DARPA

Cyber Assured Systems Engineering (CASE)

VERDICT Project Contract # N6600118C4006

General Electric Research

Style Guide & User Manual V0

October 15, 2019

Prepared by

Michael Durling, [durling@ge.com](mailto:durling@ge.com)

Kit Siu, [siu@ge.com](mailto:siu@ge.com)

Baoluo (Paul) Meng, [baoluo.meng@ge.com](mailto:baoluo.meng@ge.com)

With contributions from

Daniel Prince, GEAS

Abha Moitra, GE Research

Heber Herencia-zapana, GE Research

John Interrante, GE Research

Cesare Tinelli, University of Iowa

Omar Chowdhury, University of Iowa

Daniel Larraz, University of Iowa

Moosa Yahyazadeh, University of Iowa

Fareed Arif, University of Iowa

Contents

[Introduction to the VERDICT Tool 1](#_Toc22310242)

[Overview 1](#_Toc22310243)

[Model Based Architecture Analysis & Synthesis (MBAAS) 2](#_Toc22310244)

[Cyber Resiliency Verifier (CRV) 2](#_Toc22310245)

[Model Based Architecture Analysis & Synthesis 2](#_Toc22310246)

[Required elements and their placement 2](#_Toc22310247)

[Properties 2](#_Toc22310248)

[Mission Requirements 4](#_Toc22310249)

[Cyber Relations 5](#_Toc22310250)

[AADL Language Elements Currently Supported by VERDICT 6](#_Toc22310251)

[Simple Delivery Drone Example 6](#_Toc22310252)

[Running the Model Based Architecture Analysis & Synthesis (MBAAS) 7](#_Toc22310253)

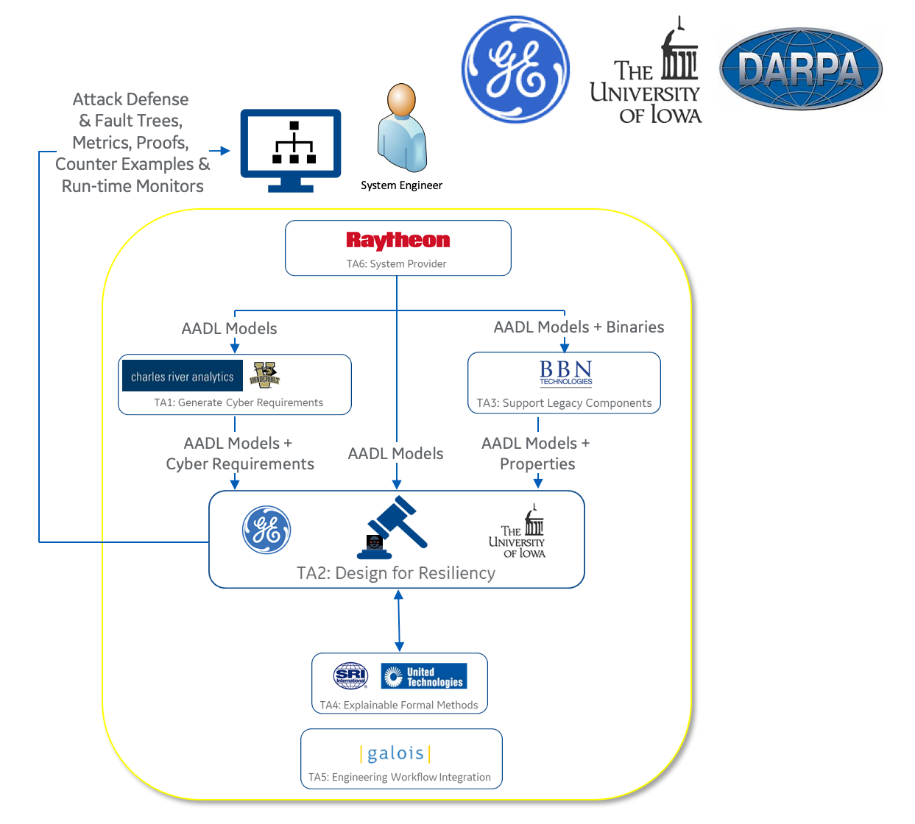
[VERDICT annex 10](#_Toc22310254)

