AI-Enabled Mission Engineering

Michael Pennock, Jesse Ponnock, Trac Bannon, Judith Dahmann

The MITRE Corporation

AI4SE & SE4AI Research Application Workshop 2025

September 17-18, 2025

MITRE

Project Team

Intentionally interdisciplinary team of engineers and scientists: Data Scientists, Mission, Simulation, Human Factors, Systems, and Software Engineers

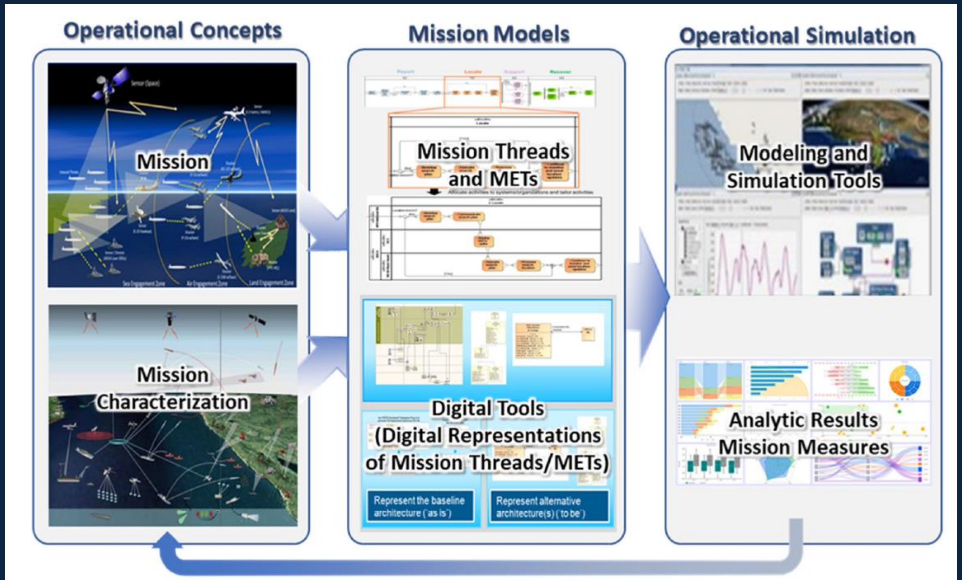
- Amy Marsh
- Benjamin Yam
- Bridget Musselman
- Christina Thompson
- Faith Morgan
- Gabriela Parasidis
- Jerry Schweiger
- Jesse Ponnock
- Joshua Lee
- Judith Dahmann
- Justin Lieffers
- Katja Sednew
- Mia Barger
- Michael Briggs
- Michael Pennock
- Nicholas Kinberg
- Patty McDermott
- Rob Scheible
- Saina Shibili
- Trac Bannon
- William Macke

Government Sponsor Advocate: **Mr. Elmer Roman**, DASD for Mission Integration

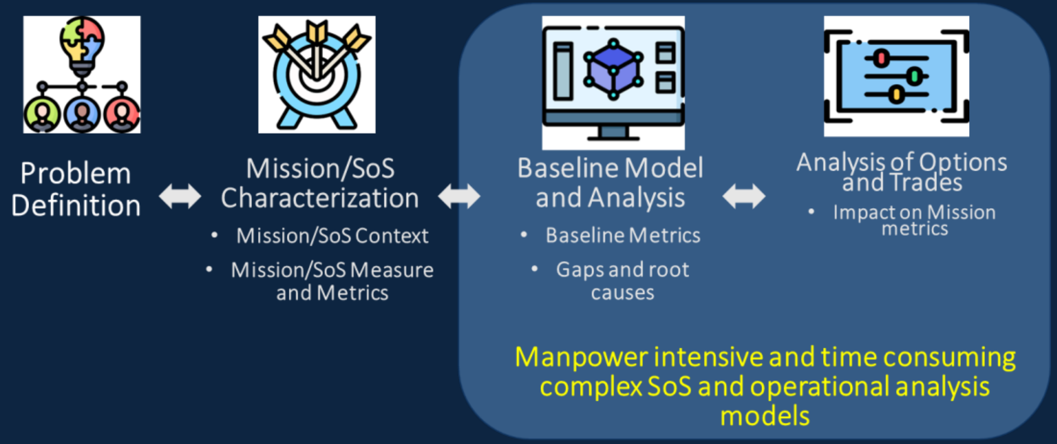
MITRE

Project Summary

- Mission engineering (ME) is essential for the US DoD to make the right investments to ensure warfighter success
- Today, mission engineering takes longer than the time available to make an investment decision
- Our project applies AI to key steps in the ME workflow to improve the speed and quality of mission engineering work
- Immediate impact for defense sponsors including OUSD, CAPE, and components that are increasingly looking to take a mission focused analytic approach to decision making



