



**International Council on Systems Engineering**  
*A better world through a systems approach*

# A Knowledge Graph Framework for Failure Analysis and Prevention

Madison Urquhart and Janet Six, Ph.D.

Tom Sawyer Software



# Today's Agenda

- Global & Problem Context
- How Can It Be Done Better?
- Human-systems Integration
- Knowledge Graphs for Failure Detection and Analysis
- Detection of Emerging Failure
- Future Work

# Global Context

- Space is booming
- Data is becoming more complex
- AI is quickly becoming more powerful
- How can we use this to reduce failures?

# Problem Context

- Failure can get expensive fast
- Pushing your system to the limits characterizes it in a clearer manner
- Rigid parameters will not always catch issues before they arise, and neither will a person

# How can it be done better?



# Knowledge Graphs Support Earlier Failure Detection

- Model the system
- Humans and computers work together to determine failure types and detect failures
- Integrated data visualization, analytics, AI, and machine learning
  - Field data
  - Digital twins
  - Post situation analysis
  - Simulations
- Trend detection
- Communication of key results and findings

# Knowledge Graphs Support Earlier Failure Detection

- Failure location
- Failure type(s)
- Concurrent or related failures
- Root cause
- Cataloging of failure types and causal conditions
- Discover emerging failure patterns and take corrective actions

**“Composite AI represents the next phase in AI evolution. It involves combining AI methodologies — such as machine learning, natural language processing and knowledge graphs — to create more adaptable and scalable solutions.”**

## **Gartner**

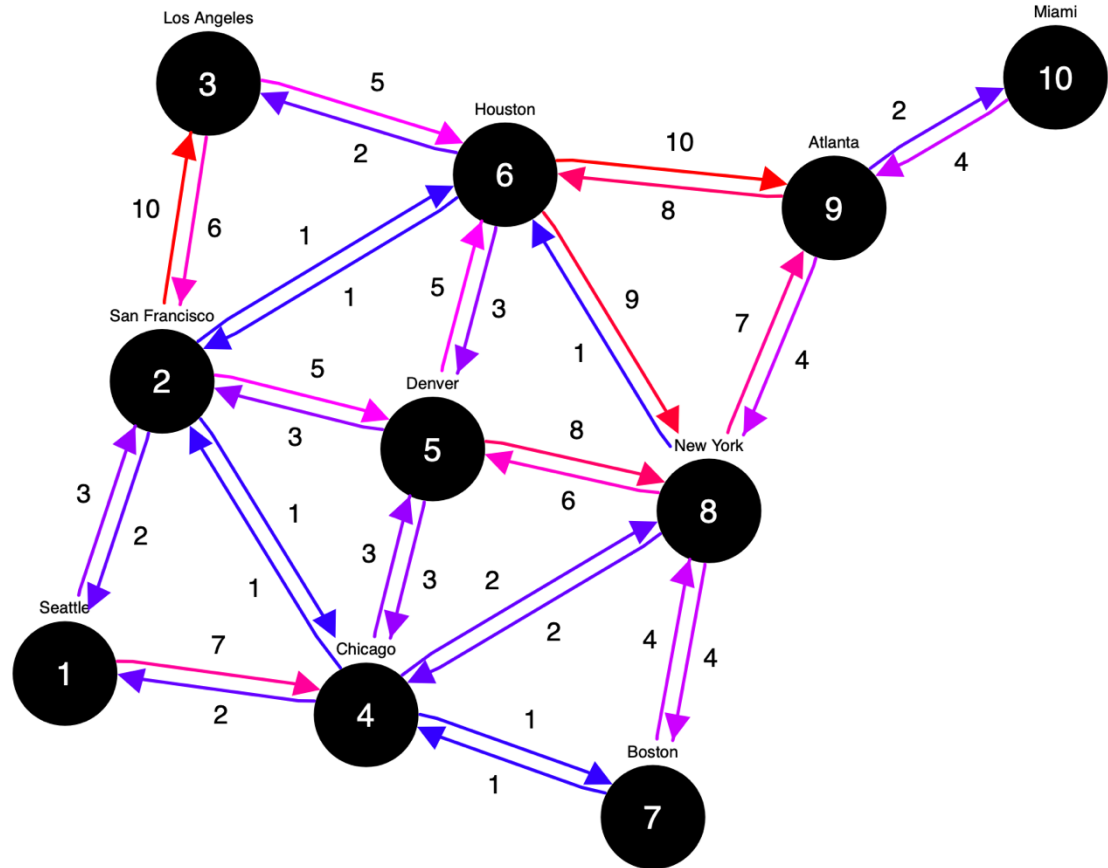
“Explore Beyond GenAI on the 2024 Hype Cycle for Artificial Intelligence”



# Knowledge Graphs

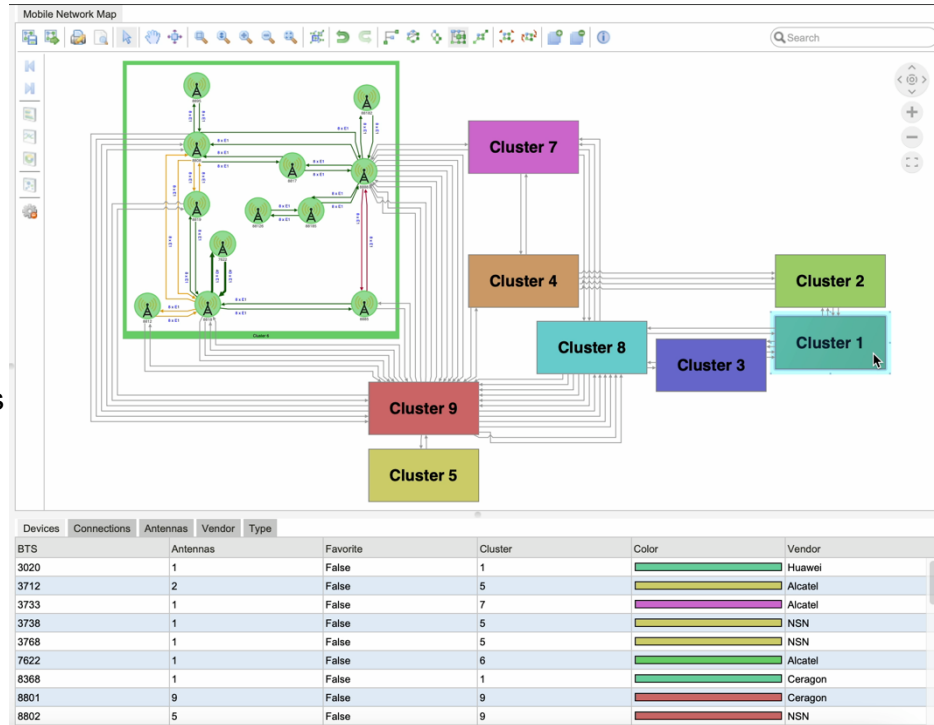
# Graphs

- Nodes (or vertices)
- Relationships (or edges)



