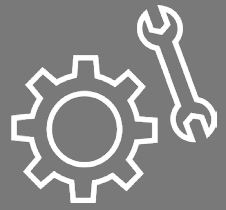
How Do We Implement DDSE?

- Since we need to cover all aspects of systems engineering in a data-driven approach, we need tools to support this activity. This is necessary as we cannot maintain this vast amount of data without using database technologies.
- Those tools should include requirements analysis, functional/object analysis, modeling and simulation, verification and validation, risk analysis, cost analysis, and schedule analysis as a minimum.
- The tools should also support modern computer architectures, with cloud computing being the most common today.
  - Secure clouds are currently available at all levels of security for any Department or Agency.
- The tools also need to have an open architecture to enable easy movement of data between tools, as we cannot expect the world to adopt only one tool!
- So, let's look at tools today.

Today's Systems Engineering Tools – *All based on 20th Century Technology*

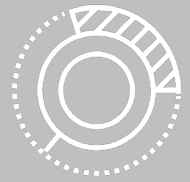
Difficult to Use

- Learning full SysML/UML often takes months
- Dozens of options are present on many views
- Drawings usually must be redrawn (ex. the Sequence, Activity, and BDD)
- Required desktop software installation to modify the models



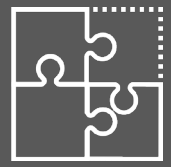
The Models Do Not Execute

- Lack of built-in variance simulation (Monte Carlo Simulation)
- Activity Diagrams do not have to be executable



Limited Lifecycle Management

- Focus is typically only on drawing models
- Most organizations use additional tools for requirements management, test management, and Monte Carlo simulation



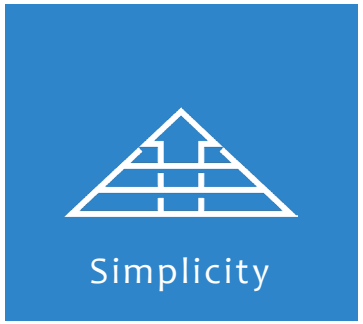
Lack of Collaboration

- Model conflicts during commits (no real-time collaboration)
- Models typically only communicate to the SE domain
- Server collaboration software often does not scale and users suffer long download/commit times (>20 min)

Is there a set of tools that can meet this need?

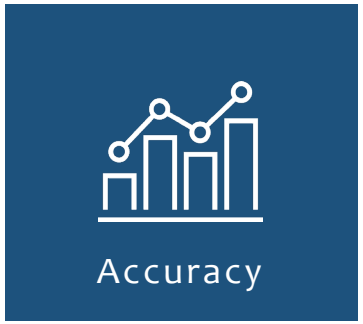
Innoslate™ – A 21st Century Solution

- First release: 2012
- Cloud-Native
- AI/NLP
- Agile



- Common options at the forefront
- Primary tool language is easy to learn (LML)
- Works with no installation, in a web browser
- ***Autogenerated diagrams and dashboards (concordance)***

- Real-Time Collaboration (Group Chat/Real Time)
- Easy Communication to other stakeholders (not just SEs)
- Proven Model-Based or Data-Based Review (MBR/DBR) capability



- Full Discrete Event Simulator which simulates cost, schedule, and performance is integrated into the Action Diagrams
- Full Monte Carlo Simulator to simulate variance
- NLP/ML for Requirements Quality and Traceability
- Intelligence View

- Tested to over 10 Million entities and 500 simultaneous users
- Cloud hardware auto-scales
- Software designed to scale

- Full requirements analysis and management capability with Requirements View
- Full modeling capability (SysML/LML/IDEF0)
- Test Center (Test Suites and Test Cases)
- Documents View (CONOPS, Project Plan, and Test Plan)

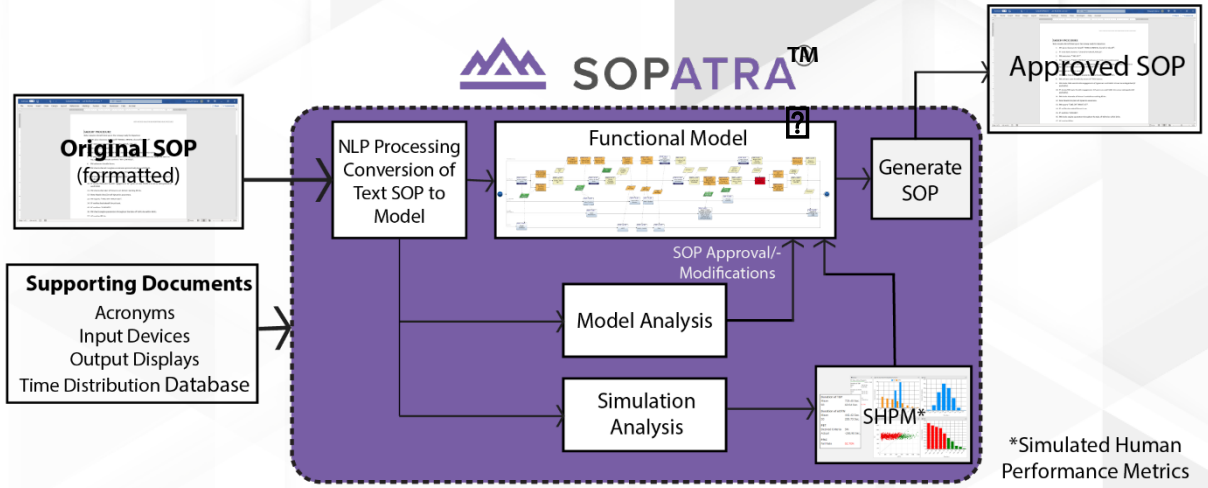
- Automatically generate and/or use other representations (SysML, DoDAF)
- Import from other RM tools (IBM DOORS CSV and ReqIF, Excel CSV)
- Integrations with STK, MatLab, and GitHub
- Open Java, REST, and JavaScript APIs

Sopatra™

We designed Sopatra with the overall goal to demonstrate the improved creation of Standard Operating Procedures (SOPs) by the automated creation and verification of SOPs using a digital assistant (DA)

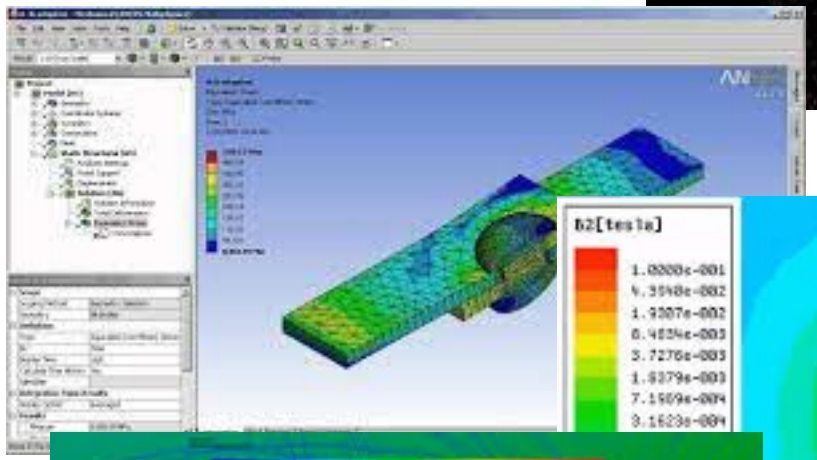
- Uses Natural Language Processing (NLP) for interacting with the systems engineering development environment (i.e. creating the behavior model)
- Provides configuration management and revision management of massive models
- Develops executable simulation of the behavior model
- Built on Innoslate®

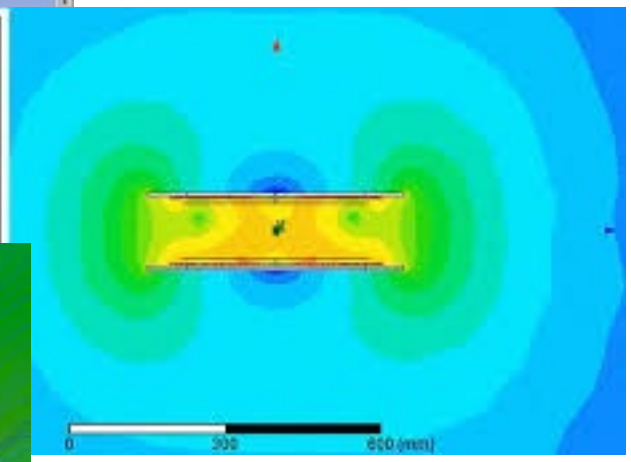
Sopatra is a digital assistant that will convert text to an LML Action Diagram and execute the simulation automatically to check all possible paths through the procedure.

