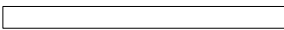


# EVALUATING CYBERSECURITY TOOLS FOR USE IN AN EARLY STPA-SEC FLOW

As Recommended by the DAU Cybersecurity Best Practices Guidebook for Major Acquisitions

15 OCT 2023

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.  
OPSEC #7992

  
Daniel W. Newport  
Branch Chief, Cyber Technology Development (CTD),  
Ground Systems Cyber Engineering (GSCE),  
Ground Vehicle Systems Center (GVSC),  
U.S. Army Combat Capabilities Development Command (CCDC)  
daniel.w.newport.civ@army.mil

David Hetherington  
Principal  
System Strategy, Inc  
dhetherington@systemxi.com

# ABSTRACT

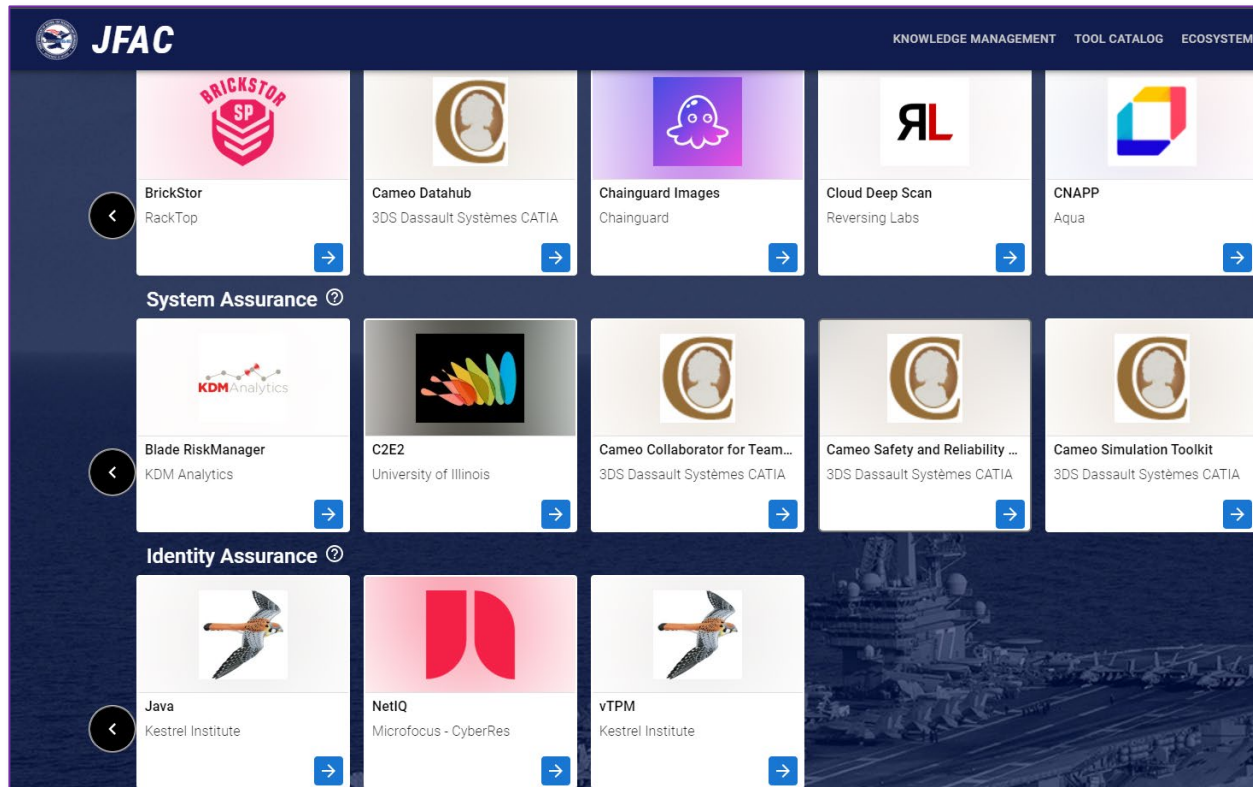


This presentation will review a recent effort performed by the Army Ground Vehicle Systems Center (GVSC) and sponsored by the Joint Federated Assurance Center (JFAC) to evaluate cybersecurity tools with an eye to integration with the Digital Engineering environment. General findings will be presented. Specific tool results will not be covered.

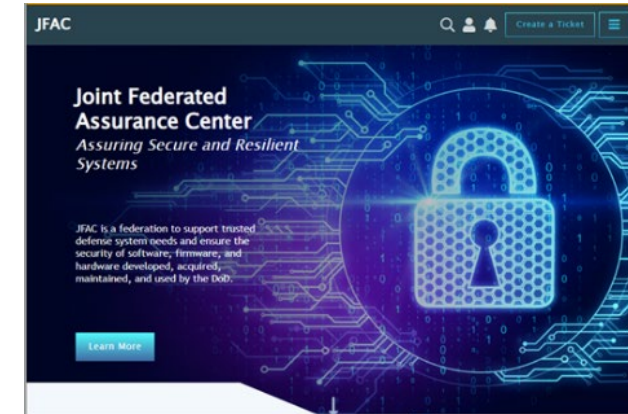
# OVERVIEW OF THE INVESTIGATION



One of the key initial goals was to collect a broad set of data about cybersecurity and digital engineering tools for the JFAC tool portal.



<https://jfac.apps.dso.mil/tools>



This market investigation of cybersecurity design tools was performed by GVSC and System Strategy, Inc on behalf of the Joint Federated Assurance Center (JFAC).



# NARROWING IN ON THE TOOLS TO INVESTIGATE



Digital Engineering & Cybersecurity = 180 tools

Cybersecurity = 107 tools

New System Design = 23 tools

Model-Based = 9 tools

1. Key goal from JFAC was to investigate cybersecurity tools in a digital engineering context.
2. Most of the cybersecurity tools were focused on managing patches for existing IT operations.
3. Of the tools focused on new system design, 9 seemed to be model-based.
4. We talked with all 9 suppliers.
5. We were able to complete the evaluation on three tools.

