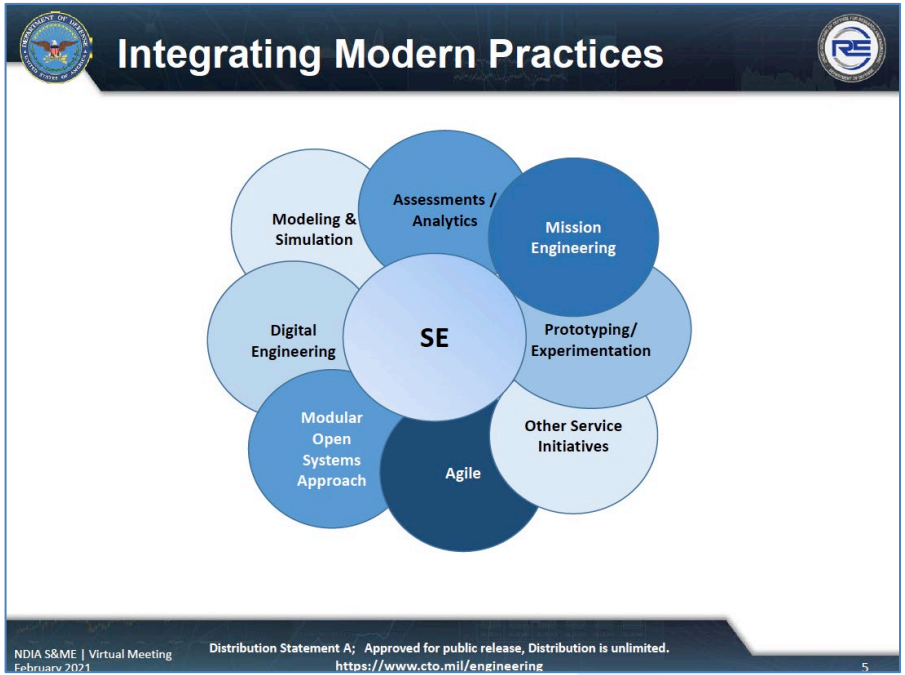
Can Data-Driven Systems Engineering Meet the Goals of SE Modernization?

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

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



SE Modernization – SERC 's View

aka "The Bomb Cyclone Diagram"

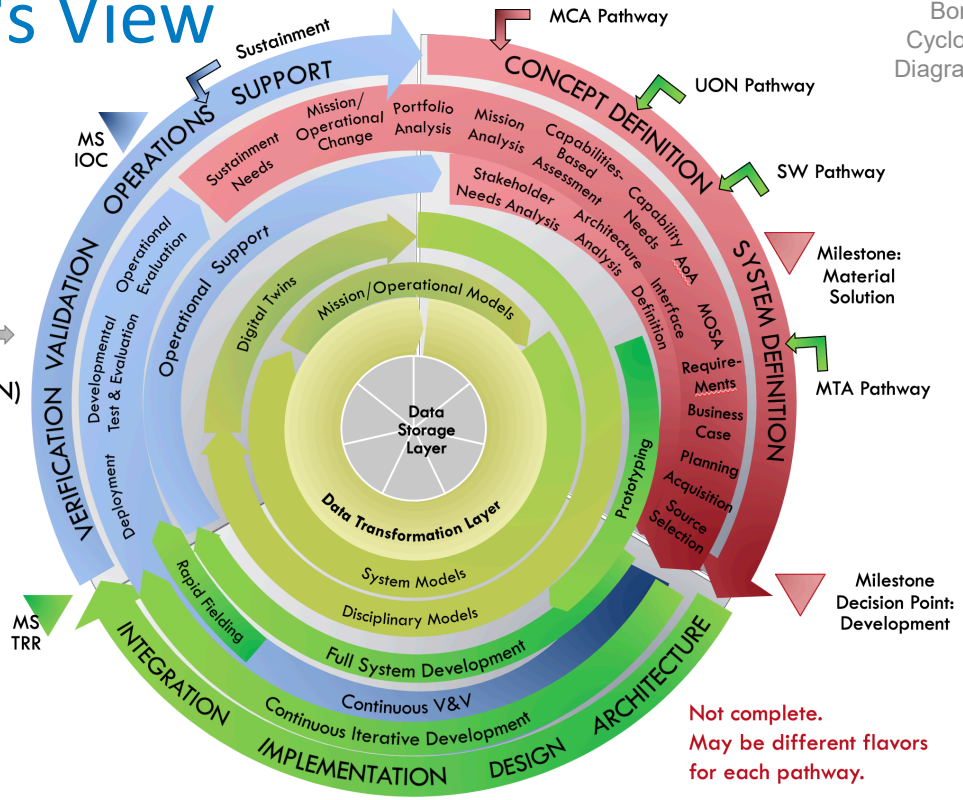
From https://sercuarc.org/systems-engineering-modernization/?utm_source=newsletter&utm_medium=email&utm_content=Systems%20Engineering%20%28SE%29%20Modernization&utm_campaign=SERC%20UPDATES%20JUN%202022 accessed 6/22/2022