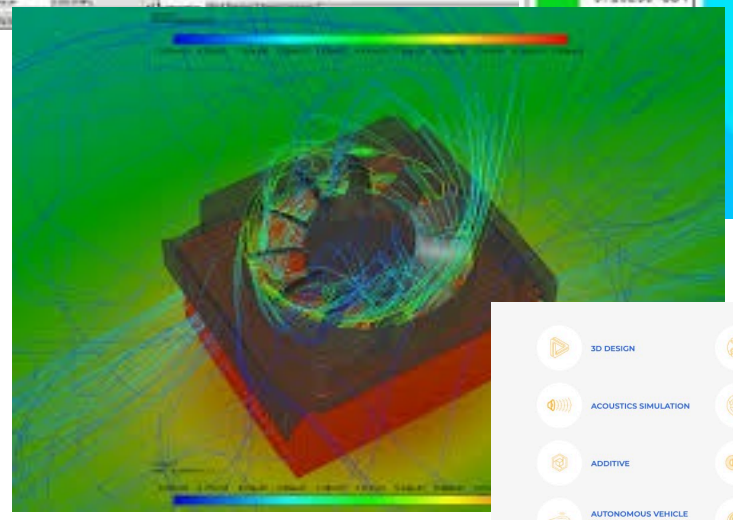


Ansys

- The Ansys tool suite is the “gold standard” of design engineering tools.
- With their recent acquisition of AGI, they brought the premier geospatial analysis tool (STK) into their toolset.
- With the recent acquisition of Phoenix Integration, they brought an amazing way to integrate a variety of common modeling and simulation tools – ModelCenter.
- They are exploring what they can do to cloud computing environments, however, for systems engineering, we need to mostly be concerned with providing the design engineers with good requirements and receiving the results of the design engineering activities.