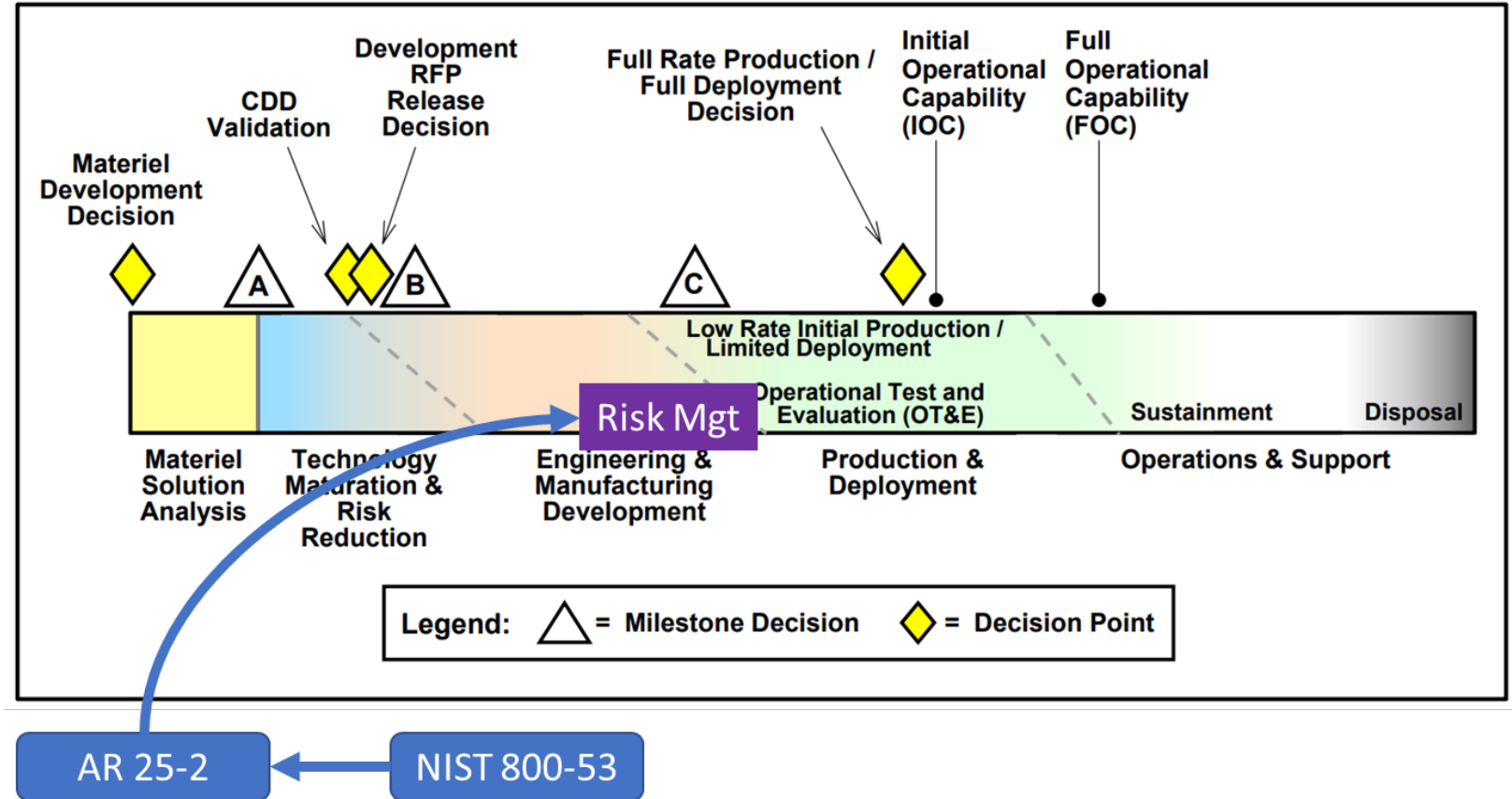
Note: 180 does NOT indicate “all tools in existence”. This was the set we could identify with a manageable amount of effort in a reasonable timeframe.

# CURRENT PRACTICES – RISK MANAGEMENT



Army combat vehicles engineering includes an implementation of the [NIST 800-53] risk management framework as tailored by [AR 25-2] and related policies. This activity occurs from the middle to end of the engineering phase for the system. It does not cover the early phases of the system architecture.

The GVSC RMF team did not feel like they had any particular tool gaps.