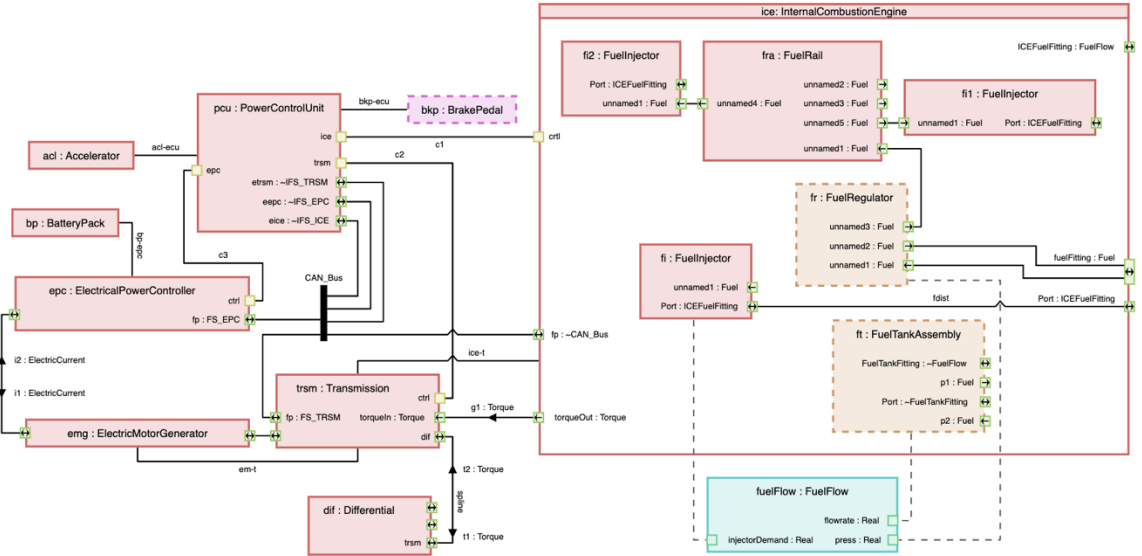
# Graphs

- Nested Drawings
  - Graph inside a node
  - Graph inside an edge
  - Edges to nodes in other nested drawings

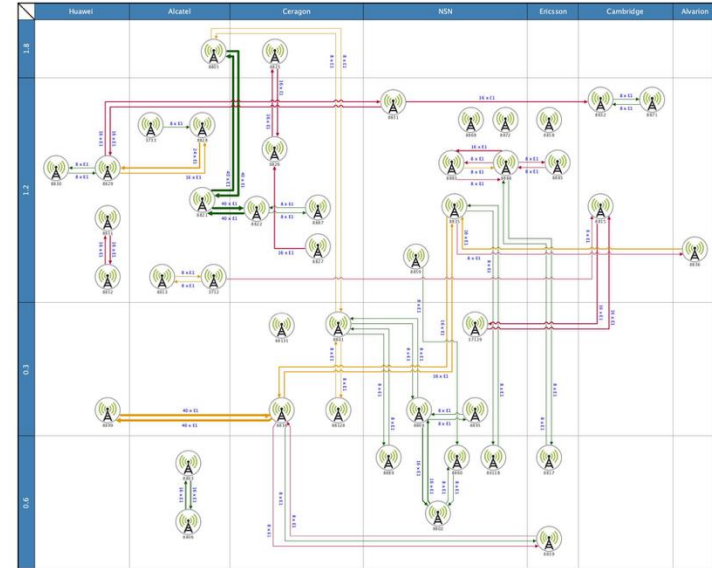


# Graphs

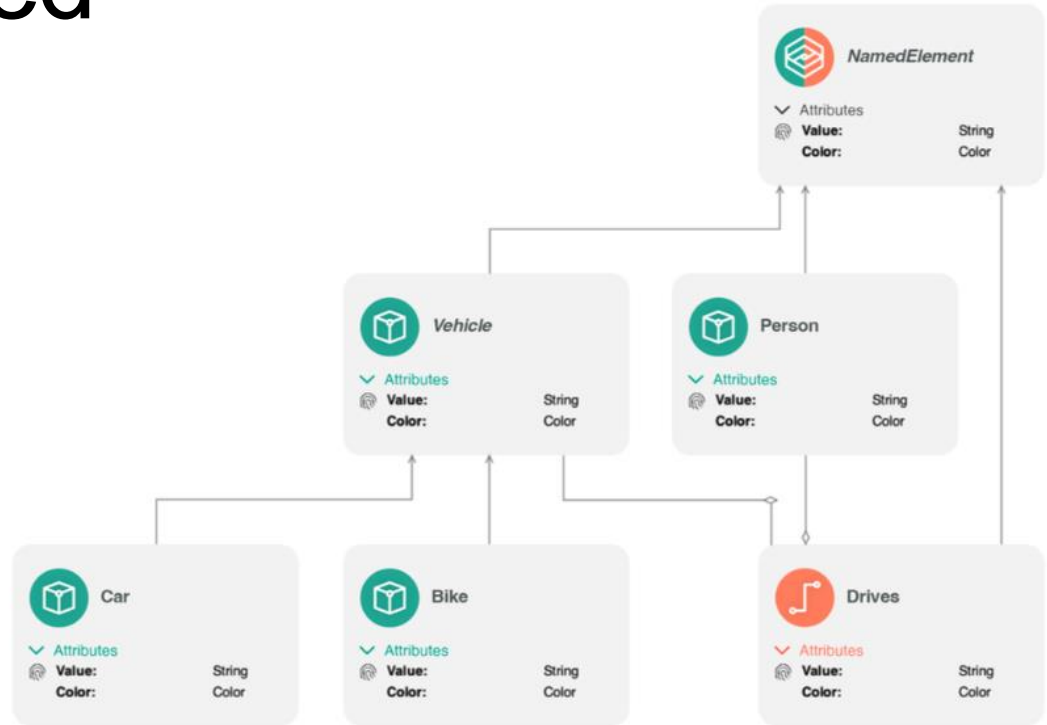
- Labels
  - Node labels
  - Edge labels
  - Connector labels
  - Edge decoration labels



- Two-Dimensional Swimlanes



# Knowledge Graphs have an expected structure

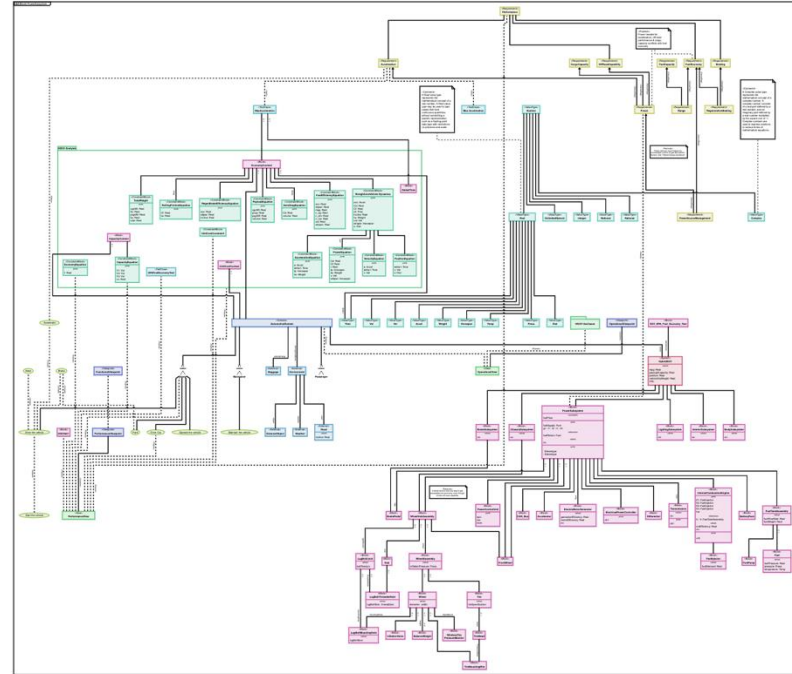


# Knowledge Graphs for Modeling and Analysis

- Support users in modeling their systems as graphs
- Apply graph visualizations and analyses to discover areas of interest in their data
- Apply graph visualizations and analyses to optimize their systems

