

Data-Driven Test & Evaluation

October 18, 2023

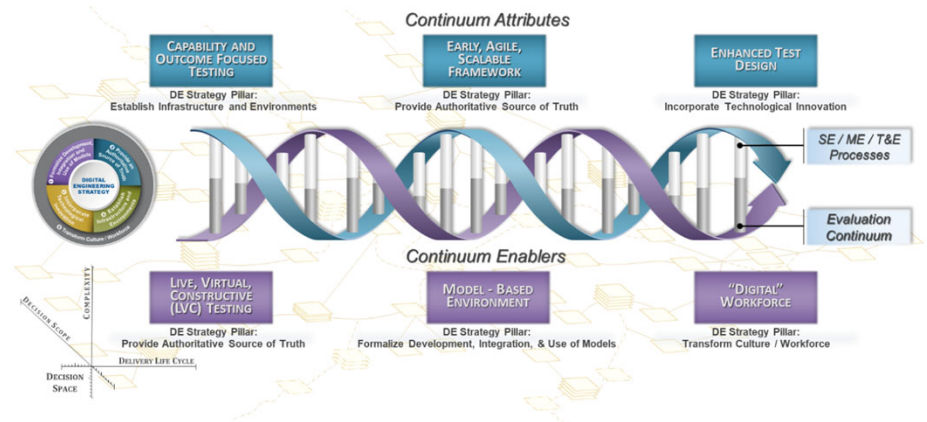
Presented by Steven H. Dam, Ph.D., ESEP

Agenda

- What is the T&E Continuum?
- Why is an ontology needed?
- Why choose LML?
- How does it aid in integrating T&E with digital engineering?
- How can we implement this language?
- How does this provide for the T&E continuum?

What is the Test and Evaluation as a Continuum Concept?

- Recently, Mr. Christopher Collins and Mr. Kenneth Senechal published an article in the ITEA Journal* that proposed:
 - “a critical change in how test and evaluation (T&E) support capability delivery.”
- The new approach uses T&E continually throughout the lifecycle to inform development and fielding decisions



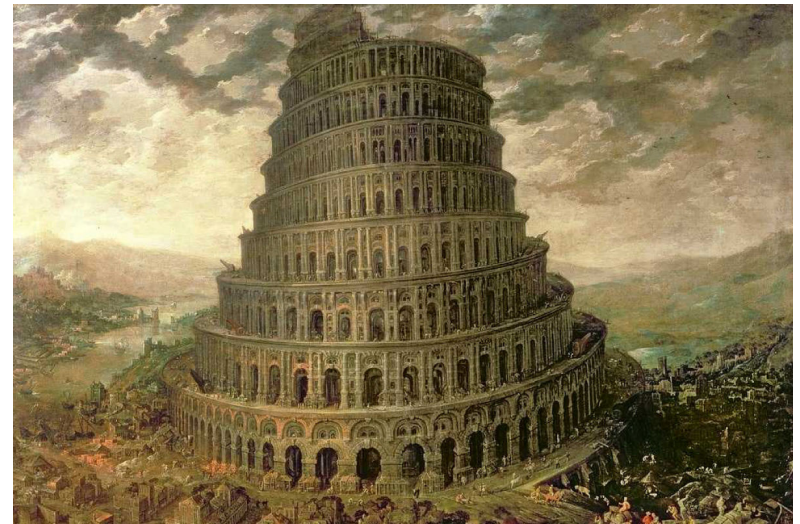
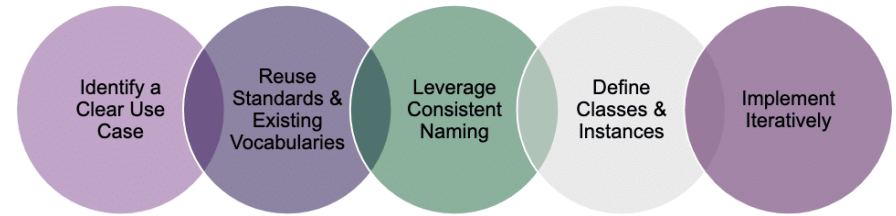
But how can we implement this concept?

