# The Future of Systems Engineering Continued

STEVEN H. DAM, PH.D., ESEP

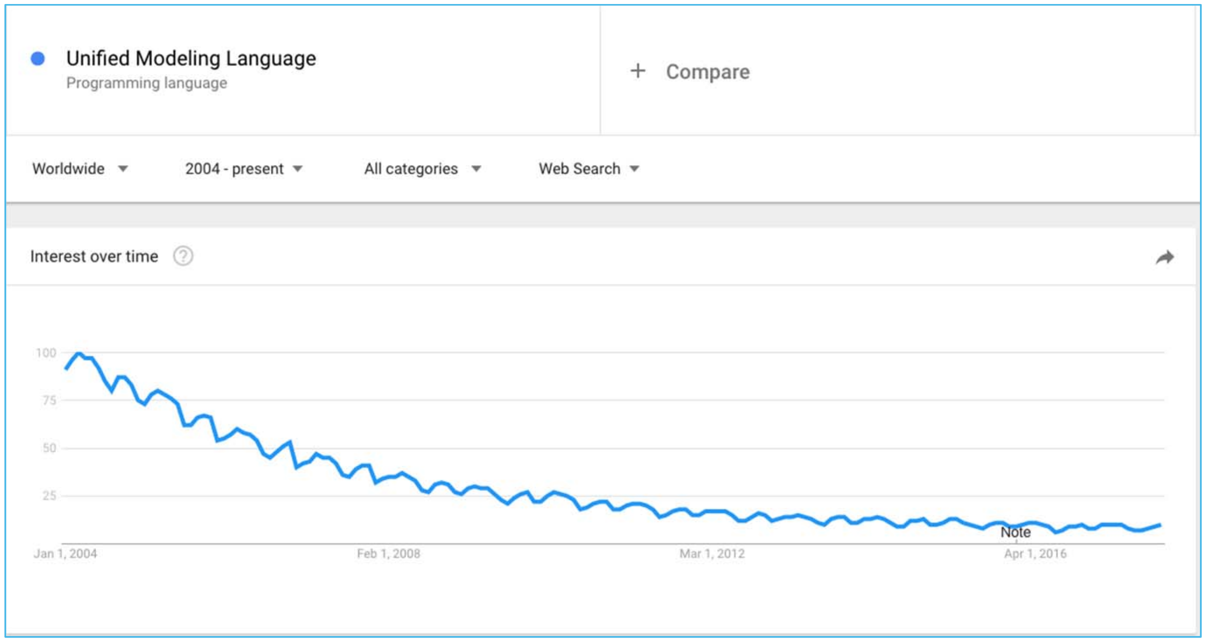
OCTOBER 23, 2019

## Agenda

- Why Isn’t SysML Enough?
- How Will Technology Improvements Enable Better SE?
- How Can We Use these New Technologies to Deal with Technology and Security Issues?
- How Can We Support the New Digital Engineering Environment Better?
- How Can We Use these New Capabilities to Enhance Systems Engineering Education and Training?

## Why Isn’t SysML Enough?

- Systems Modeling Language was developed to extend the software focused Unified Modeling Language (UML) to systems
- Interest in UML peaked in 2004
- Software developers have moved on to Agile, which requires *functional requirements*
- Both SysML and UML require experts to create and interpret
- Systems Engineering requires communications with *all* stakeholders



*From Google Trends retrieved 11/17/2017*

*If you have to be an expert in SysML’s lexicon and diagram specifications, who are you communicating with?*

## Why Isn’t SysML Enough?

- But it’s worse than just not being easy to understand
- SysML is lacking many of the programmatic pieces of information: risk, issues, decisions, schedule, cost, … as explicit diagrams or entities
- The lack of an ontology has been noted and is in the process of being developed
- But what if there was already a language that provided an ontology for SysML and filled in the missing pieces?