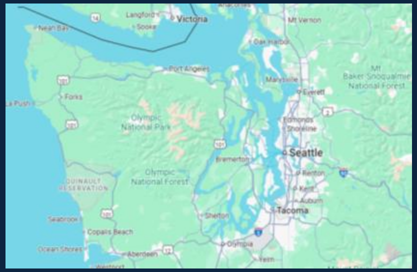
Source: DoD MEG 2.0, 2023

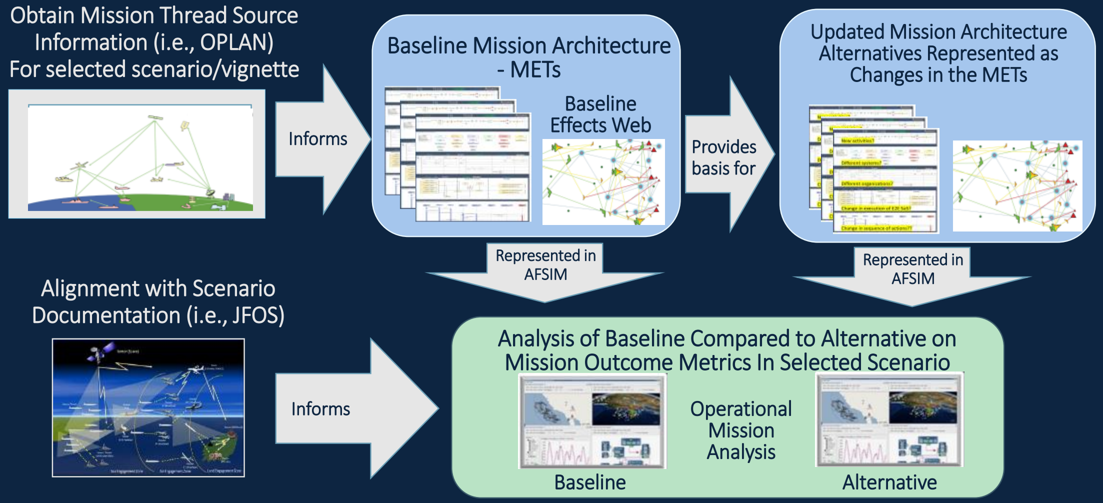


DoD's ME Challenge

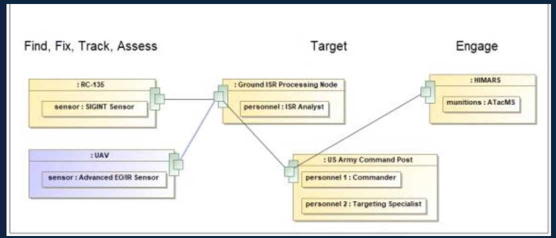
Digital engineering models are central to the mission engineering workflow

Defense of Seattle*



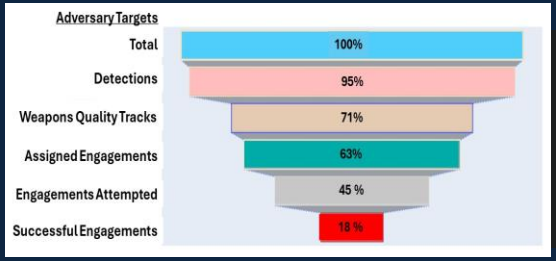


Model the SoS in an architecture tool



Model and analyze the relevant portions of the SoS using an operational simulation tool

* Source: Dahmann, D. J. S., & Parasidis, G. I. (2024). Mission Engineering. *The ITEA Journal of Test and Evaluation*, *45*(3).




```
class CodeExample():
    def printStatement(self):
        print('Hello World!')

def main():
    classEx = CodeExample()
    classEx.printStatement()
if __name__ == "__main__":
    main()
```