We need to understand how the data collected during T&E influences decisions early in the lifecycle ... we need an ontology.

*<https://itea.org/journals/volume-44-1/test-and-evaluation-as-a-continuum/>

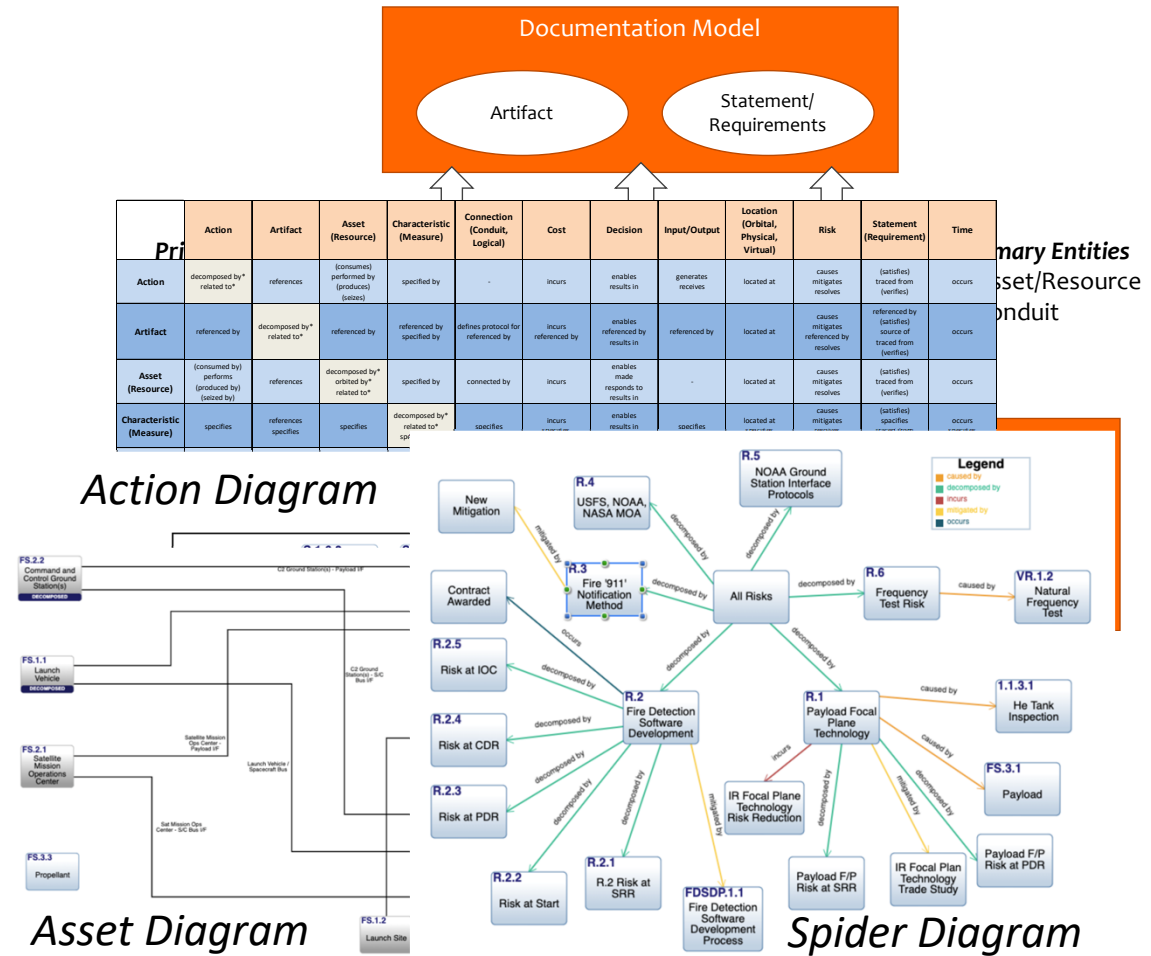
Why is an ontology needed?