"…this view emphasizes the DE transformation using a layered model with <u>data storage and transformation</u> 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	



Not complete.
May be different flavors for each pathway.

So to implement this approach we need to be data-driven in systems engineering

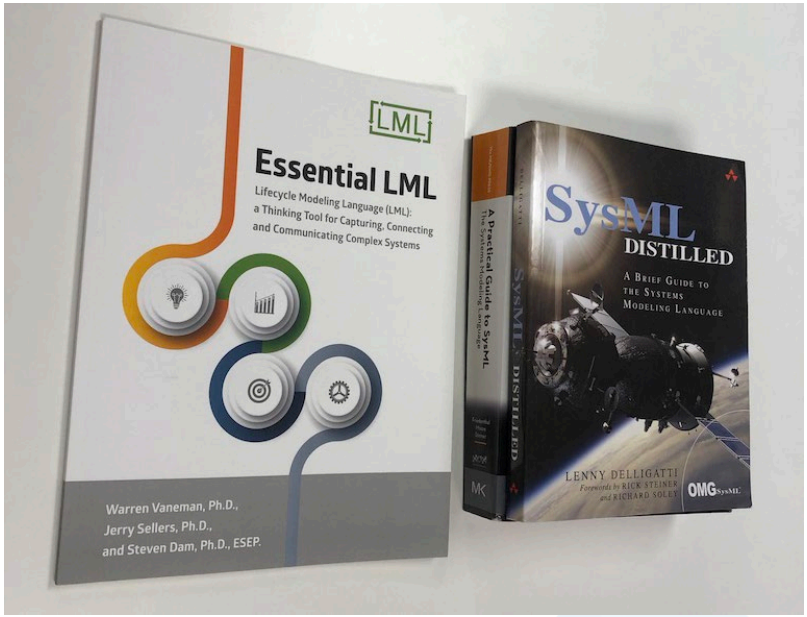
What's Data-Driven Systems Engineering (DDSE)?

- 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 – hence DDSE
- We define DDSE as:
  - *the transformation of user needs to requirements for design engineering and the transformation of design engineering data into verified and validated system-level information for decision makers to make better decisions throughout the lifecycle*
- This definition refocuses us on the underlying basis for systems engineering and explicitly identifies the benefit to all stakeholders
  - Design engineers get clear, easy to understand requirements
  - Decision makers get the information they need to make good decisions
- Fortunately there already exists such a language to meet this need: the Lifecycle Modeling Language (LML)

Lifecycle Modeling Language (LML)

LIFECYCLE MODELING LANGUAGE

- 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 current managed by the not-for-profit organization: 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, Air Force Academy
- LML is easy to learn, use, and extend
- Visit http://www.lifecyclemodeling.org