# Introduction to the VERDICT Tool

This Wiki provides a technical description of the Verification Evidence and Resilient Design in Anticipation of Cybersecurity Threats (VERDICT) project which is part of the DARPA Cyber Assured Systems Engineering (CASE) program. The VERDICT team is composed of researchers and engineers from General Electric Research (GRC) and General Electric Aviation Systems (GEAS), and the University of Iowa. The team is creating the VERDICT tool to generate a Attack-Defense tree with quantifiable likelihood of successful attack metrics, prove behavioral models meet formal cyber properties, identify the need for and place run time monitors, and generate cyber test cases. This Wiki is intended to help end users of the tool understand how to create models and run the tool.

## Overview

The Overall DARPA CASE Toolchain is shown below is designed to enable System Engineers to analyze security along with other desired properties (e.g., safety, performance, cost, weight) at design time.



The program has seven Technical Areas (TA). TA 6 are the System Providers. They provide the systems to analyze including requirements, models, and code. TA 5 are the Systems Integrators. They facilitate integration of all the tools into a tool-chain. TA 4 is focused on Explainable Formal Methods. They are developing technology and tools that enables system engineers to benefit from formal methods technology without being formal methods experts. TA 3 supports Legacy Components such as binary and source code. TA 3 is working on tools that will extract models and properties from legacy code, so that it may be understood more clearly and reused. TA 2 is Design for Resiliency. The TA 2 performers are developing design tools to model, analyze and verify improvements in cyber-resiliency properties. The TA 1 performers are generating cyber-resiliency properties from the content provided to them by the TA 6 system providers.

## Model Based Architecture Analysis & Synthesis (MBAAS)

The GE team is working on TA 2, Design for Resiliency. The GE VERDICT tool has two major functions. The first is the Model Based Architecture Synthesis (MBAS). The MBAS tool, takes in architecture models, mission and cyber-resiliency requirements, then generates attack-defense trees with resiliency metrics. The MBAS tool uses the attack-defense tree information along with cybersecurity requirements and constraints as inputs to a synthesis function that in Phase 2 of the program will generate an architecture that meets predefined resiliency design constraints. The MBAS tool is built as an extension to a Fault Tree modeling and synthesis tool named [SOTERIA](https://github.com/ge-high-assurance/SOTERIA) that was developed previously for NASA. The MBAS tool enables the system engineer to model components, then synthesize architectures that meet both safety (based on fault tree analysis) and security (based on attack-defense tree analysis) design goals.

## Cyber Resiliency Verifier (CRV)

The second major function in the GE TA 2 Design for Resiliency project is the Cyber-Resiliency Verifier (CRV). The CRV takes in cyber-resiliency properties/requirements, architecture, design and code models - performs a formal analysis using an improved version of the Kind2 model checker developed at the University of Iowa, then returns design proof evidence, localized design errors, suggested run time monitors, and test cases.

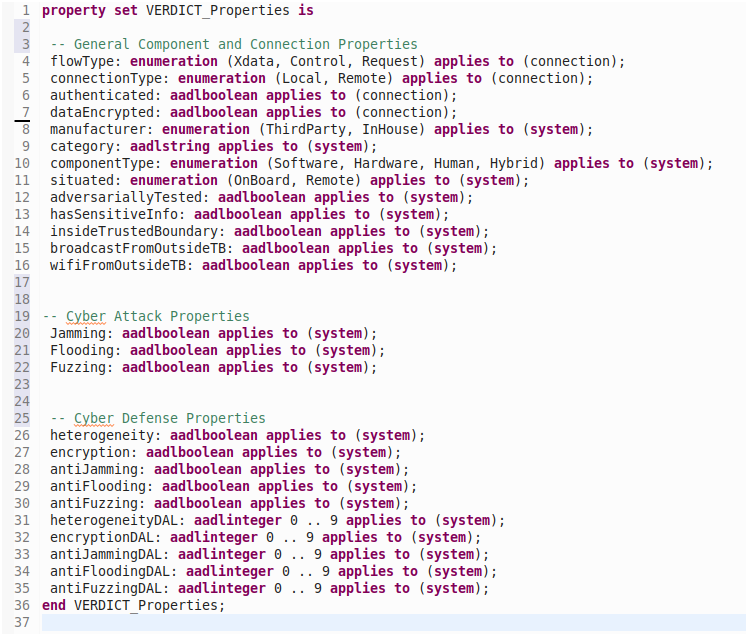
# Model Based Architecture Analysis & Synthesis

This section of the document introduces the critical content required in the AADL model in order to use the VERDICT MBAS analysis capability. See [OpenAADL](http://www.openaadl.org/) for general modeling in AADL guidelines. The VERDICT analysis engine requires: 1) Properties, 2) Mission Requirements, and 3) Cyber Relations to be defined in the AADL model.