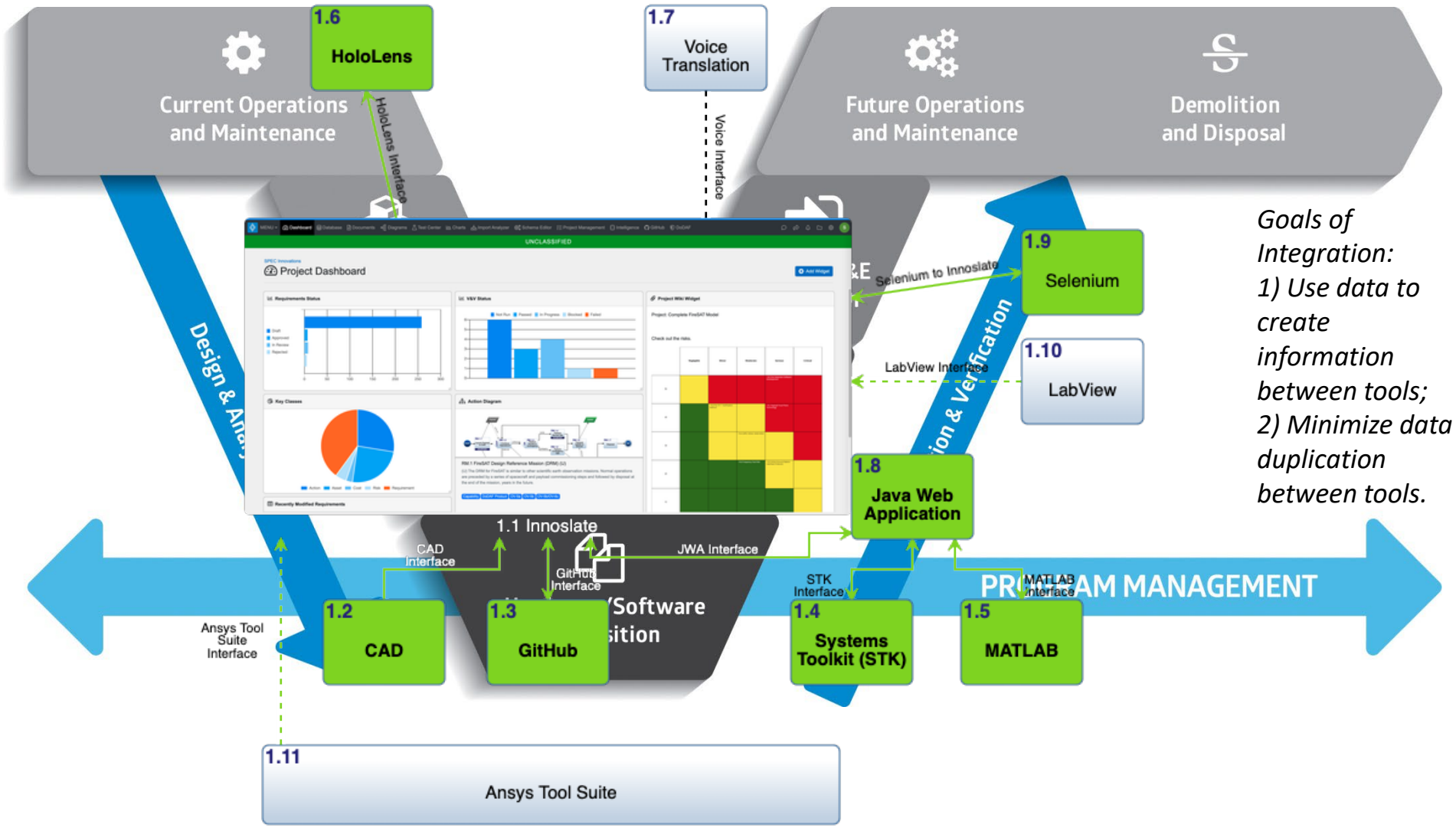




Innoslate Integrates Systems Engineering Activities Across the Entire Lifecycle

Innoslate's Digital Ecosystem

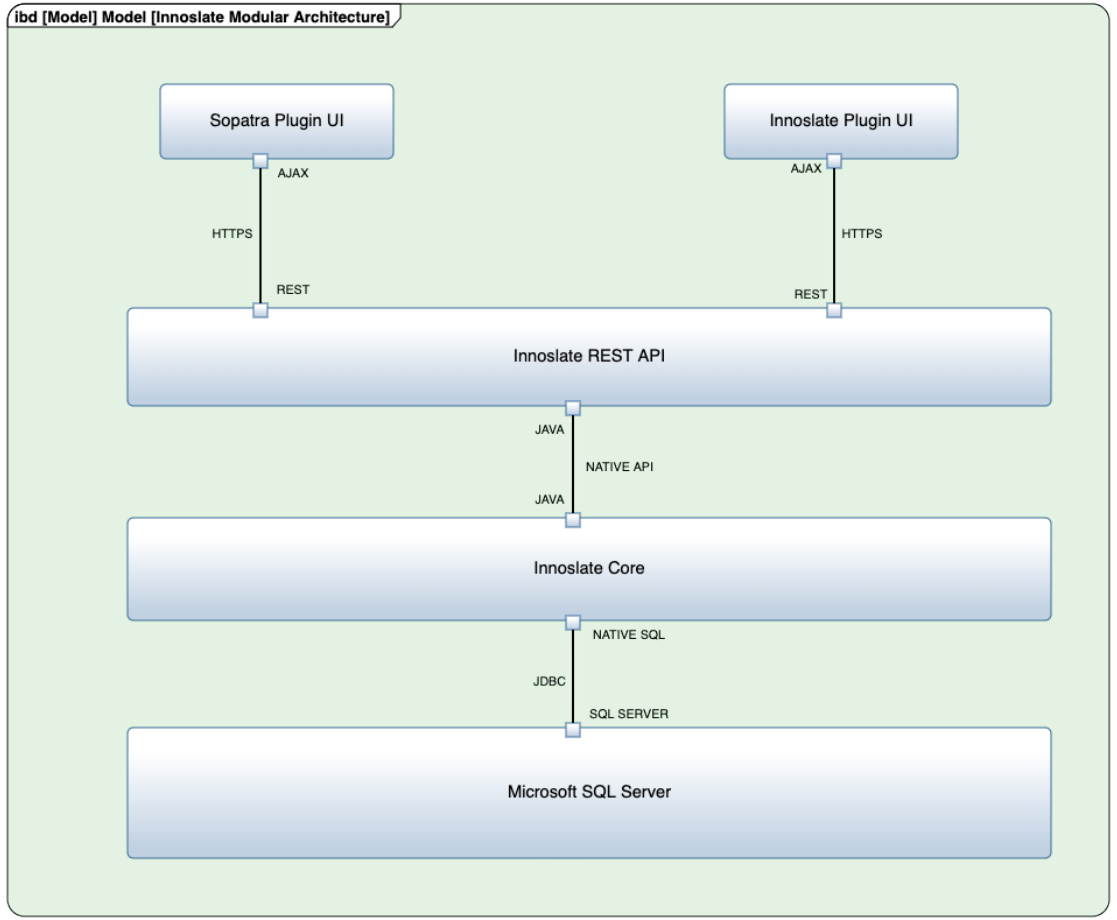
- Innoslate® provides a complete DDSE/DDE environment.
- A JWA is used to interface between the Innoslate cloud tool and desktop tools.
- We have directly integrated a number of key design engineering technologies to complete the digital thread.
- Several other tools are being added to enhance the current capability.



Goals of Integration:
1) Use data to create information between tools;
2) Minimize data duplication between tools.

Innoslate Uses a MOSA Architecture

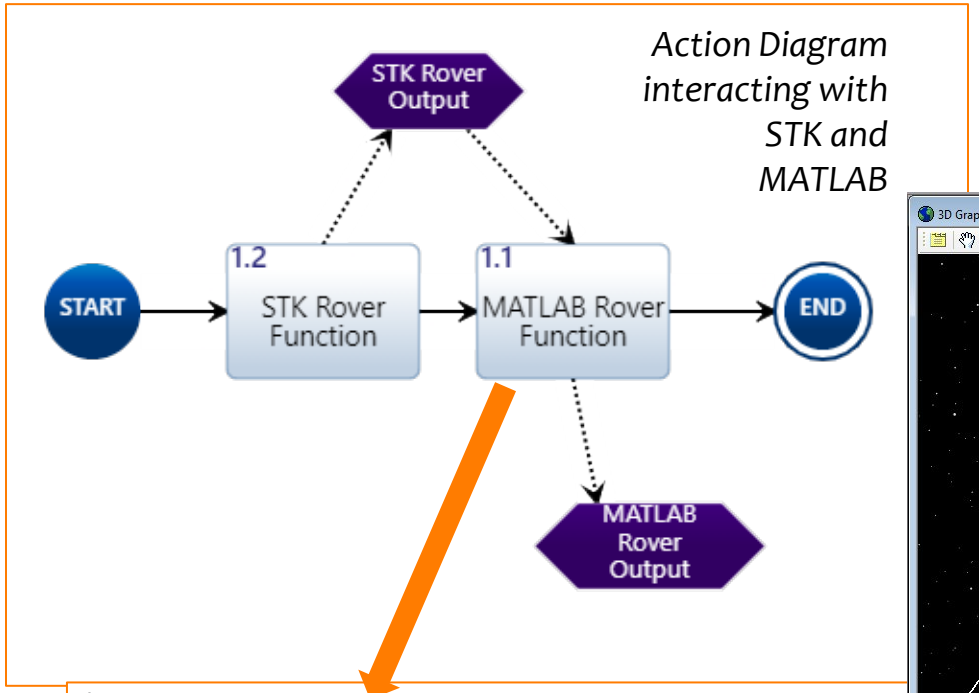
- Plugins are viewpoints of the Innoslate database.
- Plugin features:
  - Not a standalone application (requires Innoslate Core)
  - All authentication is through Innoslate Core with the options for:
    - Single-Sign-On CAC (Default)
    - Native Email/Password (Optional)
    - LDAP (Optional)
  - All data is stored in the U.S. Government managed SQL database using Innoslate Core (***no data lock***).
  - Innoslate REST API facilitates plugin data exchange.



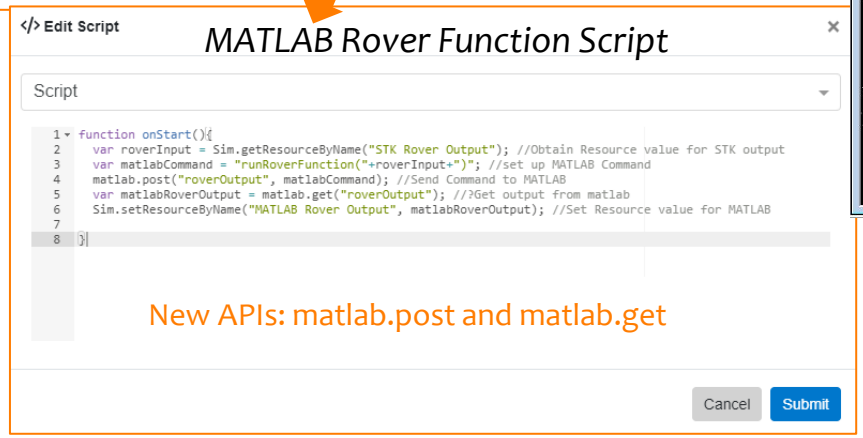
Modular Open Systems Approach (MOSA) Architecture Enables Architecture to Operations (DEVOPS)

Innoslate v.4.4

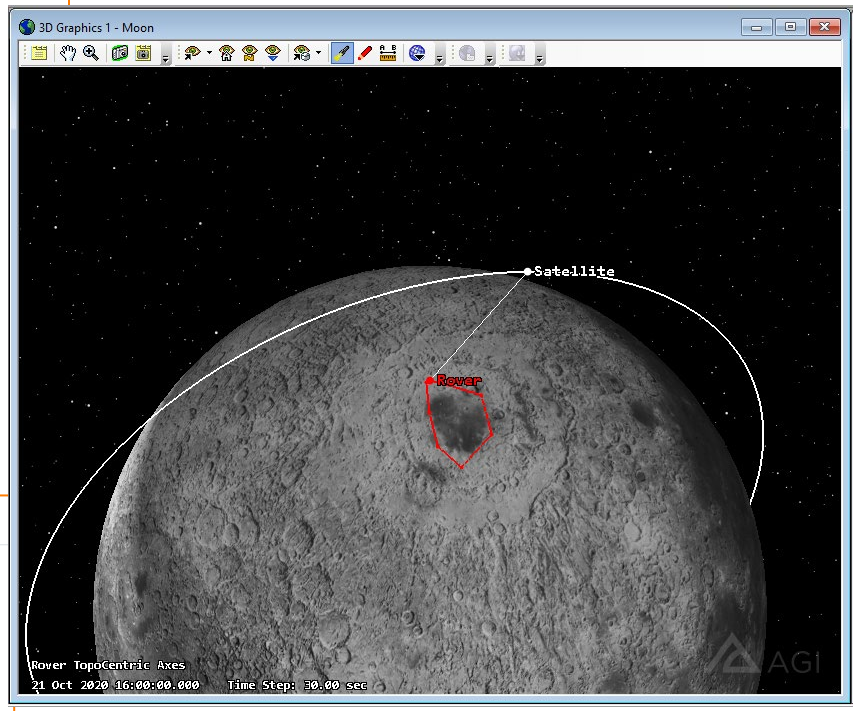
- Released in January 2021
- Provides capability to interface with MATLAB and STK through scripting APIs for co-simulation
- Only a small amount of code is required due to the added APIs performing most of the work for the user.
- Information can be easily passed to and from design engineering tools and Innoslate's discrete event simulator (Enterprise only feature) using this mechanism.