# LML Provides the Missing Ontology

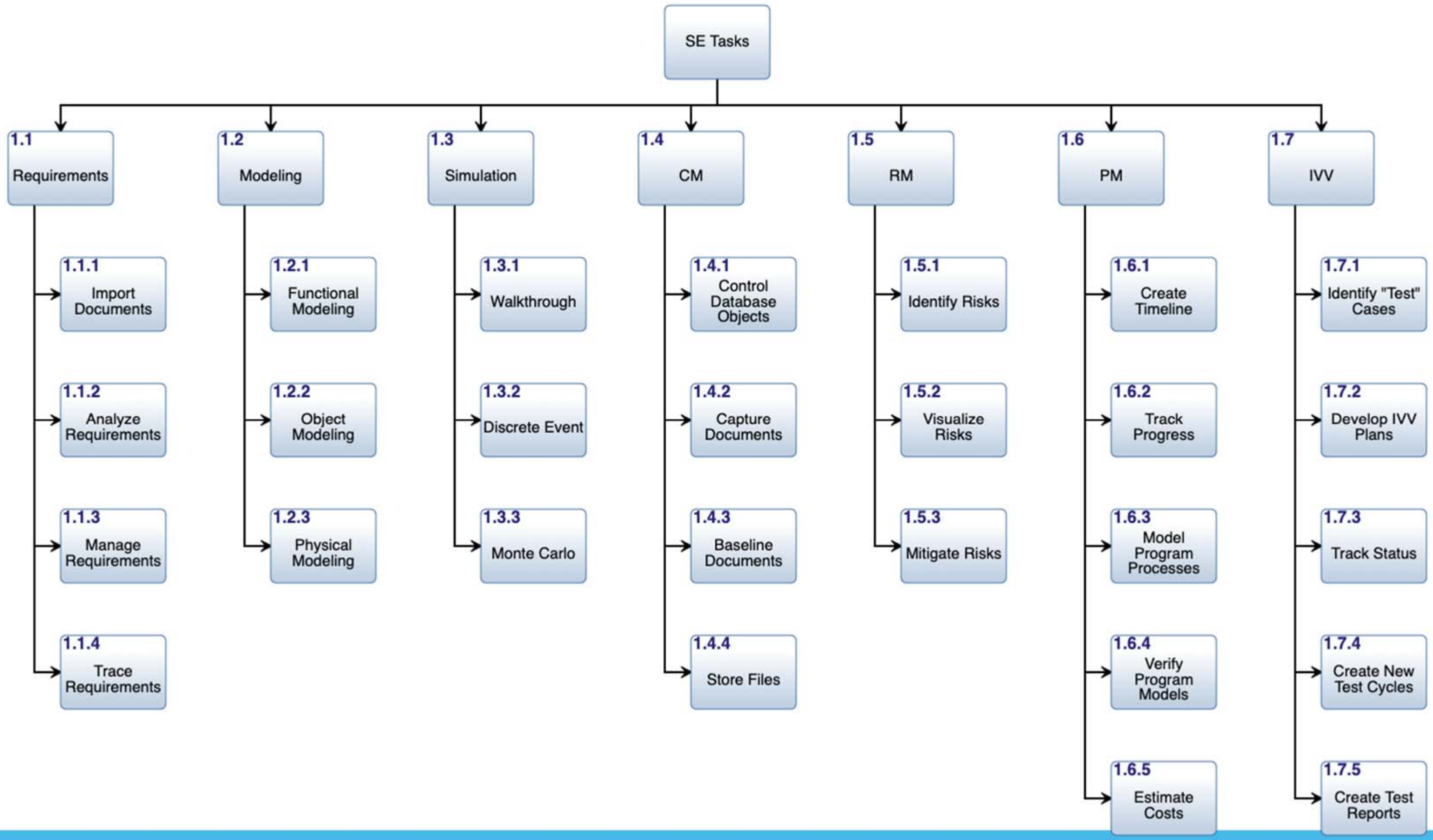
- The Lifecycle Modeling Language (LML) provides a starting point for your language
  - It’s an open standard, free for use
  - It’s designed to be the “80%” solution
  - It’s a simple language that can be extended it to meet your particular needs
- LML 1.1 was extended in 2014 to provide an ontology for SysML (www.lifecyclemodeling.org)
- LML is a *hybrid* functional and object language
- The Innoslate tool demonstrates that this ontology can generate SysML diagrams from the data, as well as other diagram types (IDEF0, N2, Class Diagrams, I2, Layer Diagram, …) as a proof of principle

## Comparing SysML and LML

- An analysis was performed by the author to compare using an ad hoc technique, implemented with MS Office tools (MS Word, MS Excel, MS PowerPoint/Visio, MS Access, MS Project) as the baseline
- The SysML-based technique included using DOORS, MagicDraw, Python/Cameo Simulation Toolkit, Risk Register, MS Project, and MS SharePoint
- The LML-based technique used only Innoslate
- A set of standard systems engineering tasks associated with systems engineer was developed, based on categories identified by Mr. Joe Elm