# Knowledge Graphs for Modeling and Analysis

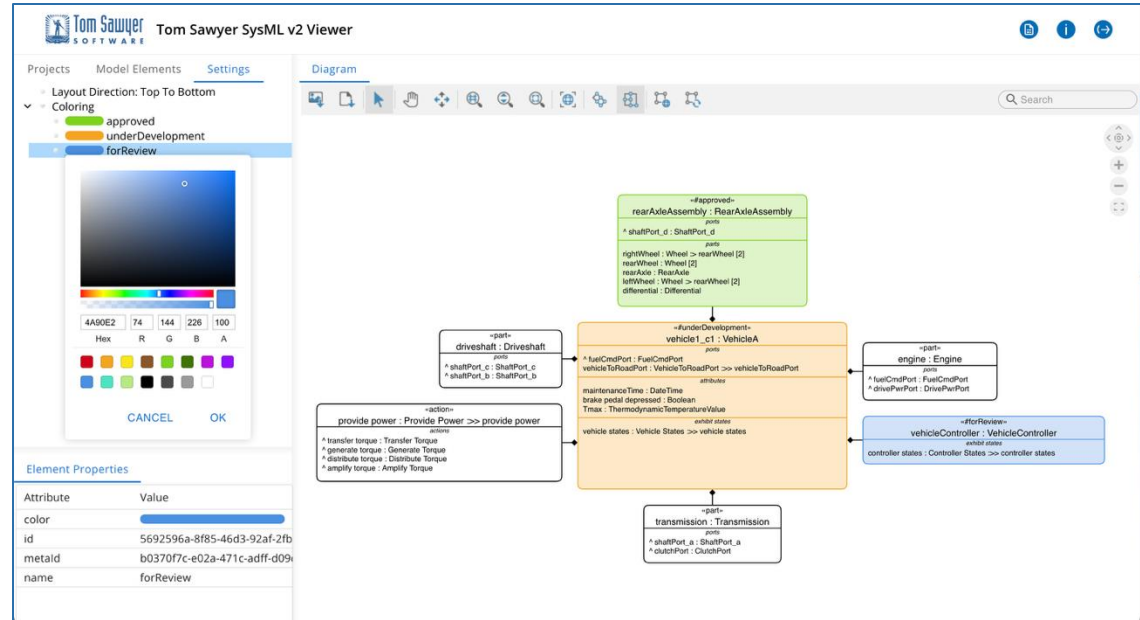
- Support users in modeling their systems as graphs





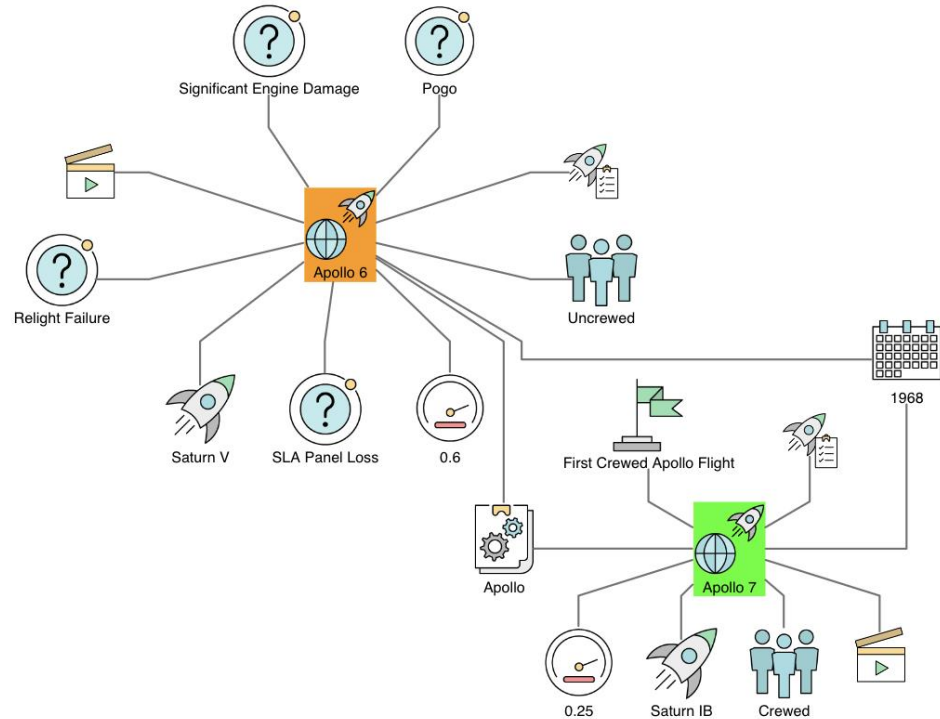
# Knowledge Graphs for Modeling and Analysis

- Support users in modeling their systems as graphs



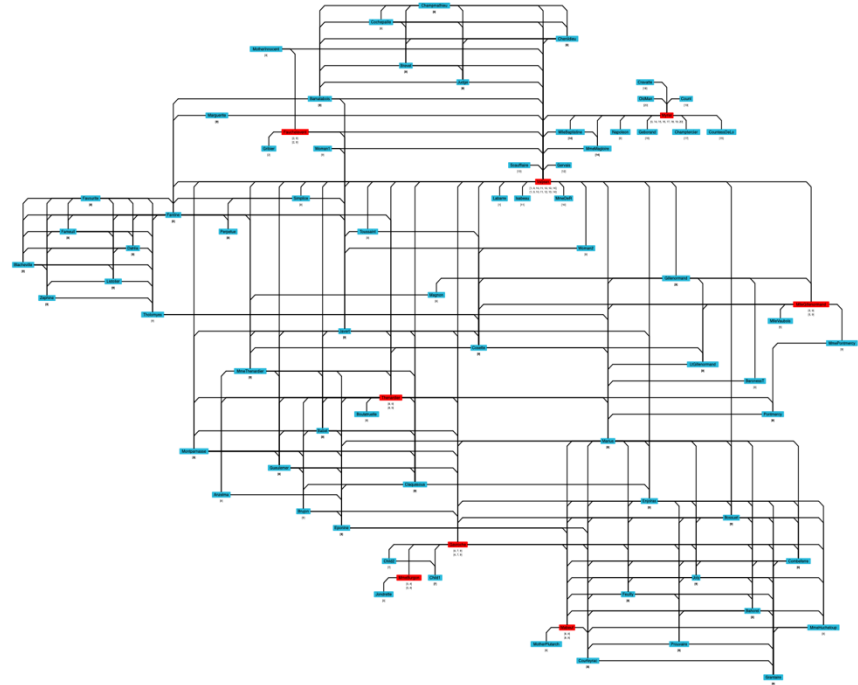
# Knowledge Graphs for Modeling and Analysis