Action Diagram interacting with STK and MATLAB



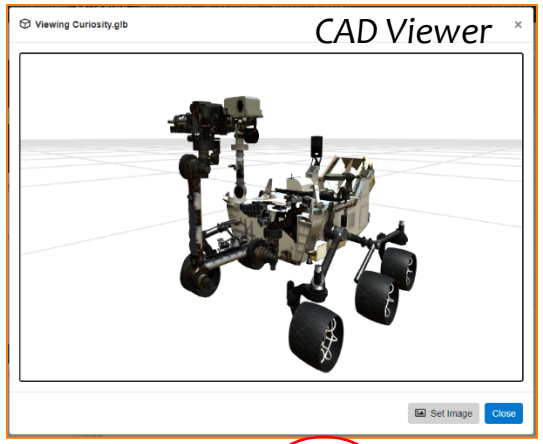
MATLAB Rover Function Script



STK Rover Model with Search Pattern and Communications Satellite

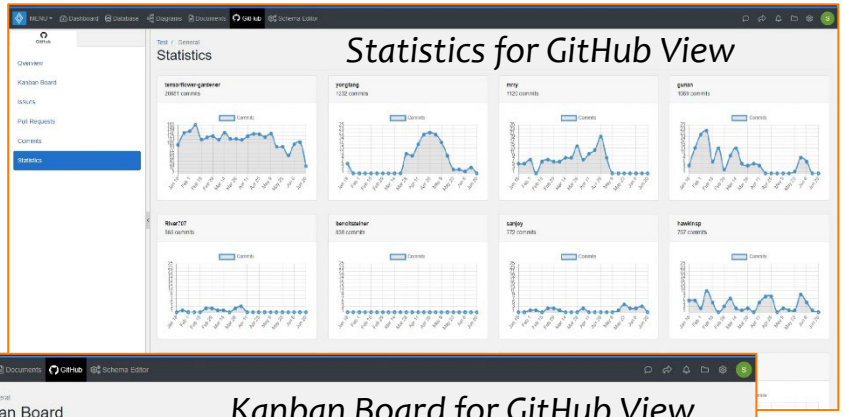
Innoslate v.4.4 (continued)

- GitHub View provides the capability to directly interface with GitHub using a special set of dashboards.
- Provides entire viewpoint to enable program managers and systems engineers to more effectively monitor software development progress and capture results
- CAD Viewer provides the capability to view CAD files and convert CAD data into Innoslate objects/BOM.
- Added other filetypes: GLB, glTF, 3DS, PLY, and ZIP (with OBJ and MTL combined)

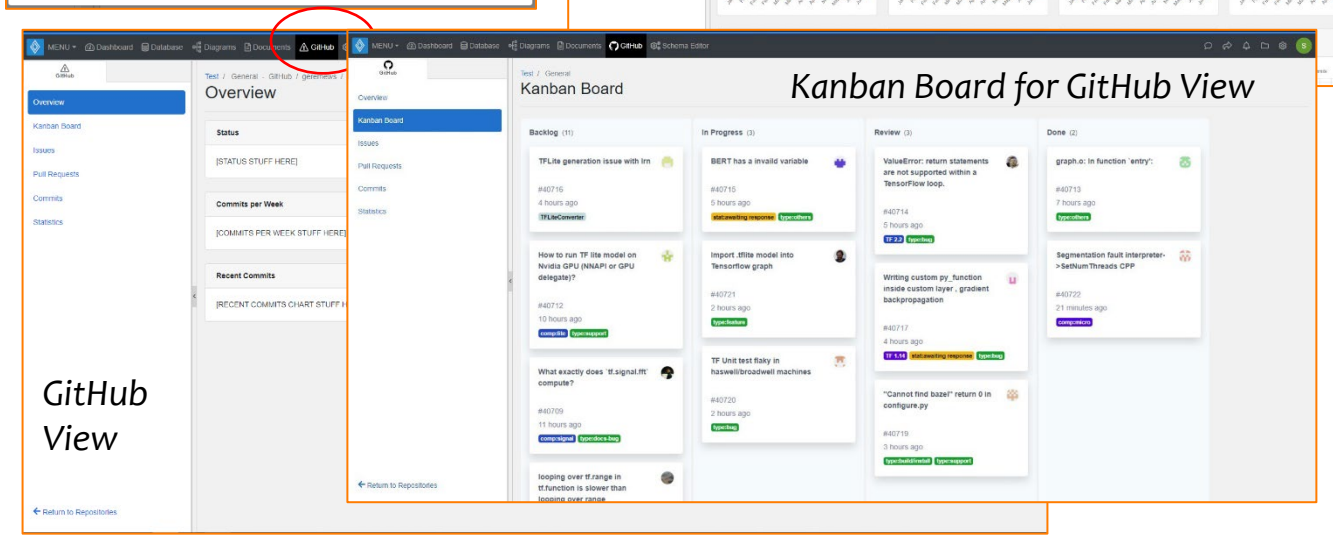


CAD Viewer

Not depicted: GitHub Issues Tracker, Pull Requests, Repository, and Commits views