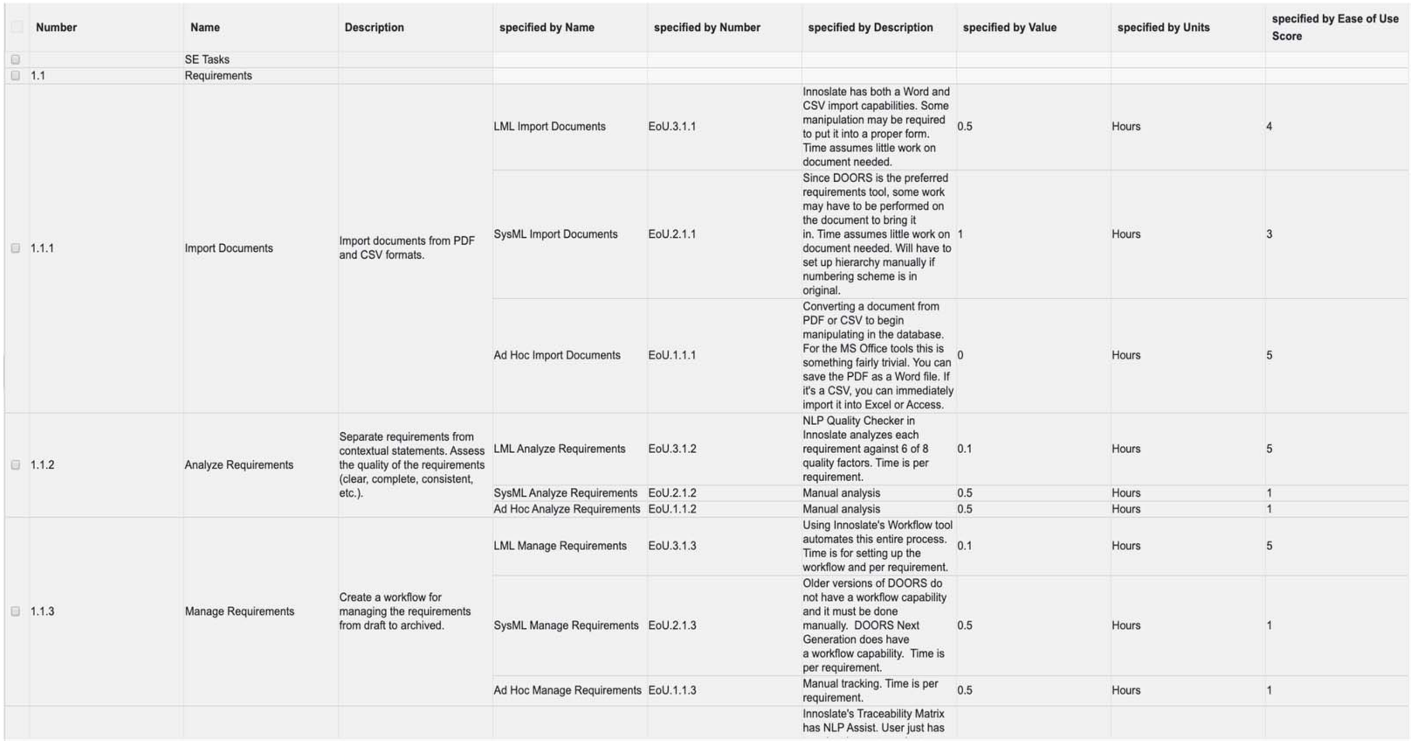
## Systems Engineering Task Decomposition

- We estimated the time it took for each task
- These estimates were then assigned an "Ease of Use" score of 0 to 5, with 5 being the least time to execute the task
- The 0 was only used if the toolset could not facilitate the task



## Comparison Chart

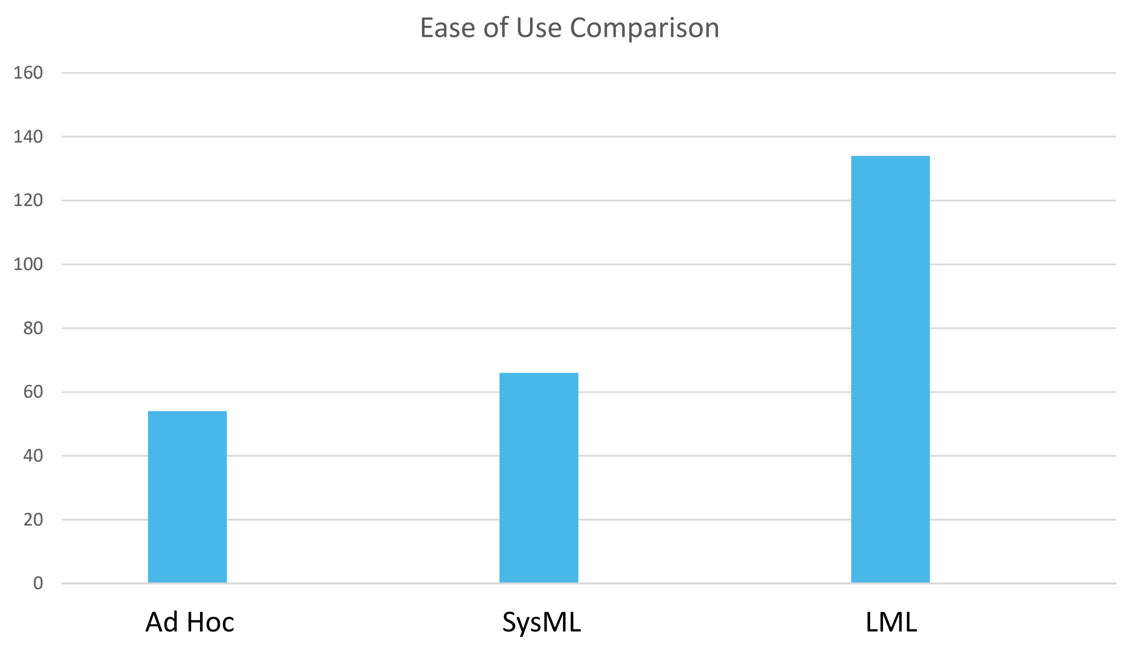
- Detailed descriptions we developed for each task
- These descriptions and scores were reviewed and adjusted by personnel with extensive experience in DOORS, MagicDraw, and other SE tools