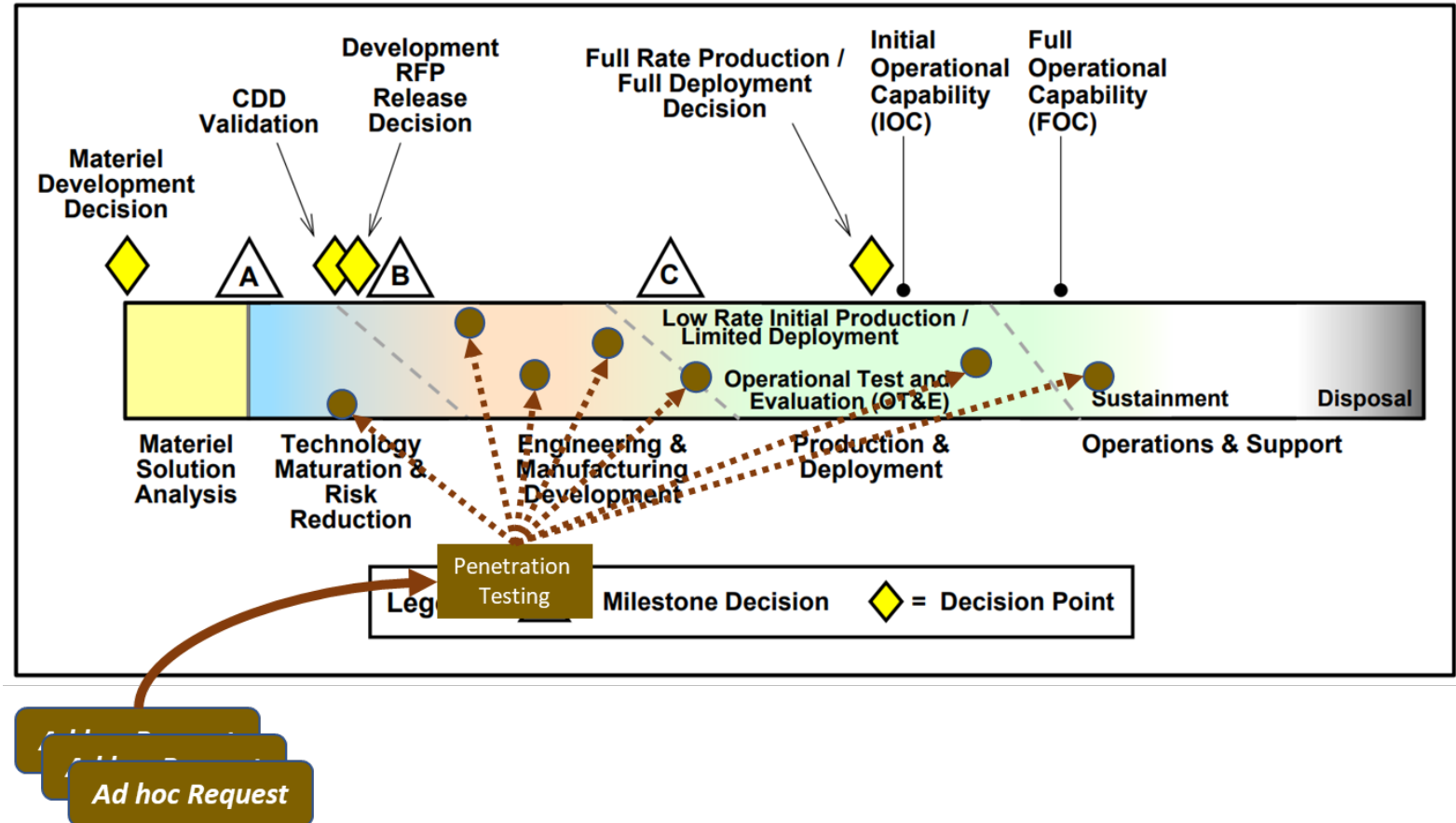


# CURRENT PRACTICES – PENETRATION TESTING



GVSC maintains a team of penetration testing subject matter experts. This team operates as a service to programs and responds to ad hoc requests for penetration testing and vulnerability assessment.

The GVSC penetration testing team did not feel like they had any particular tool gaps.



# DAU GUIDEBOOKS – GENERAL



DAU produces a very helpful series of guidebooks covering the program management and systems engineering of defense systems.

One of these is a cybersecurity best practices guidebook.

<https://aaf.dau.edu/guidebooks/>

**DAU** Home Pathways Policies **Guidance** AAFDID AAF Feedback

## ACQUISITION GUIDEBOOKS

### Acquisition Guidebooks & References

The Defense Acquisition Guidebook has been retired and replaced by a modern set of guidebooks aligned with our new acquisition policies. Identified below are twelve different functional areas fundamental to the operation of the defense acquisition process. Click on them to access specific guidebooks and where available, additional relevant reference materials.

**Cost Estimating**  
[AoA Cost Handbook](#)  
[Additional References](#)

**Cybersecurity**  
[Cybersecurity Best Practices Guidebook](#)  
[Cybersecurity in the AAF](#)

**Engineering**  
[Engineering of Defense Systems Guidebook](#)  
[Systems Engineering Guidebook](#)  
[Additional References](#)

**Human Systems Integration**  
[HSI Guidebook](#)  
[Additional References](#)

[See what's changed recently.](#)



# DAU CYBERSECURITY BEST PRACTICES GUIDEBOOK



**INTRODUCTION**

The National Defense Strategy and the DoD Cyber Strategy both highlight the imperative for the Joint Force to be capable of operating in a contested cyber environment. The Acquisition and Sustainment community has a key role to play in ensuring the weapon systems meet validated cybersecurity requirements and are cyber hardened to deal with cyber threat presented in Validated Online Lifecycle Threat (VOLT) Reports in compliance with DoDI 5000.90, “Cybersecurity for Acquisition Decision Authorities and Program Managers.”

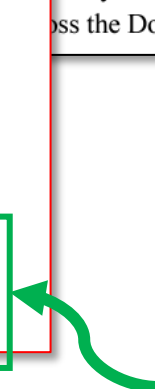
Cyber hardening weapon systems is a daunting challenge for two main reasons. First, program offices have to comply with a lot of cybersecurity policy. By one estimate, there are nearly 23,000 pages of cybersecurity documents that are cybersecurity policies or references to policies<sup>1</sup>. The purpose of this Best Practices Guide is to provide programs with observed effective approaches to complying with DoD the advanced persistent cyber threat. The second challenge is the growing with cybersecurity policies is recognized to be insufficient to stop the advanced ss the DoD.

Red Team exercises good but not scalable.



7.	<b>Red Team Exercises</b> .....	12
7.1	Key Points .....	12
7.2	What is a Red Team Exercise? .....	12
7.3	Red Team EXERCISE Weaknesses .....	13
7.4	Red team EXERCISE Conclusions.....	13

8.	<b>STPA-Sec</b> .....	15
8.1	Key Points .....	15
8.2	Case Study.....	15



23,000 pages of DoD cybersecurity guidance.

Use STPA early and iteratively



# EXAMPLE OF EARLY CYBERSECURITY LOSS IDENTIFICATION



## Credit card skimming at gas stations appears to be increasing

Experts offer several strategies to help motorists protect themselves the next time they fill up their tanks



 **Kristen Dalli**  
Reporter



NOVEMBER 15, 2022

## Vulnerabilities of electric vehicle charging infrastructure

by Sandia National Laboratories



We can identify cybersecurity losses (thief steals credit card number) before we have got down to the detail of whether we are building a gasoline or electric vehicle!

# WHAT IS STPA?



## Chapter 2: How to Do a Basic STPA Analysis

John Thomas

### STPA Method Overview

The steps in basic STPA are shown in Figure 2.1 along with a graphical representation of these steps.