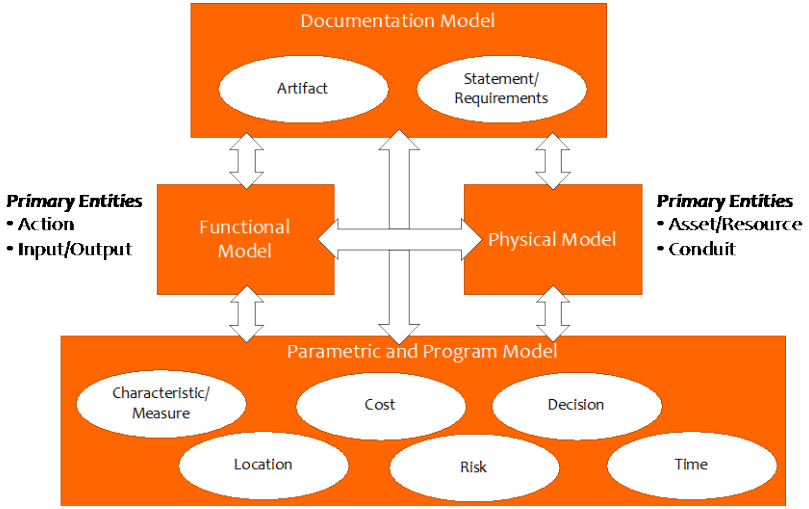


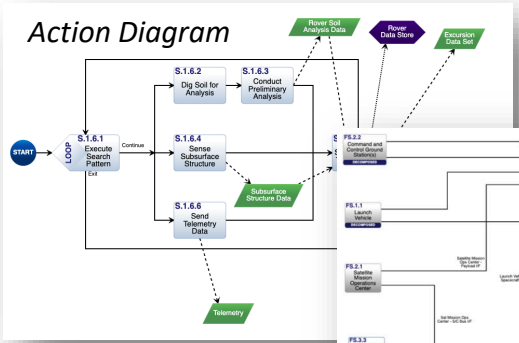
LML has proven to provide a strong ontology for systems engineering and program management and forms the basis for Data-Driven Systems Engineering (DDSE)

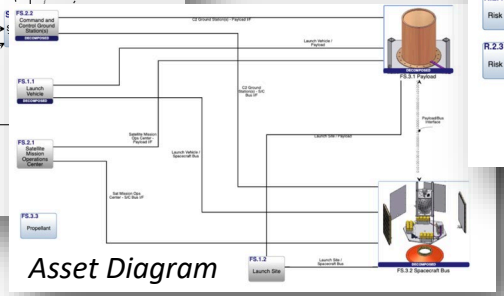
SPEC INNOVATIONS

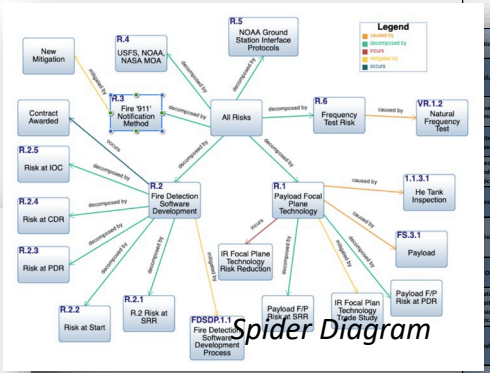
LML Overview

- 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









	Action	Artifact	Asset (Resource)	Characteristic (Measure)	Connection (Conduit, Logical)	Cost	Decision	Input/Output	Location (Orbital, Physical, Virtual)	Risk	Statement (Requirement)	Time
	decomposed by* related to*	references	(consumer) performed by (producer) (wires)	specified by	-	incurs	enables results in	generates receives	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
	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
	(consumed by) performs (produced by) (used 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
	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
	-	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
	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
	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
	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
	locates	locates	locates	locates specified by	locates	locates	locates	locates	decomposed by* related to*	locates mitigates	locates (satisfies) traced from (verifies)	occurs
	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
	(satisfied by) traced to (verified by)	references (satisfied by) source of 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)
	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, 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

Most of Today's Systems Engineering Tools Are *Based on 20th Century Technology*

Difficult to Use

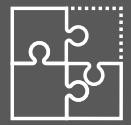
- Learning full SysML/UML often takes months
- Dozens of options are present on many views
- Drawings usually must be redrawn (ex. the Sequence, Activity, and BDD)
- Required desktop software installation to modify the models

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

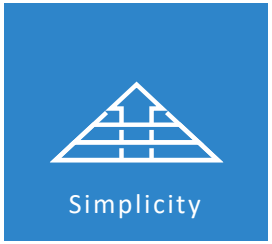
- Model conflicts during commits (no real-time collaboration)
- Models typically only communicate to the SE domain
- Server collaboration software often does not scale and users suffer long download/commit times (>20 min)

So is there a set of tools that can meet this need?