## Required elements and their placement

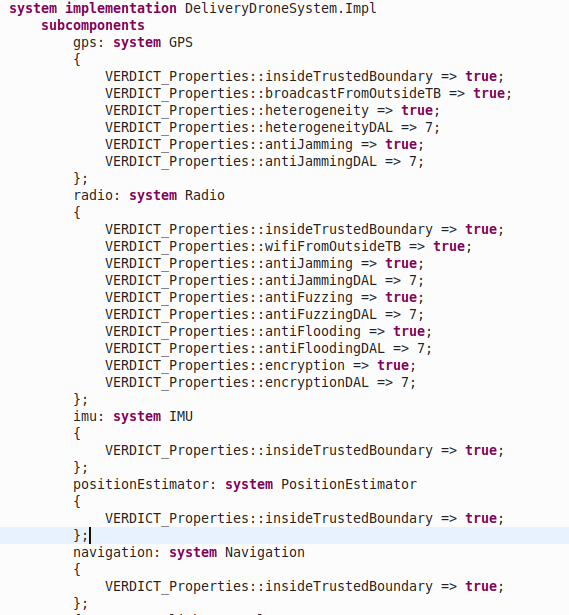
### Properties

VERDICT uses Properties associated with architectural components and connections to identify cyber vulnerabilities. Properties are defined by the system engineers in the AADL models. There are 3 Property types - 1) General Component and Connections, 2) Cyber Attack and 3) Cyber Defense. The following figure shows an example of properties declared in the VERDICT\_Properties.aadl file.

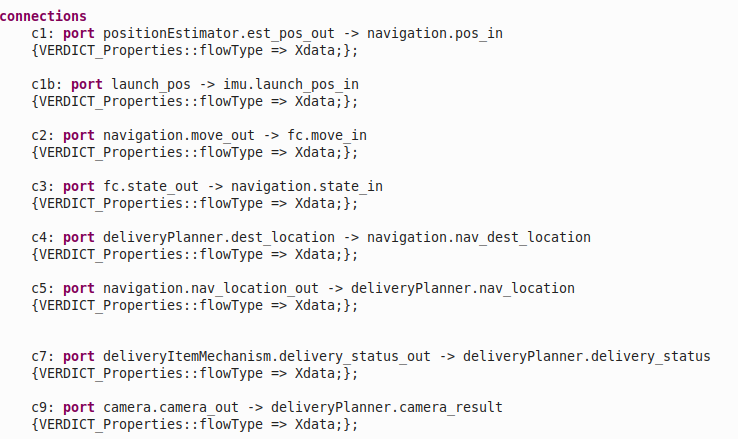


***\*\*All Properties MUST be declared in the VERDICT\_Properties.aadl file before they are used.\*\****

As shown in the figure below, VERDICT Properties are applied in the component implementation section of the AADL model.



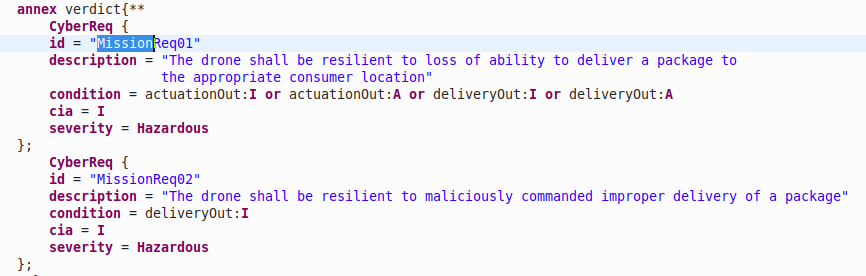
The figure below shows VERDICT Properties being set on connections in the AADL model.