- An ontology (taxonomy + relationships) provides a language for communications
- Developing one is a lot of work, but important when you want people to work together
- Each discipline has its own language
- Having different languages can cause confusion



Why choose LML?

- Lifecycle Modeling Language, first published in 2013, provides a simple ontology that covers systems engineering and program management
- The current version, v1.4, was released October 20, 2022
- Each version includes extensions, in the form of Appendices including extensions for DoDAF and SysML
- LML v1.5 will provide a mapping to SysML v2 and BPMN



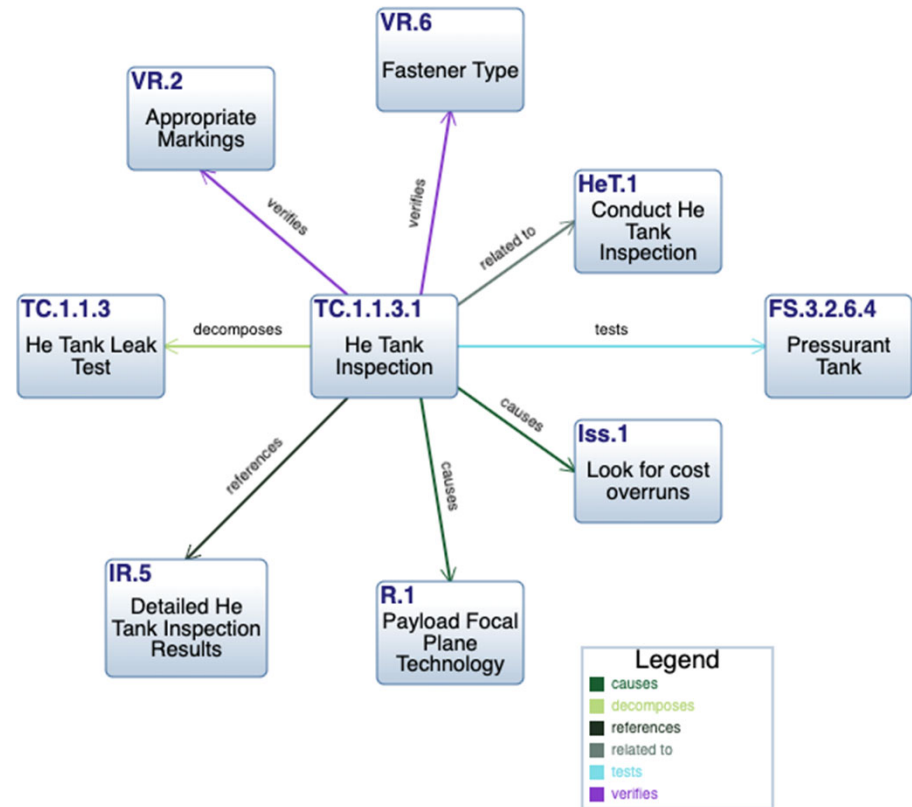
Why choose LML?

- In March 2022, version 1.2 was released and included an extension for V&V
- This extension came from users of the language proving it out on a number of programs over many years
- Test Suites act as containers for Test Cases, which are V&V tasks

Class	Parent	Description
Test Case (Verification Event)	Action	A Test Case entity specifies a verification/validation task, as well as its expected and actual results.
Test Suite	Artifact	A Test Suite entity specifies a container for Test Cases.
Verification Requirement	Statement	A Verification Requirement entity specifies what is required to confirm that a requirement is satisfied

How does it aid in integrating T&E with digital engineering?

- The V&V extension for LML has relationships to requirements and other aspects of the lifecycle model
- You can add your own relationships as well to aid in searching or reporting



How can we implement this language?