File:Hello World in Python.png - Wikimedia Commons

Our success at digital engineering has created a new challenge: large, complex digital models are difficult to review and extract valuable insights

MITRE

Today’s Real-World Challenge

An actual situation that occurred on a mission engineering project*

Project task: Ensure that the architecture and simulation models are mutually consistent

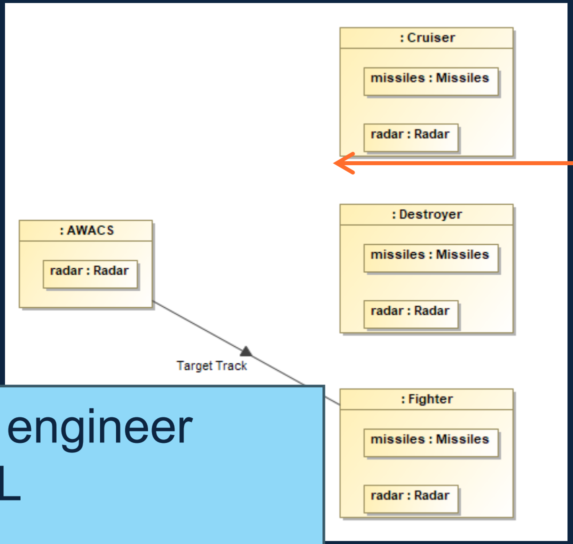
Operational Simulation Code

```
class CodeExample():
    def printStatement(self):
        print('Hello World!')

def main():
    classEx = CodeExample()
    classEx.printStatement()
if __name__ == "__ma
    main()
```

File:Hello World in Python.png - Wikimedia Commons

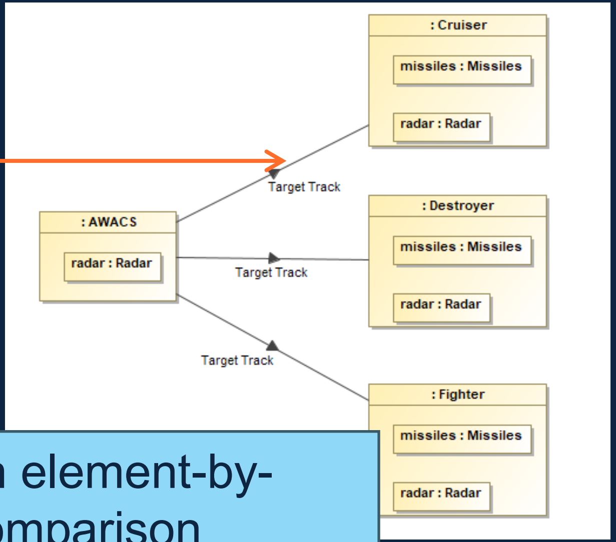
Model of Simulation



Manually reverse engineer code into a SysML representation.

Mismatch

Mission architecture model



Perform an element-by-element comparison

*Screenshots are notional

Model complexity makes this task expensive

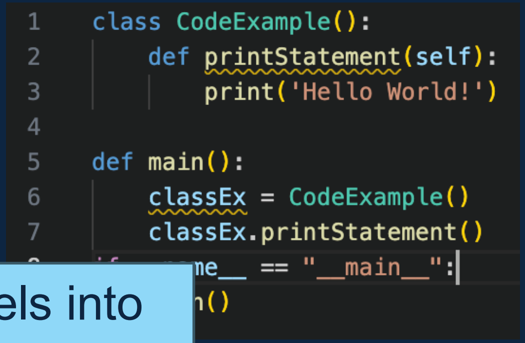
- A typical mission engineering thread includes 57 nodes, 18 control elements, 83 flow connections
- A typical ME architecture contains: ~130000 elements

MITRE

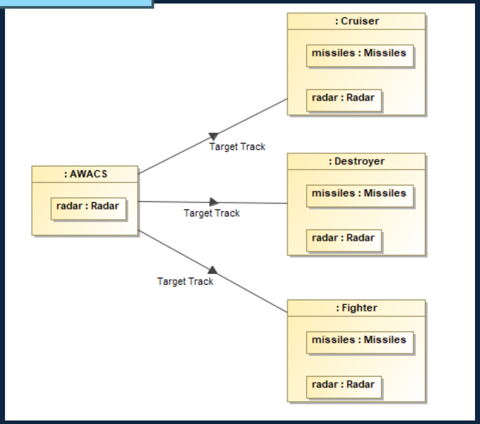
Tomorrow’s Solution Workbench

Instead, the engineer will have a conversation with the model*

File:Hello World in Python.png - Wikimedia Commons



Load models into AI system



Question: Where does the AWACS pass its target tracks?