Statistics for GitHub View

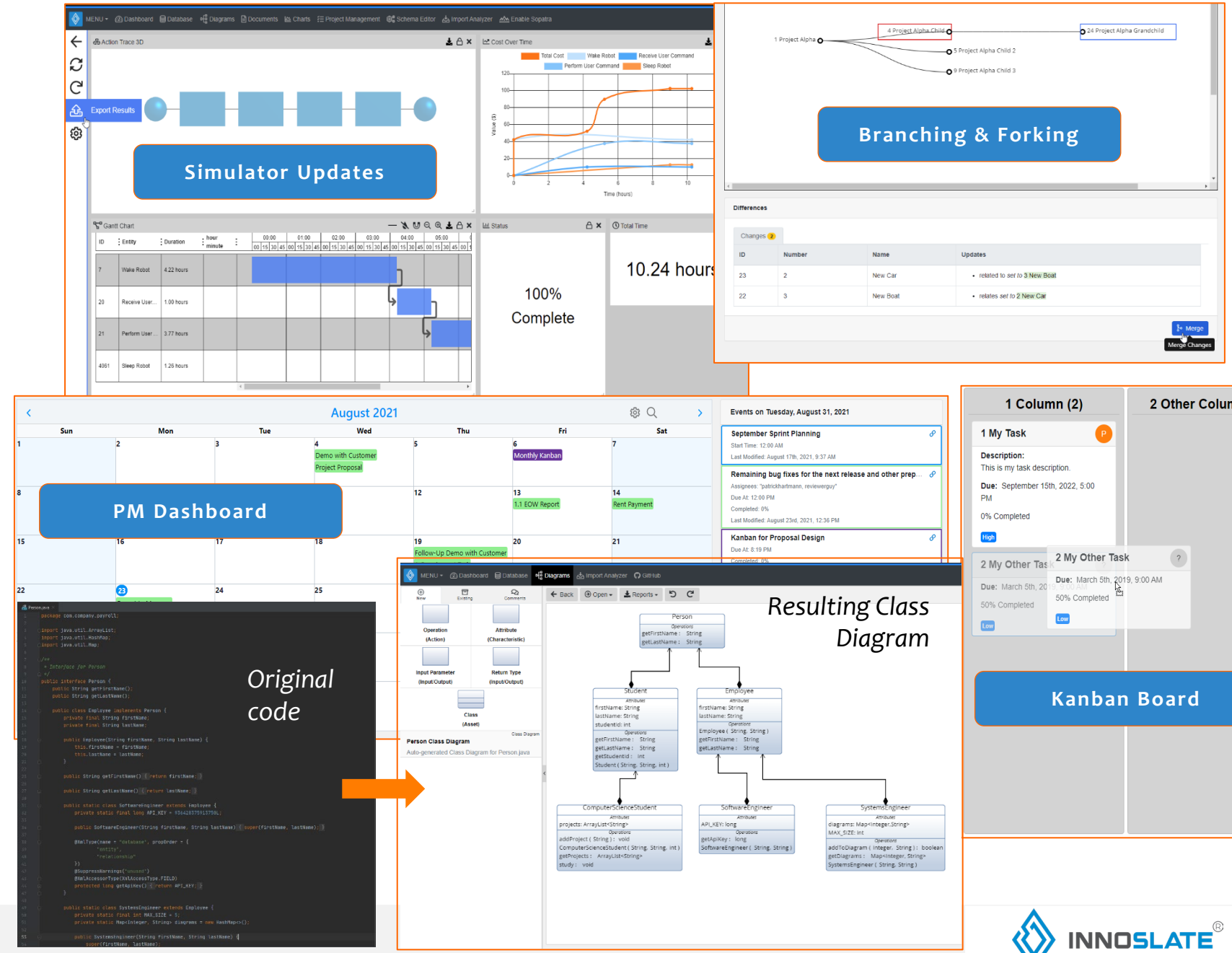


Kanban Board for GitHub View

GitHub View

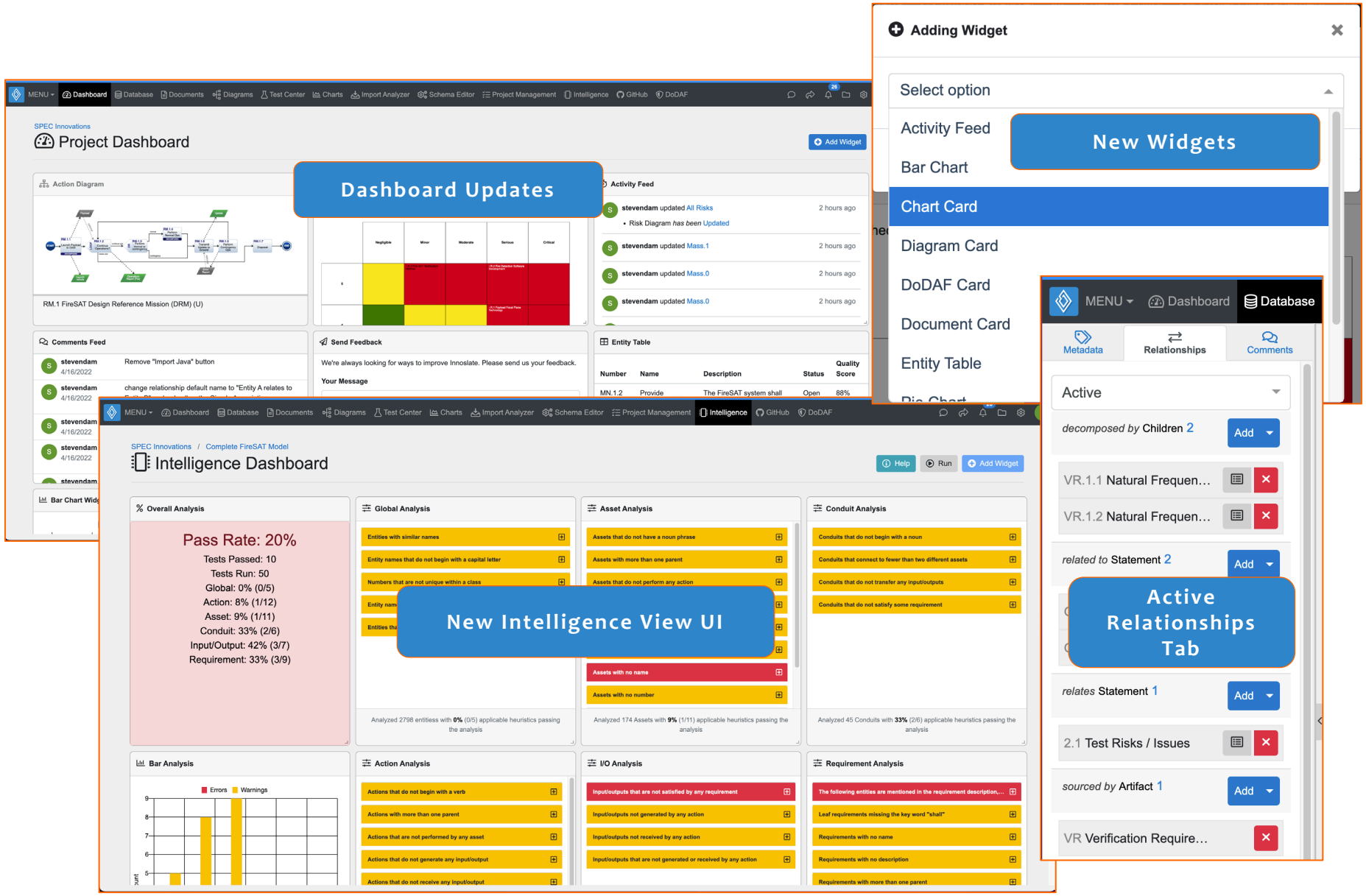
Innoslate v.4.5

- Released September 2021
- Enhanced Action Diagram
- Updated Simulators
- Program Management Dashboard
  - Kanban Boards
  - Calendar View
  - Gantt Chart
- Branching & Forking
- Class Diagram Generator
- Equation Editor
- And many bug fixes