- Support users in modeling their systems as graphs



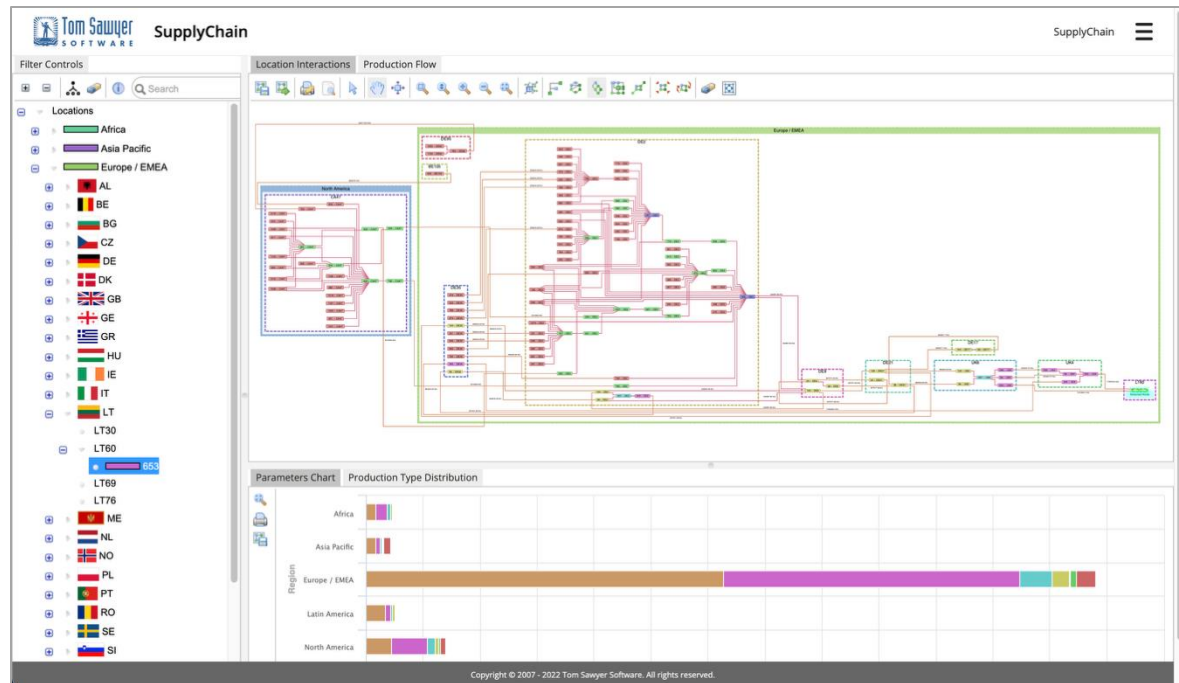
# Knowledge Graphs for Modeling and Analysis

- Apply graph visualizations and analyses to discover areas of interest in their data



# Knowledge Graphs for Modeling and Analysis

- Apply graph visualizations and analyses to optimize their systems

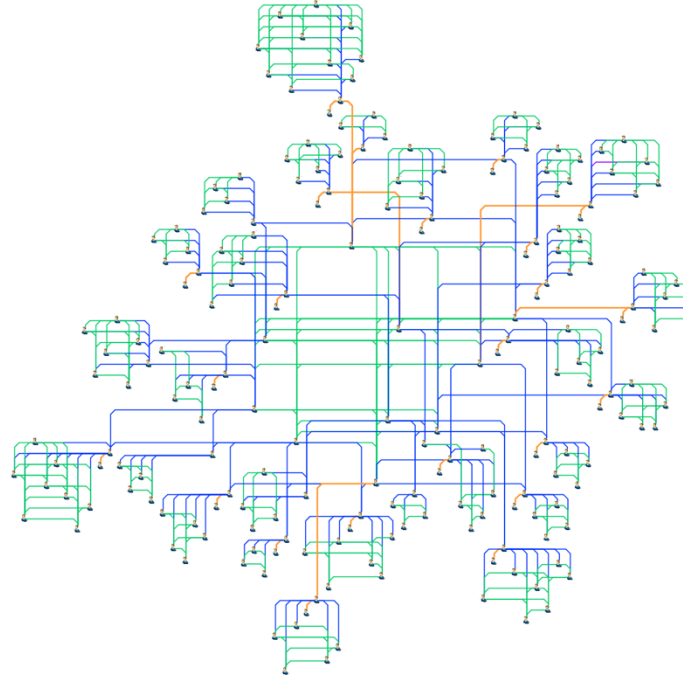


# Knowledge Graphs for Communication

- Deliver pertinent information to decision makers
- Communicate key results to stakeholders