Answer:

- In the SysML model:
  The AWACS provides t
  to the Cruiser, Destroy
  and Fighter
- In the simulation model:
  The AWACS provides tracks to the Fighter

Use the AI solution to interrogate the models

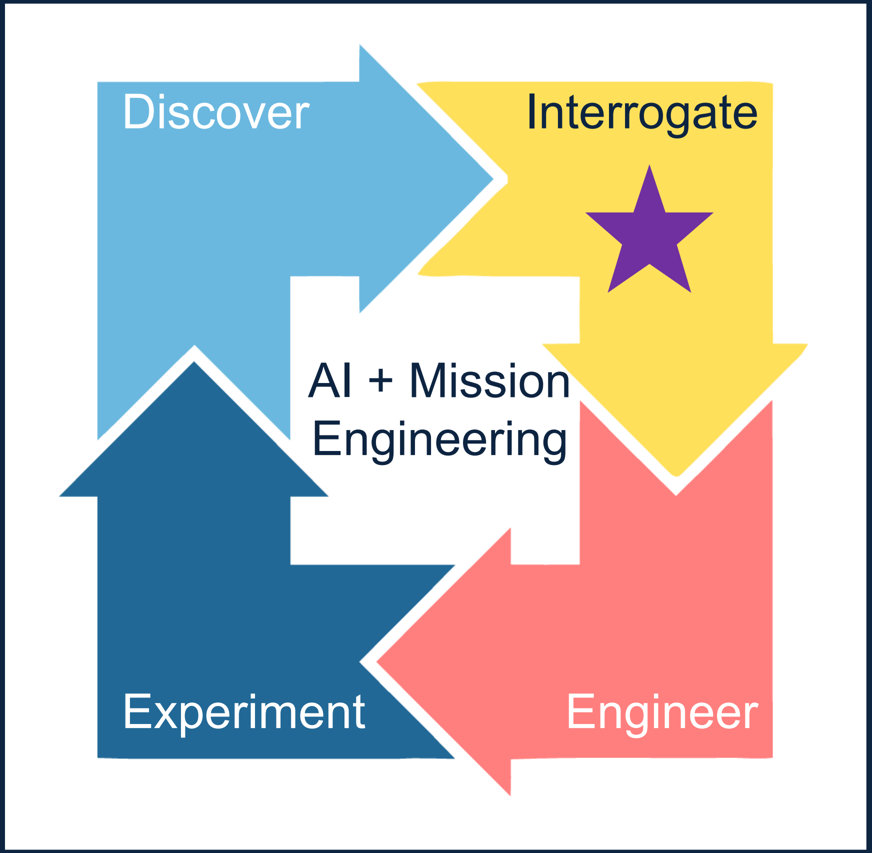
*Screenshots are notional

Answering a simple question about the model will no longer take weeks

MITRE

Problem Statement

- Enable knowledge extraction from digital mission engineering assets using AI
  - CY 25 focus on SysML models, AFSIM models, and supporting documents
- Outputs: ME model interrogation testbed - first step in ME/AI workbench
  - MITRE mission engineers
  - Sponsor resource
  - Industry and university outreach
- Opportunities to extend what we are doing to other areas of mission engineering



MITRE

Research Overview

- Research Hypothesis: Mission engineers teamed with augmented large language models can improve the speed and quality of mission engineering work

- CY 25 Research questions:
  - Can fine-tuning and/or augmentation of selected LLMs enable mission engineers to accurately search and summarize technical artifacts?
  - How much time and effort does application of these AI-enabled capabilities save?
  - Does this application improve the quality of the mission engineering models?
- CY25 Research Approach:
  - Focus testbed on RAG and Graph RAG interrogation of AFSIM and SysML ME models
  - Quantify impacts on ME using human subject experiments