| Number | Name | Description | specified by Name | specified by Number | specified by Description | specified by Value | specified by Units | specified by Ease of Use Score |
|---|---|---|---|---|---|---|---|---|
| | SE Tasks | | | | | | | |
| 1.1 | Requirements | | | | | | | |
| 1.1.1 | Import Documents | Import documents from PDF and CSV formats. | LML Import Documents | EoU.3.1.1 | Innoslate has both a Word and CSV import capabilities. Some manipulation may be required to put it into a proper form. Time assumes little work on document needed. | 0.5 | Hours | 4 |
| | | | SysML Import Documents | EoU.2.1.1 | Since DOORS is the preferred requirements tool, some work may have to be performed on the document to bring it in. Time assumes little work on document needed. Will have to set up hierarchy manually if numbering scheme is in original. | 1 | Hours | 3 |
| | | | Ad Hoc Import Documents | EoU.1.1.1 | Converting a document from PDF or CSV to begin manipulating in the database. For the MS Office tools this is something fairly trivial. You can save the PDF as a Word file. If it's a CSV, you can immediately import it into Excel or Access. | 0 | Hours | 5 |
| 1.1.2 | Analyze Requirements | Separate requirements from contextual statements. Assess the quality of the requirements (clear, complete, consistent, etc.). | LML Analyze Requirements | EoU.3.1.2 | NLP Quality Checker in Innoslate analyzes each requirement against 6 of 8 quality factors. Time is per requirement. | 0.1 | Hours | 5 |
| | | | SysML Analyze Requirements | EoU.2.1.2 | Manual analysis | 0.5 | Hours | 1 |
| | | | Ad Hoc Analyze Requirements | EoU.1.1.2 | Manual analysis | 0.5 | Hours | 1 |
| 1.1.3 | Manage Requirements | Create a workflow for managing the requirements from draft to archived. | LML Manage Requirements | EoU.3.1.3 | Using Innoslate's Workflow tool automates this entire process. Time is for setting up the workflow and per requirement. | 0.1 | Hours | 5 |
| | | | SysML Manage Requirements | EoU.2.1.3 | Older versions of DOORS do not have a workflow capability and it must be done manually. DOORS Next Generation does have a workflow capability. Time is per requirement. | 0.5 | Hours | 1 |
| | | | Ad Hoc Manage Requirements | EoU.1.1.3 | Manual tracking. Time is per requirement. | 0.5 | Hours | 1 |
| | | | | | Innoslate's Traceability Matrix has NLP Assist. User just has | | | |

## The Results

- The SysML-based approach was slightly better than the ad hoc approach
  - This result may explain why many people feel that the SysML toolset is not much better than using MS Office
- The LML-based approach scored significantly better than either of the other approaches
  - The primary reasons for this was the modern software design, automatic view generation, web-based implementation, and the extensive set of rule checkers and importers, many using Natural Language Parsing (NLP) and pre-trained Machine Learning (ML) algorithms



