# GE – U of Iowa VERDICT / DARPA CASE

## PI Meeting – February 22, 2021

**GRC:** Michael Durling, Kit Siu, Abha Moitra, Paul Meng, John Interrante, Heber Herencia-zapana, Vidhya Tekken-Valapil
**GEAS:** Daniel Prince
**University of Iowa:** Cesare Tinelli, Omar Chowdhury, Daniel Larraz

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

# Agenda

- ❑ VERDICT Program Overview
- ❑ What is the VERDICT Tool?
- ❑ What's new since last PI meeting?
- ❑ Application Lessons Learned
- ❑ Working example
- ❑ Tool availability
- ❑ Publications
- ❑ Questions

# DARPA CASE: VERDICT Program

*Verification Evidence and Resilient Design in Anticipation of Cybersecurity Threats*

## Main Objective

- Develop open-source tools that enable system engineers to design for cyber resiliency and safety under mission scenarios
- Model–based architecture synthesis function generates solutions that satisfy *both safety and security* properties
- Cyber Resiliency Verifier function proves formal cyber resiliency properties using an extension of Kind 2

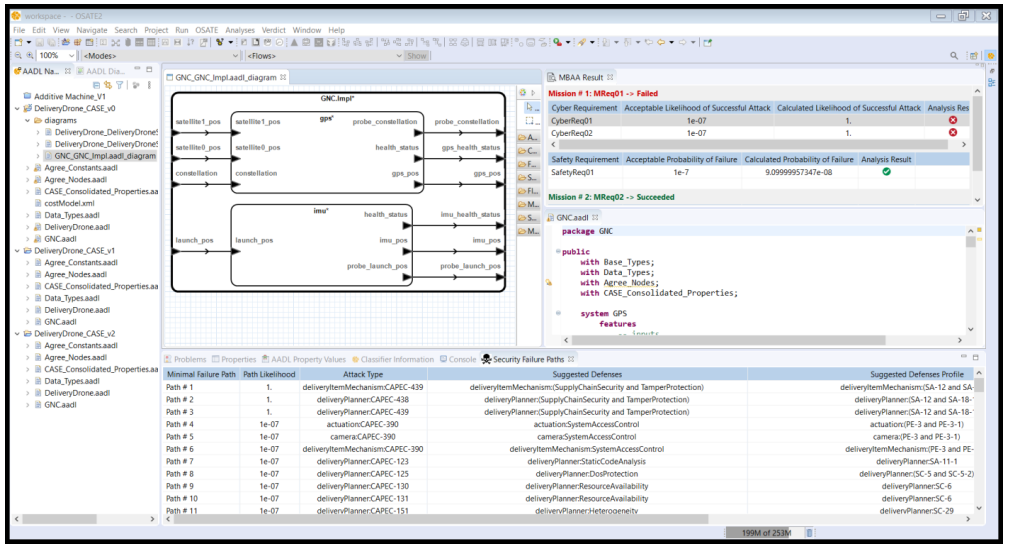
## Innovations

- Synthesis of cyber resilient architectures considering mission, safety and security
- Localized feedback of components responsible for cyber property violation
- Highly automated threat/design model instrumenter
- Ability to reason about future attacks

## Expected Impact

- Clear actionable feedback to systems engineers at design time
- Improved cyber resiliency in military systems
- Open-source tools and documentation for community

## VERDICT Cyber Resilience Design Tool



# What can we do now that we could not do before CASE?

## State of industry before CASE

× Labor-intensive, Slow

× Separate from Systems Modeling and Safety Analysis

× Often Not Threat-based

× Does not consider mission

× Strictly process assurance based

## Today with VERDICT

✓ Automated (Real-time feedback to designers on potential attacks, control suggestions, behavioral weaknesses)

✓ Integrated (Safety and Security analyses in the same tool)

✓ Built-in (Utilizes wealth of data available in models instead of collecting data from various sources)

✓ Mission-Centric (Controls suggested against specific threats that have a direct effect on the mission)

✓ Blended (Benefits from traditional and formal verification)

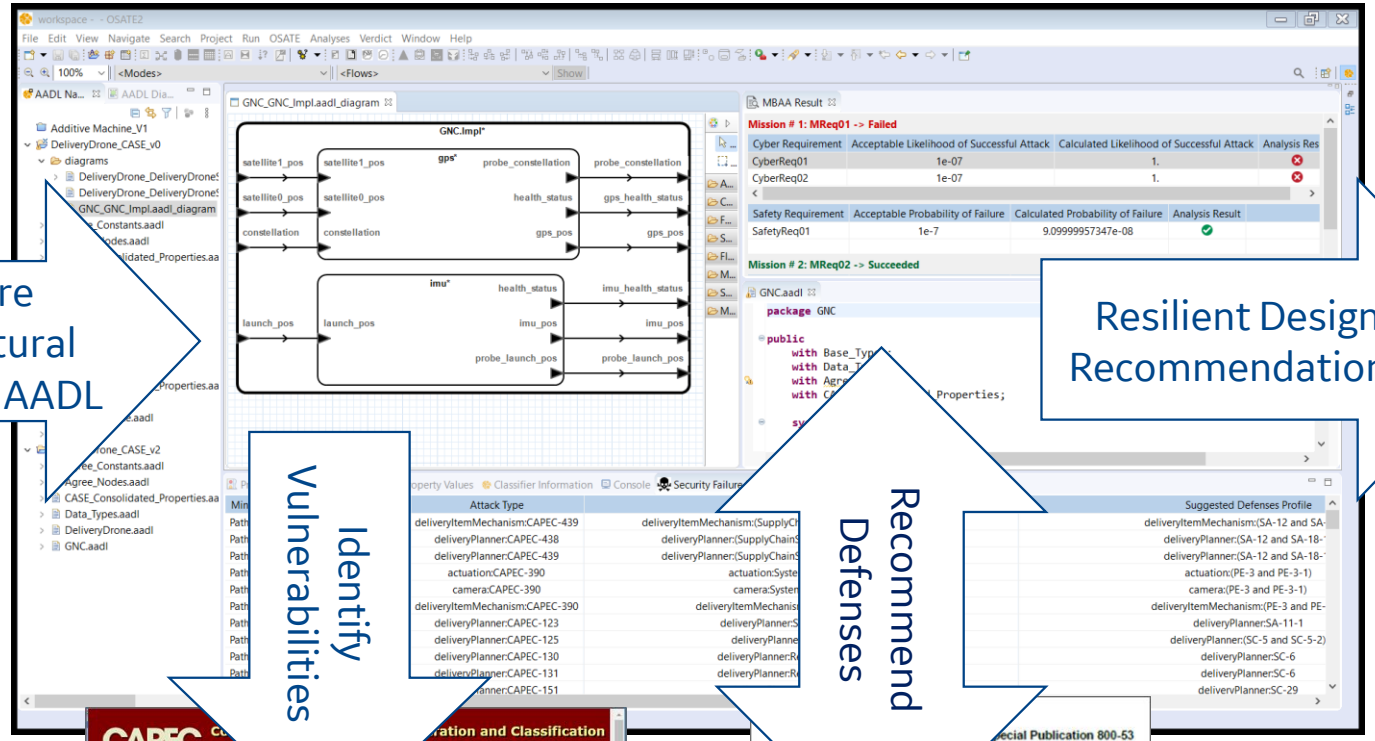
# Model-based identification of Security Threats & Mitigations to enable Cyber Resiliency