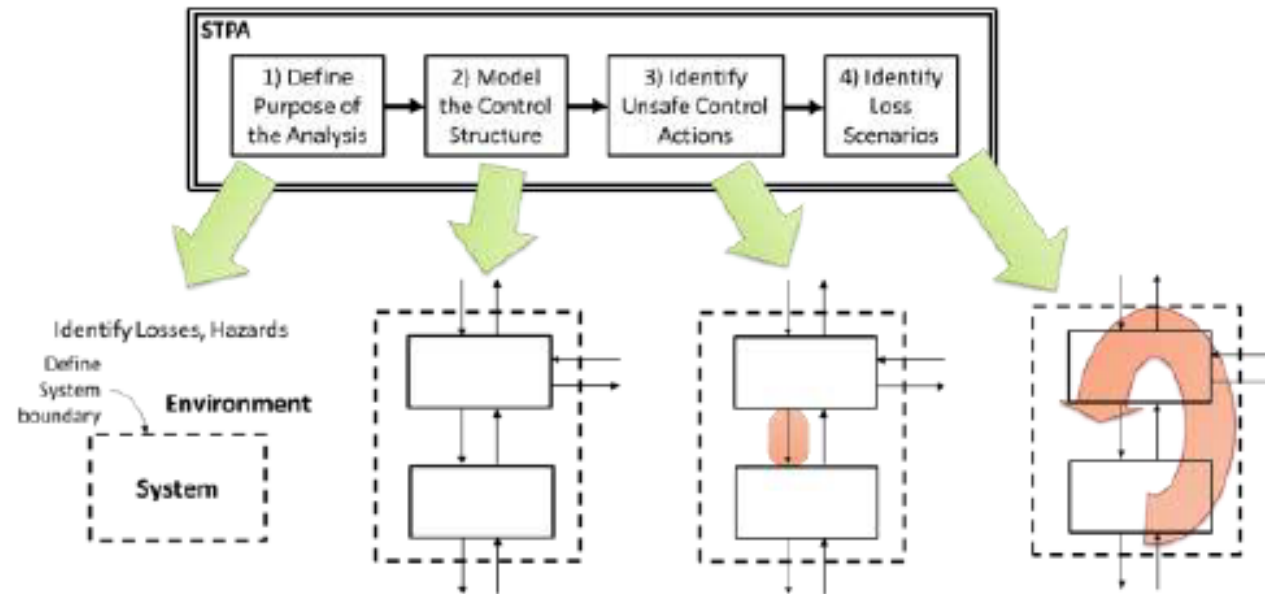


Figure 2.1: Overview of the basic STPA Method



## STPA HANDBOOK

NANCY G. LEVESON  
JOHN P. THOMAS

MARCH 2018

This handbook is intended for those interested in using STPA on real systems. It is not meant to introduce the theoretical foundation, which is described elsewhere. Here our goal is to provide direction for those starting out with STPA on a real project or to supplement other materials in a class teaching STPA.

COPYRIGHT © 2018 BY NANCY LEVESON AND JOHN THOMAS. ALL RIGHTS RESERVED. THE UNALTERED VERSION OF THIS HANDBOOK AND ITS CONTENTS MAY BE USED FOR NON-PROFIT CLASSES AND OTHER NON-COMMERCIAL PURPOSES BUT MAY NOT BE SOLD.

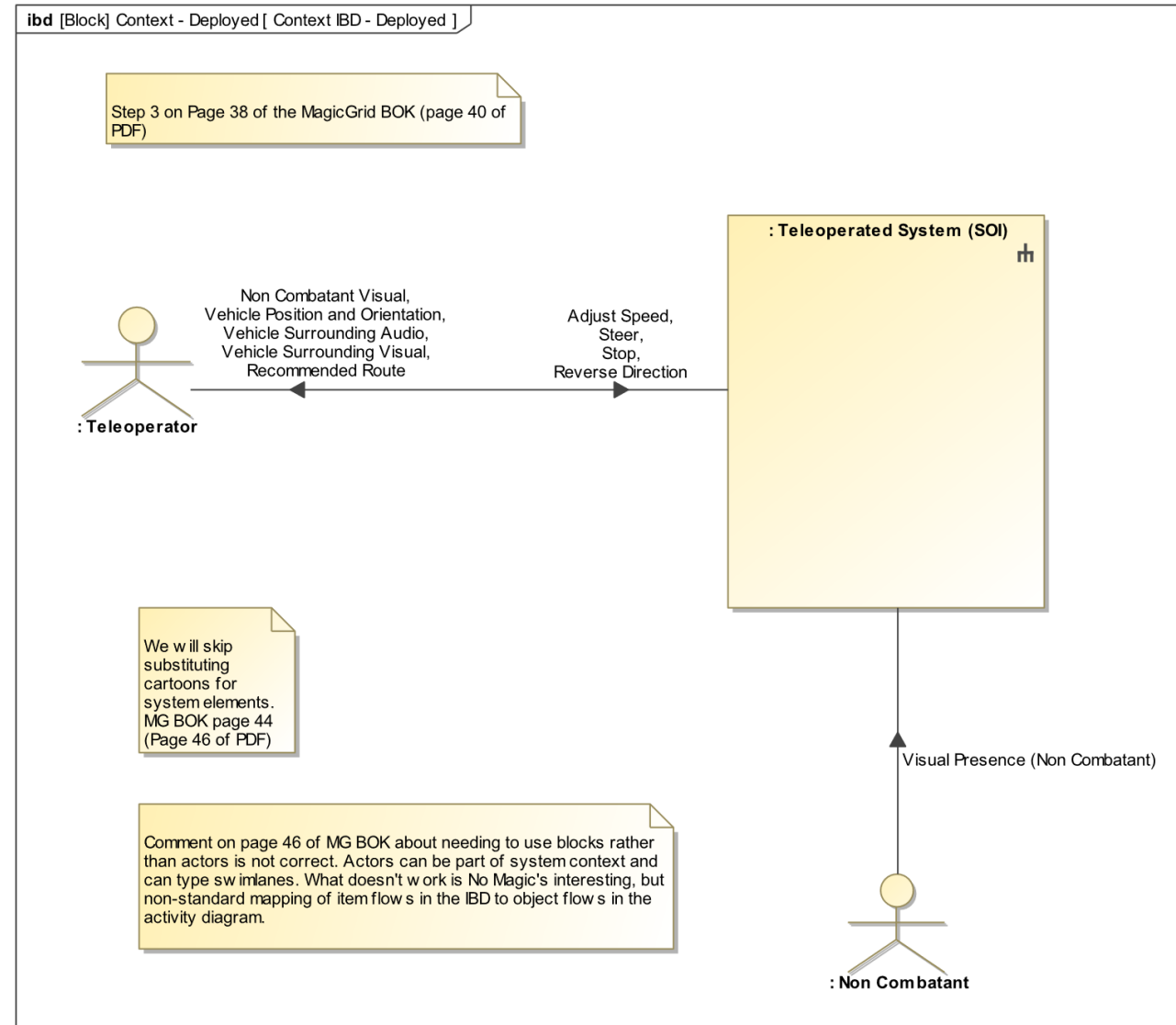
# OUR EXAMPLE SYSTEM MODEL - EXTERNAL



Our example test system is a teleoperated combat vehicle. The main components are:

1. The vehicle
2. Operator Control Unit (OCU)
3. Radio communication

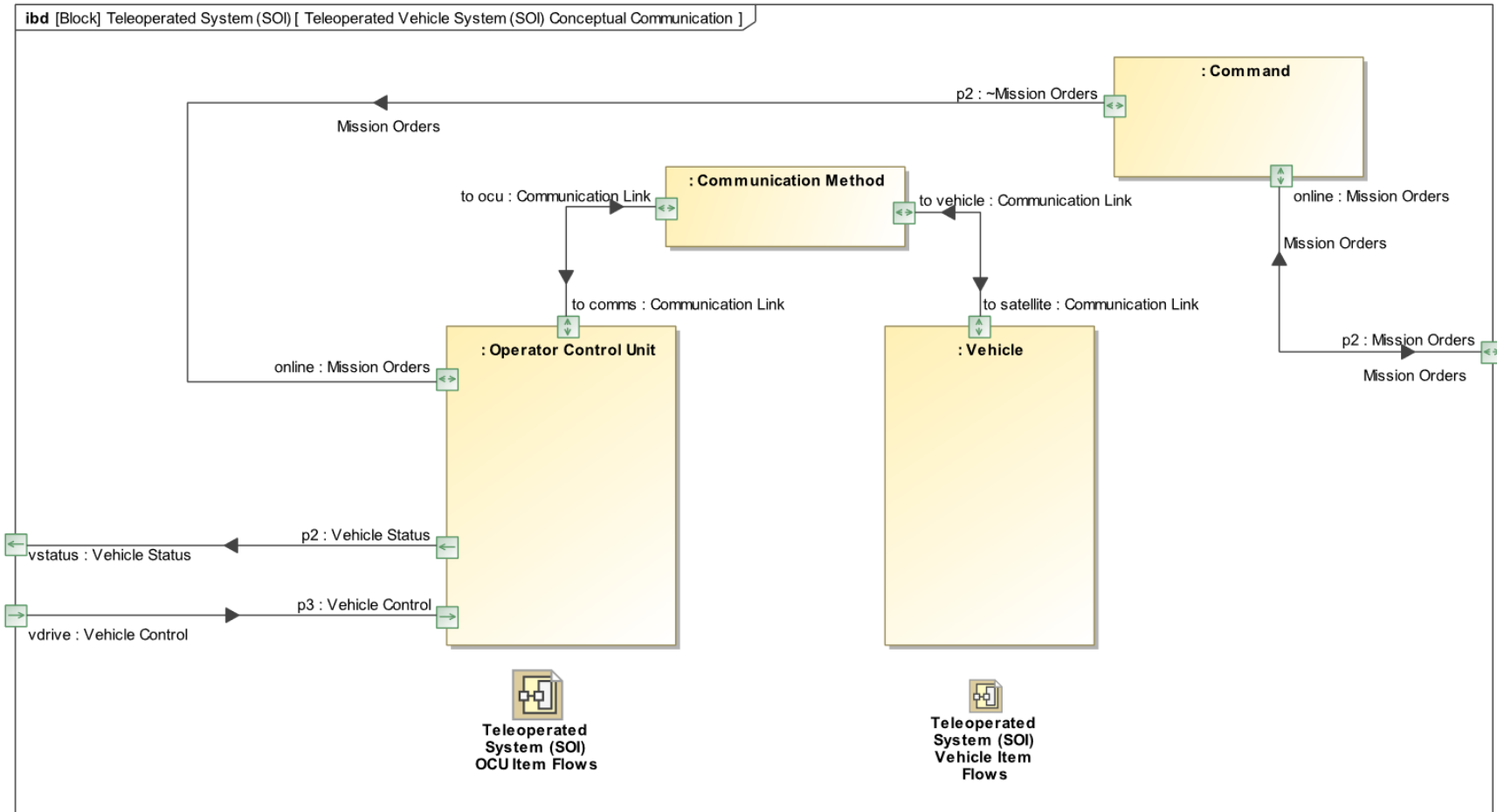
From a system point of view, these are “the system”. The teleoperator and the non combatant are external to the system.



# OUR EXAMPLE SYSTEM MODEL - INTERNAL



Looking inside the system, we see flows between elements. These will be crucial for STPA.

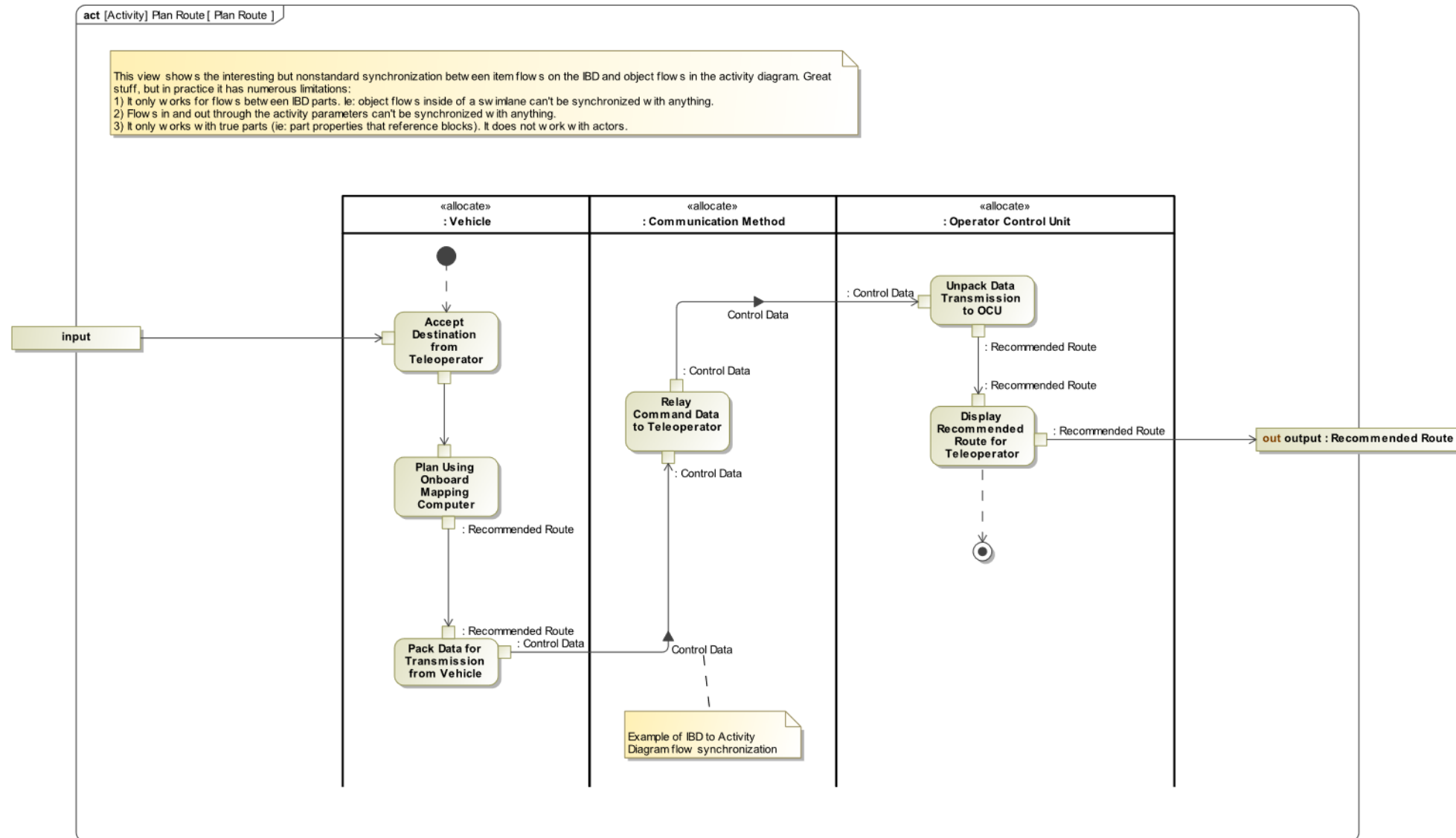


# OUR EXAMPLE SYSTEM MODEL - BEHAVIOR



Here we see the vehicle receiving a new destination, planning a route, and sending the proposed route back to the teleoperator.

