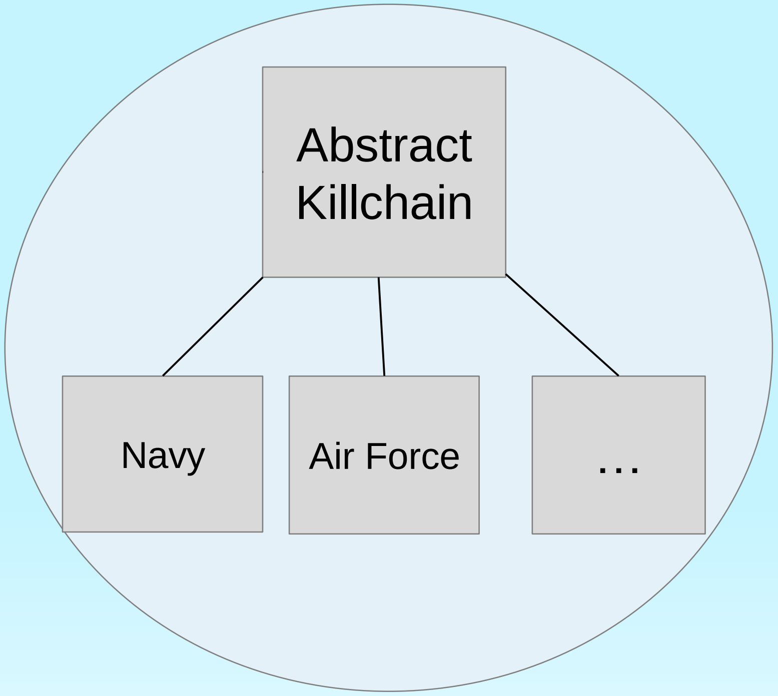
Scaling Model-Based System Engineering Practices for System of Systems Applications: Software Methods

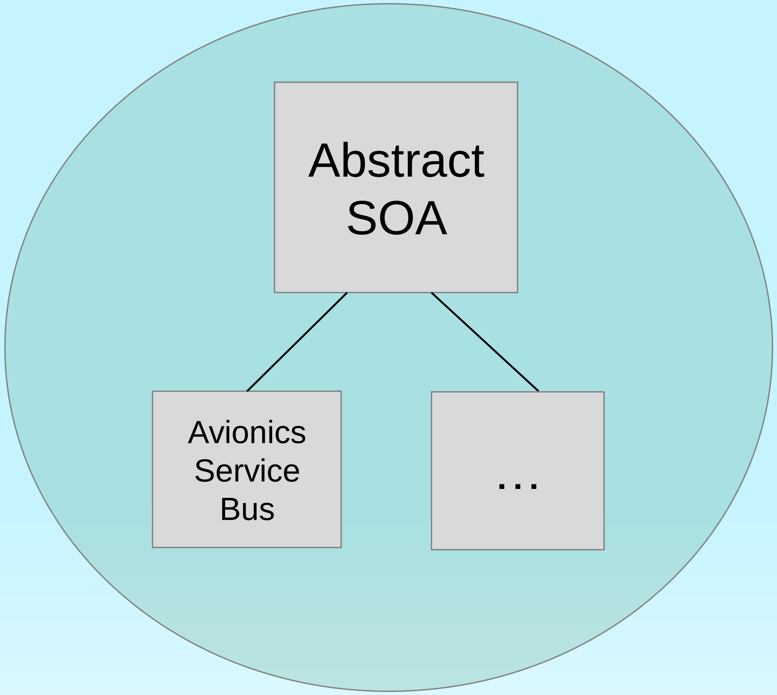
October 2017

Janna Kamenetsky jannak@mitre.org
Laura Antul lantul@mitre.org
Dr. Aleksandra Markina-Khusid amk@mitre.org
Matt Cotter mjcotter@mitre.org
Dr. Judith Dahmann jdahmann@mitre.org

NDIA 20th Annual Systems Engineering Conference
http://www.ndia.org/events/2017/10/23/20th-systems-engineering-conference

Technical Approach: Inheritable Architectures





Kill Chain Architecture

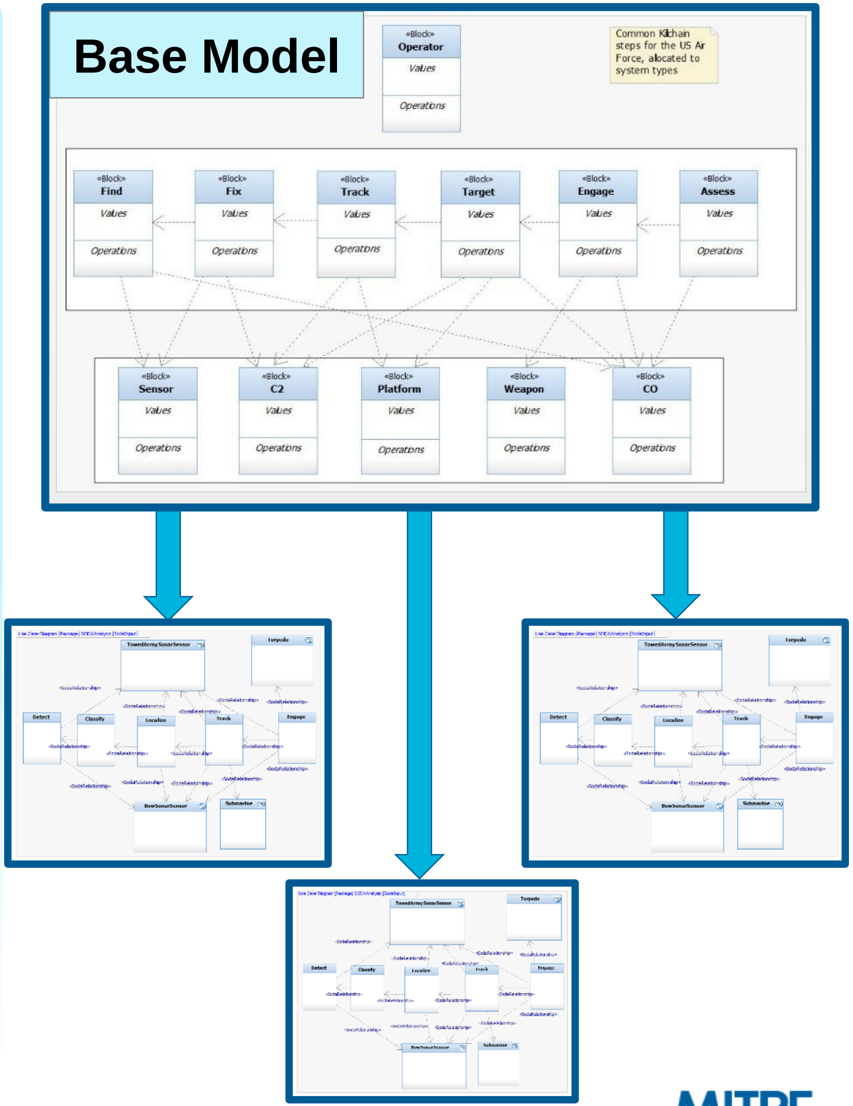
Service Oriented Architecture (SOA)