Security Requirements & Policies

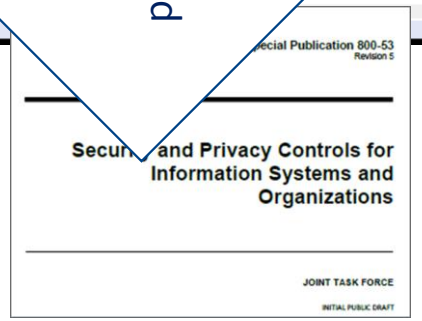
Capture architectural models in AADL

VERDICT Cyber Resilience Design Tool




Identify Vulnerabilities


Recommend Defenses



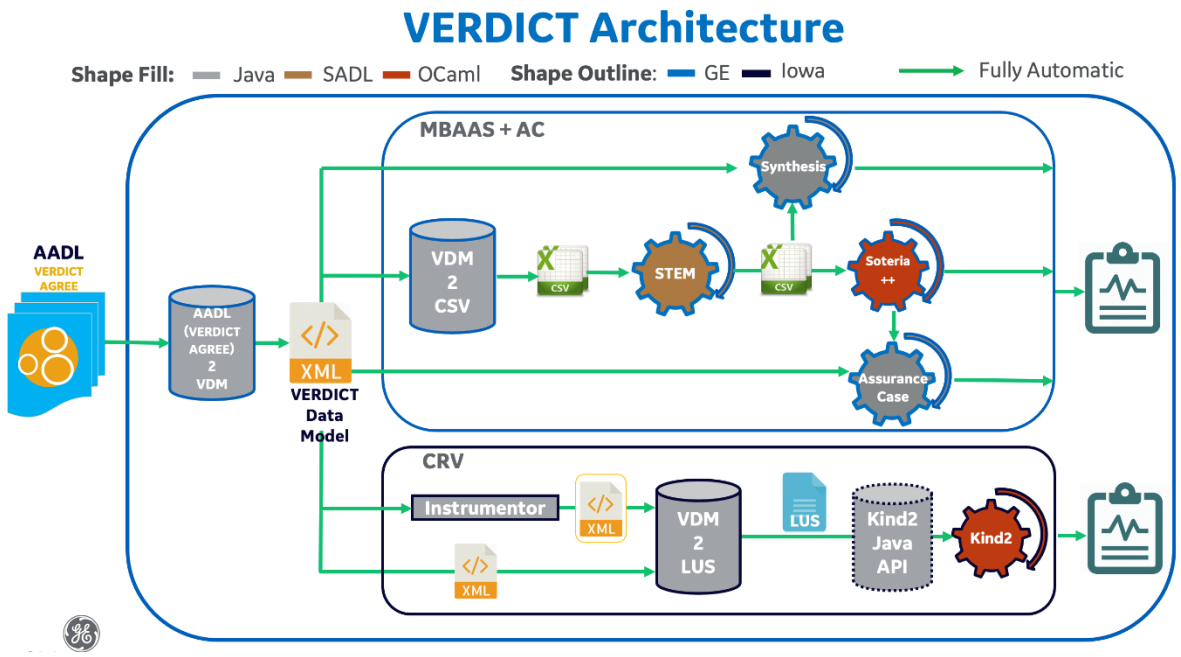

Resilient Design Recommendations



Security Requirements & Policies

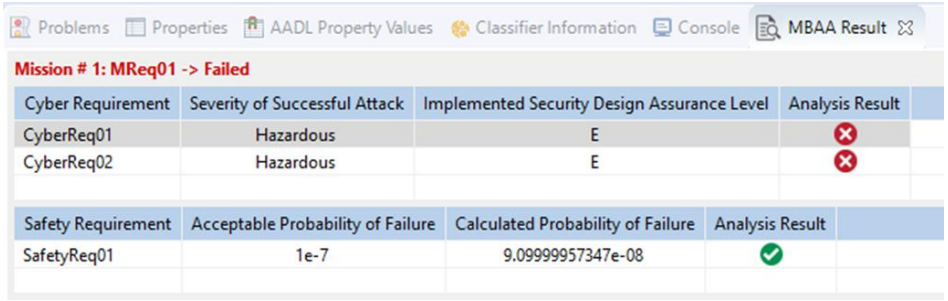
# New features since last PI meeting

## Internal Translator Network & KIND2 Java API




## MBAS User Feedback

| Acceptable Likelihood of Successful Attack | Consequence | Required Security Design Assurance Level |
|---|---|---|
| 1 e -9 | Catastrophic | A |
| 1 e -7 | Hazardous | B |
| 1 e -5 | Major | C |
| 1 e -3 | Minor | D |
| 1 | None | E |



# Aligns with RTCA DO-178C

**Table 2-1 Failure Condition Category Descriptions**

| Category | Description |
|---|---|
| Catastrophic | Failure Conditions, which would result in multiple fatalities, usually with the loss of the airplane. |
| Hazardous | Failure Conditions, which would reduce the capability of the airplane or the ability of the flight crew to cope with adverse operating conditions to the extent that there would be:<br>• A large reduction in safety margins or functional capabilities;<br>• Physical distress or excessive workload such that the flight crew cannot be relied upon to perform their tasks accurately or completely, or<br>• Serious or fatal injury to a relatively small number of the occupants other than the flight crew. |
| Major | Failure Conditions which would reduce the capability of the airplane or the ability of the crew to cope with adverse operating conditions to the extent that there would be, for example, a significant reduction in safety margins or functional capabilities, a significant increase in crew workload or in conditions impairing crew efficiency, or discomfort to the flight crew, or physical distress to passengers or cabin crew, possibly including injuries. |
| Minor | Failure Conditions which would not significantly reduce airplane safety, and which involve crew actions that are well within their capabilities. Minor Failure Conditions may include, for example, a slight reduction in safety margins or functional capabilities, a slight increase in crew workload, such as routine flight plan changes, or some physical discomfort to passengers or cabin crew. |
| No Safety Effect | Failure Conditions that would have no effect on safety; for example, Failure Conditions that would not affect the operational capability of the airplane or increase crew workload. |

| Acceptable Likelihood of Successful Attack | Consequence | Required Security Design Assurance Level |
|---|---|---|
| 1 e -9 | Catastrophic | A |
| 1 e -7 | Hazardous | B |
| 1 e -5 | Major | C |
| 1 e -3 | Minor | D |
| 1 | None | E |

**Software Level Definition**

This document recognizes five software levels, Level A to Level E. For the example failure condition categories listed in section 2.3.2, the relationships between these software levels and failure conditions are:

a. Level A: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft.

b. Level B: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a hazardous failure condition for the aircraft.

c. Level C: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a major failure condition for the aircraft.

d. Level D: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a minor failure condition for the aircraft.

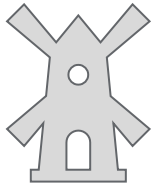
e. Level E: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function with no effect on aircraft operational capability or pilot workload. If a software component is determined to be Level E and this is confirmed by the certification authority, no further guidance contained in this document applies.

The applicant should always consider the appropriate certification guidance and system development considerations for categorizing the failure condition severity and the software level.

Source: RTCA DO-178C and DO-356-A

# Application lessons learned

## Applications

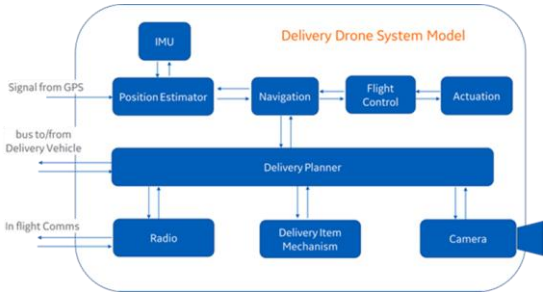





## Capture Initial Models


Domain Expert


Modeling Expert

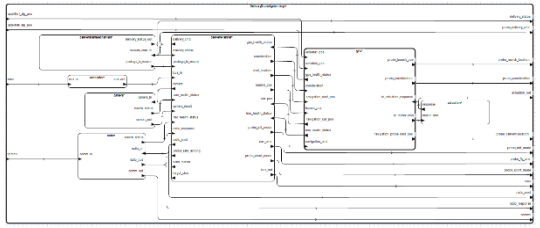



- Develop initial model - key architectural components & information flow
- Make thoughtful choices on signal names and level of abstract
- Create and document the story of how the system works (Use Case)
- Start at high level and refine model
- Identify “mission requirements” and consequence of successful attack
- Modeling learning curve is modest
- Include open ports and physical security

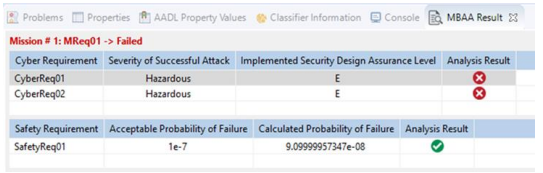
## Capture AADL Models


Domain Expert


Modeling Expert



```
radio: system Radio
{
    -- VERDICT Component Properties

    CASE_Consolidated_Properties::canReceiveConfigUpdate => true;
    CASE_Consolidated_Properties::canReceiveSWUpdate => true;
    CASE_Consolidated_Properties::componentType => Hybrid;
    CASE_Consolidated_Properties::hasSensitiveInfo => true;
    CASE_Consolidated_Properties::insideTrustedBoundary => true;
    CASE_Consolidated_Properties::pedigree => COTS;
```



Problems | Properties | AADL Property Values | Classifier Information | Console | MBAA Result

Mission # 1: MReq01 -> Failed

| Cyber Requirement | Severity of Successful Attack | Implemented Security Design Assurance Level | Analysis Result |
|---|---|---|---|
| CyberReq01 | Hazardous | E | ✗ |
| CyberReq02 | Hazardous | E | ✗ |

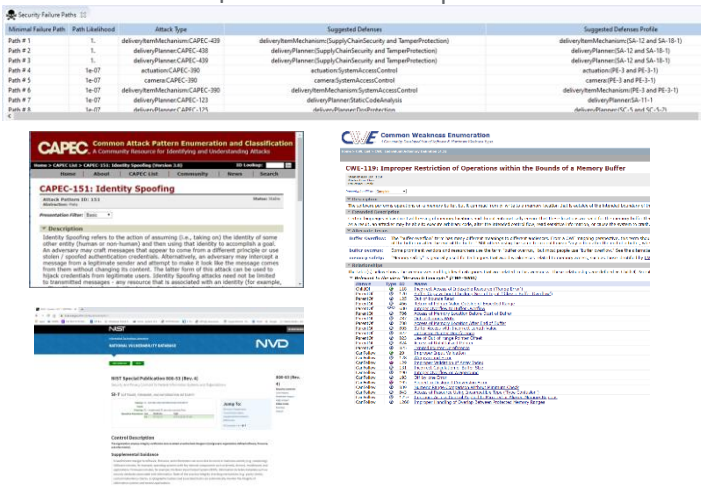
| Safety Requirement | Acceptable Probability of Failure | Calculated Probability of Failure | Analysis Result |
|---|---|---|---|
| SafetyReq01 | 1e-7 | 9.09999957347e-08 | ✓ |

- Set the CASE Properties on the components and connections
- Run Architectural Analysis
- Experiment with settings