- Innoslate[®] uses LML as it's primary language
- Innoslate[®] acts as a test bed for the language and research tool for digital and systems engineering
- Innoslate[®]'s Test Center provides a dashboard to roll up results

The screenshot displays the Innoslate Test Center interface. On the left, a sidebar shows the test hierarchy for 'System Acceptance Test Suite'. The main area contains a table of test results with columns for Entity, Expected Result, Actual Result, Status, and Status Roll-Up. A detailed view of the '1 System Acceptance Test' is shown, including its description and a flowchart diagram.

Entity	Expected Result	Actual Result	Status	Status Roll-Up
1 System Acceptance Test Final Test to ensure system meets all requirements	Meets all acceptance criteria		Failed	1 5 1 2
1.1 Propulsion Module Acc...	Meets all propulsion module acceptance criteria	See File full results	Failed	1 2 2
1.1.1 Propellant Tank Leak Test	Less than 2 parts/million detected	1.785 parts/million detected	Passed	1
1.1.1.1 Propellant Tank Inspection Include inspecting appropriate markings	All seams appear complete	All seams inspected using x-ray are solid and complete. Tank markings correct.	Passed	Passed
1.1.2 Propulsion Module Struct...	Must pass "shake and bake" test	Meets all vibration and temperature requirements.	Passed	Passed
1.1.3 He Tank Leak Test	Less than 10 parts/million He		Failed	

But building test cases early in the lifecycle costs money!

How can we implement this language? (continued)

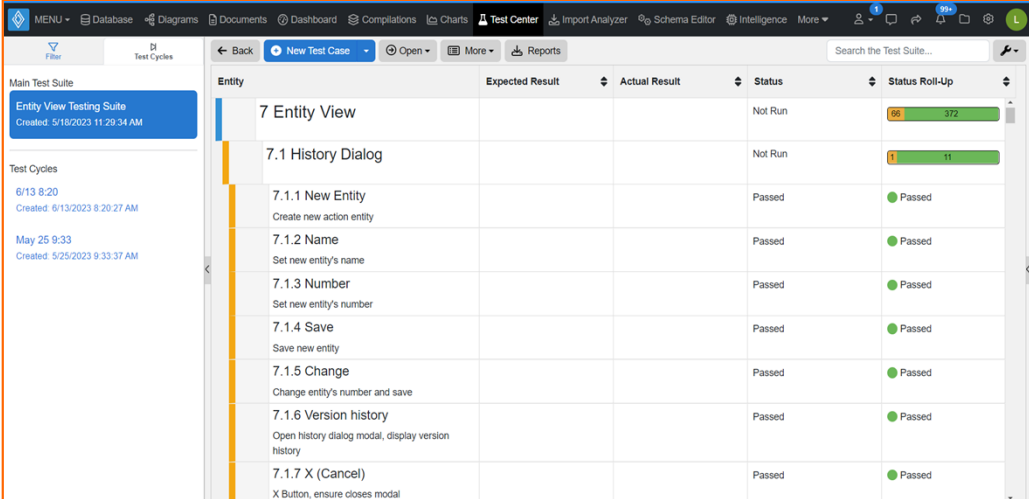
- To create test cases from requirements has always been a difficult task to perform early in the lifecycle because it adds more time (and cost)
- Recently, ChatGPT was added to Innoslate and may provide a means to speed up the test case development from requirements
- Providing a verifiable requirement to ChatGPT can result in a suggested test case