Enables Model Re-use corresponding to different architecture patterns

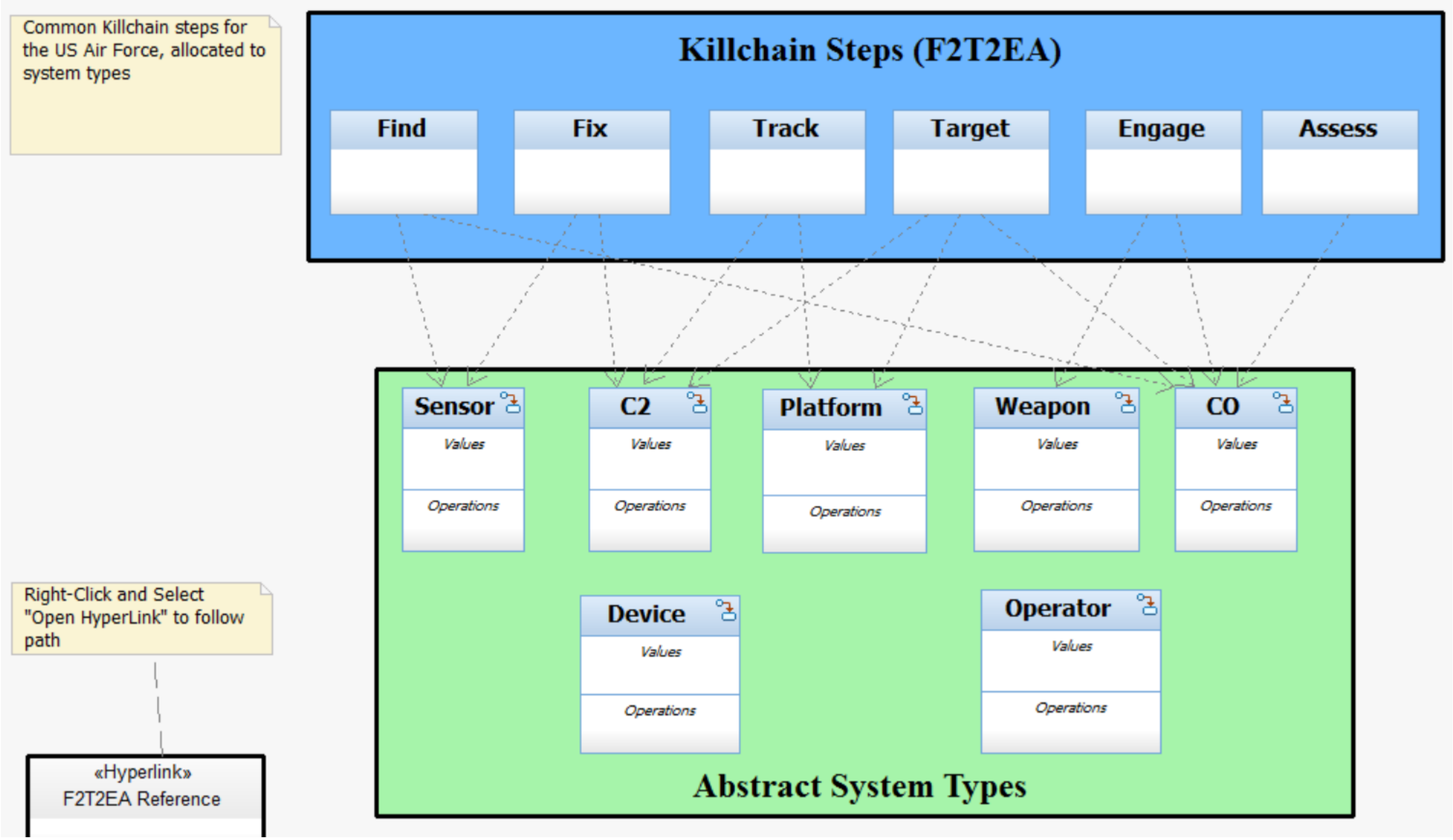
Base Model Architecture

- **Base/Derivative Model Framework**
  - **Base Model captures key functional SoS architecture**
  - **Derivative model represent domain-specific behavior**

- **This approach helps:**
  - **Accelerate domain model development via Base Model reuse**
  - **Rapidly evaluate different options utilizing predefined stereotypes and analysis engines**
  - **Iterative design to continuously refine common SoS functions**