## Analyze Results & Update Models


Domain Expert


Cybersecurity Expert



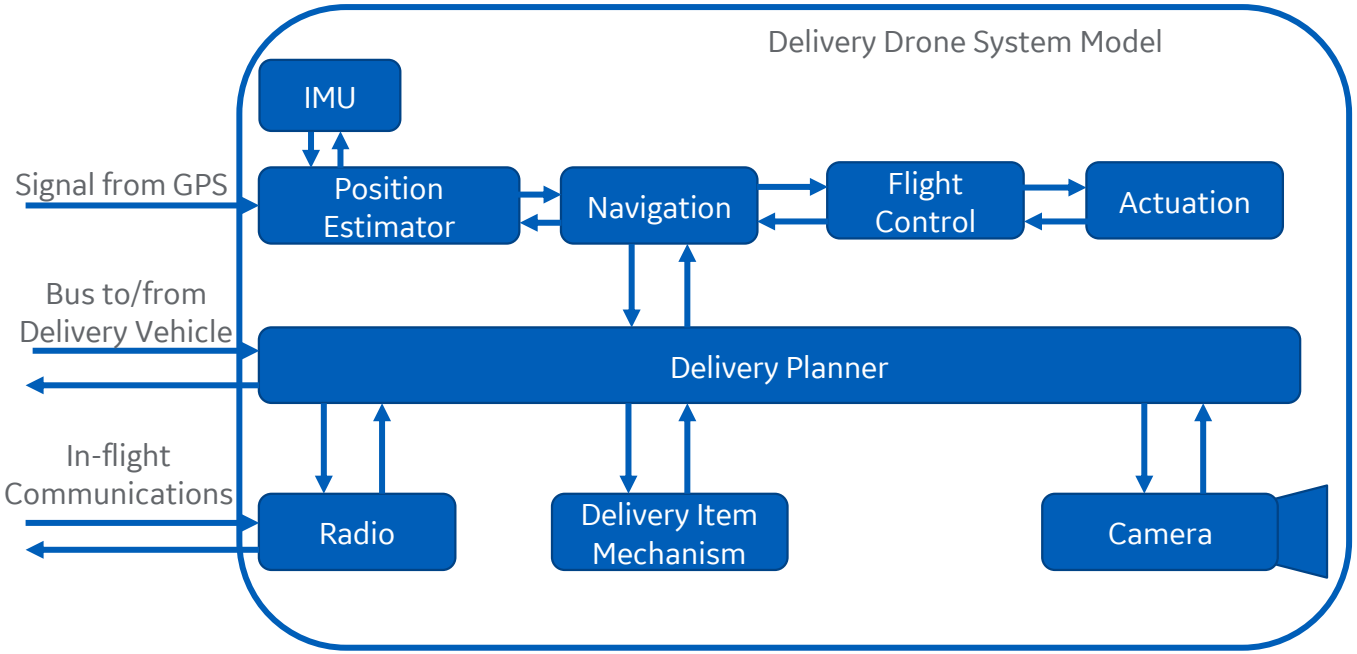

- Drill down in CAPEC’s to CWE’s – excellent resources
- Drill down in NIST 800-53 compare notes with CAPEC and CWE recommendations
- VERDICT information appropriate for Threat Assessment document and implementation requirements
- Review results with cybersecurity expert
- Add formal properties, behavioral model, run CRV
- Use VERDICT feedback to annotate and analyze implementation code

VERDICT enables, guides and facilitates system resiliency analysis … it does not blindly automate it

# Working Example

# Demo: Delivery Drone

- Scenario: a truck with packages to be delivered using one or more delivery drones.
- Truck arrives at a location close to multiple delivery sites. Delivery drones are initialized with their current position, delivery locations, and loaded with the package.
- Drone uses:
  - Inputs from GPS and IMU to navigate
  - Camera to capture an image of receiving site and to confirm site is free of obstacles
  - Radio to get confirmation from truck operator if delivering a high-value package
- Delivery Planner activates the Delivery Item Mechanism to drop off the package.



# Model Based Architecture Analysis (MBAA) Capability

# Develop the Architectural Model in AADL

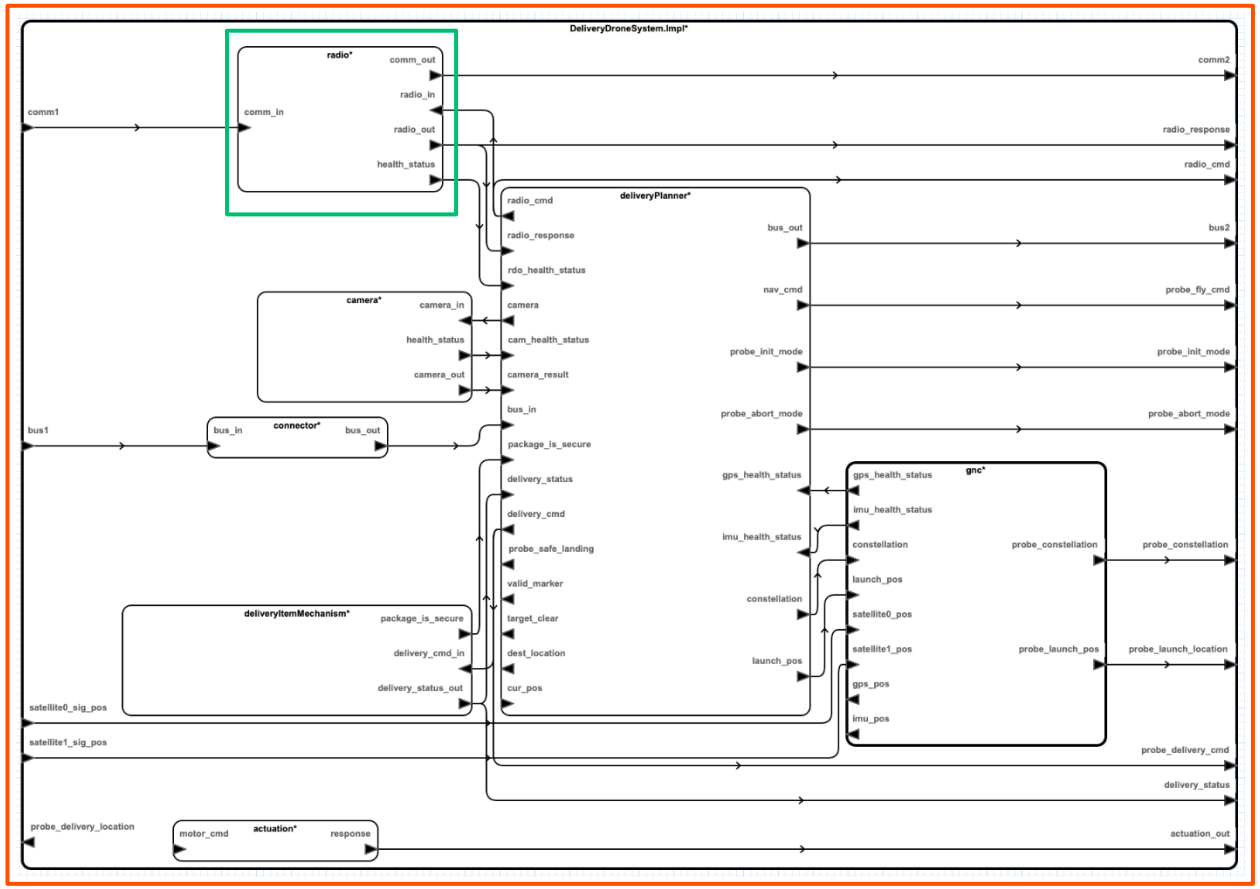
## Define architectural model in AADL

- Define components interfaces, subcomponents of the system, and internal connectivity

```
system Radio
    features
        comm_in: in data port Data_Types::RadioResponse.impl;
        comm_out: out data port Base_Types::Boolean;
end Radio;
```

```
system DeliveryDroneSystem
    features
        comm1: in data port Data_Types::RadioResponse.impl;
        comm2: out data port Base_Types::Boolean;
end DeliveryDroneSystem;

system implementation DeliveryDroneSystem.Impl
    subcomponents
        radio: system Radio;
    connections
        c16: port comm1 -> radio.comm_in;
        c17: port radio.comm_out -> comm2;
end DeliveryDroneSystem.impl;
```

## AADL Auto-generated diagram for Delivery Drone



# Capture Mission, Cyber and Safety Requirements with VERDICT annex

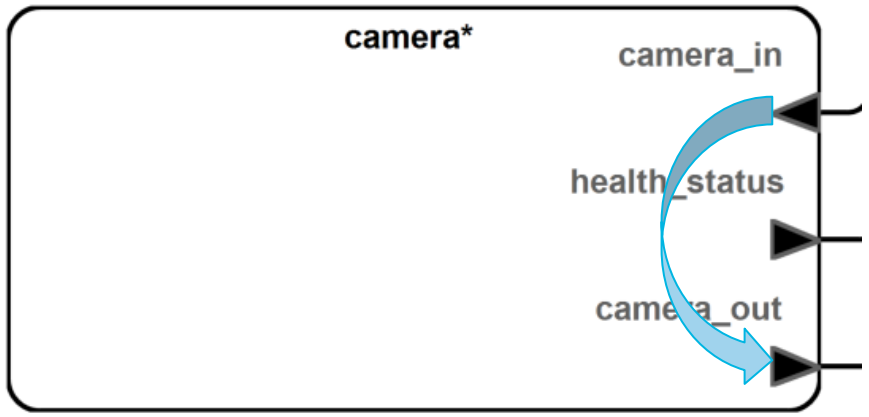
```
annex verdict{**
    CyberReq {
        id = "CyberReq01"
        description = "The drone shall be resilient to loss of
                  ability to deliver a package to
                  the appropriate consumer location"
        condition = actuation_out:I or actuation_out:A or
                   delivery_status:I or delivery_status:A
        cia = I
        severity = Hazardous
    };
    SafetyReq {
        id = "SafetyReq01"
        description = "Loss of actuation shall be less than 1e-7 pfh"
        condition = actuation_out:A
        targetProbability = 1e-07
    };
    MissionReq {
        id ="MReq01"
        description = "Deliver a package to the intended location."
        reqs = "CyberReq01", "SafetyReq01"
    };
**};
```

Guidance for Setting **Severity** for Cyber Requirements

| Consequence | Required Security Design Assurance Level (DAL) | Acceptable Likelihood of Successful Attack |
|---|---|---|
| Catastrophic | A | 1e-9 |
| Hazardous | B | 1e-7 |
| Major | C | 1e-5 |
| Minor | D | 1e-3 |
| None | E | 1e-0 |

Terminologies used in avionic software development

# Set Cyber and Safety Relations




- **CyberRel:** security vulnerability propagations
- **SafetyRel:** safety vulnerability propagations
- **Event:** undesirable events

Cyber and Safety Relations and Events Modeling for Camera

```
system Camera
    features
        camera_in: in data port Base_Types::Boolean;
        camera_out: out data port Base_Types::Integer;
        health_status: out data port Base_Types::Boolean;
    annex verdict {**

        CyberRel "camera_out_I" = camera_in:I => camera_out:I;
        CyberRel "camera_out_A" = camera_in:A => camera_out:A;

        Event {
            id = "ued_event"
            probability = 1.0e-9
            comment = "undetected erroneous data of the Camera"
            description = "UED"
        }
        SafetyRel "camera_out_UED" = happens("ued_event") or camera_in:I
                                     => camera_out:I;

    **};
end Camera;
```