### Mission Requirements

VERDICT generates Attack-Defense Trees and calculates Likelihood of Successful Attack based on Mission Requirements. The Mission Requirements represent the top of the Attack-Defense Tree. The AD Tree shows all the attacks and defenses that impact the ability to successfully achieve the Mission Requirement. VERDICT calculates the Likelihood of Successful Attack that would prevent the Mission Requirement.

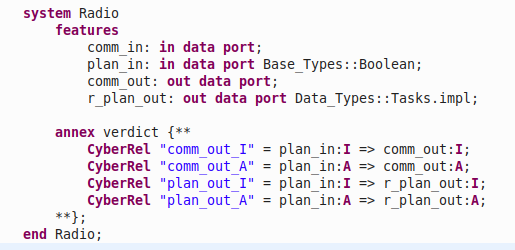
As shown below, Mission Requirements are placed in the System declaration section of the AADL model. VERDICT created an AADL annex to capture Mission Requirements. The system engineer modeler includes a description of the Mission Requirement, the condition (connection impacted by CIA), and the severity. The figure below shows a Mission Requirement that impacts the Integrity of the system at a Hazardous level. The severity level is used by VERDICT as an acceptance criterion for the Likelihood of Successful Attack.



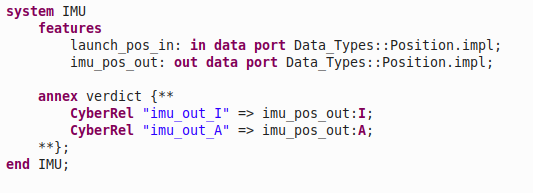
### Cyber Relations

VERDICT requires the user to declare the Cyber Relations for each of the components in the AADL model. Cyber Relations represent the relationship of the input and output signals of a component. Cyber Relations are defined in the declaration section of the AADL model using the verdict annex. The figure below shows the Cyber Relations for a Radio component.

The comm\_out integrity is impacted by the plan\_in integrity, etc. VERDICT has a Wizard to automatically generate Cyber Relations and place them in the AADL model.



Connections at the outer edges of the AADL need to be "terminated". The following figure shows how to capture Cyber Relations for connections at the outer edges of an AADL model to support VERDICT analysis.



### AADL Language Elements Currently Supported by VERDICT

At the time of the 2019 PI Meeting VERDICT 19.0 Release, the VERDICT tool supported a subset of the AADL language which is shown in the following list.