*Ease of Use Means Saving Time and Money, While Enhancing Quality*

## How Will Technology Improvements Enable Better SE?

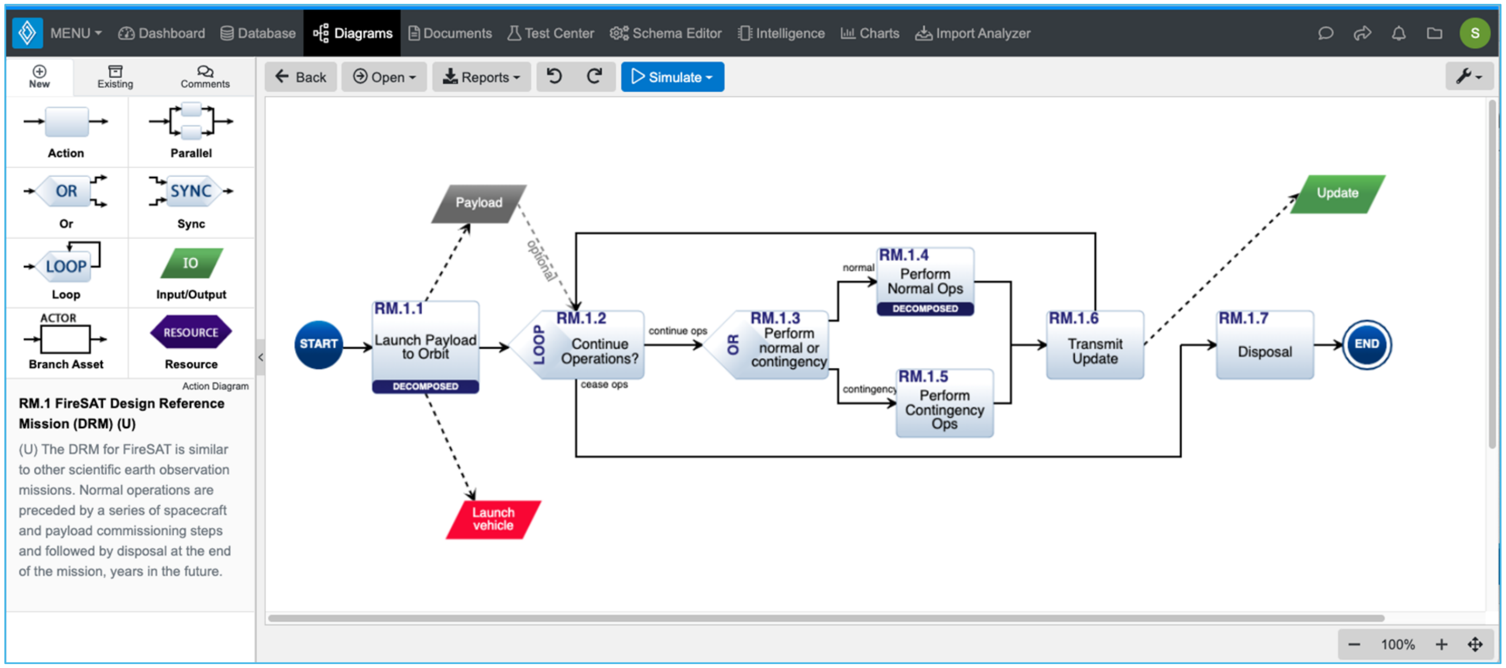
- Some emerging/available technologies of the future:
  - Cloud computing (already here!)
  - Artificial Intelligence (Natural Language Process is already here!)
  - Graph Databases (already here!)
  - Optical Computing (coming soon)
- How can they help us?
  - Cloud computing provides a means to collaborate worldwide today … SE tools need to take advantage of this capability
  - Artificial Intelligence can help us find design problems or potential problems early
  - Graph Databases enable greater storage capacity
  - Optical Computing will enable create speed of computations, thus allowing for higher fidelity modeling and simulation

## How Can We Use these New Technologies to Deal with Technology and Security Issues?

- One technology of great interest is artificial intelligence
- How can we model and predict AI behavior?
  - The same way we model and predict human behavior – behavior models
- Behavior modeling is a term used by popular tools, such as Ascent Logic’s RDD-100, Vitech CORE, and Innoslate
  - This modeling was originally developed by TRW in the late 1960s
- Behavior modeling can also support other concerns, such as cybersecurity
- LML took the behavior model one step beyond previous versions through the Action Diagram

# Modeling AI and Decisions

- The LML Action Diagram uses decision points as a special case of the Action class entity
- Innoslate added Resources explicitly to the Action Diagram