# Set VERDICT Properties

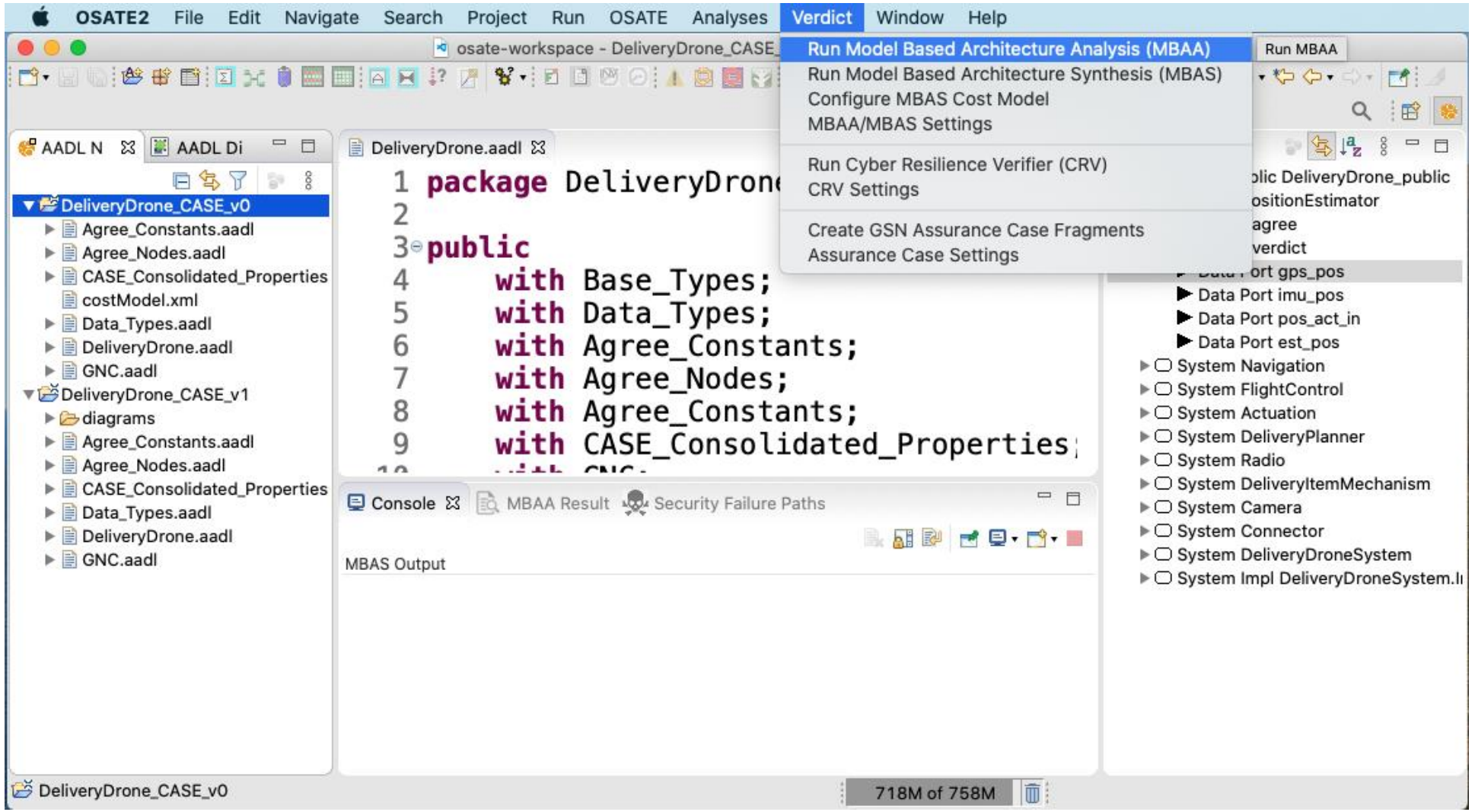
## Set Properties in AADL for deliveryPlanner

```
deliveryPlanner: system DeliveryPlanner
{
    -- VERDICT Component Properties
    CASE_Consolidated_Properties::canReceiveSWUpdate => true;
    CASE_Consolidated_Properties::componentType => Software;
    CASE_Consolidated_Properties::hasSensitiveInfo => true;
    CASE_Consolidated_Properties::insideTrustedBoundary => true;
    CASE_Consolidated_Properties::pedigree => Sourced;

    -- VERDICT Cyber Defense and DAL Mitigations
    CASE_Consolidated_Properties::antiJamming => 7;
    CASE_Consolidated_Properties::dosProtection => 7;
    CASE_Consolidated_Properties::failSafe =>7;
};
```

## Guide for Setting DAL for Defense Properties

| Required Security Design Assurance Level | DAL Number for Defense Property |
|---|---|
| A | 9 |
| B | 7 |
| C | 5 |
| D | 3 |
| E | 0 |

# Run VERDICT MBAA



# Review VERDICT tool user feedback

## MBAA Result

Console | MBAA Result

**Mission # 1: MReq01 -> Failed**

| Cyber Requirement | Severity of Successful Attack | Implemented Security Design Assurance Level | Analysis Result |
|---|---|---|---|
| CyberReq01 | Hazardous | E | ✗ |
| CyberReq02 | Hazardous | E | ✗ |

| Safety Requirement | Acceptable Probability of Failure | Calculated Probability of Failure | Analysis Result |
|---|---|---|---|
| SafetyReq01 | 1e-7 | 9.09999957347e-08 | ✓ |

| Acceptable Likelihood of Successful Attack | Consequence | Required Security Design Assurance Level |
|---|---|---|
| 1 e -9 | Catastrophic | A |
| 1 e -7 | Hazardous | B |
| 1 e -5 | Major | C |
| 1 e -3 | Minor | D |
| 1 | None | E |