MITRE

Analysis of ME Workflows have Yielded Target Use Cases

		User questions		
		What are the contents of a model?	Which elements of the model would need to be adapted or restructured?	Is the model consistent with other models and artifacts?
Users	Experienced Mission Engineer	- Assess progress and completeness before delivery - Extract data from models to use in a new model	- Understand what needs to change to architect and analyze alternatives to the baseline - Identify gaps in externally provided models	- Check architecture and simulation models for mutual consistency
	New ME Team Member	- Learn organization and components of project models	- Learn which parts the model are relevant to work assignments	-Check model changes for consistency with references
	Reviewer/Sponsor	- Check if model contains required components	- Make recommendations for model improvement	-Check if model is consistent with authoritative sources

Yellow text = CY 25 Priority

MITRE

Three Integrated Efforts

Mission Engineering Work Analysis

- Analyze User Needs
- Collect relevant artifacts
- Generate ME tasks and questions

SysML Interrogation

- Analyze SysML model data
- Experiment and solution
- Implement SysML test capability

AFSIM Interrogation

- Analyze AFSIM model data
- Experiment and solution
- Implement AFSIM test capability

Human Subjects Experiments

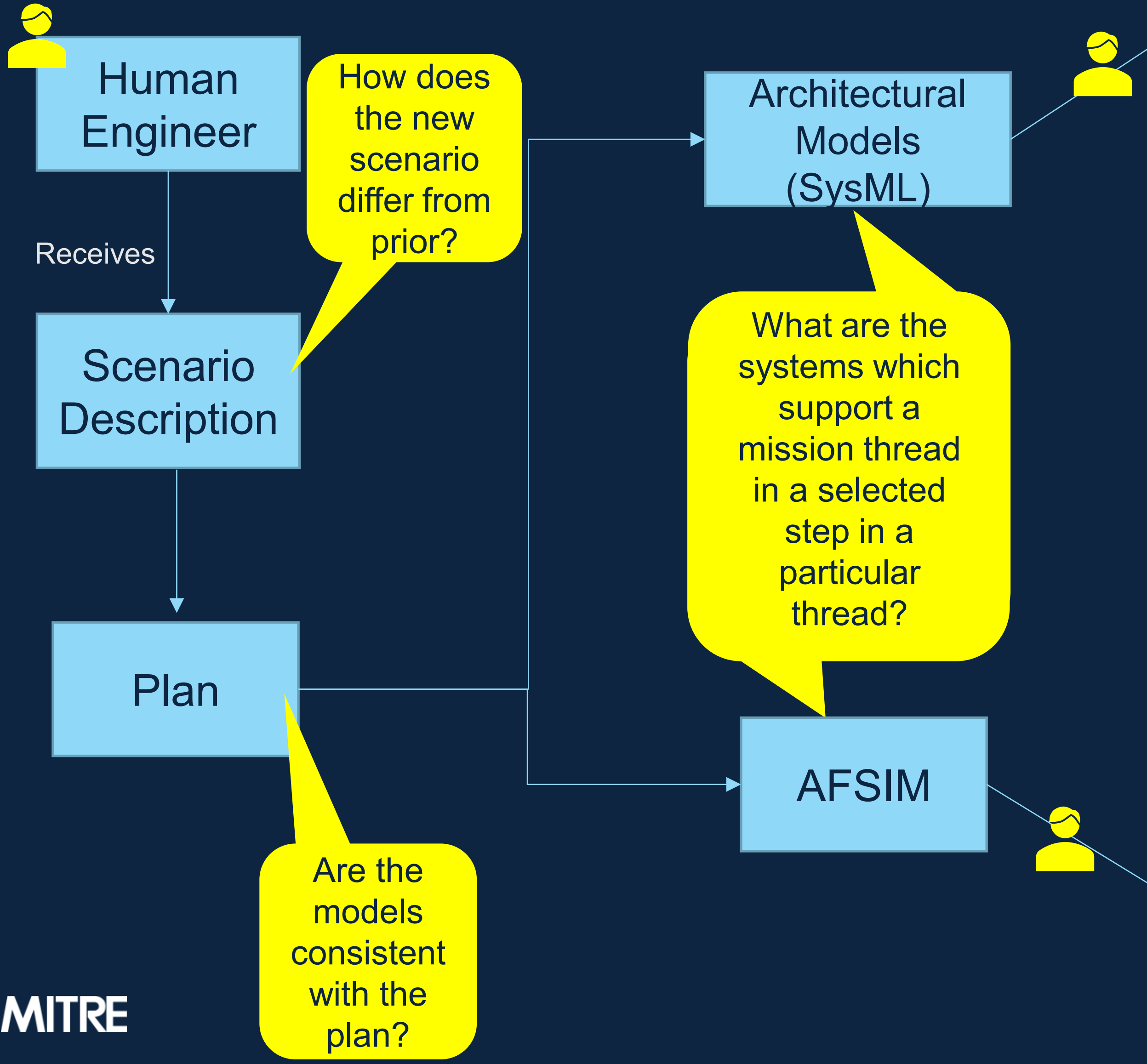
Model Interrogation Testbed

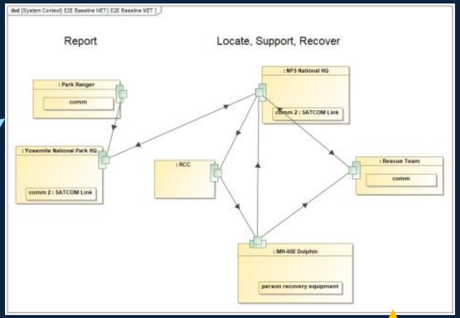
- Model integration workflow
- User starter kit
- Curated data corpus
- Automated ME test battery

MITRE

How a Mission Engineer Thinks

What are the contents of a model?





GraphRAG

Prompt ID	Proposed Decomposed Prompt	Dependency	Abstraction	contextual Variability	Tot Score	Complexity	Target Solution
Q006-A	What are the <<systems>> assigned to perform activities in <<thread_name>> at <<step_number>>?	2	1	2	5	M	GraphRAG
Q006-B	Which <<mission elements>> are allocated to <<functions>> at <<step_number>> in <<thread_name>>?	2	2	2	6	M	GraphRAG
Q006-C	What <<functions>> are required at <<step_number>> in <<thread_name>>, and which <<systems>> support them?	3	2	2	7	H	GraphRAG
Q006-D	Are there any <<systems>> missing or undefined at <<step_number>> of <<thread_name>>?	2	2	3	7	H	GraphRAG
Q006-E	In <<AFSIM scenario>>, what <<platforms>> are simulated at <<step_number>> in <<thread_name>>?	2	1	2	5	M	AFSIM RAG
Q006-F	How do <<systems>> interact at <<step_number>> of <<thread_name>> (data flows, triggers, dependencies)?	3	3	2	8	H	GraphRAG