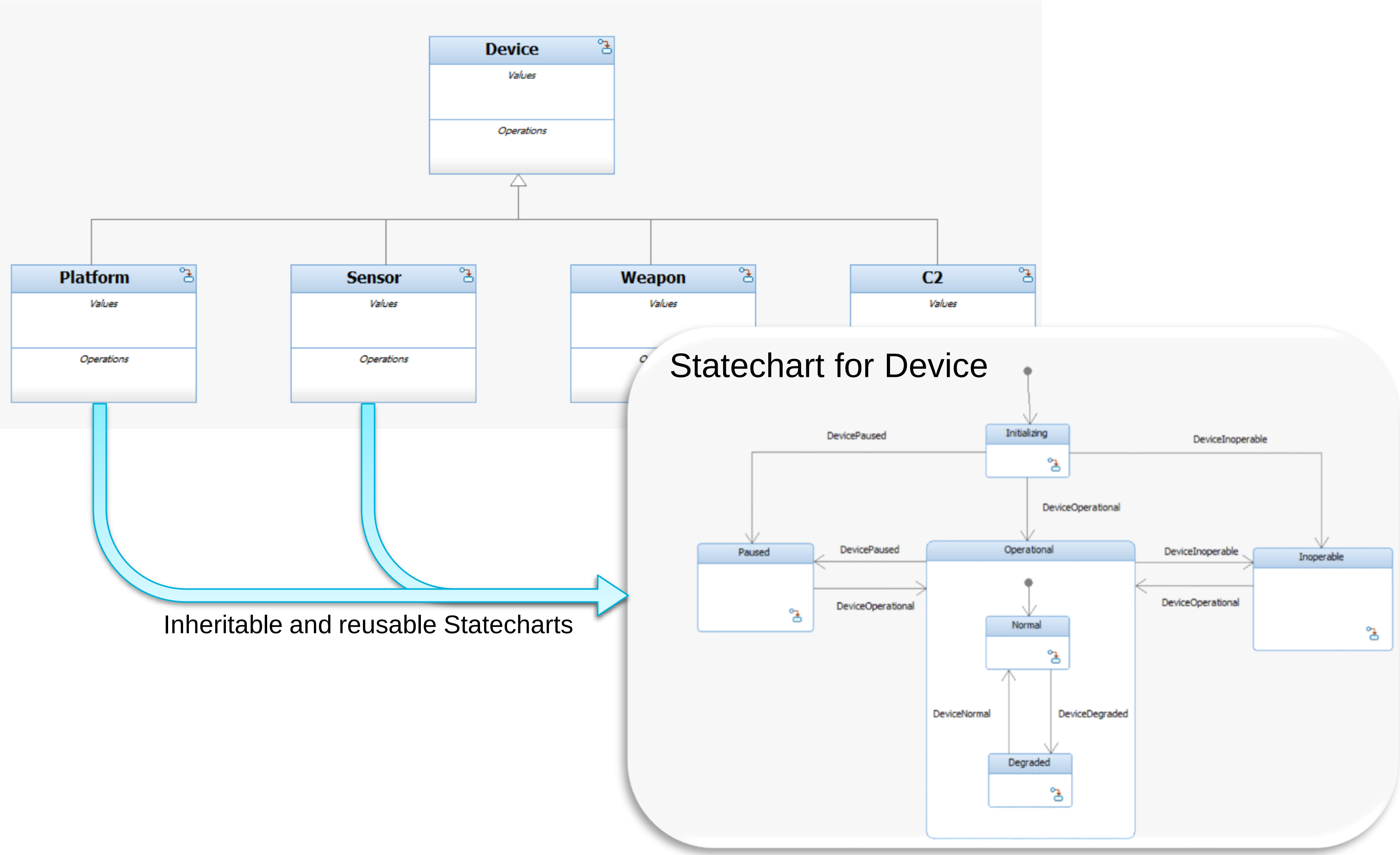
Base Model: High Level Structure



Common Killchain steps for the US Air Force, allocated to system types

Right-Click and Select "Open HyperLink" to follow path

Base Model: Inheritance Structure



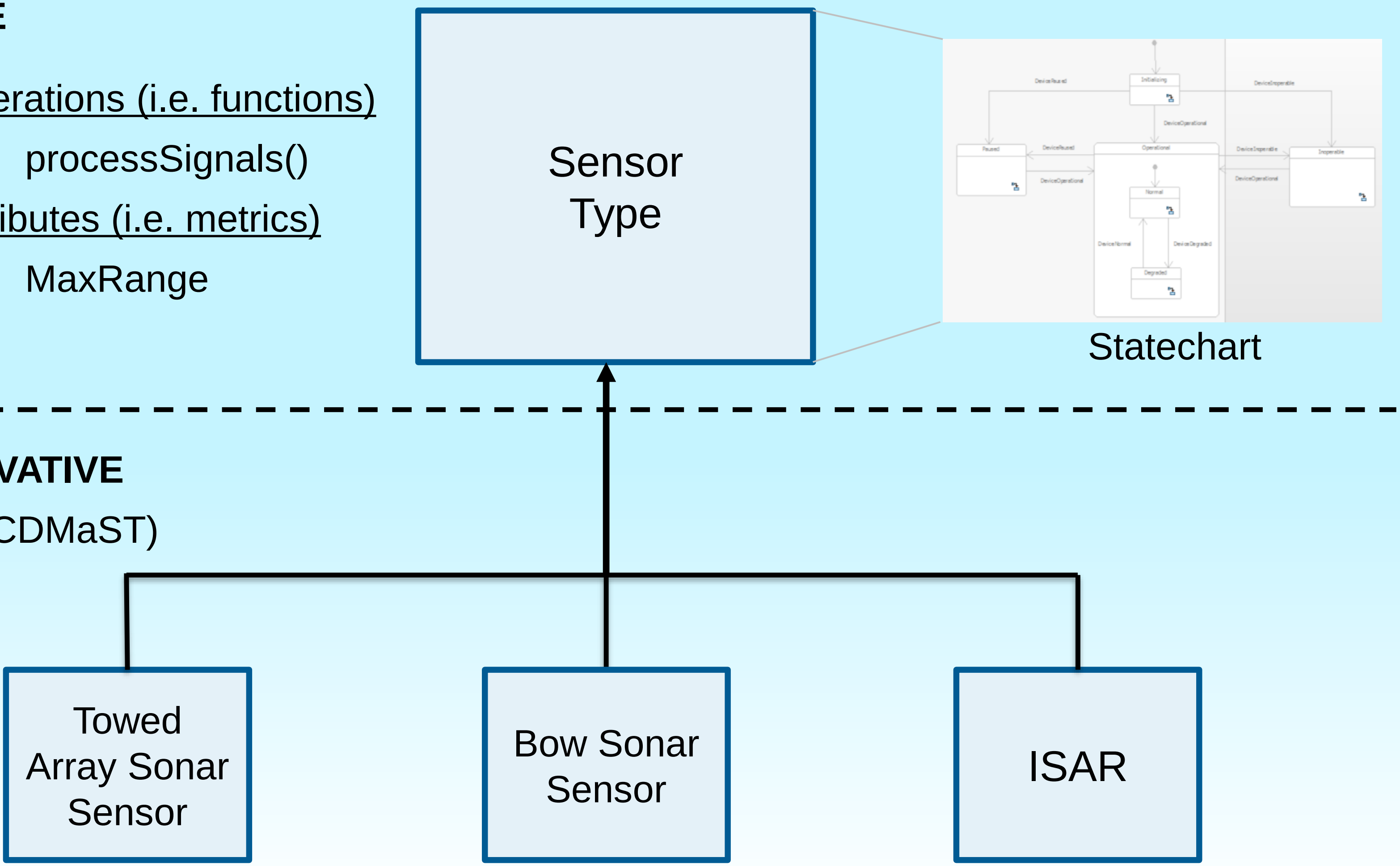