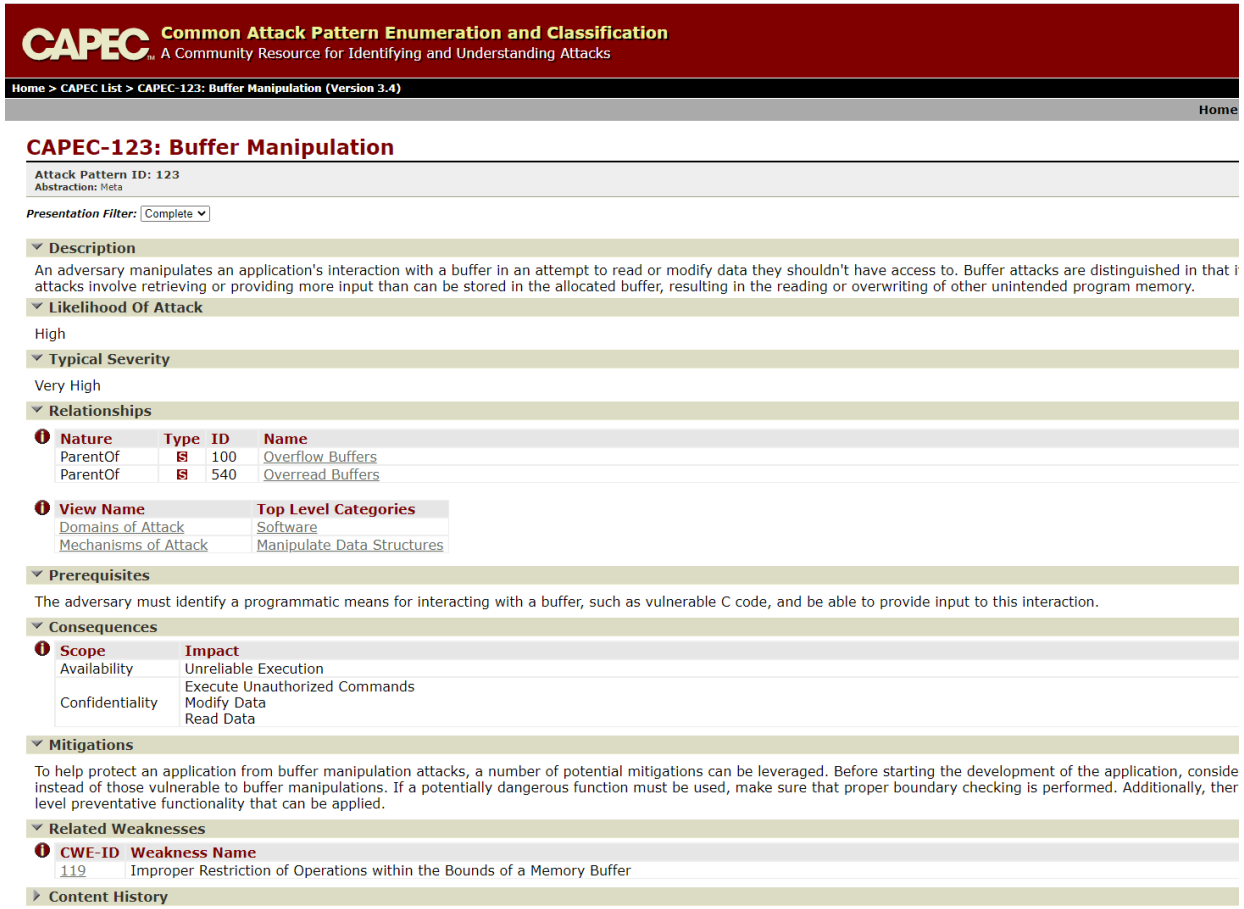
## Failure Paths

Console | MBAA Result | Security Failure Paths

| Minimal Failure Path | Path Likelihood | Attack Type | Suggested Defenses | Suggested Defenses Profile | Implemented Defenses |
|---|---|---|---|---|---|
| Path # 1 | 1. | deliveryPlanner:CAPEC-123 | deliveryPlanner:StaticCodeAnalysis | deliveryPlanner:SA-11-1 | |
| Path # 2 | 1. | deliveryPlanner:CAPEC-25 | deliveryPlanner:StaticCodeAnalysis | deliveryPlanner:SA-11-1 | |
| Path # 3 | 1. | deliveryPlanner:CAPEC-26 | deliveryPlanner:StaticCodeAnalysis | deliveryPlanner:SA-11-1 | |
| Path # 4 | 1. | deliveryPlanner:CAPEC-74 | deliveryPlanner:(InputValidation and StaticCodeAnalysis) | deliveryPlanner:(SA-11-1 and SI-10 and SI-10-5) | deliveryPlanner:inputValidation |
| Path # 5 | 1e-07 | actuation:CAPEC-390 | actuation:SystemAccessControl | actuation:(PE-3 and PE-3-1) | actuation:systemAccessControl |
| Path # 6 | 1e-07 | camera:CAPEC-390 | camera:SystemAccessControl | camera:(PE-3 and PE-3-1) | camera:systemAccessControl |
| Path # 7 | 1e-07 | deliveryItemMechanism:CA... | deliveryItemMechanism:SystemAccessControl | deliveryItemMechanism:(PE-3 and PE-3-1) | deliveryItemMechanism:systemA... |
| Path # 8 | 1e-07 | deliveryPlanner:CAPEC-125 | deliveryPlanner:DosProtection | deliveryPlanner:(SC-5 and SC-5-2) | deliveryPlanner:dosProtection |
| Path # 9 | 1e-07 | deliveryPlanner:CAPEC-130 | deliveryPlanner:ResourceAvailability | deliveryPlanner:SC-6 | deliveryPlanner:resourceAvailability |
| Path # 10 | 1e-07 | deliveryPlanner:CAPEC-131 | deliveryPlanner:ResourceAvailability | deliveryPlanner:SC-6 | deliveryPlanner:resourceAvailability |
| Path # 11 | 1e-07 | deliveryPlanner:CAPEC-151 | deliveryPlanner:Heterogeneity | deliveryPlanner:SC-29 | deliveryPlanner:heterogeneity |
| Path # 12 | 1e-07 | deliveryPlanner:CAPEC-28 | deliveryPlanner:InputValidation | deliveryPlanner:(SI-10 and SI-10-5) | deliveryPlanner:inputValidation |
| Path # 13 | 1e-07 | gps:CAPEC-390 | gps:SystemAccessControl | gps:(PE-3 and PE-3-1) | gps:systemAccessControl |
| Path # 14 | 1e-07 | imu:CAPEC-390 | imu:SystemAccessControl | imu:(PE-3 and PE-3-1) | imu:systemAccessControl |
| Path # 15 | 1e-07 | positionEstimator:CAPEC-123 | positionEstimator:StaticCodeAnalysis | positionEstimator:SA-11-1 | positionEstimator:staticCodeAnal... |
| Path # 16 | 1e-07 | positionEstimator:CAPEC-125 | positionEstimator:DosProtection | positionEstimator:(SC-5 and SC-5-2) | positionEstimator:dosProtection |
| Path # 17 | 1e-07 | positionEstimator:CAPEC-130 | positionEstimator:ResourceAvailability | positionEstimator:SC-6 | positionEstimator:resourceAvaila... |

Clear feedback on each mission, cyber and safety requirement – localized to components

# CAPEC library



CAPEC Common Attack Pattern Enumeration and Classification
A Community Resource for Identifying and Understanding Attacks

Home > CAPEC List > CAPEC-123: Buffer Manipulation (Version 3.4)

Home

## CAPEC-123: Buffer Manipulation

**Attack Pattern ID: 123**
**Abstraction:** Meta

*Presentation Filter:* Complete

**Description**

An adversary manipulates an application's interaction with a buffer in an attempt to read or modify data they shouldn't have access to. Buffer attacks are distinguished in that i attacks involve retrieving or providing more input than can be stored in the allocated buffer, resulting in the reading or overwriting of other unintended program memory.

**Likelihood Of Attack**

High

**Typical Severity**

Very High

**Relationships**

| Nature | Type | ID | Name |
|---|---|---|---|
| ParentOf | S | 100 | Overflow Buffers |
| ParentOf | S | 540 | Overread Buffers |

| View Name | Top Level Categories |
|---|---|
| Domains of Attack | Software |
| Mechanisms of Attack | Manipulate Data Structures |

**Prerequisites**

The adversary must identify a programmatic means for interacting with a buffer, such as vulnerable C code, and be able to provide input to this interaction.

**Consequences**

| Scope | Impact |
|---|---|
| Availability | Unreliable Execution |
| Confidentiality | Execute Unauthorized Commands<br>Modify Data<br>Read Data |

**Mitigations**

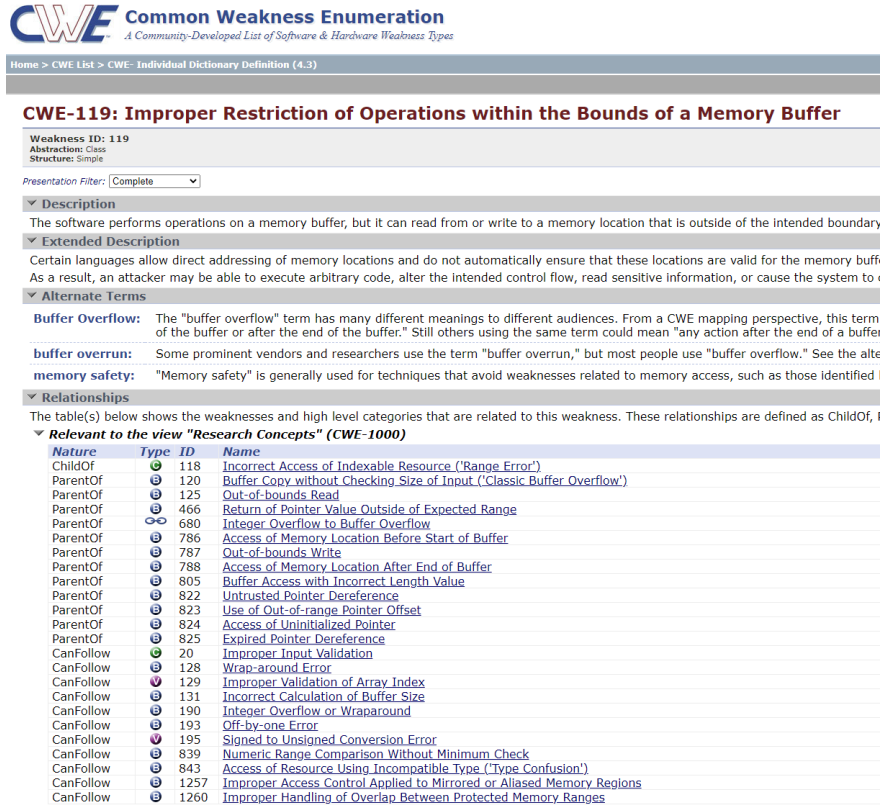
To help protect an application from buffer manipulation attacks, a number of potential mitigations can be leveraged. Before starting the development of the application, conside instead of those vulnerable to buffer manipulations. If a potentially dangerous function must be used, make sure that proper boundary checking is performed. Additionally, ther level preventative functionality that can be applied.

**Related Weaknesses**

| CWE-ID | Weakness Name |
|---|---|
| 119 | Improper Restriction of Operations within the Bounds of a Memory Buffer |

**Content History**

[https://capec.mitre.org/data/definitions/123.html](https://capec.mitre.org/data/definitions/123.html)



CWE Common Weakness Enumeration
A Community-Developed List of Software & Hardware Weakness Types

Home > CWE List > CWE- Individual Dictionary Definition (4.3)

## CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer

**Weakness ID: 119**
**Abstraction:** Class
**Structure:** Simple

*Presentation Filter:* Complete

**Description**

The software performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary

**Extended Description**

Certain languages allow direct addressing of memory locations and do not automatically ensure that these locations are valid for the memory buff
As a result, an attacker may be able to execute arbitrary code, alter the intended control flow, read sensitive information, or cause the system to

**Alternate Terms**

**Buffer Overflow:** The "buffer overflow" term has many different meanings to different audiences. From a CWE mapping perspective, this term of the buffer or after the end of the buffer." Still others using the same term could mean "any action after the end of a buffe

**buffer overrun:** Some prominent vendors and researchers use the term "buffer overrun," but most people use "buffer overflow." See the alte

**memory safety:** "Memory safety" is generally used for techniques that avoid weaknesses related to memory access, such as those identified

**Relationships**

The table(s) below shows the weaknesses and high level categories that are related to this weakness. These relationships are defined as ChildOf,