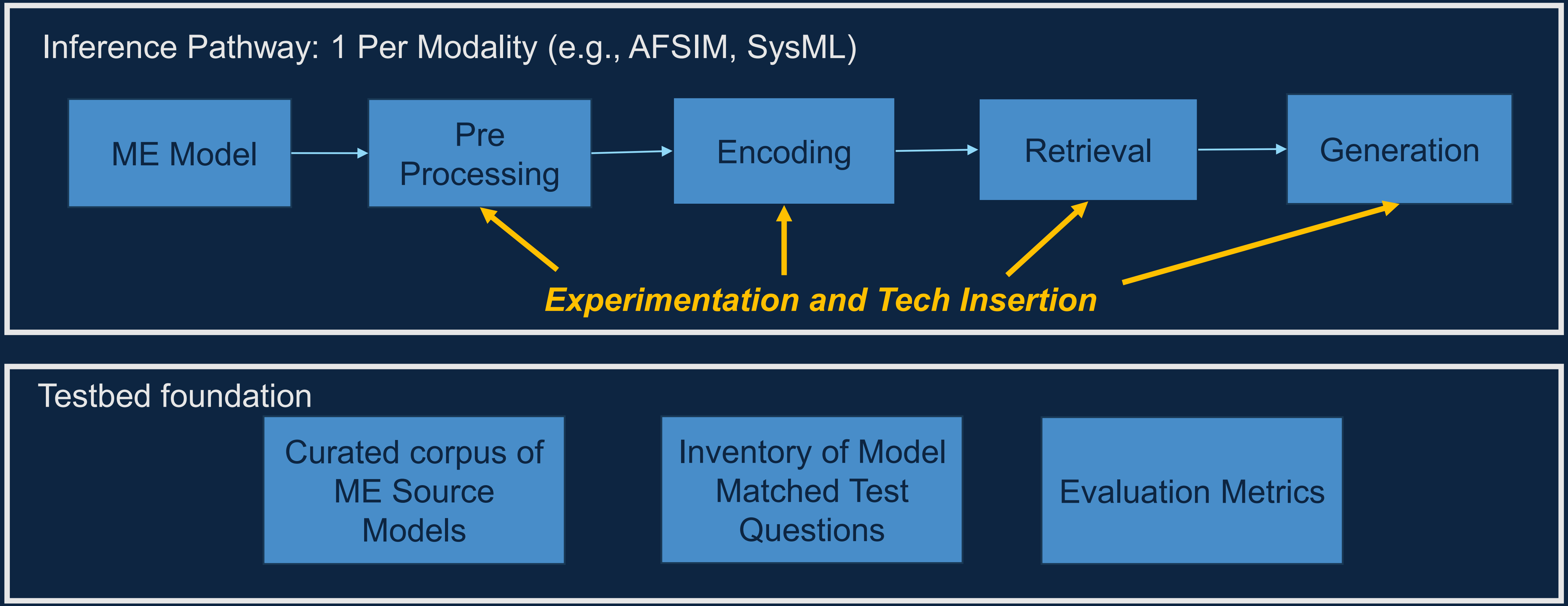
AFSIM Script Files

RAG

MITRE

Testbed Overview

Testbed Architecture



What have we learned so far?

ME Discipline

- Tabletops and analysis intended to inform the design of experiments revealed characteristics of ME
  - Highly dependent on the experience of individual
  - Highly iterative
- Impacts how we use AI to support the workflow

ME Data

- Architecture stored in proprietary data models
- Wide variety of modeling approaches
- Sparse edge information in ME architectures
- Domain knowledge implicit
- Uncommon syntax and terminology

Applying AI to ME

- Graphs proving more useful for analysis than the original tools
- GNN and community summarization on graphs show potential to address challenging aspects of ME data
- AFSIM interpretation is sensitive to model size
- Sparsity of documentation in ME models is challenging

Findings have implications for how we perform ME in the future beyond just applying AI

MITRE

Challenges and Opportunities

ME Discipline

Low level of standardization

Niche discipline

Nature of the work tends to push classification up

Small subject pool

ME Data

Vendors make data extraction difficult

Complex data structures with inconsistent and redundant labeling

Variations in organization challenge pre-processing

Applying AI to ME

Models not trained on defense specific data

Sparse edge information in ME architectures

Data sensitivity limits size of corpus and LLM options

Limited computational resources

More AI is not the solution to all of these challenges. There are implications for ME and SoSE

MITRE

Next Steps

- This year
  - Finalize mission engineering testbed AI-pipeline
  - User-based testing of AI solutions

- Next year
  - Experimental use on actual mission engineering projects
  - Expand the workbench with additional capabilities

MITRE

Backup

Illustrative Questions

Facts about Model Elements (Activities; Systems)	Which mission threads are included in this model?
	What are the key steps in each mission thread?
	Is a particular system of interest included in the scenario?
	What types of systems are included in the scenario?
	How many instances are there of a specific system and where are these located?
Facts about Relationships/ Elements	What are the systems which support a mission thread in a selected step in a particular thread?
	What communications systems are supported by a selected system?
	Can system X communicate with system Y? Directly or indirectly? What are the intermediary nodes if indirectly?
	What communications systems support this connection?
Facts about Relationships/ Threads	Which systems support multiple threads?
	What threads have any single system links?
	How many different paths exist for a selected platform weapon system?
	How many inputs and outputs are supported by each command-and-control node in the model?
	If system X is removed from the mission architecture, what mission threads and systems are impacted?
Reasoning About Threads and Relationships	How is a selected thread represented in each model? What systems support each step in the thread in the system model? In the AFSIM model? Are these aligned?
	For a thread with a single node supporting a step, what options exist to add systems to strengthen the thread.
	If we add a new system of a particular type, how could this be integrated into the architecture?
	Do the threads and supporting systems in a selected thread align with the description in the supporting scenario documentation?

MITRE