Statechart for Device

Inheritable and reusable Statecharts

BASE Model: Inheritable Types

BASE

- Operations (i.e. functions)
  - processSignals()
- Attributes (i.e. metrics)
  - MaxRange

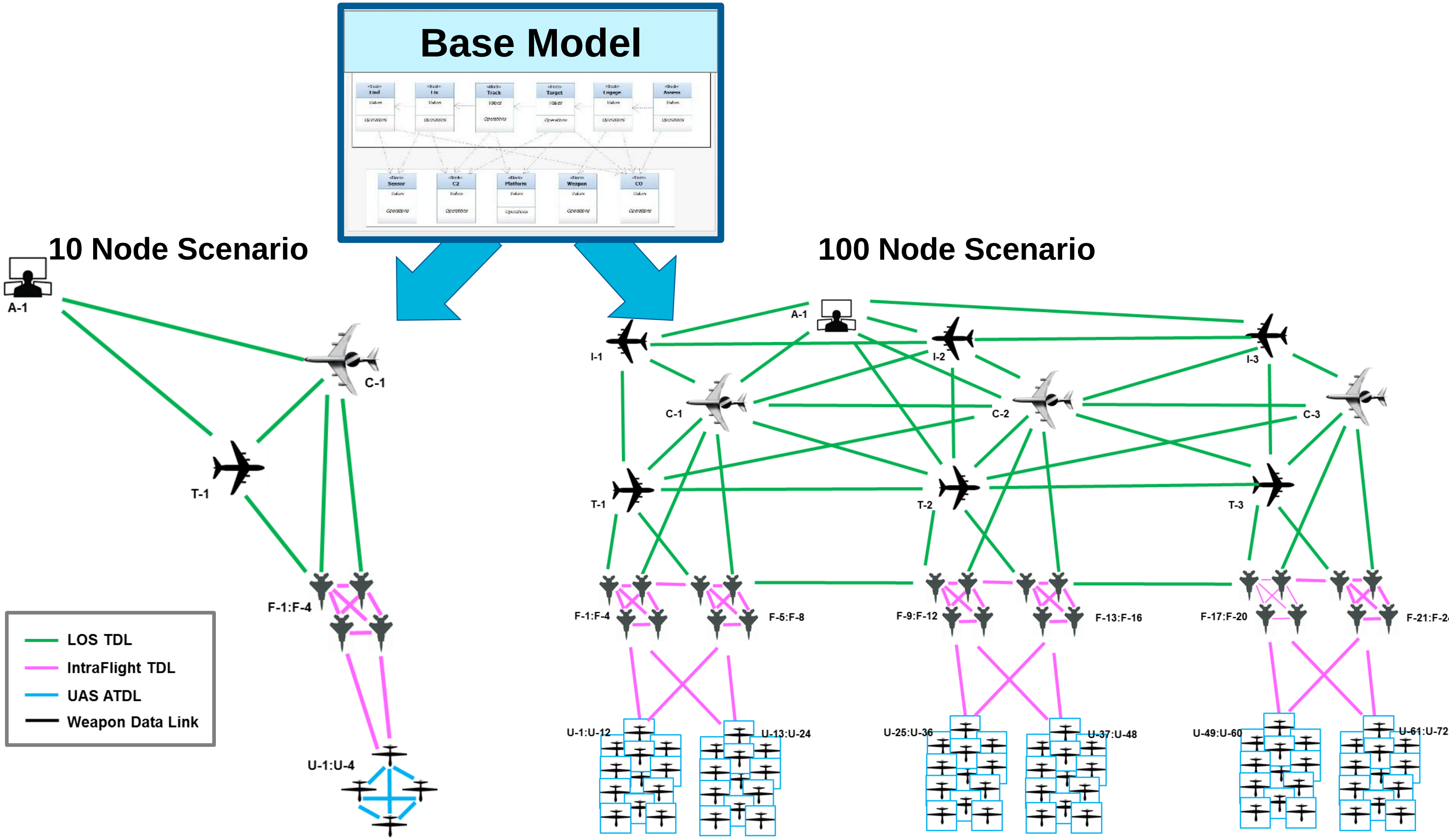


Statechart

DERIVATIVE

(e.g. CDMaST)