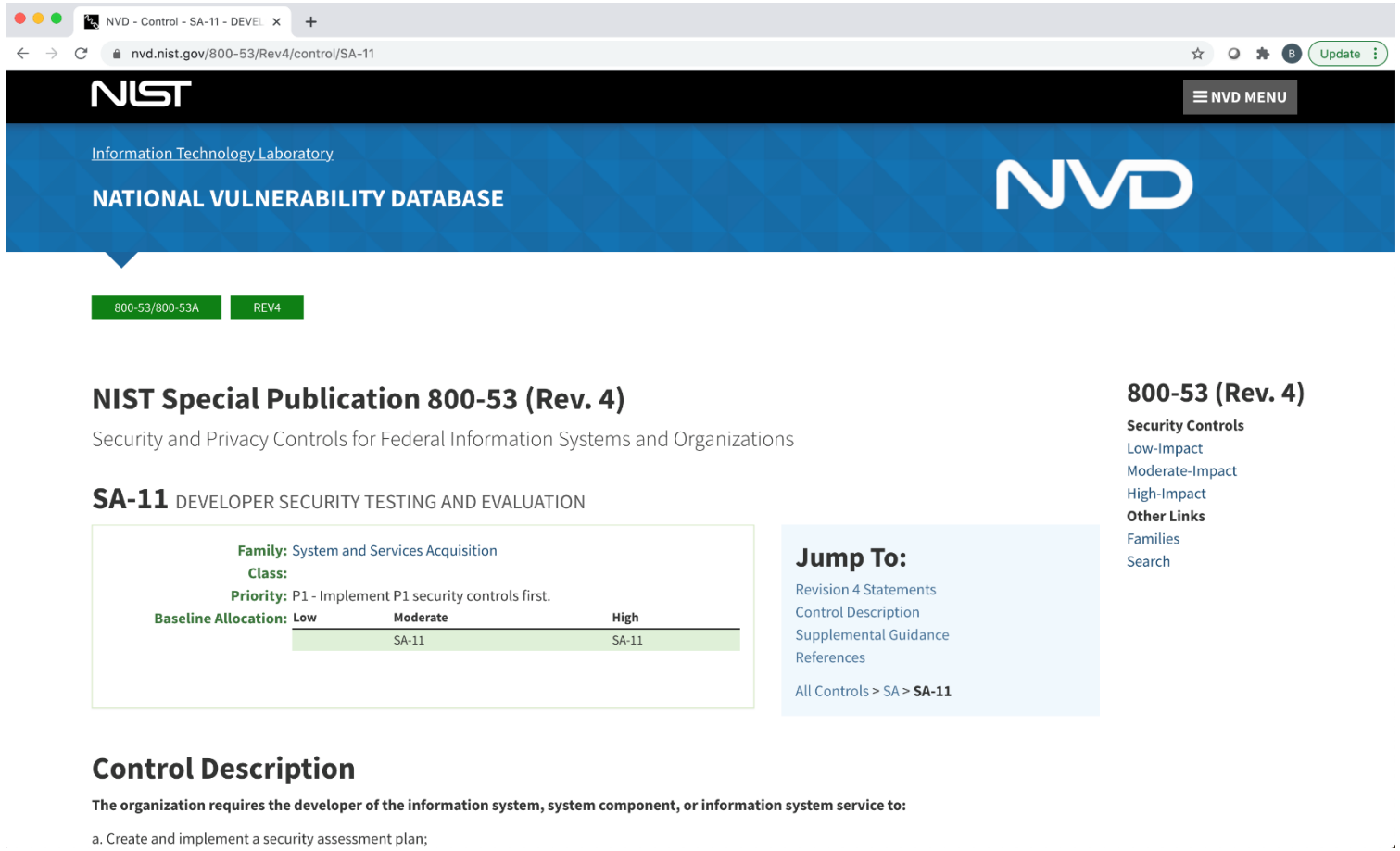
***Relevant to the view "Research Concepts" (CWE-1000)***

| Nature | Type | ID | Name |
|---|---|---|---|
| ChildOf | C | 118 | Incorrect Access of Indexable Resource ('Range Error') |
| ParentOf | B | 120 | Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') |
| ParentOf | B | 125 | Out-of-bounds Read |
| ParentOf | B | 466 | Return of Pointer Value Outside of Expected Range |
| ParentOf | ∞ | 680 | Integer Overflow to Buffer Overflow |
| ParentOf | B | 786 | Access of Memory Location Before Start of Buffer |
| ParentOf | B | 787 | Out-of-bounds Write |
| ParentOf | B | 788 | Access of Memory Location After End of Buffer |
| ParentOf | B | 805 | Buffer Access with Incorrect Length Value |
| ParentOf | B | 822 | Untrusted Pointer Dereference |
| ParentOf | B | 823 | Use of Out-of-range Pointer Offset |
| ParentOf | B | 824 | Access of Uninitialized Pointer |
| ParentOf | B | 825 | Expired Pointer Dereference |
| CanFollow | C | 20 | Improper Input Validation |
| CanFollow | B | 128 | Wrap-around Error |
| CanFollow | V | 129 | Improper Validation of Array Index |
| CanFollow | B | 131 | Incorrect Calculation of Buffer Size |
| CanFollow | B | 190 | Integer Overflow or Wraparound |
| CanFollow | B | 193 | Off-by-one Error |
| CanFollow | V | 195 | Signed to Unsigned Conversion Error |
| CanFollow | B | 839 | Numeric Range Comparison Without Minimum Check |
| CanFollow | B | 843 | Access of Resource Using Incompatible Type ('Type Confusion') |
| CanFollow | B | 1257 | Improper Access Control Applied to Mirrored or Aliased Memory Regions |
| CanFollow | B | 1260 | Improper Handling of Overlap Between Protected Memory Ranges |

From CAPEC – drill down to CWE

[https://cwe.mitre.org/data/definitions/119.html](https://cwe.mitre.org/data/definitions/119.html)

# NIST 800-53 Defense Controls




Our recommended defenses serve as implementation requirements to CAPEC-123

# Update AADL Model

**Based on feedback from VERDICT**

- Set Cyber Defense properties
- Change component properties
- Move Trust Boundary

## Update DAL

```
deliveryPlanner: system DeliveryPlanner
{
    -- VERDICT Component Properties
    CASE_Consolidated_Properties::canReceiveSWUpdate => true;
    CASE_Consolidated_Properties::componentType => Software;
    CASE_Consolidated_Properties::hasSensitiveInfo => true;
    CASE_Consolidated_Properties::insideTrustedBoundary => true;
    CASE_Consolidated_Properties::pedigree => Sourced;

    -- VERDICT Cyber Defense and DAL Mitigations
    CASE_Consolidated_Properties::staticCodeAnalysis =>7;
};
```

## Update and rerun tool until satisfied with the results

Console | MBAA Result

**Mission # 1: MReq01 -> Succeeded**

| Cyber Requirement | Severity of Successful Attack | Implemented Security Design Assurance Level | Analysis Result |
|---|---|---|---|
| CyberReq01 | Hazardous | B | ✓ |
| CyberReq02 | Hazardous | B | ✓ |

| Safety Requirement | Acceptable Probability of Failure | Calculated Probability of Failure | Analysis Result |
|---|---|---|---|
| SafetyReq01 | 1e-7 | 9.09999957347e-08 | ✓ |

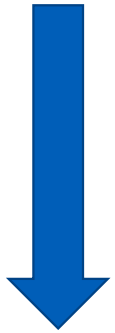
# Cyber Resiliency Verifier (CRV) Capability

# Capture Formal Properties

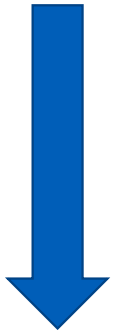
**Architecture Model**

**Formal Property**

**P7:Do no deliver** the package to an off-limits location



AADL



Expressed

Assume-Guarantee Contracts (AADL/AGREE)

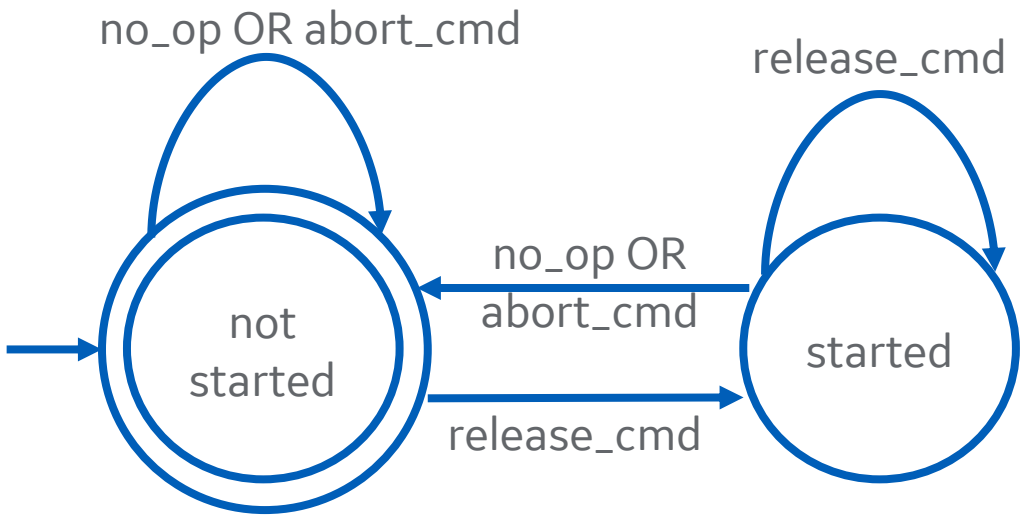
```
guarantee "P7: The drone never initiates packet release to an off-limits location":
    started => not Agree_Nodes::InRestrictedArea(probe_delivery_location);
```