(Lots of obvious cyber concerns here)



# SYSTEM MODEL NEEDS – DIDS



- **DI-SESS-8230** – Is for planning of the MBSE activity. In addition to staffing considerations, it covers the choice of modeling framework in 3.3.3. For making a CONOPS-level model that can later be used for cybersecurity and/or safety analysis (with out without STPA) we recommend the choice of:
  - 3.3.3.2. Unified Architecture Framework (UAF)
  - 3.3.3.6. Object Oriented Systems Engineering Methodology (OOSEM)
  - 3.3.3.7. MagicGrid Framework for Systems Modeling Language (SysML) by NoMagic
  
- **DI-SESS-82364** – Covers the content that must be in the model. Again, particular attention would need to be paid to sections: 3.6.2.2, 3.6.3.5, 3.6.3.7, 3.6.5.2, 3.6.5.4, 3.7.1.5, 3.7.2.5, 3.9.3, and 3.9.4



# SYSTEM MODEL NEEDS – THE IMPORTANCE OF BEHAVIOR



- Most modeling teams create a lot of detail about the physical breakdown of the system.
- The reason that most models are overly focused on structure is that structure is very easy to think about.
- Asked to model behavior, many model teams will revert immediately to block structure thinking and break down the functions in the same manner as the physical pieces. (SysML allows this sort of modeling)
- The difficulty is that structure does little or nothing for cybersecurity.
- Cybersecurity (and many safety) problems come out of the interactions in the system. Who does what and when? What information flows back and forth in which sequence?
- This behavior is critical for STPA, but also for cybersecurity and safety in general.
- Modelers **MUST** define use cases, user stories, external interfaces, stakeholders, interactions, and the like early. ....even when being forced to do so makes their heads hurt.

# STPA TOOL GOALS



Goal	
<b>1.4.1 Nesting of Elements</b>	STPA shows nesting of some elements such as hazards. SysML and requirements tools support nesting as well. STPA tools should support nesting, preferably across all elements.
<b>1.4.2 Traceability and Element Structure</b>	In a MBSE environment, elements such as hazards should have ID and separate “name” and “text” fields to be manageable in a project explorer or on a diagram.
<b>1.4.3 Interactive Element Creation</b>	(Minimum) Tools should be able to interactively create each of the necessary STPA element types.
<b>1.4.4 Spreadsheet Element Import</b>	For project scalability and interaction with the digital engineering environment, STPA tools should be able to import all STPA element types from a spreadsheet.
<b>1.4.5 Spreadsheet Element Export</b>	For project scalability and interaction with the digital engineering environment, STPA tools should be able to export all STPA element types to a spreadsheet.
<b>1.4.6 Diagram Export</b>	In order to support the creation of cyber assurance case artifacts, STPA tools should be able to export individual diagrams.
<b>1.4.7 Traceability to Main System Model</b>	STPA tools should be able to trace STPA elements back to corresponding elements in the main systems engineering model.
<b>1.4.8 Relationship Mapping</b>	STPA tools need to be able to conveniently assign all of the different STPA element-to-element relationships.
<b>1.4.9 Define System Boundary</b>	(Shown in Figure 2.2 of the STPA Handbook) STPA tools should support definition of the system boundary.
<b>1.4.10 Diagram Control Structure</b>	STPA tools should be able to produce the core STPA controller diagrams.
<b>1.4.11 Reduced Need for Specialized Skills</b>	STPA tools should reduce the need for scarce specialists. Both Cyber and SysML skills are scarce.

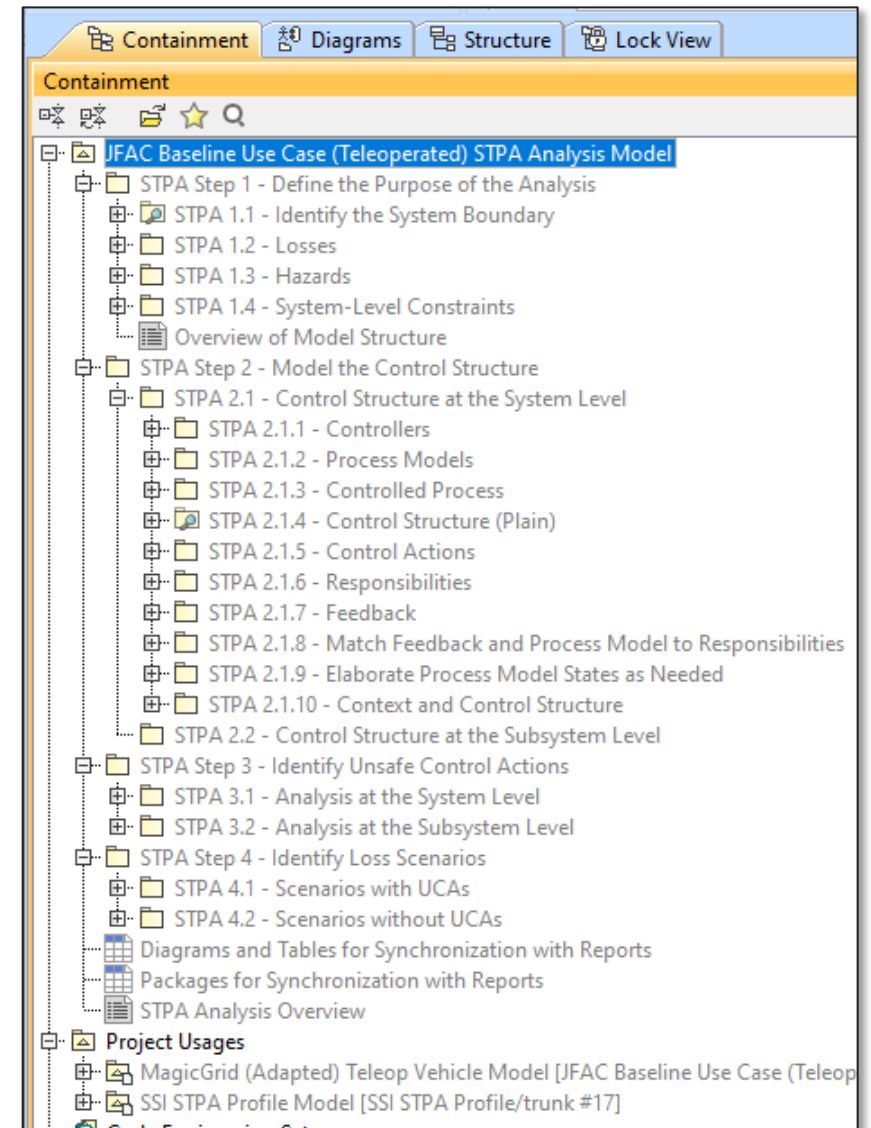
# OUR EXAMPLE STPA ANALYSIS MODEL



The STPA Handbook presents four main steps for STPA.