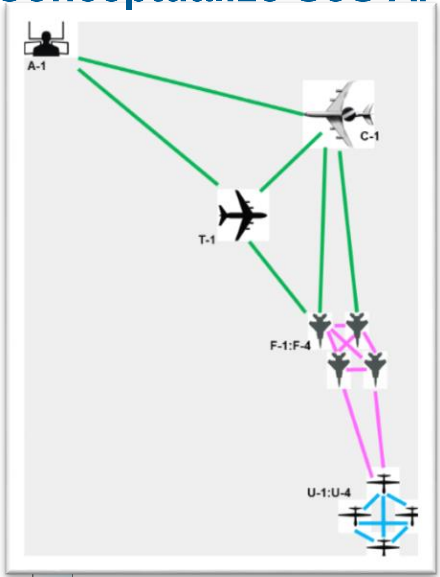
Base Model CSV Importer



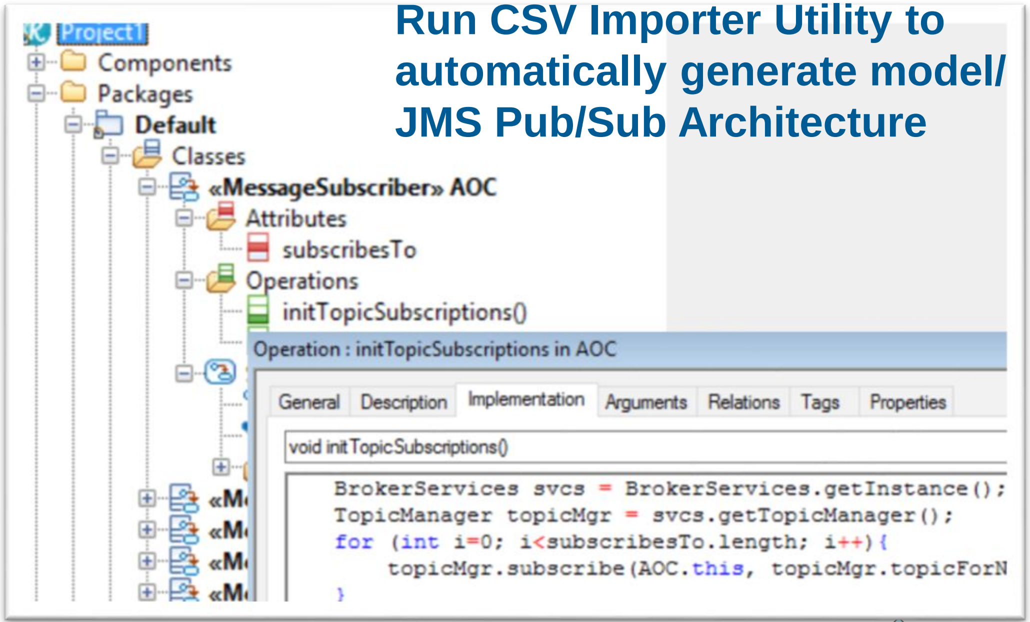
MBE Utility to reduce development effort associated with modeling large SoS complex networks

CSV Importer Utility

Conceptualize SoS Architecture



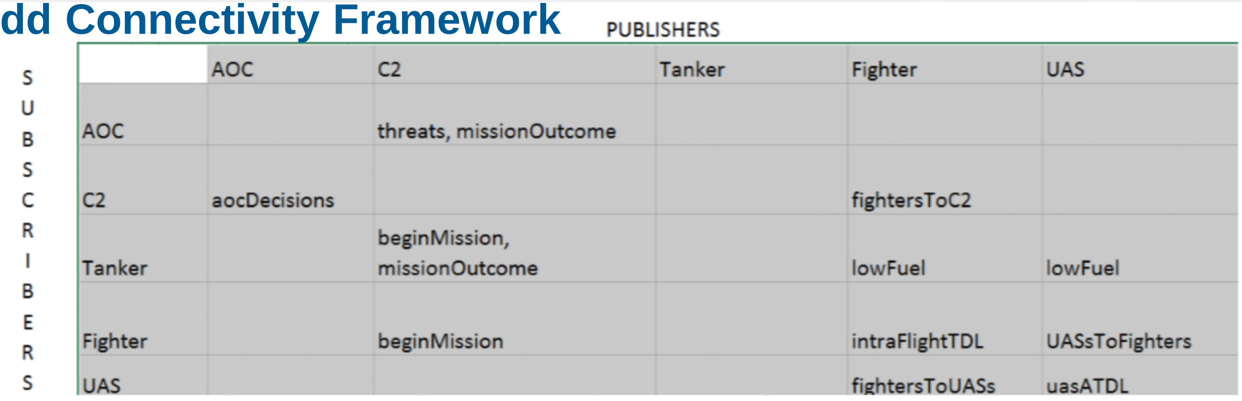
Run CSV Importer Utility to automatically generate model/ JMS Pub/Sub Architecture



```
void initTopicSubscriptions()
```

```
BrokerServices svcs = BrokerServices.getInstance();
TopicManager topicMgr = svcs.getTopicManager();
for (int i=0; i<subscribesTo.length; i++){
    topicMgr.subscribe(AOC.this, topicMgr.topicForN
}
```