\* system

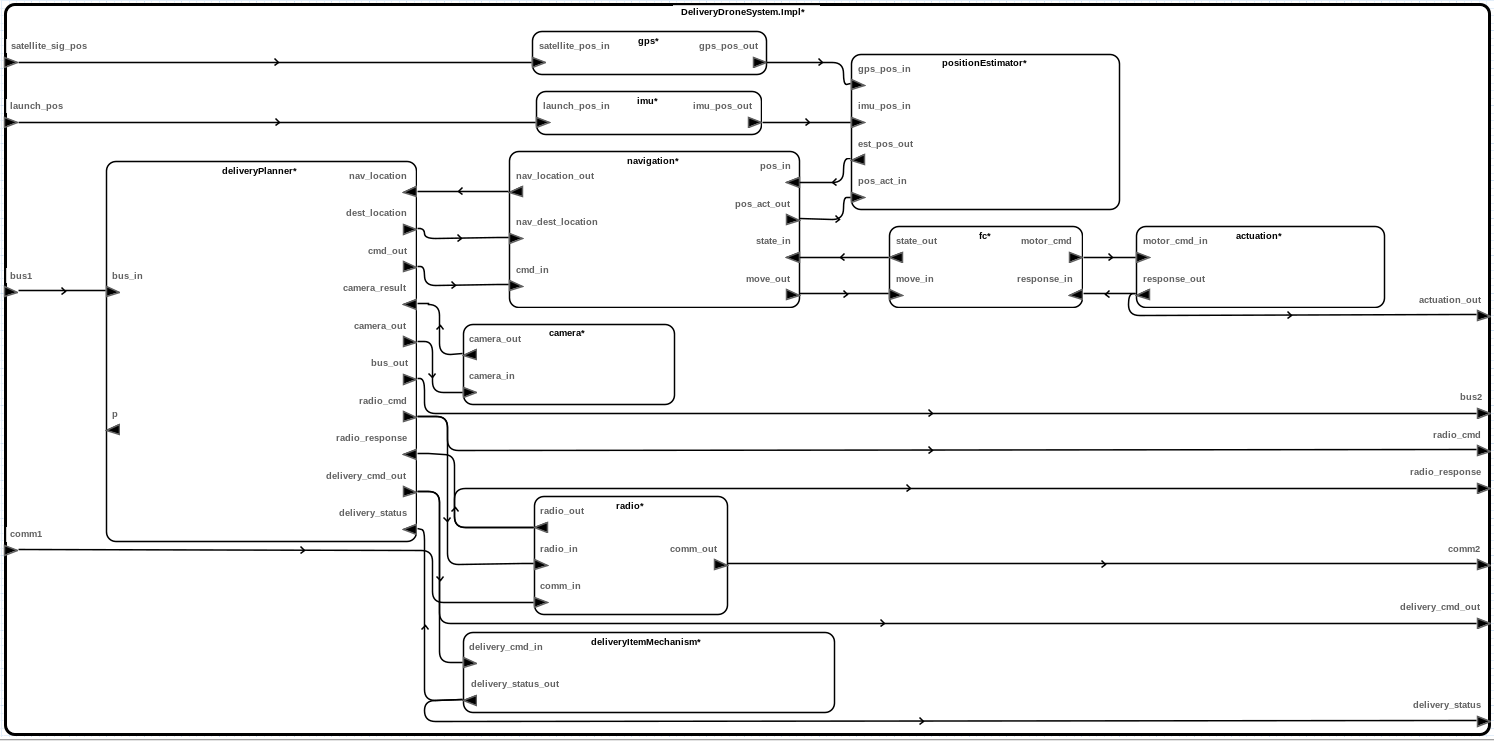
\* in data port

\* out data port

\* No support for hierarchical models

## Simple Delivery Drone Example

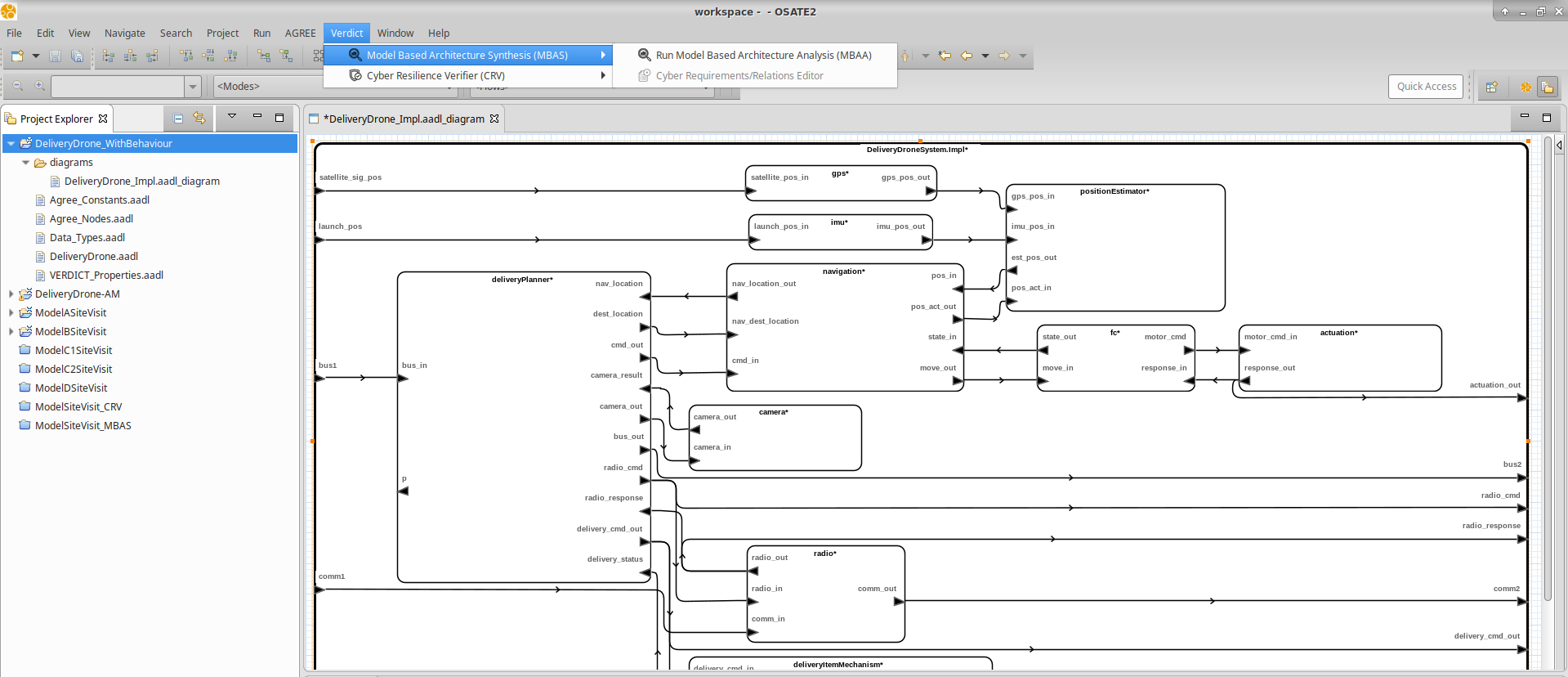
The VERDICT team created the Delivery Drone model to demonstrate the tool functionality. The figure below shows an AADL model diagram of a delivery drone that may be used to deliver packages to residential locations.