# Capture Behavioral Model

**Architecture Model** → Expressed → AADL

**Formal Property** → Expressed → Assume-Guarantee Contracts (AADL/AGREE)

**Behavior Model** → Expressed → Assume-Guarantee Contracts (AADL/AGREE)

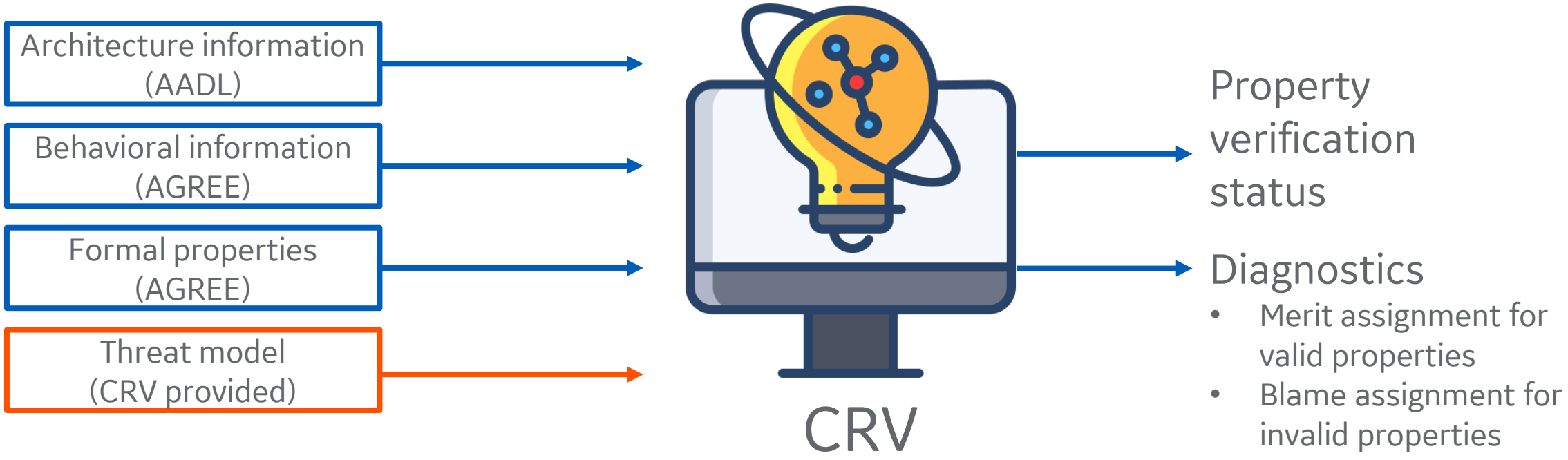



DeliveryItemMechanism State Machine Example

```
annex agree {**
    eq release_cmd: bool = (delivery_cmd_in = Agree_Constants::RELEASE_PACKAGE_CMD);
    eq abort_cmd: bool = (delivery_cmd_in = Agree_Constants::ABORT_DELIVERY_CMD);
    eq no_op_cmd: bool = (delivery_cmd_in = Agree_Constants::NO_OPERATION_CMD);

    guarantee "Initially, delivery status is NOT_STARTED":
        Agree_Nodes::InitiallyX(delivery_status_out = Agree_Constants::NOT_STARTED_STATUS);

    guarantee "if no op or abort command have received then delivery status gets re-started":
        true -> (no_op_cmd or abort_cmd => (delivery_status_out = Agree_Constants::NOT_STARTED_STATUS));

    guarantee "If delivery command is issued, delivery status is different from NOT_STARTED":
        true -> (release_cmd => delivery_status_out <> Agree_Constants::NOT_STARTED_STATUS);
**};
```