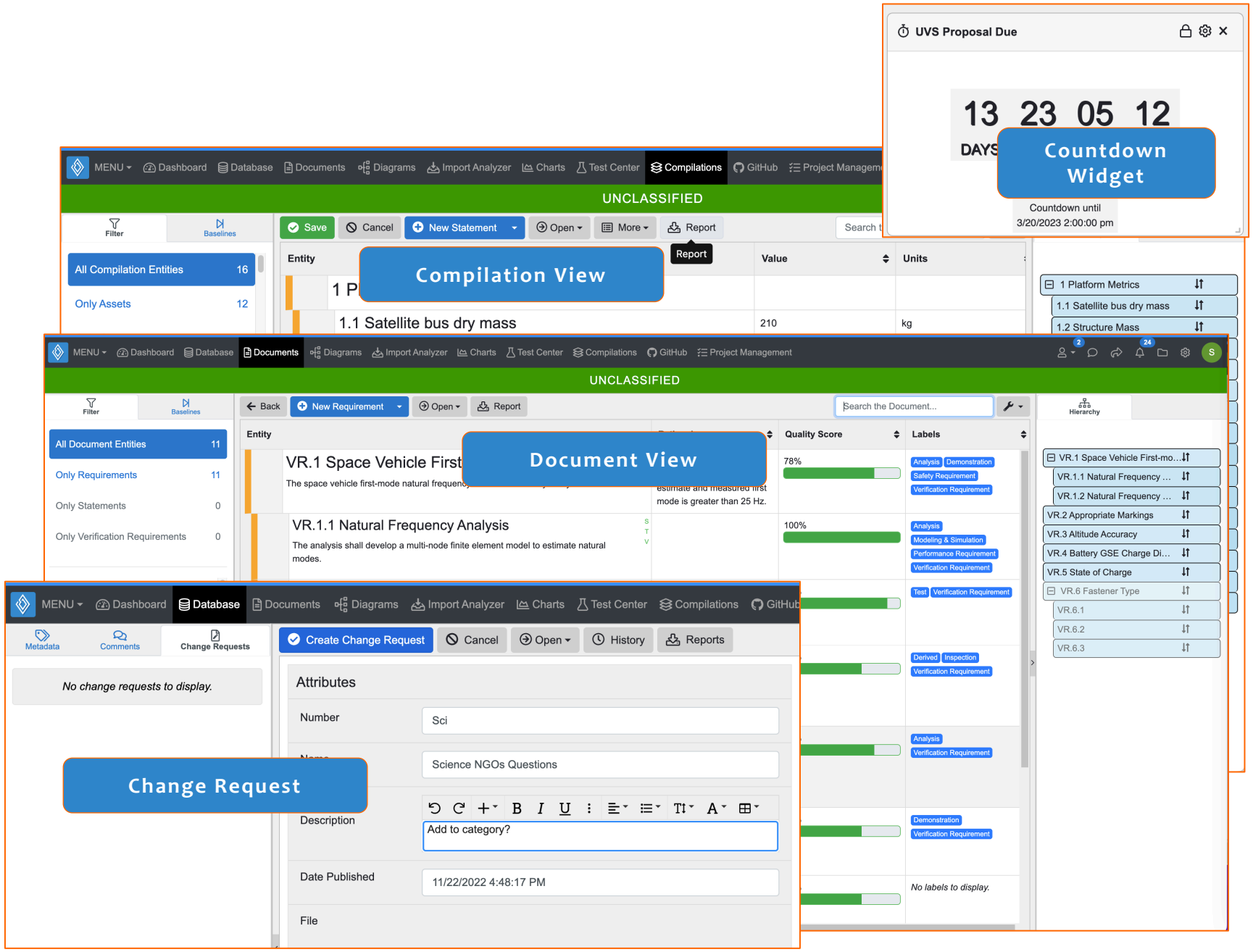
Innoslate v.4.6

- Released May 2022
- New widgets for Project and all other dashboards
- Updated dashboards for all views, including Admin dashboard
- New Intelligence View User Interface
- New relationships panel tab: Active (replaces “popular”)
- Enhanced Guided Tour
- Many minor enhancements
- And over 500 issues resolved



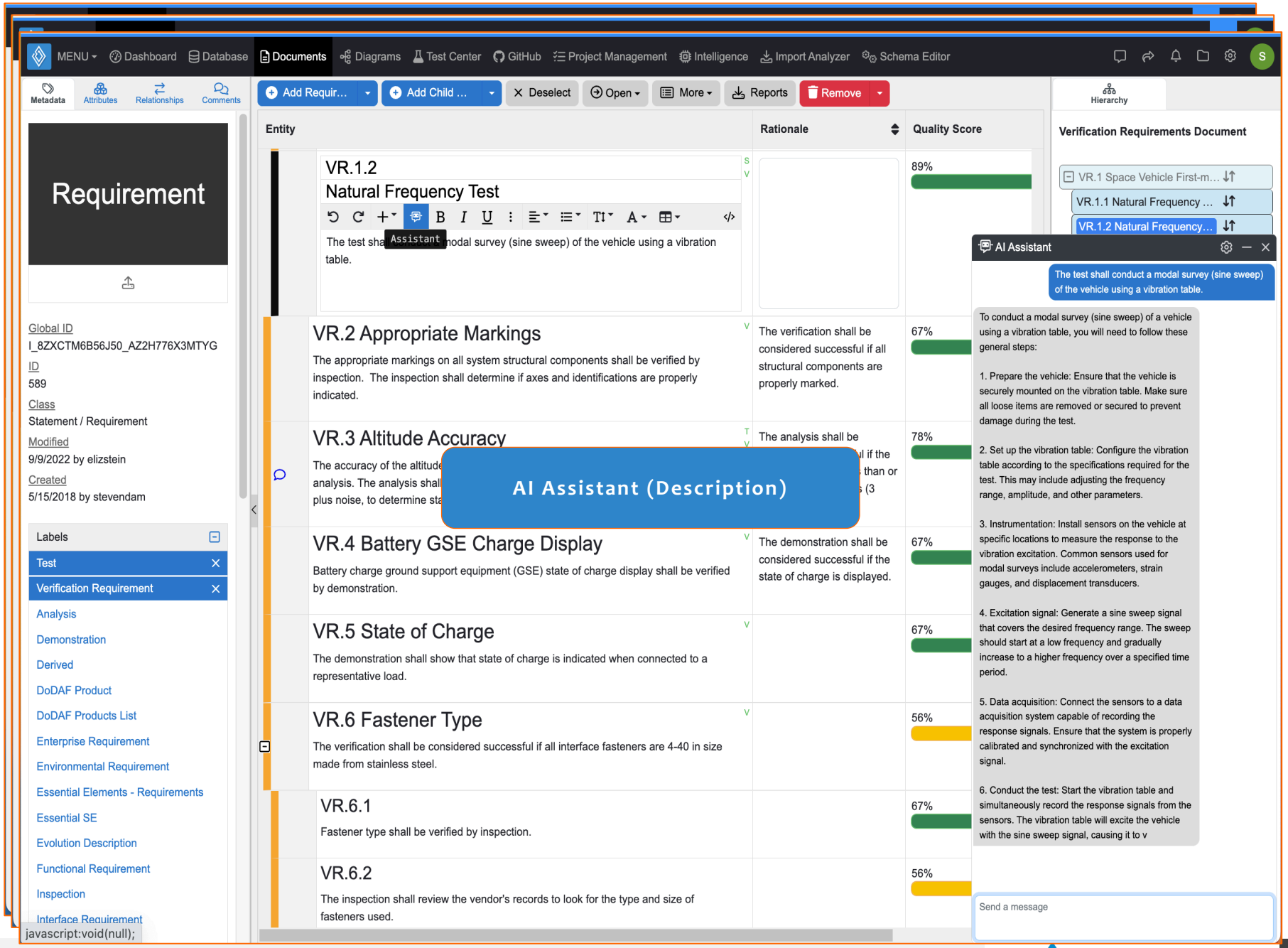
Innoslate v.4.7

- Released February 2023
- Redesigned Documents View & Test Center for ”infinite scrolling” and meta-numbering, drag & drop
- New Compilation View shows any entity class in a document form
- Diagram export
- Change requests now available in Entity View
- GitHub integration enhanced to enable requirements creation and traceability maintenance
- Numerous other enhancements and bug fixes



Innoslate v.4.8

- Released July 2023
- Added Generative AI features
  - AI Assistant (General)
  - Innoslate Help
  - AI Assistant (Description Attribute)
- Improved: Chat messages restyled to make the colors and sender easier to read
- Improved: Updated Redactor's edit image modal to a bootstrap modal
- Improved: Reset Test Suites without creating a baseline
- Numerous Bug Fixes



When Should We Integrate Tools?

- Lacking a common ontology, we need to carefully select the tools we want to integrate with.
- For the most part, we can rely on the export of a common format, such as CSV, XML, or ReqIF, to move data between tools; APIs can also be used to move the data between tools.
- Note that this data usually flows in one direction well, but not bi-directionally.
- As systems engineers, we mainly want to produce specifications for the design engineers, take the results of the design engineering tools, then import them into our SE database.
- A completely seamless integration between our SE tools and DE tools may be a bridge too far at this time, but that shouldn't stop us from trying to make it work.

Careful integration is critical to avoiding "garbage in, garbage out."

When Should We Not Integrate Tools?