On closer inspection, however, we identified more than 40 atomic steps to actually complete the analysis.

(Most of the packages shown at the right contain SysML diagrams to support the completion of several atomic steps).



# CREATING THE DETAILED TEST PROCEDURE



At each of the 40+ atomic STPA steps, each tool is evaluated against the relevant subset of the 11 overall tool goals for compliance.

(We found many cases where a tool would meet a goal in one place, but not in others)

Goal	Achieved	Comments
1.4.3 Interactive Element Creation	Yes	Control action elements can be created in the [CEA] containment tree.
1.4.4 Spreadsheet Element Import	Yes	Control action elements can be copied from a spreadsheet and pasted into a [CEA] generic table.
1.4.5 Spreadsheet Element Export	Yes	Control action elements can be exported from either a [CEA] generic table or a [CEA] requirements matrix to a spreadsheet.
1.4.6 Diagram Export	Yes	[CEA] can export all diagrams and tables as SVG files.




#	Id	Name	Text
1	CA-1	 Update Destination	Give vehicle a new destination. This might be the initial destination or an update to a previous destination.
2	CA-2	 Approve Route	Approve the route proposed by the vehicle.
3	CA-3	 Stop	Emergency stop of the vehicle for safety or other reasons. Suspends the current route.

Figure 25 – Control action elements can be copied from a spreadsheet into a CEA generic table

# NARROWING DOWN THE TOOL SELECTION



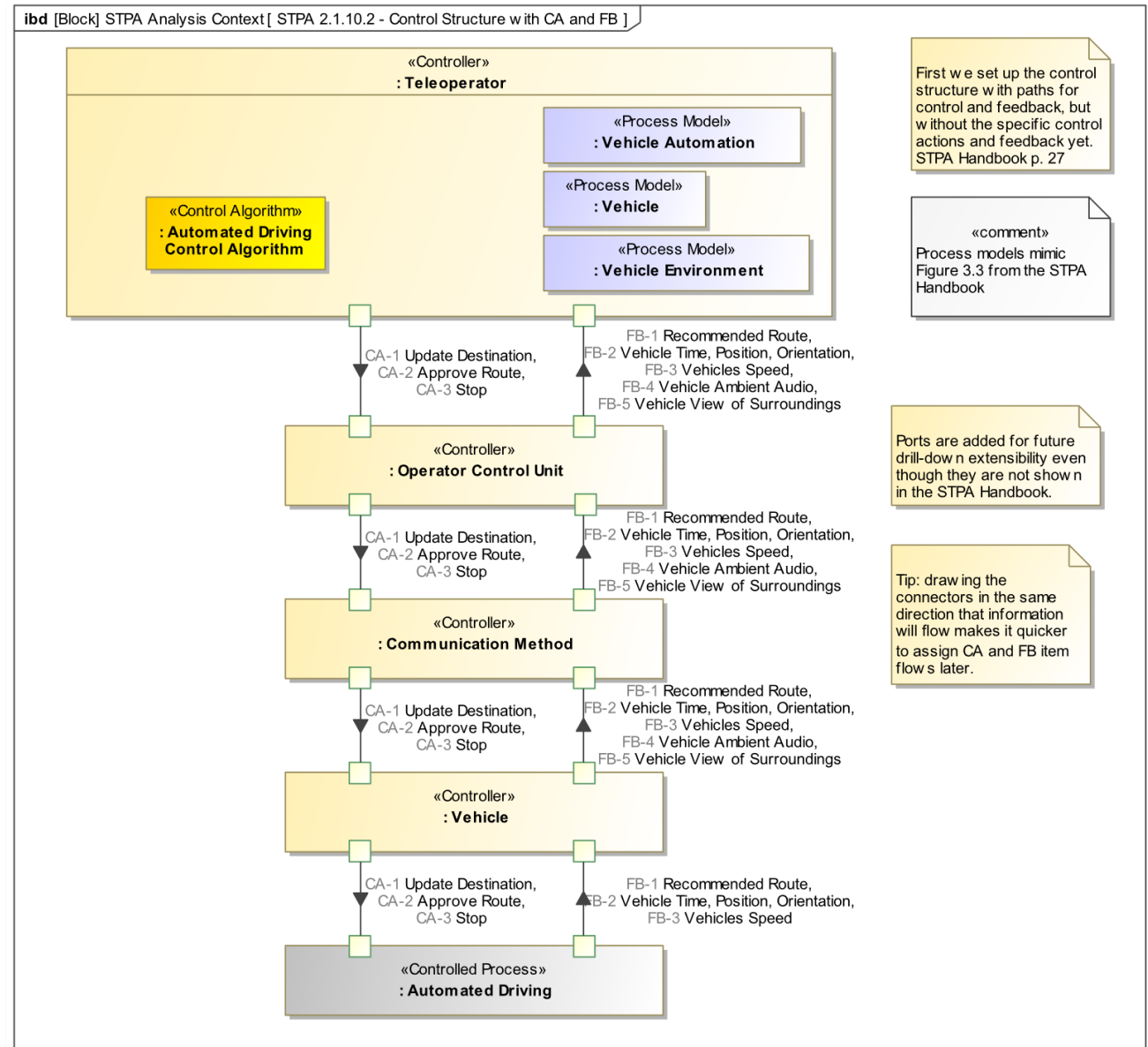
Tool	Phase 2 Result
Tool 1	Evaluation completed
Tool 2	Evaluation completed
Tool 3	Evaluation completed
Tool 4	We had very productive discussions with this supplier. Ultimately concerns about confidentiality of their not-yet-released tool features prevented us from including them in the final evaluation.
Tool 5	We had very productive discussions with this tool supplier. Ultimately, use in a STPA analysis would require some additional features that they were not yet ready to develop and we did not include them in the final evaluation.
Tool 6	This tool is a pure fault tree analysis (FTA) tool. Early on we thought this sort of function would be very helpful for STPA Step 4 “Identify Loss Scenarios”. However, study of the broader set of publications around STPA yielded the insight that the MIT team has strong feelings about FTA not being part of STPA. In order to keep the testing scope clear, we decided to exclude fault tree methods from the evaluation.
Tool 7	Manpower and staffing concerns from the team supporting this tool made it impractical to include this tool in the evaluation.
Tool 8	This supplier’s assessment was that they would need to invest in new features to support STPA and that they had all the business that they could handle from the automotive market from their current tool.
Tool 9	While this open source tool has some interesting concepts, it has only one developer at the moment and additional funding would be needed to bring it to the needed level of function and robustness.