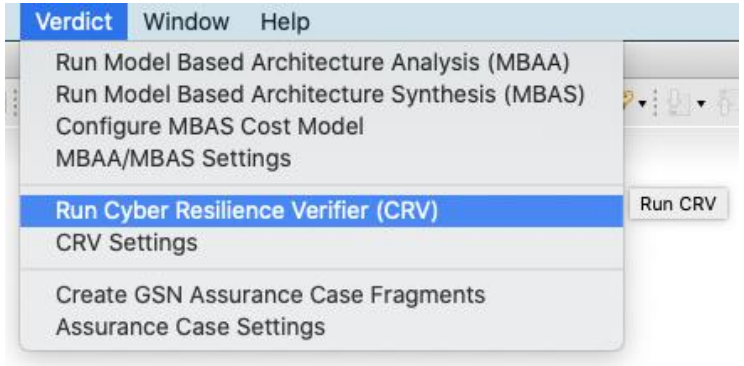
AADL/AGREE Modeling

# CRV Analysis

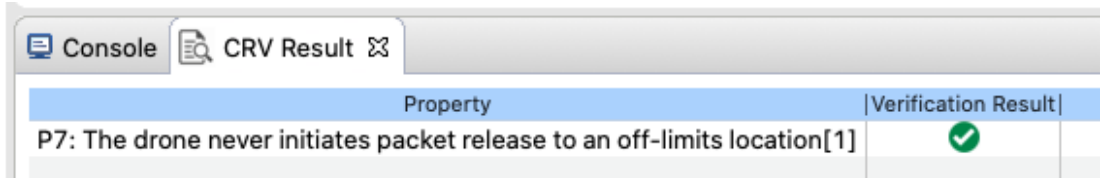


# Analyze Model in Benign Case

## CRV Menu Entry




## Benign Case Result

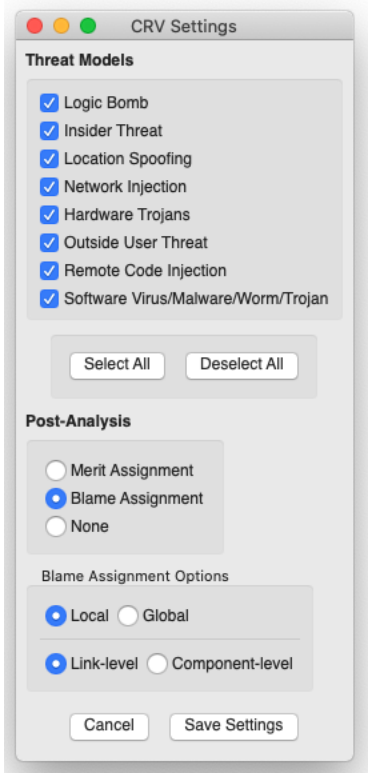



- No threats enabled
- Property P7 is valid

# Analyze Model Instrumented with Threats

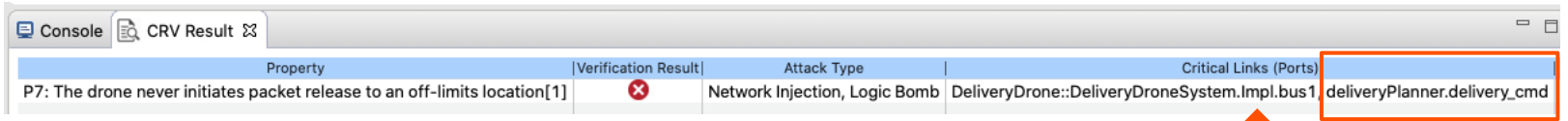
1

CRV Setting Panel



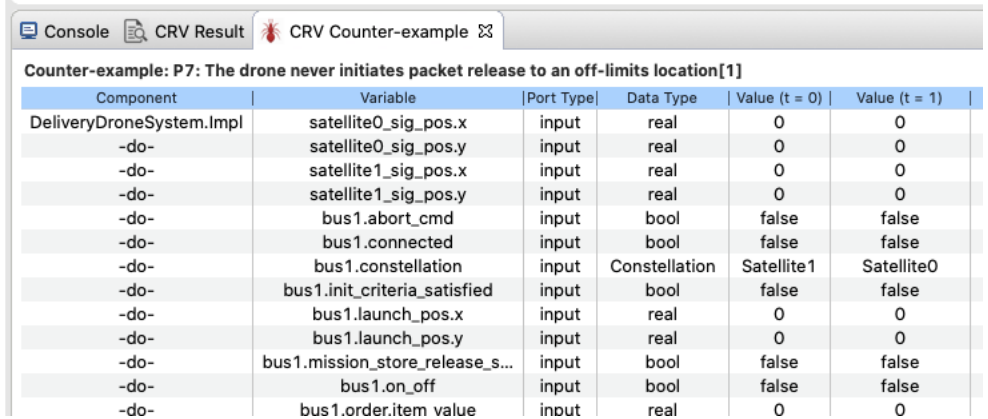

2

CRV Results with Instrumented Threats



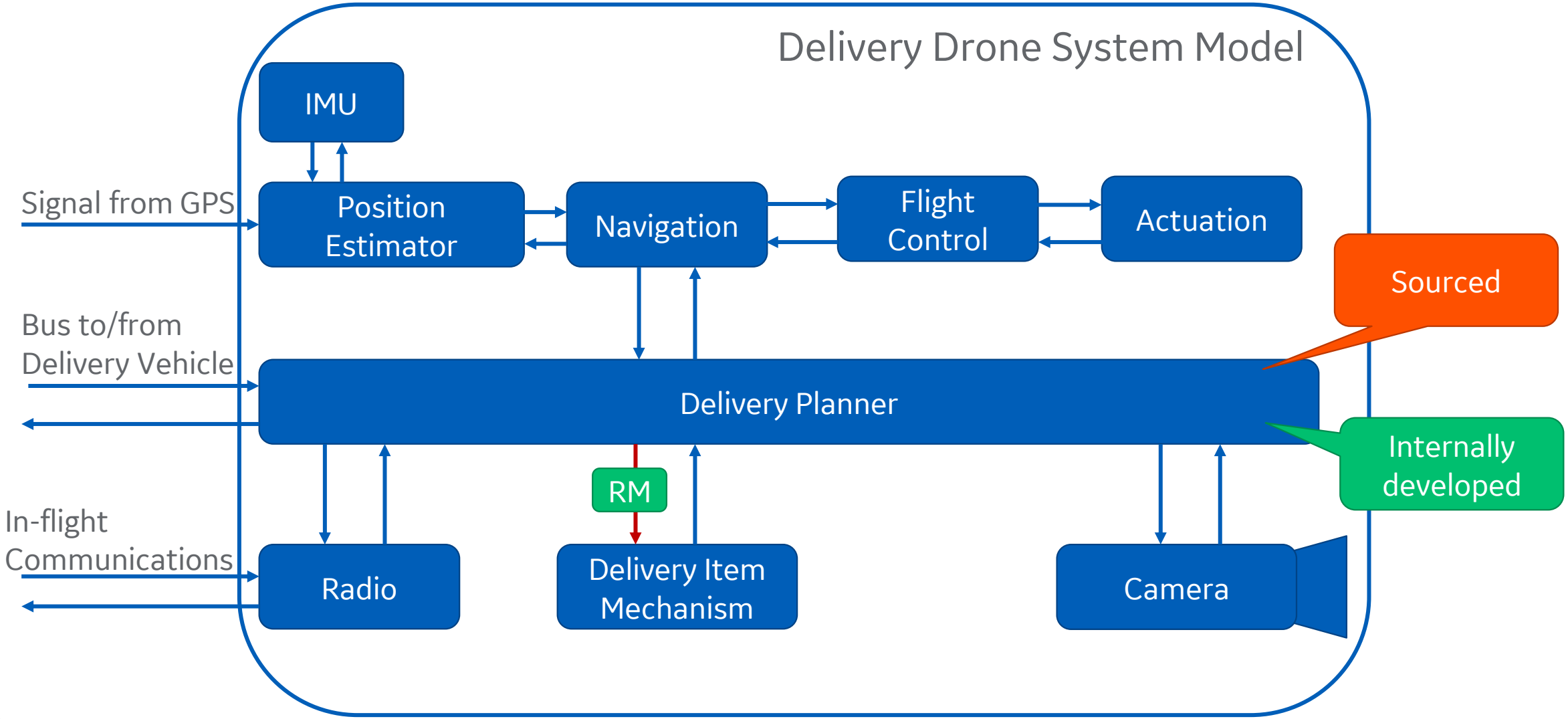

3

Counter-example




Blame Assignment Info

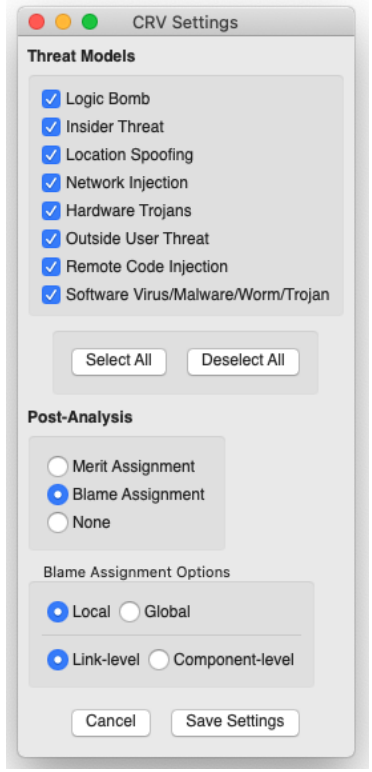
# Update Design



# Analyze Updated Design

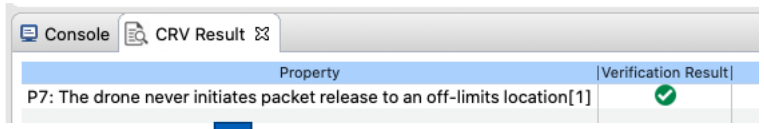
1

CRV Setting Panel



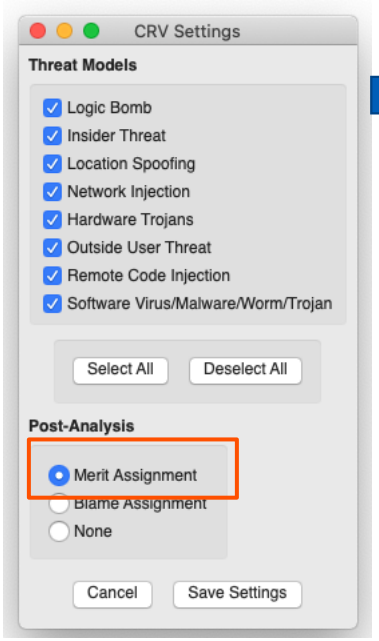

2

CRV Result for the Updated Model



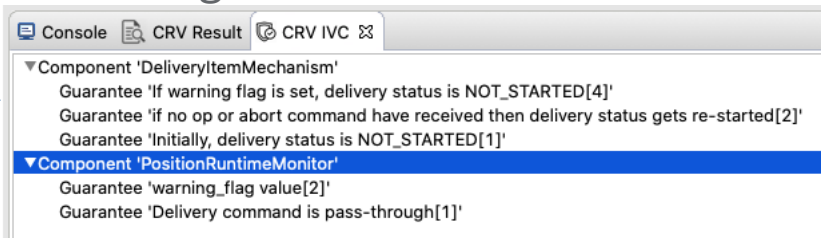

3

CRV Setting Panel



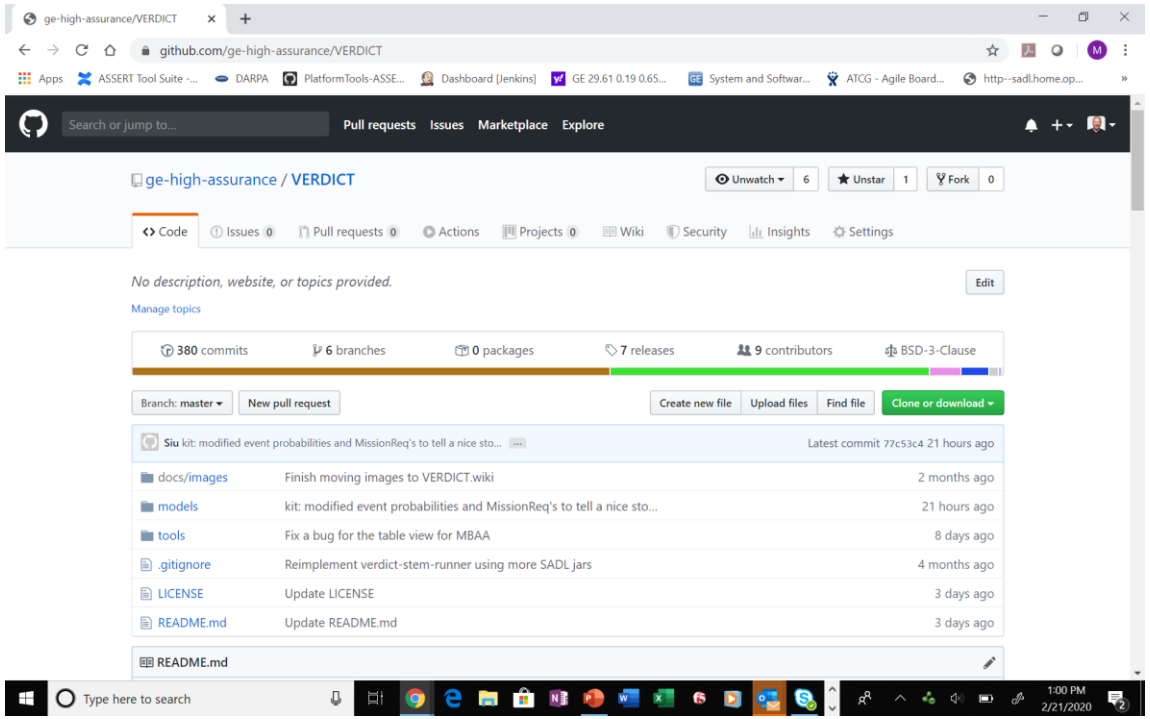

4

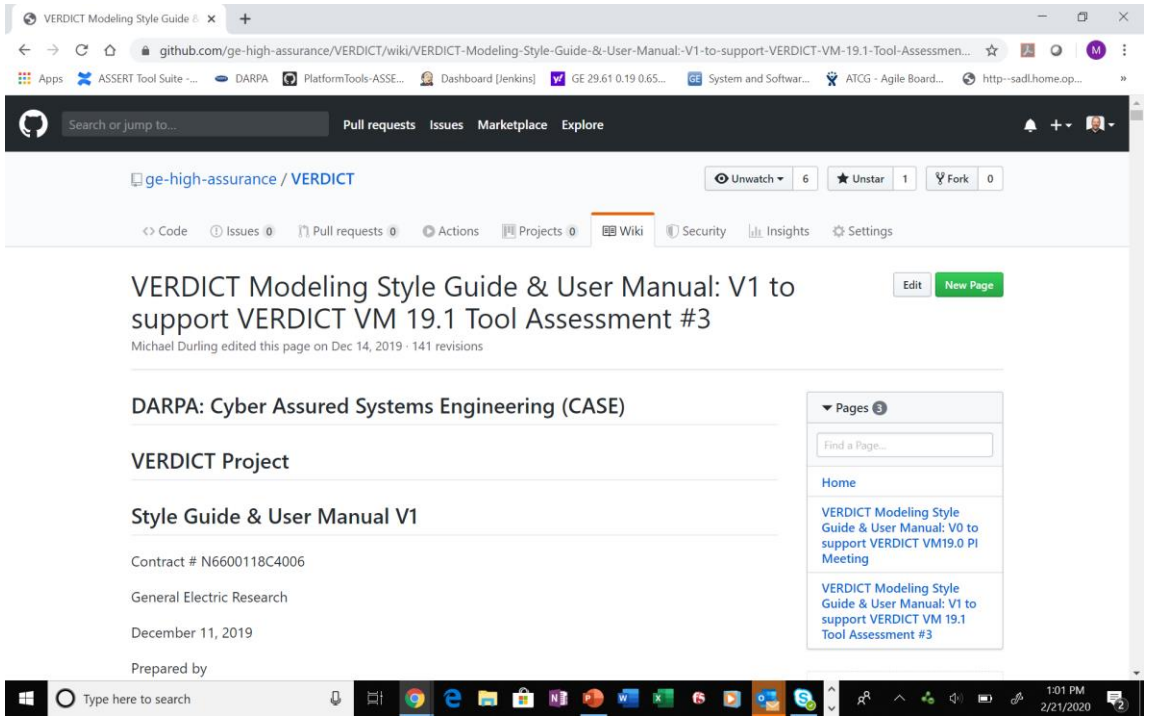
Merit Assignment



# VERDICT Open Source on GitHub

https://github.com/ge-high-assurance/VERDICT




YouTube High Assurance System MBAAS Demo: https://www.youtube.com/watch?v=UDKbDyukmlw
YouTube High Assurance System CRV Demo: https://www.youtube.com/watch?v=1_35EPh5Fp8

# VERDICT Publications

- “Architectural and Behavior Analysis for Cyber Security”, *Digital Avionics Systems Conference (DASC)*, September 2019.
- “A Model Based Framework for Analyzing the Security of System Architectures”, *Reliability and Maintainability Symposium (RAMS)*, January 2020.
- “DARPA Project Producing Tool to Help Anticipate Military and Industrial Systems’ Cyber Threats”, *NextGov.com*, April 2020.
- “Threat Identification and Defense Control Selection”, accepted by *SAE International Journal of Transportation Cybersecurity and Privacy*, June 2020.
- “Towards Developing Formalized Assurance Case”, accepted by *Digital Avionics Systems Conference (DASC)*, October 2020.
- “Expat: Expectation-based Policy Analysis and Enforcement for Appified Smart-Home Platforms”, ACM SACMAT 2019.
- “PatrIoT: Policy-assisted Resilient Programmable IoT system”, Runtime Verification (RV), 2020.
- “VERDICT: A Language and Framework for Cyber Resilient Systems Engineering”, Systems Journal, 2021. (to appear)
- “Experience in Designing for Cyber Resiliency in Embedded DoD Systems”, Annual INCOSE International Symposium, 2021. (to appear)

Building a world that works