Add Connectivity Framework



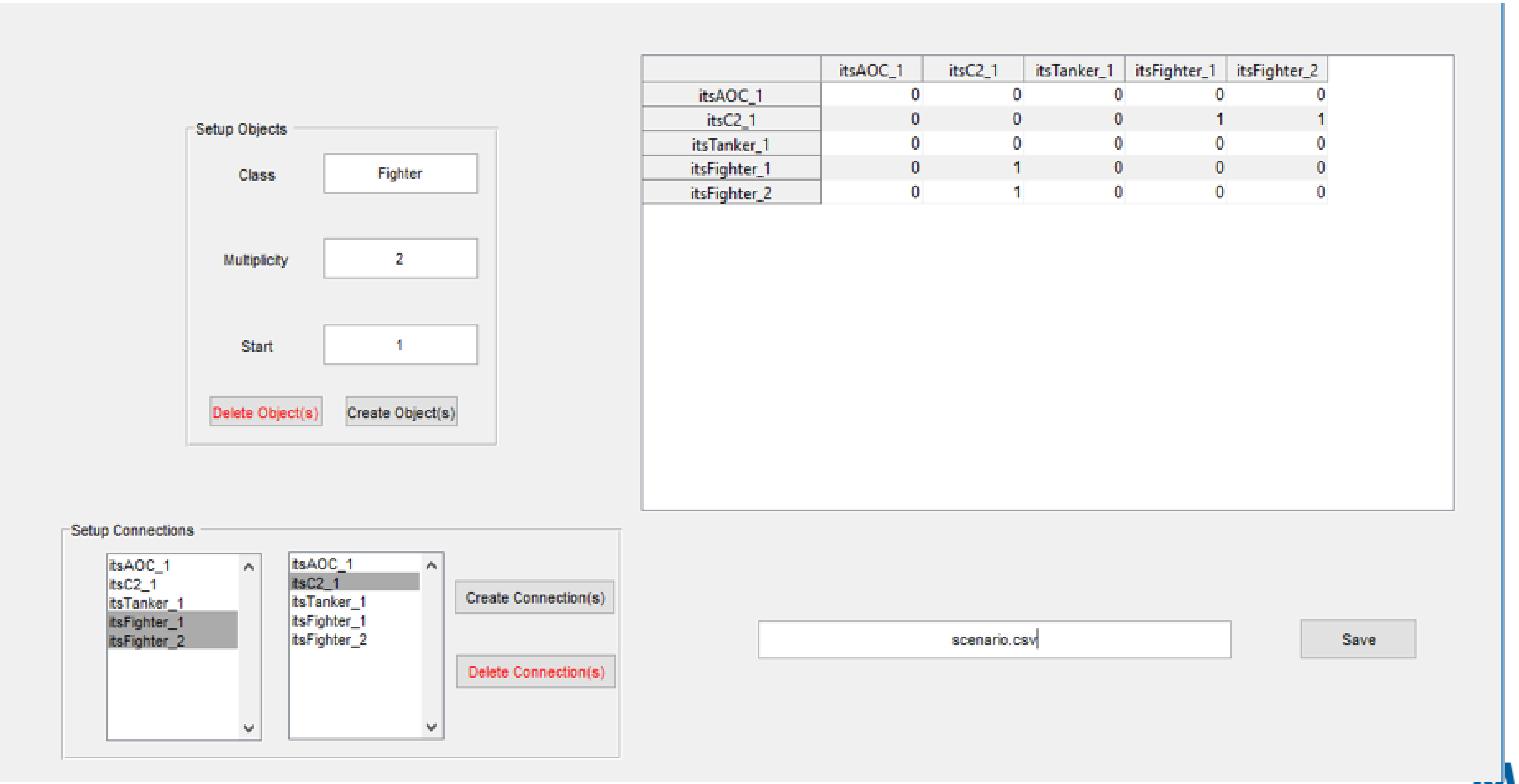
PUBLISHERS

SUBSCRIBERS

	AOC	C2	Tanker	Fighter	UAS
AOC		threats, missionOutcome			
C2	aocDecisions			fightersToC2	
Tanker		beginMission, missionOutcome		lowFuel	lowFuel
Fighter		beginMission		intraFlightTDL	UASsToFighters
UAS				fightersToUASs	uasATDL

Base Model GUI

- A MATLAB GUI has been built to simplify the process of populating a connectivity matrix
- The tool outputs a CSV file that can then be imported into the architecture model



Setup Objects

Class: Fighter

Multiplicity: 2

Start: 1

	itsAOC_1	itsC2_1	itsTanker_1	itsFighter_1	itsFighter_2
itsAOC_1	0	0	0	0	0
itsC2_1	0	0	0	1	1
itsTanker_1	0	0	0	0	0
itsFighter_1	0	1	0	0	0
itsFighter_2	0	1	0	0	0