# PROCEDURE EXAMPLE – COMPLETED CONTROL STRUCTURE



In the SysML STPA analysis model, we were able to produce a complete model-based representation of the control structure, control actions, and feedback as shown in the STPA handbook.

The three evaluated tools were able to produce similar diagrams.





# PROCEDURE EXAMPLE – MAPPING OF FEEDBACK TO CONTROLLER RESPONSIBILITIES



Here we have a mapping in the STPA analysis model of feedback to controller responsibilities.

This mapping demonstrates a key strength of the STPA analysis: our system model does not include nearly enough feedback to support all of the controller responsibilities identified in an earlier step!

Legend		STPA 2.1.7 - Feedback				
↗ Supports Responsibility		FB-1 Recommended Route	FB-2 Vehicle Time, Position, Orientation	FB-3 Vehicles Speed	FB-4 Vehicle Ambient Audio	FB-5 Vehicle View of Surroundings
STPA 2.1.6 - Responsibilities						
R-1.1 Monitor Vehicle Health			1	1	3	3
R-1.2 Monitor Vehicle Fuel						
R-1.3 Avoid Problematic Terrain						
R-1.4 Monitor Vehicle Weapons						
R-1.5 Monitor Vehicle Ammunition						
R-2 Mask Vehicle Location						
R-2.1 Mask Vehicle Location (Vehicle)						
R-2.2 Mask Vehicle Location (Communication)						
R-3.1 Mask Teleoperator Location (Operator Control Unit)						
R-3.2 Mask Teleoperator Location (Communication)						
R-4.1 Protect Crew During Mount	2				↗	↗
R-4.2 Protect Crew During Dismount	2				↗	↗
R-5.1 Avoid Noncombatant	4	↗	↗	↗	↗	↗
R-6 Hide Crew Identity						
R-6.1 Hide Crew Personal Identity Information						
R-6.2 Hide Crew Secure Credentials						
R-7 Hide Teleoperator Identity						
R-7.1 Hide Teleoperator Personal Identity Information						
R-7.2 Hide Teleoperator Secure Credentials						

# OVERVIEW OF THE TOOL TESTING RESULTS



Goal	SysML Profile Approach	Tools Tested	Comments
1.4.1 Nesting of Elements	Yes	Partial	Some nesting of elements possible in some places. None of the tools supported universally.
1.4.2 Traceability and Element Structure	Yes	Partial	Some structuring of elements with IDs, Name, and Text in some places. None of the tools supported this structure universally.
1.4.3 Interactive Element Creation	Yes	Yes	All tools have methods to create STPA analysis elements one-at-time.
1.4.4 Spreadsheet Element Import	Yes	Partial	One tool was pretty strong. The other two had limited or no spreadsheet import capability.
1.4.5 Spreadsheet Element Export	Yes	Partial	One tool could export only the entire project. One tool had fairly consistent export. One had limited or no export.
1.4.6 Diagram Export	Yes	Partial	One tool could export a few specific diagrams but did not have a general diagram export mechanism. Another tool mostly only exported Excel. Another had a custom XML export.
1.4.7 Traceability to Main System Model	Yes	No	[CEA] can do this several different ways. None of the other STPA tools tested had a capability to implement federated traceability back to a SysML model.
1.4.8 Relationship Mapping	Yes	Partial	Most of the tools had methods to set relationships. In many cases they were less flexible/convenient than the satisfy matrix approach of [CEA]. In some cases, they could not establish the relationship we needed at all – specifically the mapping of a UCA to a feedback item as specified in [ <a href="#">SAE J3187</a> ].
1.4.9 Define System Boundary	Yes	No	[CEA] can link back to the main system model for the system boundary. The other tools exhibited no features to support definition of a system boundary.
1.4.10 Diagram Control Structure	Yes	Yes	All tools had some sort of diagram that looked reasonably like the [STPA] example diagrams. One tool's diagram was awkward, but usable.
1.4.11 Reduced Need for Specialized Skills	No	Partial	Only one tool gave the impression of being easier to use than the SysML profile supported by [CEA]. The other two tools seemed to be at least as difficult to work with as SysML and [CEA].

# FURTHER DEVELOPMENT OF STPA ITSELF



STPA represents a brilliant paradigm shift away from patching of symptoms at the end of the cycle to thinking about mission loss at the beginning of the cycle. Work remains, however, to continue to shape it into a methodology that can be used in a DoD Digital Engineering environment.

- 1. Supply Chain Partitioning** – While STPA recognizes that subsystems exist, it does not yet present a well-partitioned process that allows the analysis to be handed off from integrator to supplier down the supply chain.
- 2. Control Loop Paradigm** – The control loop paradigm is a huge advance in thinking compared to the hardware fault / security bug mentality that preceded it. Nevertheless, some problems do not fit the control loop mold. In the cyber domain, information leaks are an example that does not fit well into the control loop mold. The system is running fine. All control actions are being executed. All feedback appears normal. Nevertheless, information is leaking. There really isn't a way to install a sensor at enemy headquarters to provide information on cyber leaks.

# EXECUTIVE SUMMARY / CONCLUSIONS



The DAU's Cybersecurity best practices recommendation to apply STPA-Sec early and iteratively is right on target. The core of STPA is a huge advance in thinking about the cybersecurity problem from a mission loss perspective. That having been said, STPA itself and the STPA-based cybersecurity tools are still evolving. Further work will be needed for the method and its associated tools to work smoothly in a DoD digital engineering environment.



# THANK YOU.

Daniel W. Newport  
Branch Chief, Cyber Technology Development (CTD),  
Ground Systems Cyber Engineering (GSCE),  
Ground Vehicle Systems Center (GVSC),  
U.S. Army Combat Capabilities Development Command (CCDC)  
[daniel.w.newport.civ@army.mil](mailto:daniel.w.newport.civ@army.mil)

David Hetherington  
Principal  
System Strategy, Inc  
[dhetherington@systemxi.com](mailto:dhetherington@systemxi.com)



**U.S. ARMY**