Entity	Rationale	Quality Score	Labels
VR.1.2 Natural Frequency Test The test shall conduct a modal survey (sine sweep) of the vehicle using a vibration table.		89%	Test X Verification Req. X
VR.2 Appropriate Markings The appropriate markings on all system structural components shall be verified by inspection. The inspection shall determine if axes and identifications are properly indicated.	The verification shall be considered successful if all structural components are properly marked.	67%	Derived Inspection Verification Requirement
VR.3 Altitude Accuracy The accuracy of the altitude determination system estimates shall be verified by analysis. The analysis shall use Monte Carlo simulations of expected sensor accuracy, plus noise, to determine statistical distribution error.	The analysis shall be considered successful if the predicted error is less than or equal to 0.01 degrees (3 sigma).	78%	Analysis Verification Requirement
VR.4 Battery GSE Charge Display Battery charge ground support equipment (GSE) state of charge display shall be verified by demonstration.	The demonstration shall be considered successful if the state of charge is displayed.	67%	Demonstration Verification Requirement
VR.5 State of Charge The demonstration shall show that state of charge is indicated when connected to a representative load.		67%	No labels to display.
VR.6 Fastener Type The verification shall be considered successful if all interface fasteners are 4-40 in size made from stainless steel.		56%	Inspection Verification Requirement
VR.6.1		67%	No labels to display.

With this application, we are just scratching the surface of what Generative AI might be able to do for us.

How does this provide for the T&E continuum?

- Customers have interfaced Innoslate® with LabView® for dynamic hardware-in-the-loop testing
- SPEC Innovations has recently interfaced Innoslate® with Selenium® for software test tracking
 - Uses Innoslate's Java SDK to make changes to entities (editing, creating, deleting, etc.)



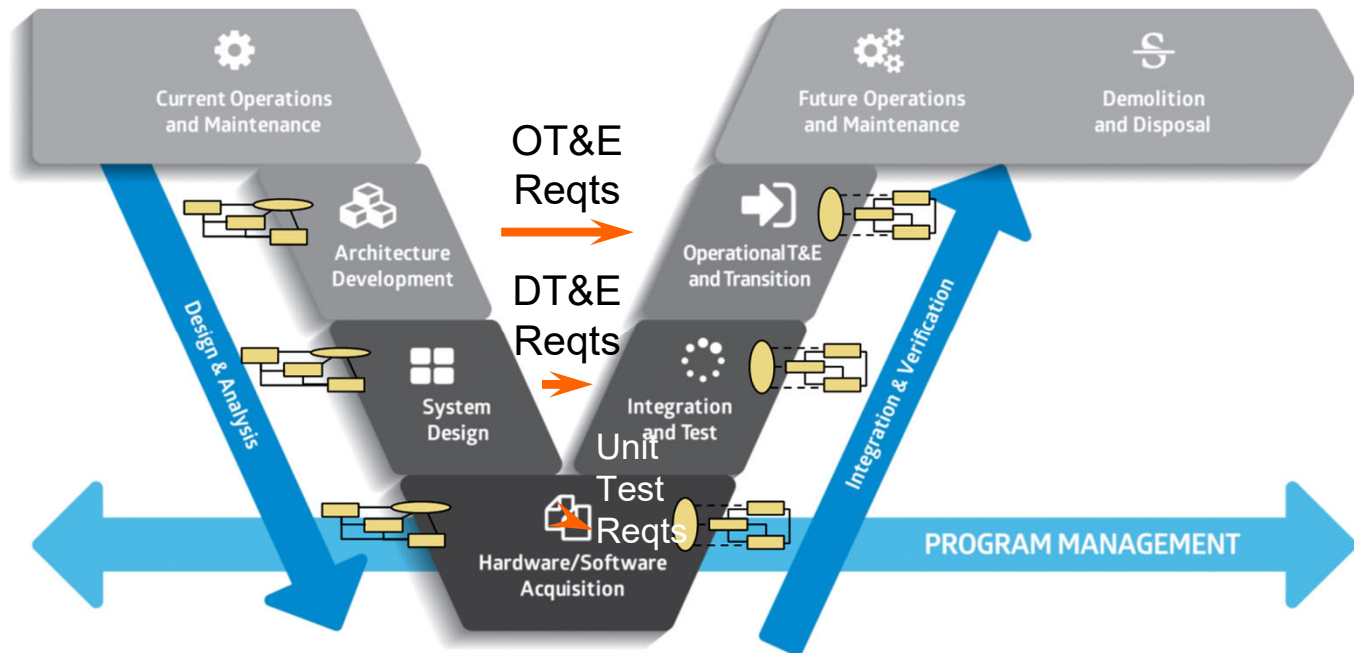
The screenshot shows the Test Center interface with a table of test cases. The table has columns for Entity, Expected Result, Actual Result, Status, and Status Roll-Up. The test cases are listed as follows:

Entity	Expected Result	Actual Result	Status	Status Roll-Up
7 Entity View			Not Run	0% 372
7.1 History Dialog			Not Run	1% 11
7.1.1 New Entity Create new action entity			Passed	Passed
7.1.2 Name Set new entity's name			Passed	Passed
7.1.3 Number Set new entity's number			Passed	Passed
7.1.4 Save Save new entity			Passed	Passed
7.1.5 Change Change entity's number and save			Passed	Passed
7.1.6 Version history Open history dialog modal, display version history			Passed	Passed
7.1.7 X (Cancel) X Button, ensure closes modal			Passed	Passed

With this integration, we have created a seamless interface between systems engineering and software engineering!

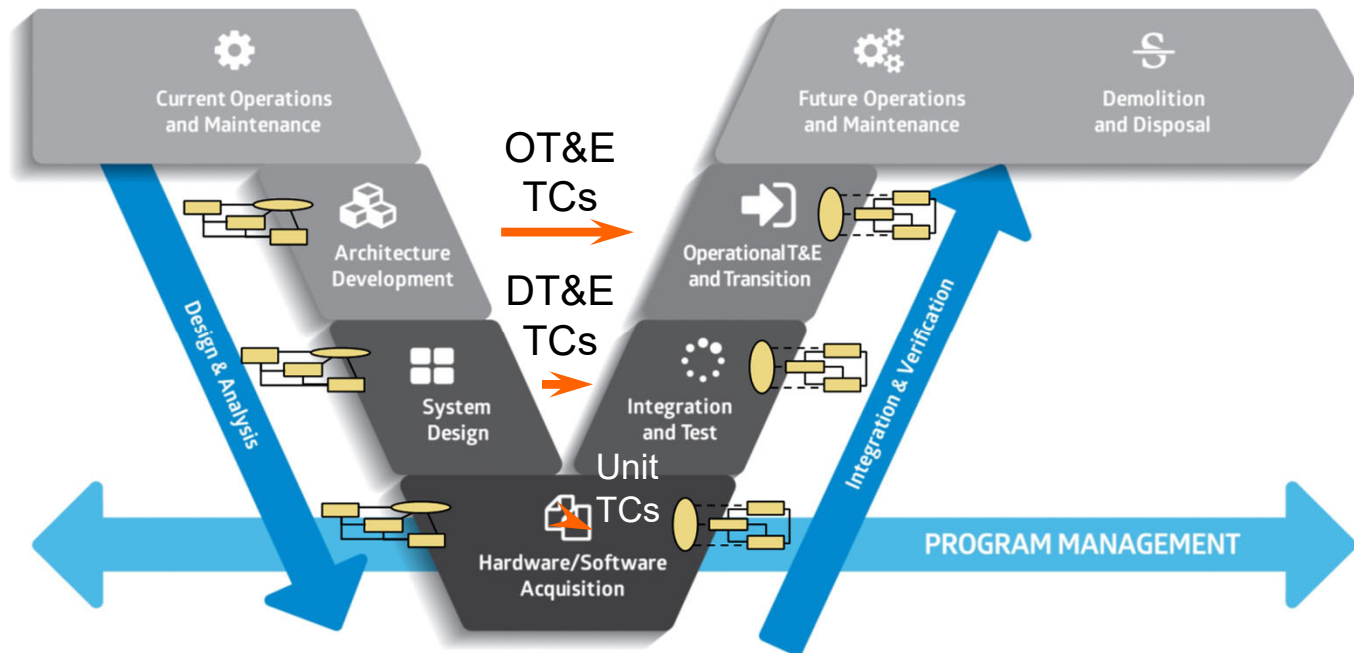
How does this provide for the T&E continuum?