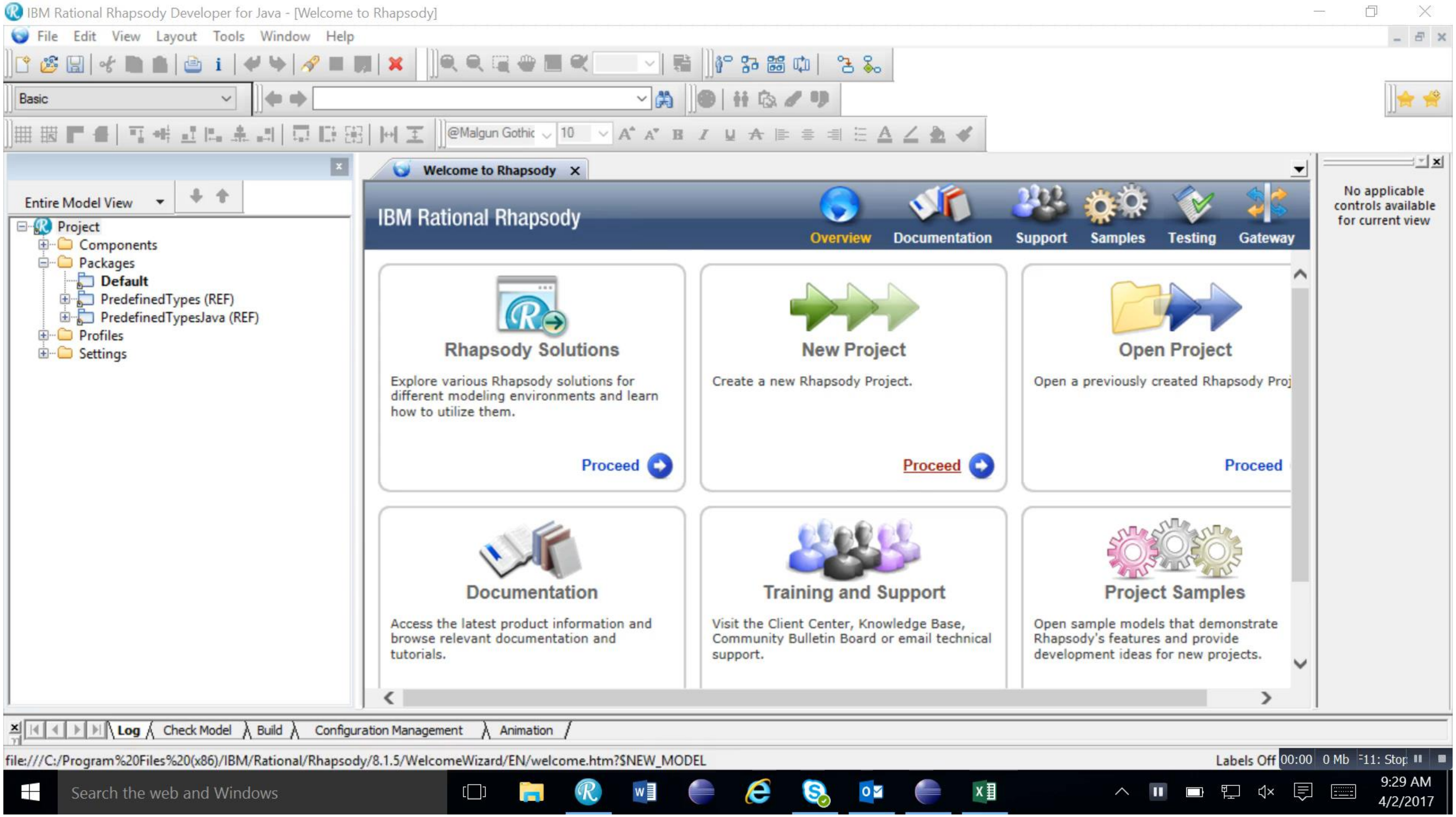
Setup Connections

itsAOC_1
itsC2_1
itsTanker_1
itsFighter_1
itsFighter_2

itsAOC_1
itsC2_1
itsTanker_1
itsFighter_1
itsFighter_2

scenario.csv

Demonstration



Rhapsody Solutions

Explore various Rhapsody solutions for different modeling environments and learn how to utilize them.

Proceed

New Project

Create a new Rhapsody Project.

Proceed

Open Project

Open a previously created Rhapsody Proj

Proceed

Documentation

Access the latest product information and browse relevant documentation and tutorials.

Training and Support

Visit the Client Center, Knowledge Base, Community Bulletin Board or email technical support.

Project Samples

Open sample models that demonstrate Rhapsody's features and provide development ideas for new projects.

file:///C:/Program%20Files%20(x86)/IBM/Rational/Rhapsody/8.1.5/WelcomeWizard/EN/welcome.htm?$NEW_MODEL

Q2 Metrics – Experiments

- **Qualitative**
  - Experiment 1: Give the base model to MITRE employees to use on their projects as they see fit. Collect feedback.
    - Likes, dislikes, pain points, time savings estimates, description of use case, experience level
    - Time Cost: 30 min interview
- **Quantitative**
  - Experiment 2: Give MITRE employees a sample coms network and have them create it by hand and by using the CSV importer
    - Networks of different sizes
    - Measure time to complete exercise
    - Time Cost: Approx. 45 min per data point
  - Experiment 3: Randomized control trial with ~20 new interns
    - Group A: Create reference model from scratch
    - Group B: Create reference model using base model

Metrics – Experiment 1 Results

- **Project 1:**
  - 3 reviewers
  - Not adopted
- **Feedback:**
  - "…This base model would be a great reference, e.g., utilizing the package structure framework used, with the inheritable architectures and the focus on reuse."
  - "…We expect to draw ideas from it as we build our own model."
  - "We intend to focus more on activity diagrams than state charts."
  - "Our project is not in the context of the Air Force, so we would have to change the block and activity names."
  - "Overall it is not a good fit for [our project]."

- **Project 2:**
  - 1 reviewer
  - Adopted
- **Feedback:**
  - Qualitative

    Base Model state charts look too "in-depth", "specific", need to take a closer look to see if they will work for my use case. But if they work, "that would be awesome", it will save tons of time.
  - Pseudo - Quantitative

    Estimated time savings of 40 hours on work completed so far.
  - Update

    Base Model has proven a good fit for project and has been used extensively.