- Physics-based simulations often apply only a portion of the physics needed to fully represent a system and its environment.
  - Limited theoretical basis for physics itself
  - Approximations used due to our lack of math
  - Application to areas where we don't know the exact physics
- Physics-based simulations require extensive calibration of experimental data to be useful.
  - Simulation codes use "knobs" to adjust results to match experiments.
  - Users need to understand the boundary conditions where the results are valid.
- Pushing poorly or non-analyzed simulation results up to the systems level may introduce new errors.

In these cases, we need an "air gap" or at least an "analyst-in-the-loop."

How Does DDSE and The Innoslate/Ansys Tool Suite Support SE Modernization?

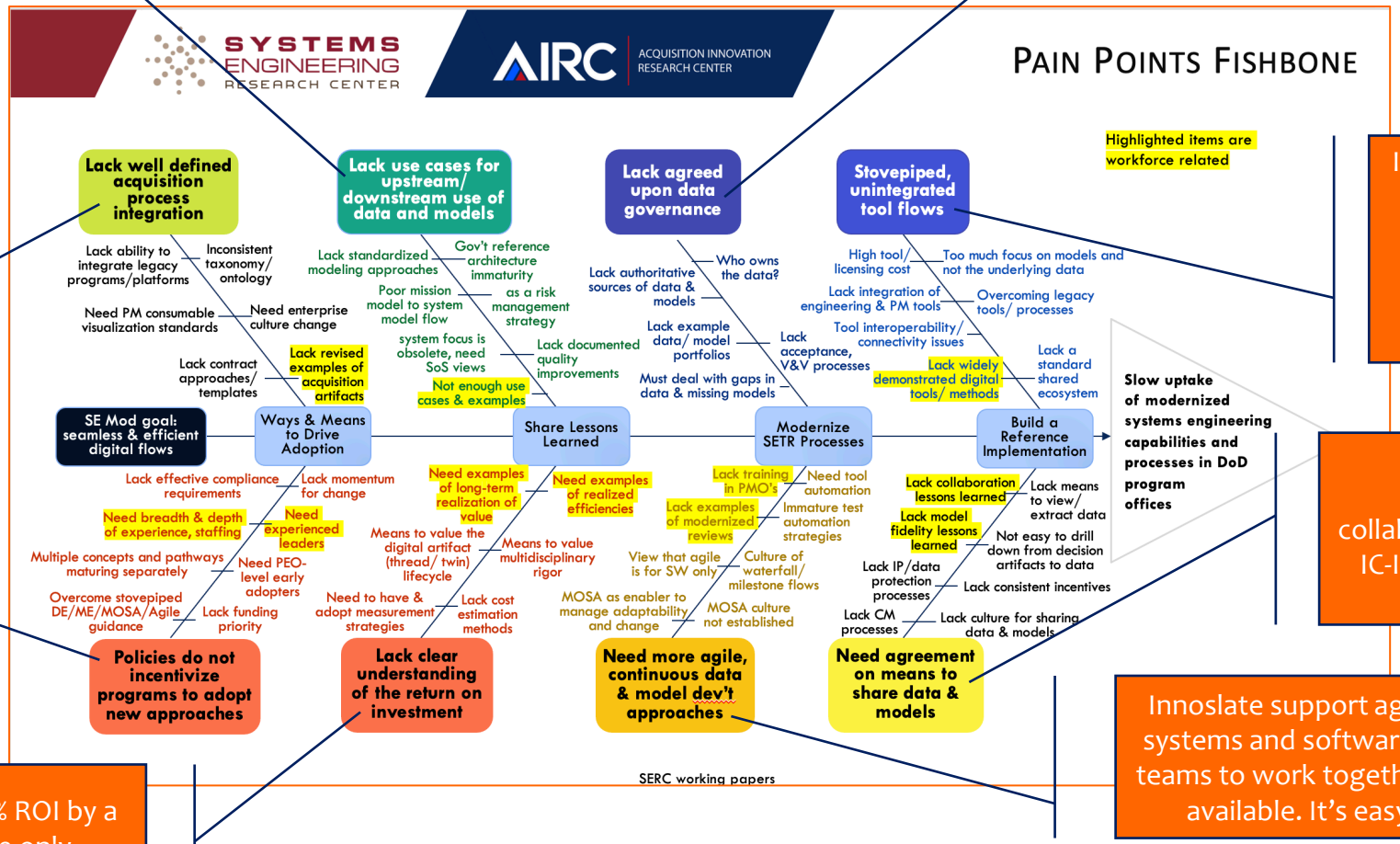
- Digital Engineering
  - Innoslate, Sopatra, and the Ansys Tool Suite provide the full range of tools needed to create digital twins.
- MOSA
  - Functional analysis focuses on functions that can be allocated in many different ways – Innoslate provides a complete functional (and object) analysis capability.
  - Innoslate has unique interface diagrams to enable a better definition of the interfaces.
  - Innoslate's MOSA architecture enables modular use of the data too.
- Mission Engineering
  - Integration of the Innoslate discrete event simulator with STK enables us to plan, analyze, organize, and integrate operational and system capabilities to achieve desired effects.
- Agile Development
  - Innoslate provides direct integration to software engineering tools (GitHub and Selenium – in the near future).
  - Can align Sprints and Epics with standard SETR events by conducting Data-Driven Reviews

Innoslate Soothes the Pain Points

Innoslate provides several "starter" models for use in its guided tour and in the Model Curation Station

Innoslate has a well defined and published schema, based on the Lifecycle Modeling Language, which provides an ontology for SysML, DoDAF, BPMN, and many other languages

Innoslate has PM tools embedded, including Cost, Risk, and Decisions. Its diagrams are easily understood by anyone



Innoslate provides a complete environment for PM/SE work and has built-in integration with key design engineering tools, including GitHub, MATLAB, and STK

Innoslate growth has been limited by certain DoD organizations effectively mandating old, foreign technology tools

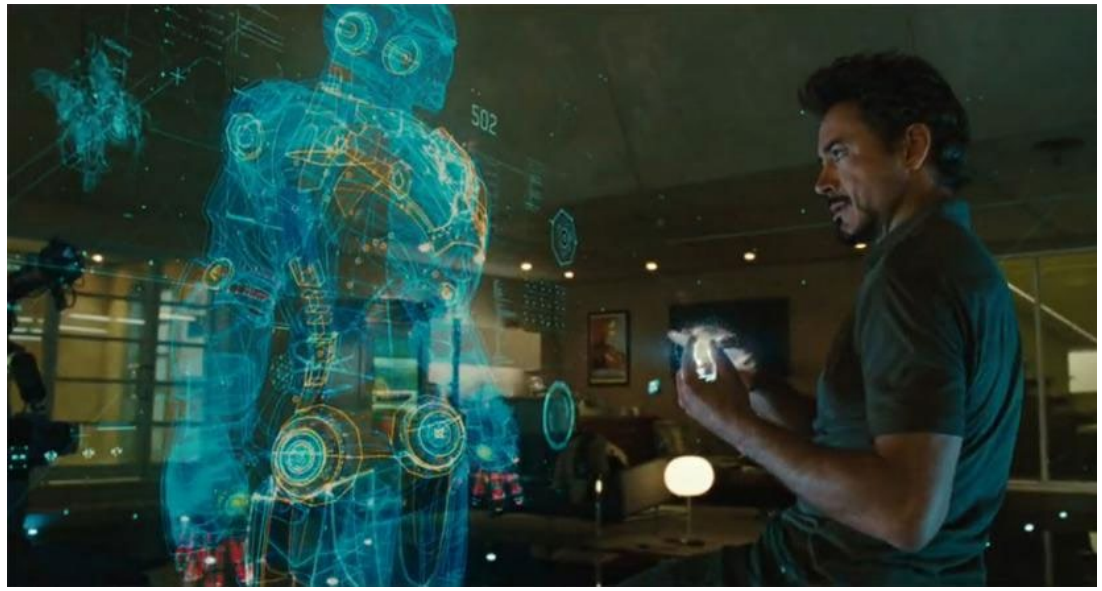
Innoslate's cloud native environment provides collaboration at NIPR, SIPR, JWICS, IC-ITE, and Iron Bank. Has Open APIs.