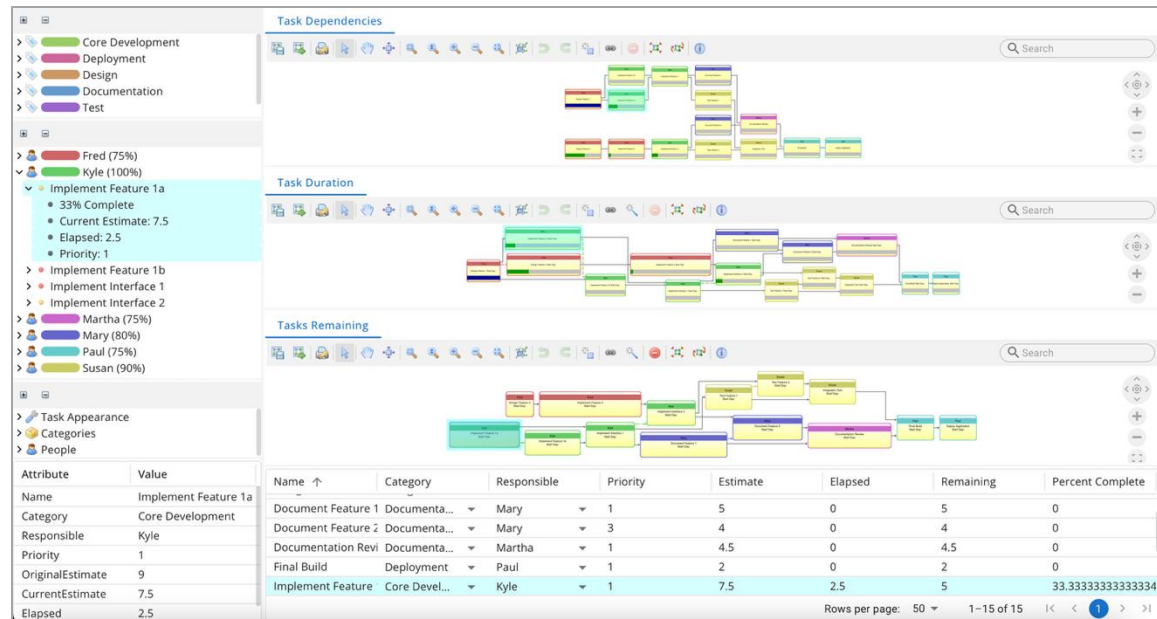
# Knowledge Graphs for Communication

- Deliver pertinent information to decision makers



# Knowledge Graphs for Communication

- Communicate key results to stakeholders

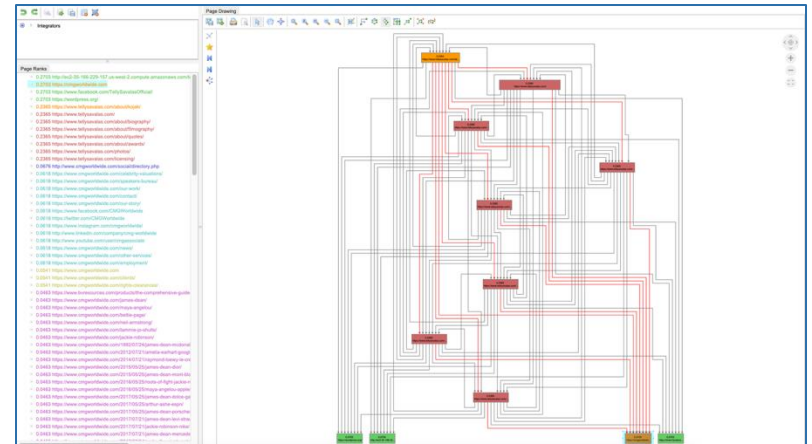
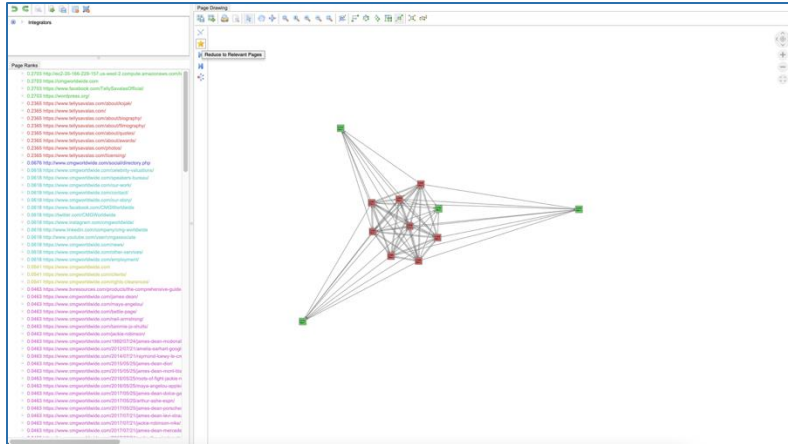


# Data Visualization for Systems Engineering

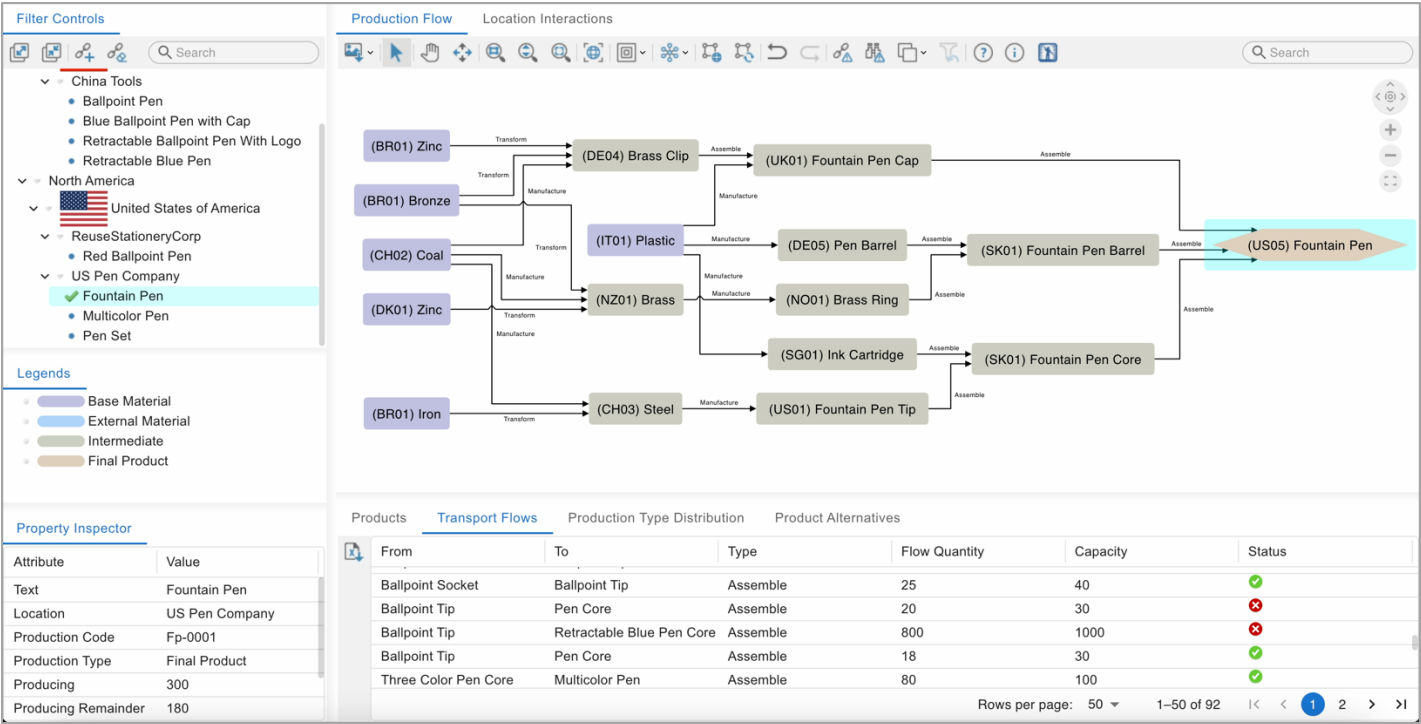
- Bring human experts back into the equation
  - “There is something interesting here, let’s look further”
  - “I have seen this pattern in another situation, a similar solution may be helpful here as well”
  - “There is an almost-pattern right there!” (almost vulnerability in a communications network, almost viable alternative for supply chain component, etc.)
  - “We can optimize the system here”
  - Multiple views into the same data provide a basis for the solution