SPEC INNOVATIONS

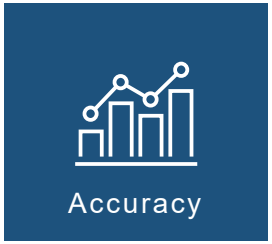
- First release: 2012
- Cloud-Native
- AI/NLP
- Agile

Innoslate – A 21st Century Solution



- 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 1,000 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

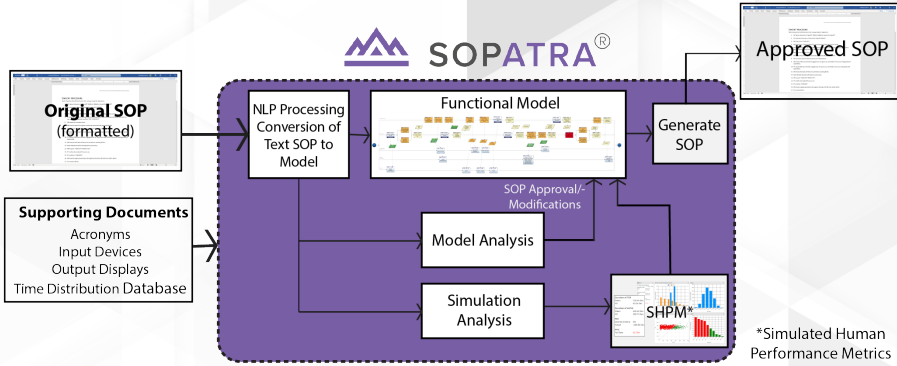
SPEC INNOVATIONS

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)

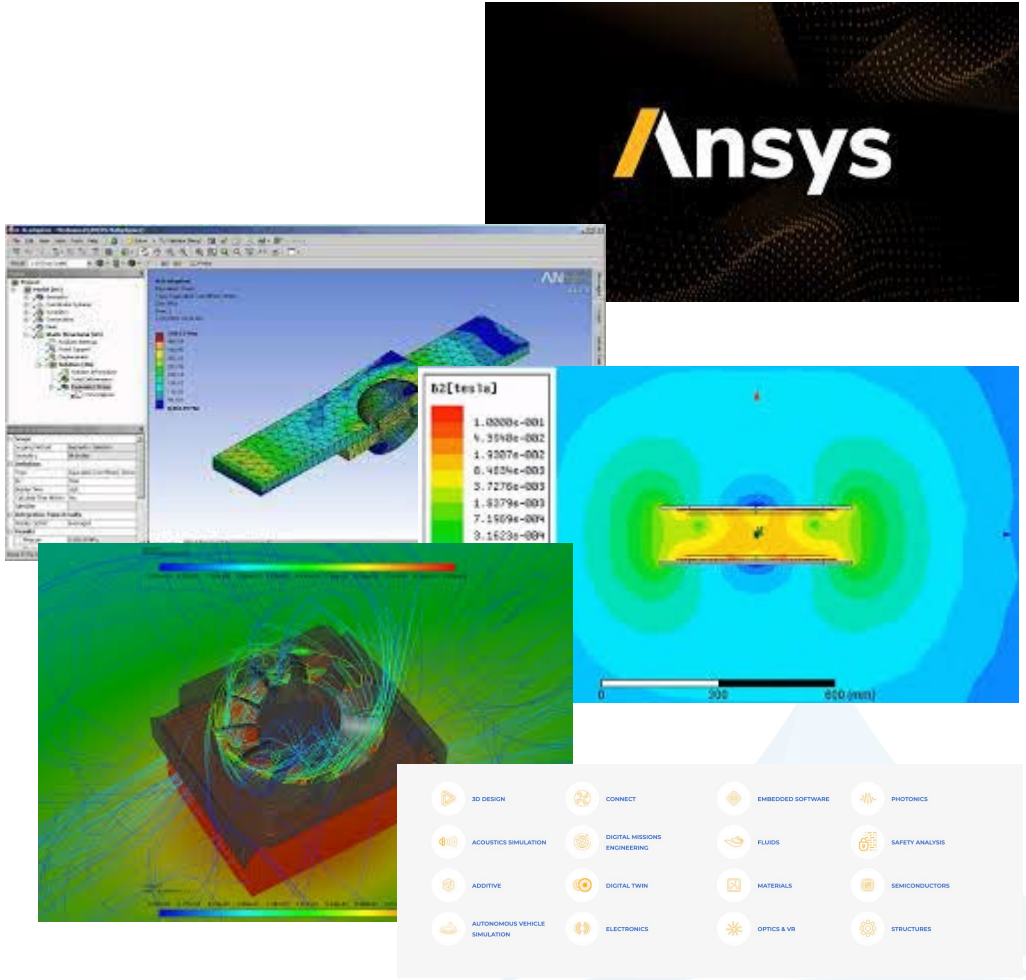
Sopatra is a digital assistant that will convert text to an LML Action Diagram and execute the simulation automatically to check all possible paths through the procedure.