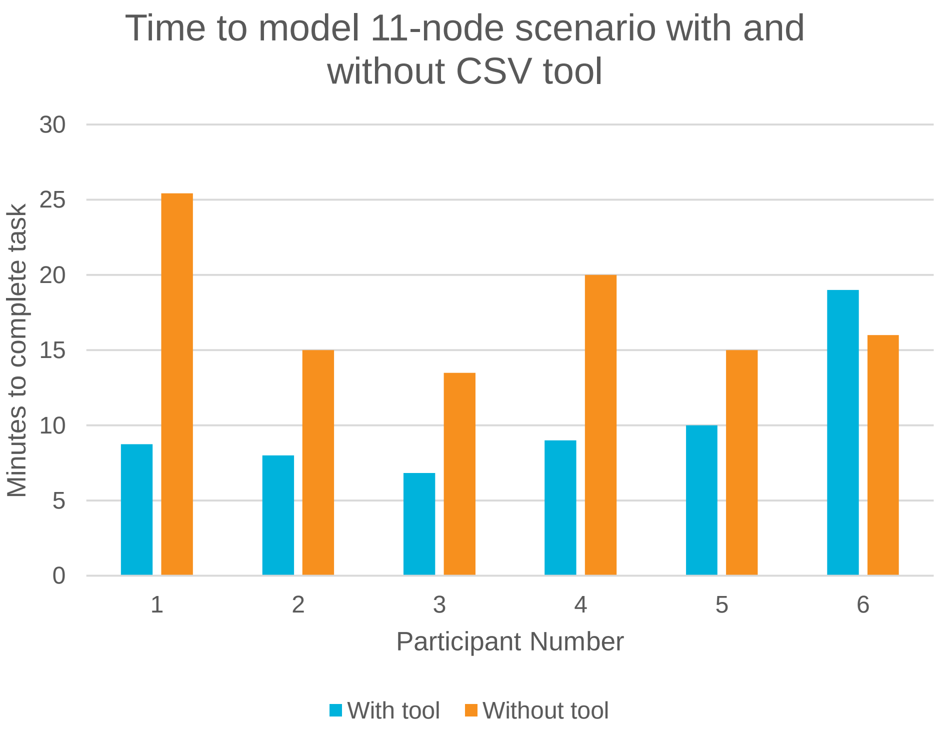
Metrics – Experiment 2 Results

The Scenario

This is a hypothetical Air Force kill-chain scenario consisting of 1 ground control station (AOC), 1 air command and control (C2), 4 Fighter Jets, 4 Unmanned Aircraft Systems (UASs), and 1 Tanker.

- AOC needs to be able to communicate with C2, since C2 alerts AOC when there is a threat and then gets its orders from the ground.
- C2 also needs to be able to communicate with all fighters and the Tanker during the mission.
- Also, all fighters and UASs need to be able to communicate with the Tanker, since they'll occasionally need to refuel during flight.
- Every fighter needs to be able to communicate with every other fighter, and
- every UAS needs to be able to communicate with every other UAS.
- Moreover, every fighter should be able to communicate with every UAS, and vice versa.

You may assume all communication channels are bi-directional (any communication matrix you set up should be symmetric with respect to rows and columns).

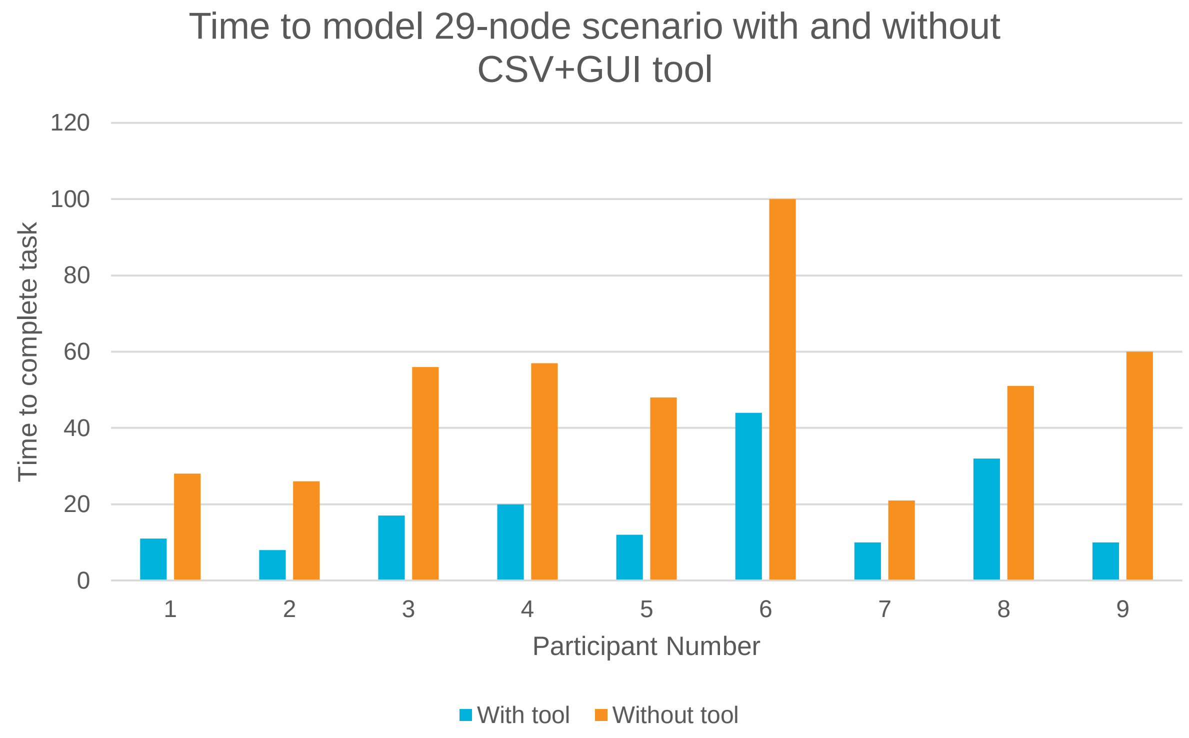


Time savings

Mean: 39%

Standard Dev: 12%

Metrics – Experiment 2 Results



Time savings

Mean: 63%

Standard Dev: 14%

Average mistakes

Without tool: 9.2

With tool: 0.8