# Post Situation Analysis

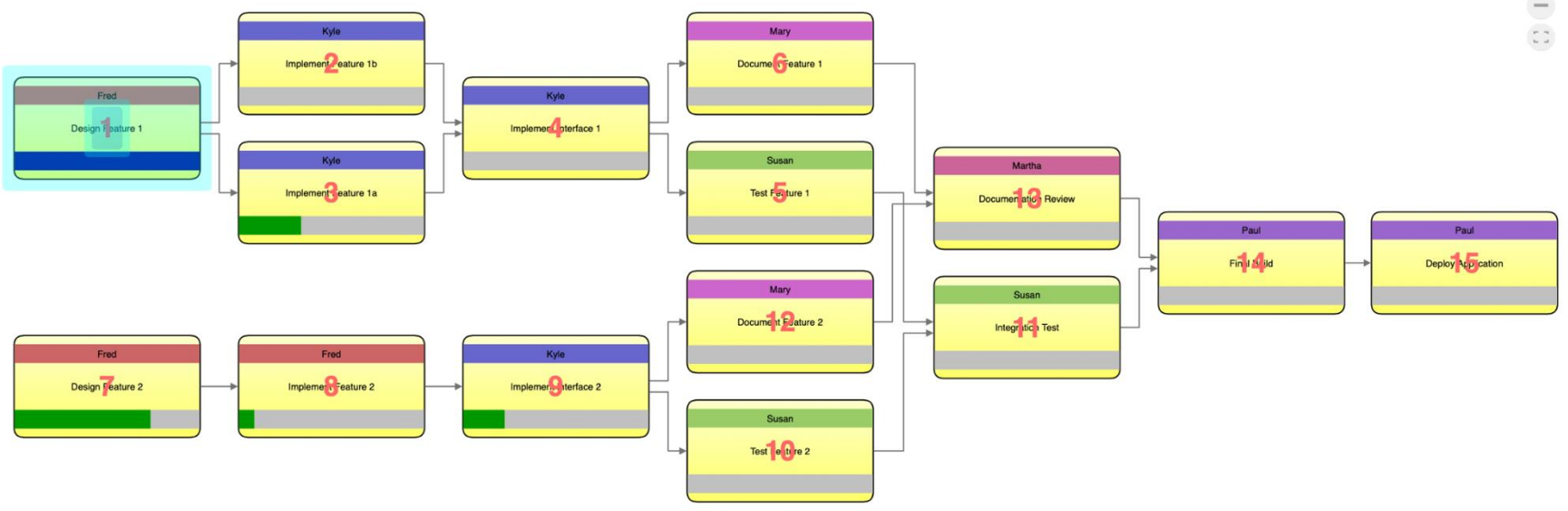


# Simulate Future Scenarios

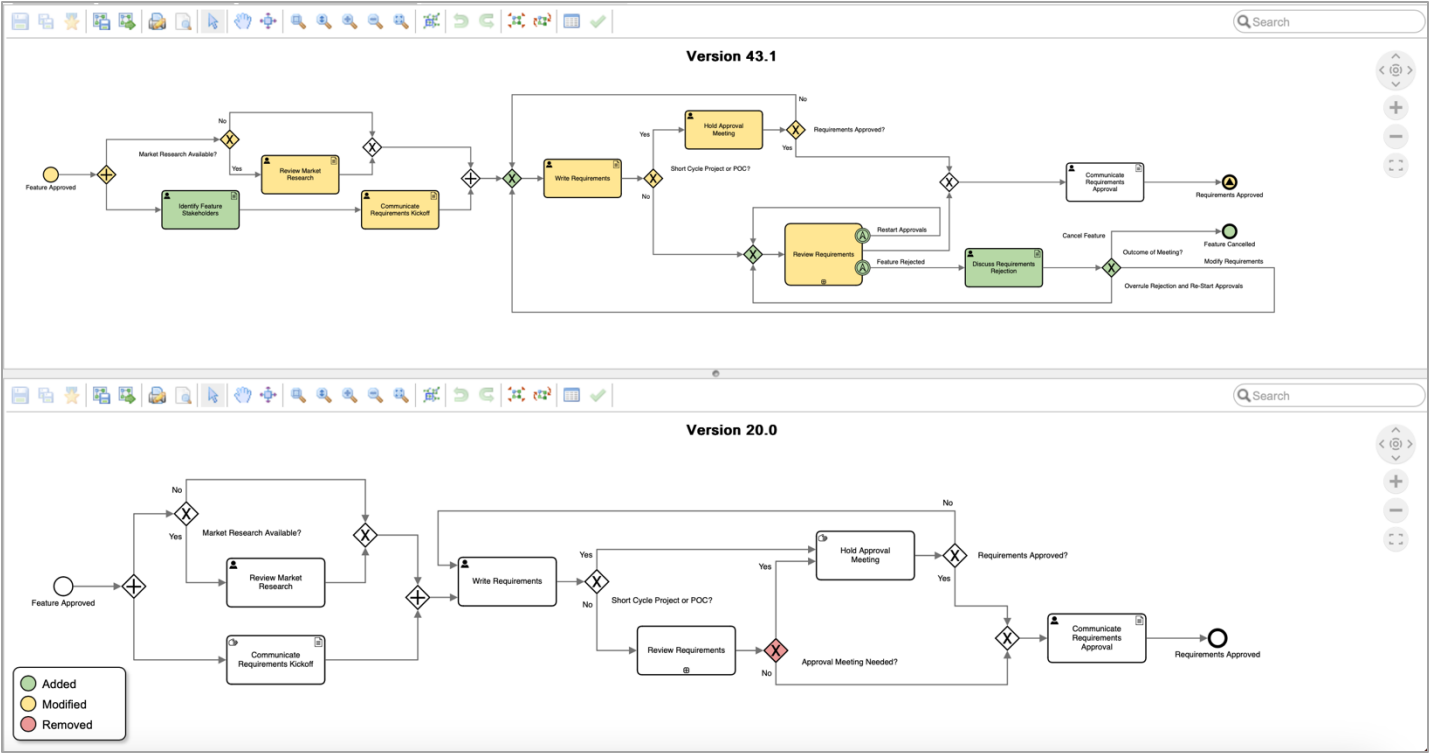




# Simulate Future Scenarios



# Discover Trends



# Knowledge Graphs for Failure Analysis

# Finding the Hidden Insights in Spaceflight Data

- Extremely complex, heterogeneous data
- Legacy data
- Difficult to find deeper insights
- Systematic approach required for accuracy and breadth of findings
- Adaptability of approach supports additional data sources and new failure types

# Human System Interaction

## The System

- Analyzes patterns in historical data.
- Notices common sequences that precede failure and the amount of times it results in an unfavorable outcome.
- Depending on the system configuration, it would either notify an operator and/or automatically implement mitigation measures.

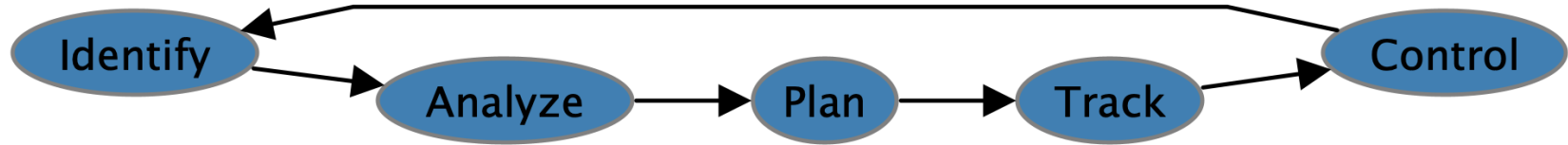
## The Human Interaction

- The system will create a visible alert that showcases need-to-know information and the suggested mitigation measures.
- The level of input the human has depends on the configuration of the system.

## Added Benefits

- Helps increase accuracy of fixed limitations to minimize the amount of failures and the amount of aborted tests that later are found to have been safe to proceed.
- Saves money and time throughout the testing phase.

# Continuous Risk Management (CRM)

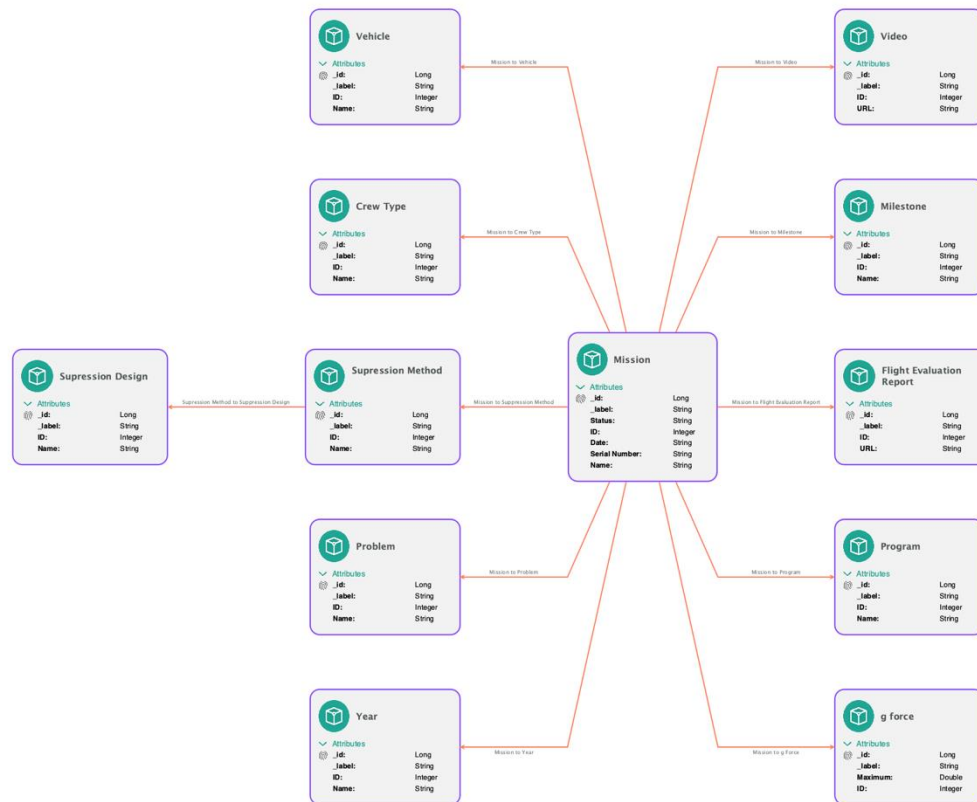




**“The CRM process manages risk by identifying specific issues that are of concern to one or more stakeholders, and which are perceived as presenting a risk to the achievement of one or more performance requirements.”**