**RM.1 FireSAT Design Reference Mission (DRM) (U)**

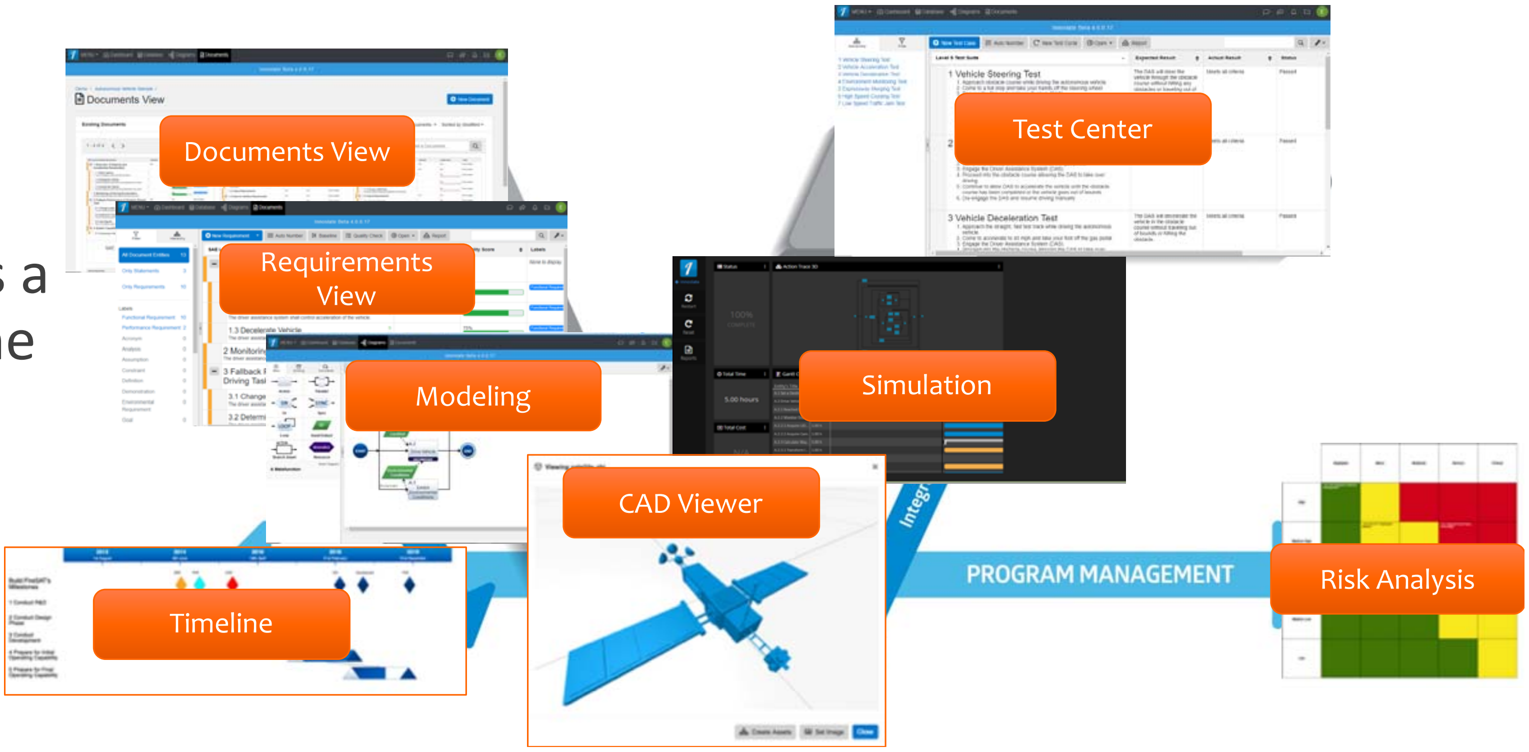
(U) The DRM for FireSAT is similar to other scientific earth observation missions. Normal operations are preceded by a series of spacecraft and payload commissioning steps and followed by disposal at the end of the mission, years in the future.