- 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®



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 tool set
- 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 porting what they can to cloud computing environments, however for system engineering, we need to mostly be concerned with providing the design engineers with good requirements and receive the results of the design engineering activities

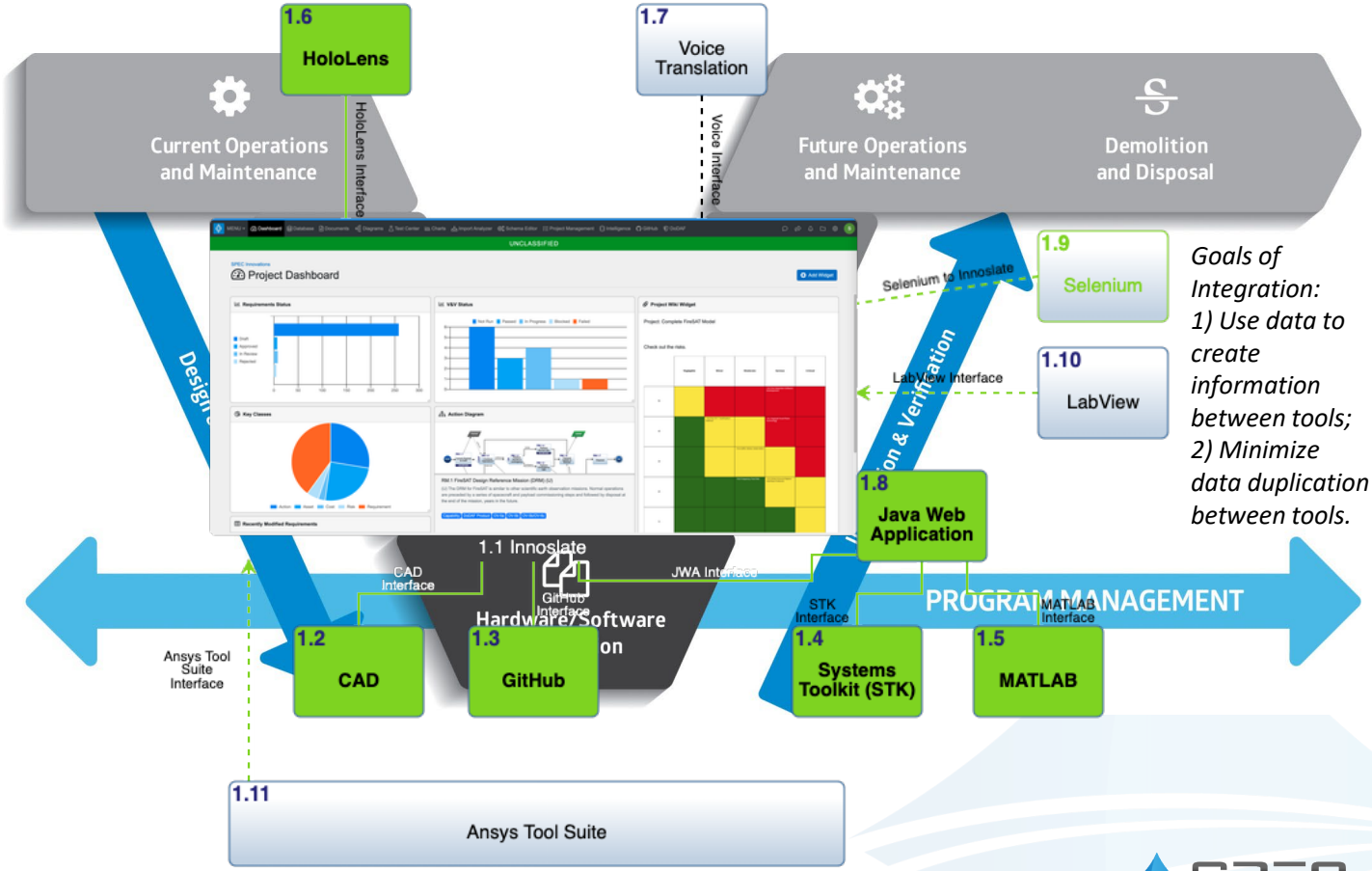


Innoslate Integrates Systems Engineering Activities Across the Entire Lifecycle

Innoslate's Digital Ecosystem

SPEC Innovations is an Ansys Partner

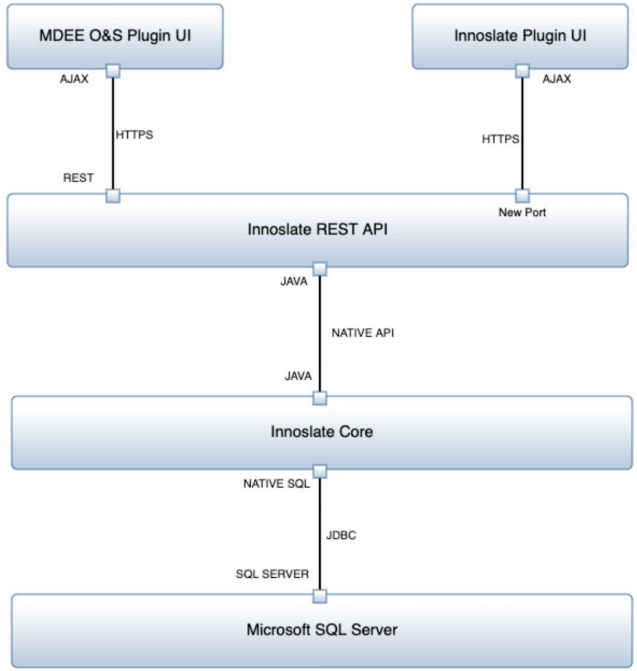
- Innoslate® provides a complete DDSE/DDE environment
- A JWA is used to interface between the Innoslate cloud tool and desktop tools
- We have directly integrated a number of key design engineering technologies to complete the digital thread