**NASA Risk Management Handbook**

# Create Knowledge Graph



# Create Knowledge Graph

Guidance System Failure	Pioneer 1
Stage Recontact at Separation	Pioneer 1
Upper Stage Ignition Failure	Pioneer 2
Electrical Failure	Pioneer 2
Premature Total Stage Propulsion Shutdown	Pioneer 1

Electrical Failure

Guidance System Failure

Pioneer 2

1958

Pioneer 1

Upper Stage Ignition Failure

Stage Recontact at Separation

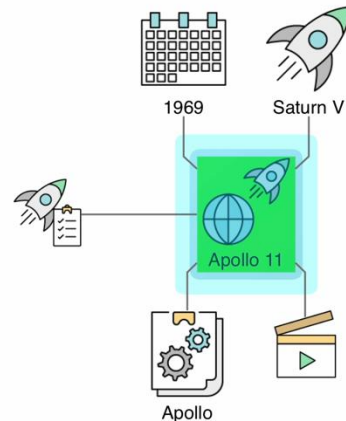
Premature Total Stage Propulsion Shutdown

Attribute	Value
No rows	

Date ↑	Name	Status
1958.10.11	Pioneer 1	Partial Failure
1958.11.07	Pioneer 2	Failure

# Support Graph RAG

Guidance System Failure	Pioneer 1
Stage Recontact at Separation	Pioneer 1
Upper Stage Ignition Failure	Pioneer 2
Electrical Failure	Pioneer 2
Premature Total Stage Propulsion Shutdown	Pioneer 1

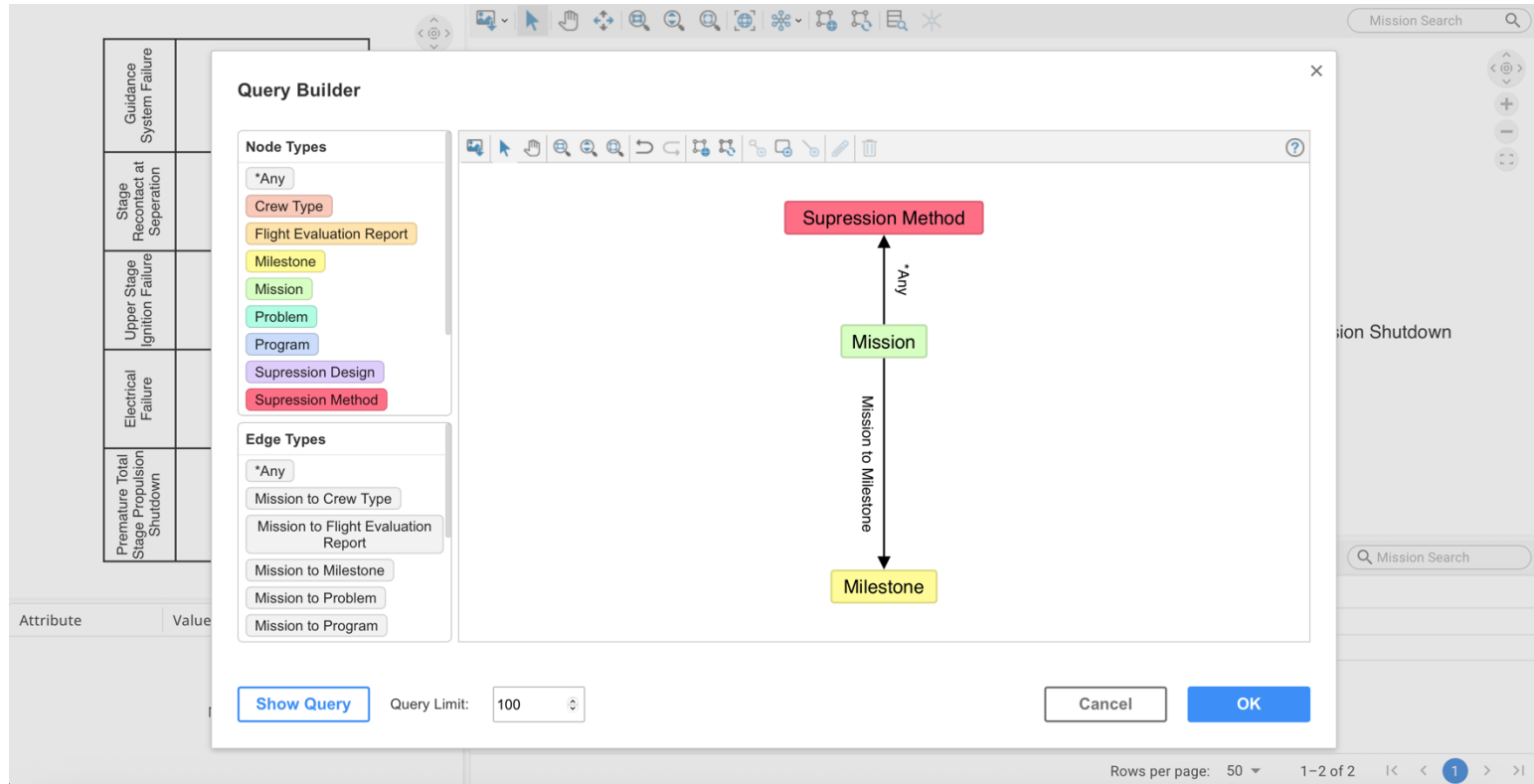


1969 Saturn V Apollo

Attribute	Value
_id	118
_label	Mission
Date	1969.07.16
ID	121
Name	Apollo 11
Serial Number	SA-506

Date ↑	Name	Status
1958.10.11	Pioneer 1	Partial Failure
1958.11.07	Pioneer 2	Failure
1969.07.16	Apollo 11	Success

# Define Advanced Graph Patterns



The screenshot displays the INCOSE Query Builder interface. On the left, a sidebar lists various system components: Guidance System Failure, Stage Recontact at Separation, Upper Stage Ignition Failure, Electrical Failure, and Premature Total Stage Propulsion Shutdown. The main area shows a graph pattern definition with three nodes: Supression Method (red), Mission (green), and Milestone (yellow). The edges are labeled: \*Any (between Supression Method and Mission) and Mission to Milestone (between Mission and Milestone). The interface includes a 'Show Query' button, a 'Query Limit' dropdown set to 100, and 'Cancel' and 'OK' buttons. The bottom status bar indicates 'Rows per page: 50' and '1-2 of 2'.

**Query Builder**

**Node Types**

- \*Any
- Crew Type
- Flight Evaluation Report
- Milestone
- Mission
- Problem
- Program
- Supression Design
- Supression Method

**Edge Types**

- \*Any
- Mission to Crew Type
- Mission to Flight Evaluation Report
- Mission to Milestone
- Mission to Problem
- Mission to Program

**Graph Pattern:**

```

graph TD
    SM[Supression Method] -- "*Any" --> M[Mission]
    M -- "Mission to Milestone" --> MI[Milestone]
  
```

**Buttons:** Show Query, Query Limit: 100, Cancel, OK

**Status Bar:** Rows per page: 50, 1-2 of 2, 1

## Communicate Key Findings to Stakeholders and Decision Makers

The screenshot displays the Mission Search tool interface, which includes a mission timeline table, a system architecture diagram, and a search results table.