You can also model defense-in-depth and command & control more accurately

*Decision Points Can Now Be Allocated to Hardware, Software, Wetware, or aiWARE*

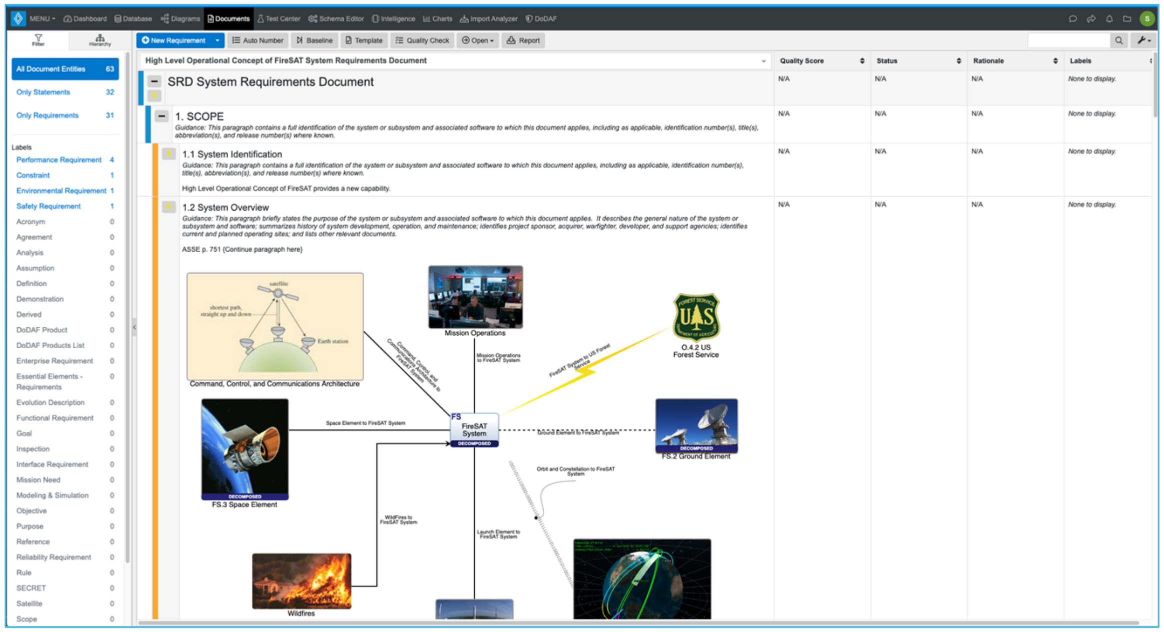
# How Can We Support the New Digital Engineering Environment Better?

- Vertically integrate with design engineering tools using APIs
- Use SE Database as a repository for all the design engineering information
- Create living documents for model-based reviews



# Living Document Example: SRD

- This SRD uses the DAU SRD Template as a basis
- The Asset Diagram provides a generator tool
- All directly related information to the entities in the diagram are used to fill-in the template
- Subsequent generations will baseline the previous version first and then create a new version of the document



MENU
Dashboard
Database
Diagrams
Documents
Test Center
Schema Editor
Intelligence
Charts
Import Analyzer
DoDAF
Filter
Hierarchy
New Requirement
Auto Number
Baseline
Template
Quality Check
Open
Report
satellite
shortest path, straight up and down
Earth station
Command, Control, and Communications Architecture
Mission Operations
O.4.2 US Forest Service
Command, Control, and Communications Architecture to FireSAT System
Mission Operations to FireSAT System
FireSAT System to US Forest Service
FS
FireSAT System
DECOMPOSED
Space Element to FireSAT System
Ground Element to FireSAT System
FS.3 Space Element
FS.2 Ground Element
Orbit and Constellation to FireSAT System
WildFires to FireSAT System
Launch Element to FireSAT System
Wildfires

| | |
|---|---|
| All Document Entities | 63 |
| Only Statements | 32 |
| Only Requirements | 31 |

Labels

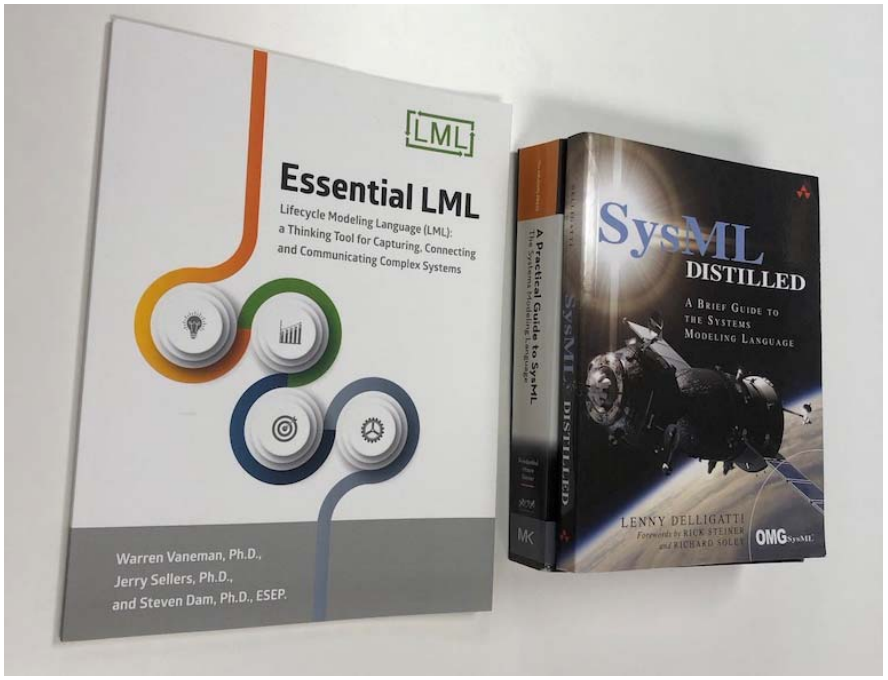
| | |
|---|---|
| Performance Requirement | 4 |
| Constraint | 1 |
| Environmental Requirement | 1 |
| Safety Requirement | 1 |
| Acronym | 0 |
| Agreement | 0 |
| Analysis | 0 |
| Assumption | 0 |
| Definition | 0 |
| Demonstration | 0 |
| Derived | 0 |
| DoDAF Product | 0 |
| DoDAF Products List | 0 |
| Enterprise Requirement | 0 |
| Essential Elements - Requirements | 0 |
| Evolution Description | 0 |
| Functional Requirement | 0 |
| Goal | 0 |
| Inspection | 0 |
| Interface Requirement | 0 |
| Mission Need | 0 |
| Modeling & Simulation | 0 |
| Objective | 0 |
| Purpose | 0 |
| Reference | 0 |
| Reliability Requirement | 0 |
| Rule | 0 |
| SECRET | 0 |
| Satellite | 0 |
| Scope | 0 |