- Several other tools are being added to enhance the current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



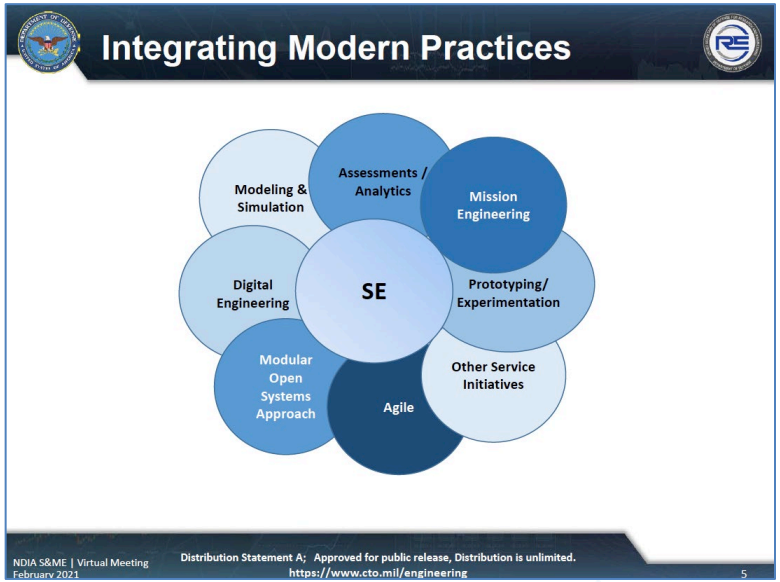
Modular Open Systems Approach (MOSA) Architecture Enables Architecture to Operations (DEVOPS)

How Does LML, 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
  - LML & Innoslate have unique interface diagrams to enable 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 near future) and provides Kanban Boards and other Agile tools
  - Can align Sprints and Epics with standard SETR events by conducting Data-Driven Reviews

Summary

- SE Modernization requires us to use 21st Century technologies to design and build the future
- We need to use these technologies to break down the walls we have erected between these modern practices
- LML and the Innoslate/Ansys digital engineering tools will help us break down those walls and create the digital ecosystem needed



It's all really about doing system engineering well!