#### Mission Timeline Table

Attribute	Value
_id	70
_label	Mission
Date	1960.12.04
ID	111
Name	Explorer S-56

#### System Architecture Diagram

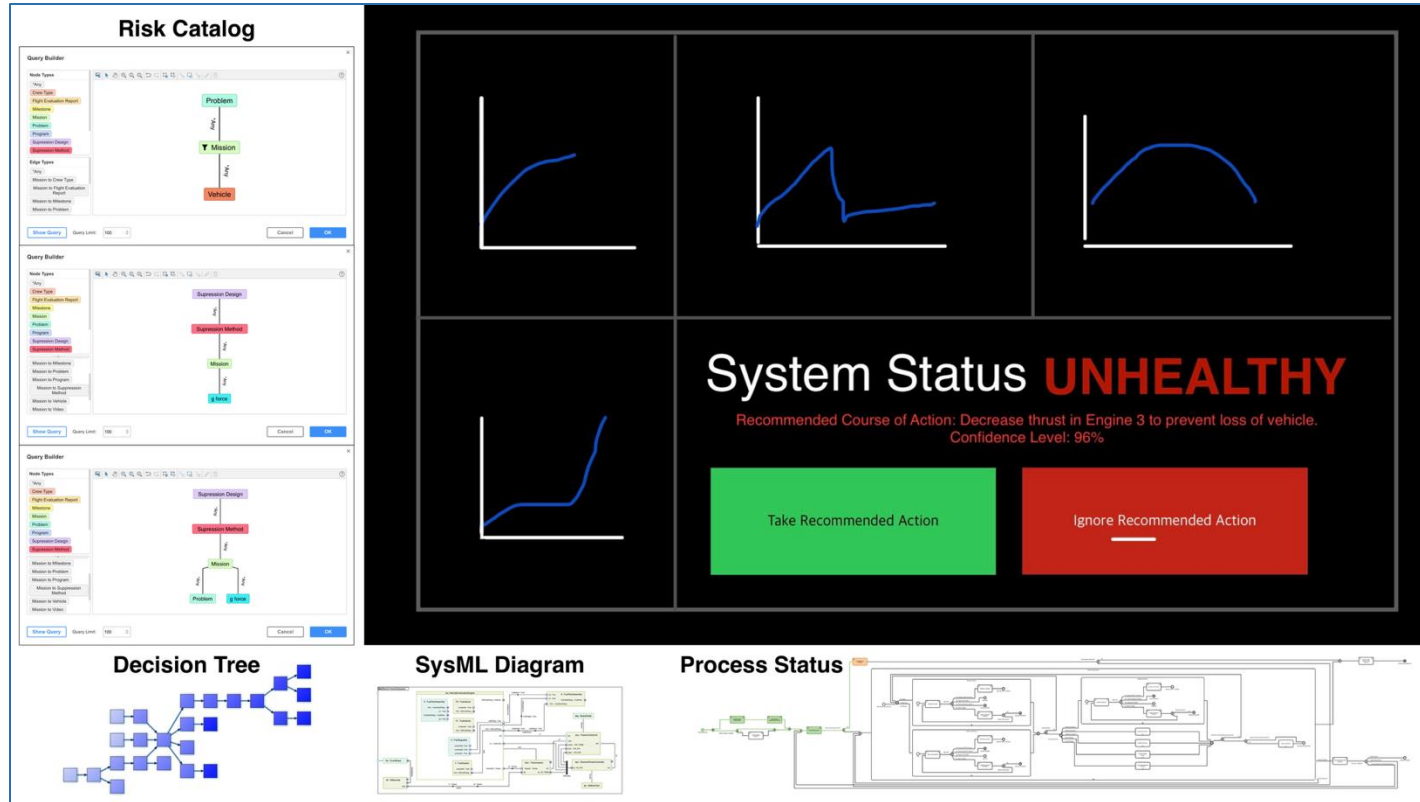
The diagram illustrates the system architecture for the Explorer S-56 mission. It shows various components such as the Explorer S-56, Mercury-Atlas 3, Explorer S-55, Explorer 13/S55A, and Mercury-Scout 1, interconnected by communication links. The diagram also includes a legend for component types and a search bar labeled "Mission Search".

#### Search Results Table

Date ↑	Name	Status
1960.12.04	Explorer S-56	Failure
1961.04.25	Mercury-Atlas 3	Failure
1961.06.30	Explorer S-55	Failure
1961.08.25	Explorer 13/S55A	Partial Failure
1961.11.01	Mercury-Scout 1	Failure

Rows per page: 50 | 1-26 of 26

# Emerging Failure Detection



# Supporting Systems Engineering Vision 2035

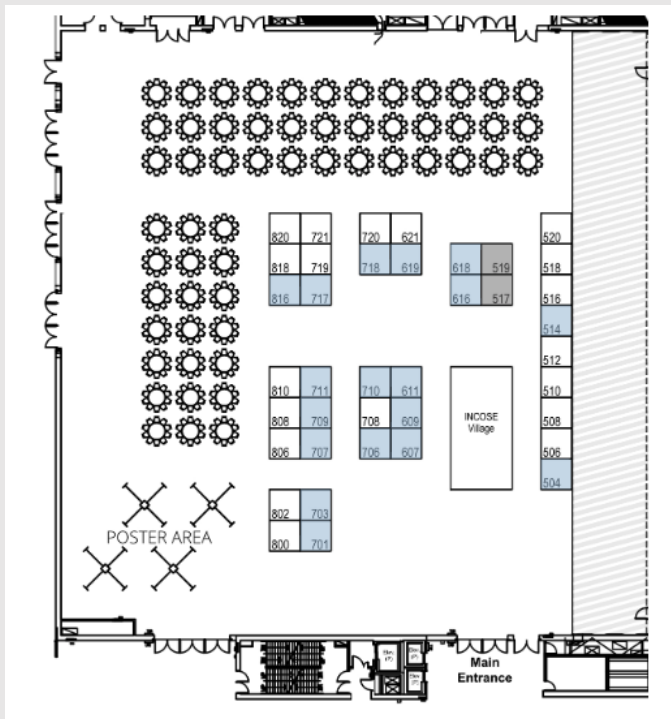
- Adaptability to evolving technology and requirements
- Increased interoperability with simulation and multi-disciplinary analysis
- Analytic framework to understand, define, and sustain increasingly complex systems
- Provide reliable and timely knowledge for decision management
- Management of complexity and risk



# Future Work

- Apply this approach to other industries including manufacturing and automotive
- Increase integration with SysML v2 modeling
- Provide insights for Risk-Informed Decision Making (RIDM)

# Let's connect



**Madison Urquhart**

INCOSE Student Member

Tom Sawyer Software

Product Management Intern

e [murquhart@tomsawyer.com](mailto:murquhart@tomsawyer.com)

**Janet Six, Ph.D.**

INCOSE Member

Tom Sawyer Software

Senior Product Manager

e [jsix@tomsawyer.com](mailto:jsix@tomsawyer.com)



# 35<sup>th</sup> Annual **INCOSE** international symposium

hybrid event

**Ottawa, Canada**  
26 - 31 July 2025