| High Level Operational Concept of FireSAT System Requirements Document | Quality Score | Status | Rationale | Labels |
|---|---|---|---|---|
| SRD System Requirements Document | N/A | N/A | N/A | *None to display.* |
| 1. SCOPE<br>*Guidance: This paragraph contains a full identification of the system or subsystem and associated software to which this document applies, including as applicable, identification number(s), title(s), abbreviation(s), and release number(s) where known.* | N/A | N/A | N/A | *None to display.* |
| 1.1 System Identification<br>*Guidance: This paragraph contains a full identification of the system or subsystem and associated software to which this document applies, including as applicable, identification number(s), title(s), abbreviation(s), and release number(s) where known.*<br>High Level Operational Concept of FireSAT provides a new capability. | N/A | N/A | N/A | *None to display.* |
| 1.2 System Overview<br>*Guidance: This paragraph briefly states the purpose of the system or subsystem and associated software to which this document applies. It describes the general nature of the system or subsystem and software; summarizes history of system development, operation, and maintenance; identifies project sponsor, acquirer, warfighter, developer, and support agencies; identifies current and planned operating sites; and lists other relevant documents.*<br>ASSE p. 751 {Continue paragraph here} | N/A | N/A | N/A | *None to display.* |

*Creating Living Documents Means We Never Have to Go of MS Office, Until We Want to Make It Shelfware*

## How Can We Use these New Capabilities to Enhance Systems Engineering Education and Training?

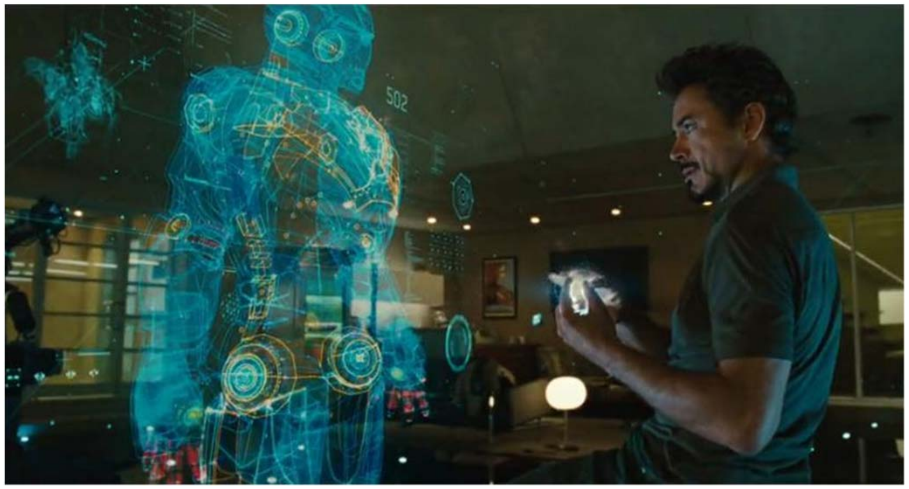
- We currently support over 200 Colleges and Universities Around the World
- LML provides a much easier to understand and use as a means of teaching key systems engineering and MBSE concepts to students
- Innoslate is free for Academic use and includes support for anyone using the tool
- Innoslate also provides a platform to extend SE research


LML
Essential LML
Lifecycle Modeling Language (LML): a Thinking Tool for Capturing, Connecting and Communicating Complex Systems
Warren Vaneman, Ph.D., Jerry Sellers, Ph.D., and Steven Dam, Ph.D., ESEP.
A Practical Guide to SysML
SysML Distilled
A Brief Guide to the Systems Modeling Language
Lenny Delligatti
OMG SysML

*Supporting Academia Is Critical to the Future of Systems Engineering*

## Summary

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



*Come to the Future of Systems Engineering Panel session Thursday at 3:30PM for more!*