The primary purpose of the “V” model was to remind systems engineers that they have a responsibility to create the verification requirements for the V&V activities



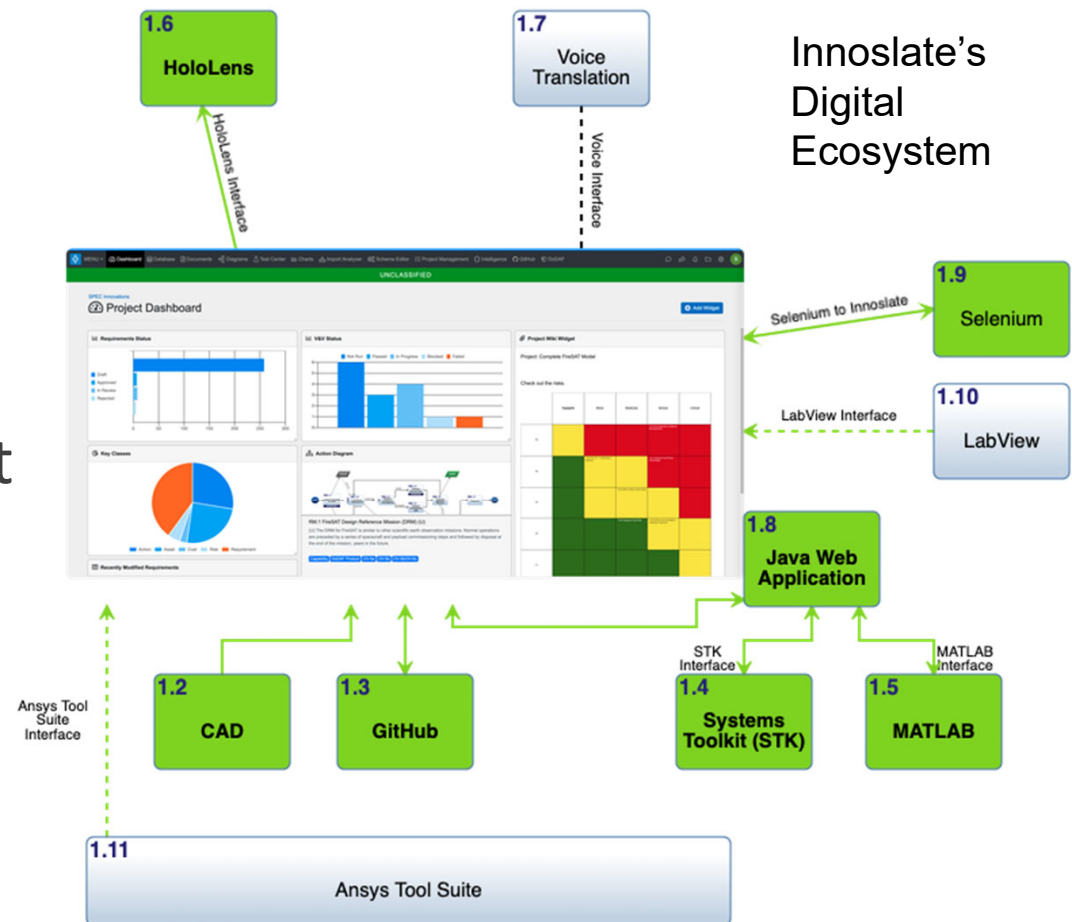
How does this provide for the T&E continuum?

- Now with Test Center we can (should?) create the test cases directly
 - Reduces time of developing separate VRs
 - Provides detailed test plans early in the lifecycle
- Will follow current software engineering best practice



How does this provide for the T&E continuum?