Innoslate was rated with a 100% ROI by a GMU study in 2019. We have only improved dramatically since then.

Innoslate support agile development for both systems and software engineering by enabling teams to work together. GitHub round-trip now available. It's easy use and easy to learn

Summary

- LML provides a path to the future of systems engineering and DDSE.
- If SE Modernization is the future, then we need to have modern cloud-based tools that implement NLP/ML, AR, and other 21st Century technologies.
- We recommend including LML in any and all SE tools to enhance tool interoperability.
- We cannot design future systems using old methods and tools.
- We need to embrace these new technologies, not hamper them.



Innoslate Version 20 = Jarvis?

We look forward to helping make "Tony Stark's lab" a reality!

Questions and Answers

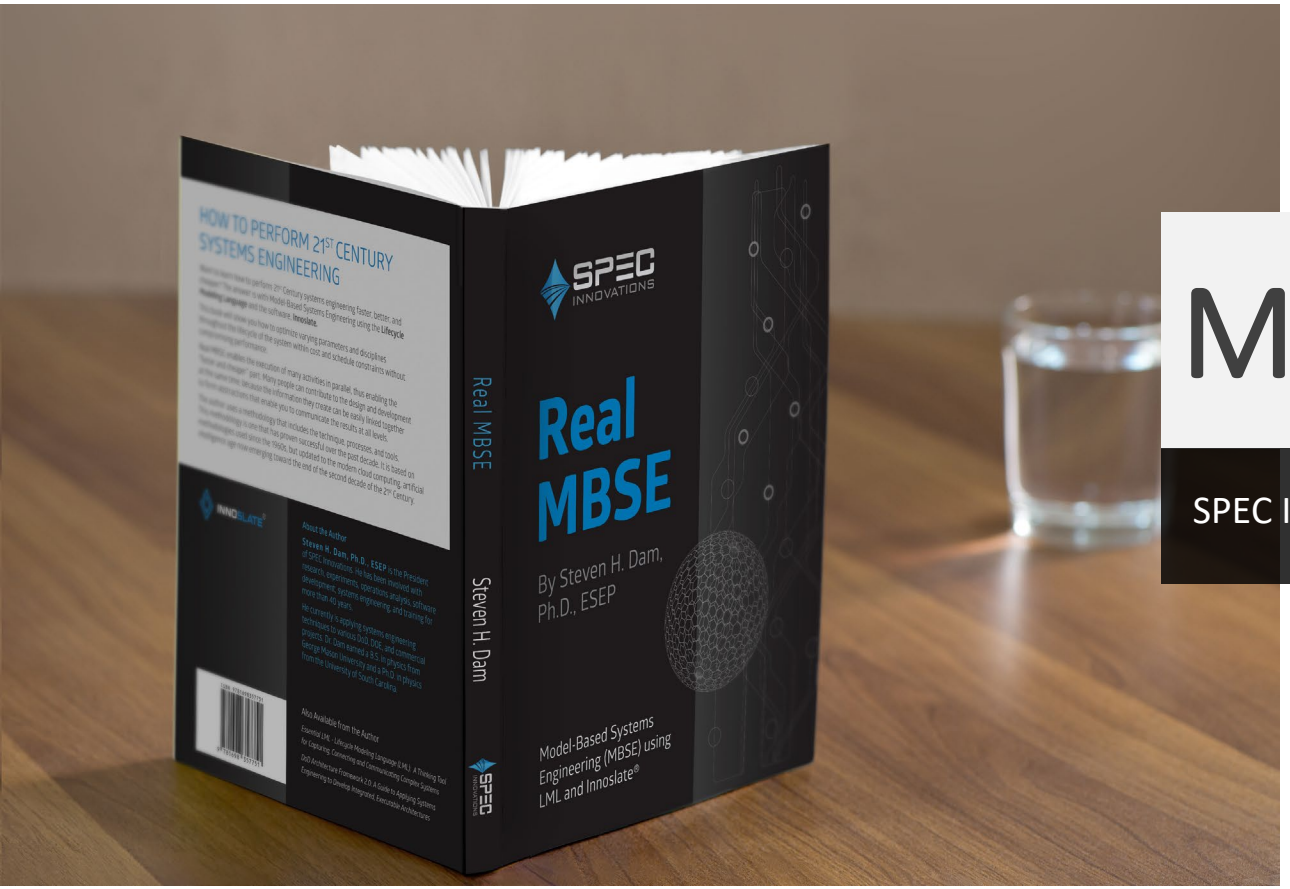
Use the panel on the right to ask your questions

SPEC INNOVATIONS

- SPEC Innovations provides **software, training, and consulting** to the defense and aerospace industries and the intelligence community.
- Our flagship software product, Innoslate, is the first cloud-native, **model-based systems engineering** software solution made solely in the United States of America.
- Our engineers built Innoslate to help systems engineers develop full lifecycle solutions for **complex system of systems.**
- Innoslate software supports **Requirements Management**, **Modeling and Simulation**, **Verification and Validation**, and more in one seamless package.

About Us

We are the experts in systems engineering



More Resources

SPEC Innovations offers training, books, videos, documentation, trials, and more

Training: specinnovations.com/training

Books: “Real MBSE” textbook and lab manual available on Amazon

Videos: Visit the SPEC Innovations Youtube channel

LinkedIn: Innoslate and Systems Engineers User Group

Documentation: help.Innoslate.com

Trial: cloud.Innoslate.com

Upcoming Webinars (tentative schedule)

Who	What	When
Kerry Lunney (Moderator)	INCOSE President-Elect Candidates Present Their Vision Statements	Tuesday 22nd August 2023 at 11am EDT
Rick Dove	Agile Systems Engineering – It's Not Your Father's Oldsmobile	Wednesday 20 September 2023 at 11am EDT

Invitations will be emailed in advance and informational updates will be placed on www.incose.org
Go to http://www.incose.org/products-and-publications/webinars for more info on the webinar series, including a way to view the last 164 Webinars and soon – this one!

Information on the webinars can now be accessed through the Professional Development Portal (PDP), at https://www.incose.org/_pdp/advanced-search-the-content-catalog - search for "Webinars"
Joining instructions will added around two weeks before each webinar is scheduled to take place.

Upcoming Events

SAE/NASA/INCOSE Energy and Mobility Conference
Cleveland, Ohio, USA
12 – 15 September 2023
https://www.energyandmobility.org/

ENERGY&MOBILITY
TECHNOLOGY, SYSTEMS AND VALUE CHAIN
CONFERENCE & EXPO
SEPT 12-15 • I-X CENTER • CLEVELAND, OHIO

Questions: *Is our modern civilization "Sustainable"? Can our aging and fragile electric grid support the coming "Transportation Electrification"? How will emerging technologies integrate EVs and electric airborne systems for urban mobility? – and… Can we hope to achieve a "NetZero" Carbon Climate without bankrupting our children?*

ATTENDEE/EXHIBITOR
REGISTRATION

INCOSE Systems Engineering Professional PDU Credit

For more information about INCOSE certification renewal, including PDU categories and limits: http://www.incose.org/certification/CertProcess/CertRenew

Claim PDUs

You can claim 1 PDU credit towards your INCOSE Systems Engineering Professional (SEP) renewal by attending this entire webinar.

Eligible Sources To Claim PDU

- Live attendance at the webinar: "Attend non-peer-reviewed Professional Technical Society event."
- Watching a recording of the webinar: "Consume SE-related media, including journal article, book, video, or audio."

Claim PDUs for Other certifications

INCOSE webinars may also apply to the PDU requirements of other organizations, depending on the subject matter.

With thanks to our Sponsors for 2023