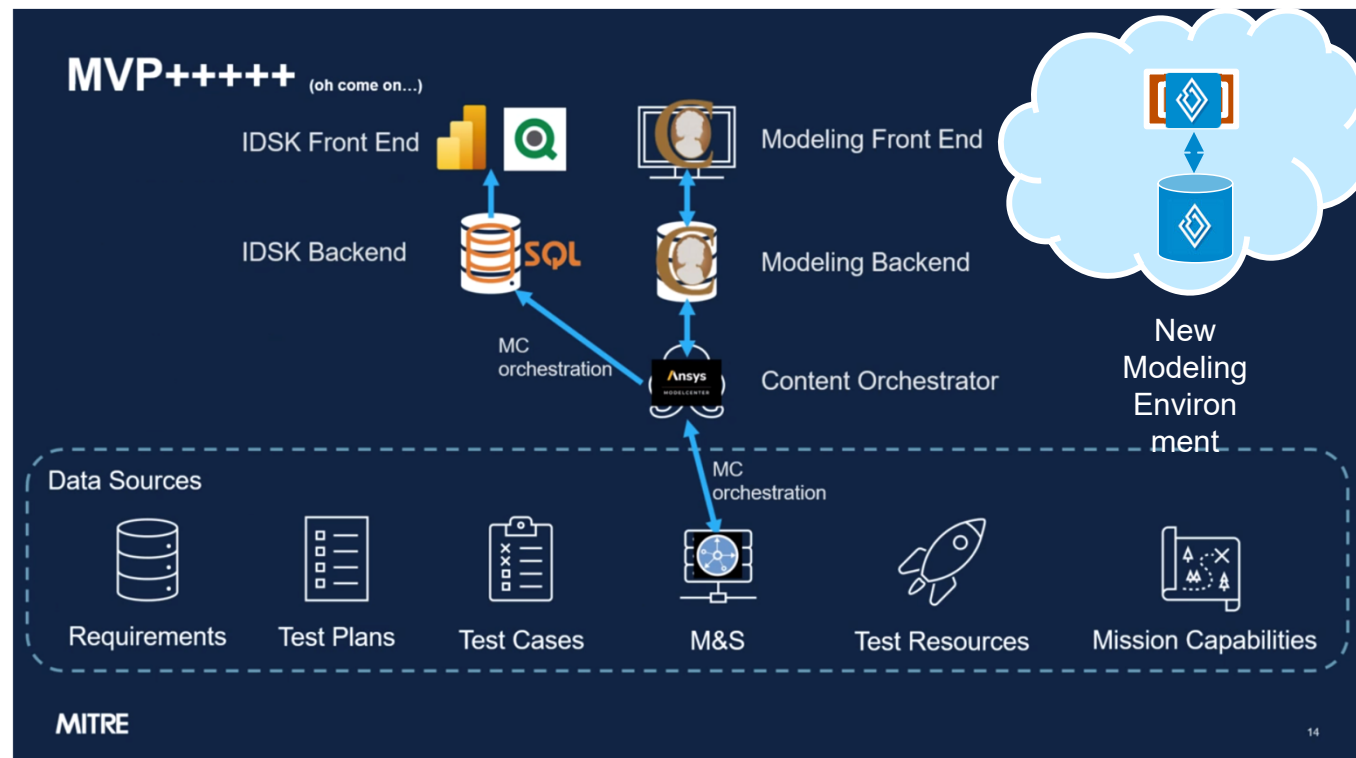
- Another thing we can do earlier in the process is conduct simulation to predict test results
- The use of simulation for this purpose is not new, but the modern tools provide the capability to conduct modeling and simulation together



How does this provide for the T&E continuum?

- Reduces “vendor lock” for the modeling environment by adding another tool
- Cloud-native and USA owned and developed
 - provides a perfect alternative modeling environment
- Adding this path to the Hack-a-Thon only requires that Innoslate integrate with Ansys ModelCenter

Proposed Addition to next Integrated Decision Support Key (IDSK) Hack-a-thon



Summary

- LML provides an ontology that connects the dots between systems engineering and test and evaluation today
- It has been tested by numerous organizations inside and outside DoD
- The language can be implemented by any tool with an open database schema
- It's time to move beyond diagrams and focus on the data ... that's where true interoperability can be attained