The Delivery Drone Model is available to download in a VM @ <https://collab.provatek.com/index.php/apps/files/?dir=/Meetings/2019-08-21-PI-03/TA2-GE%2BIowa&fileid=64318>

## Running the Model Based Architecture Analysis & Synthesis (MBAAS)

1. Select the Model and call the MBAS function from the VERDICT menu



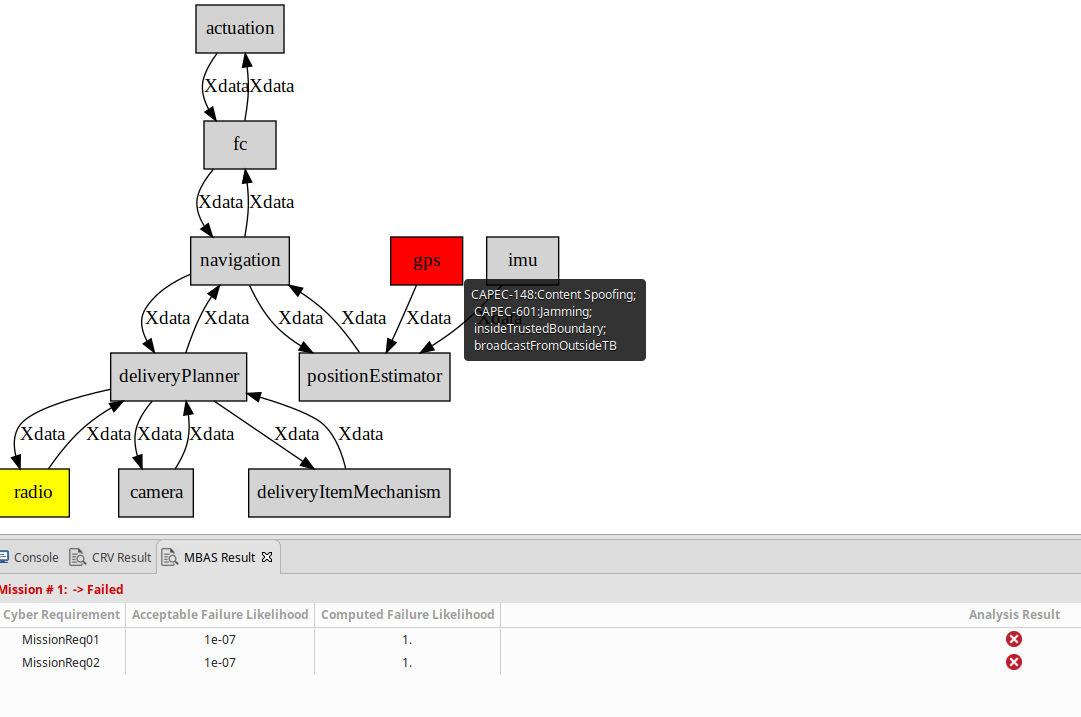
2. Explore the Results

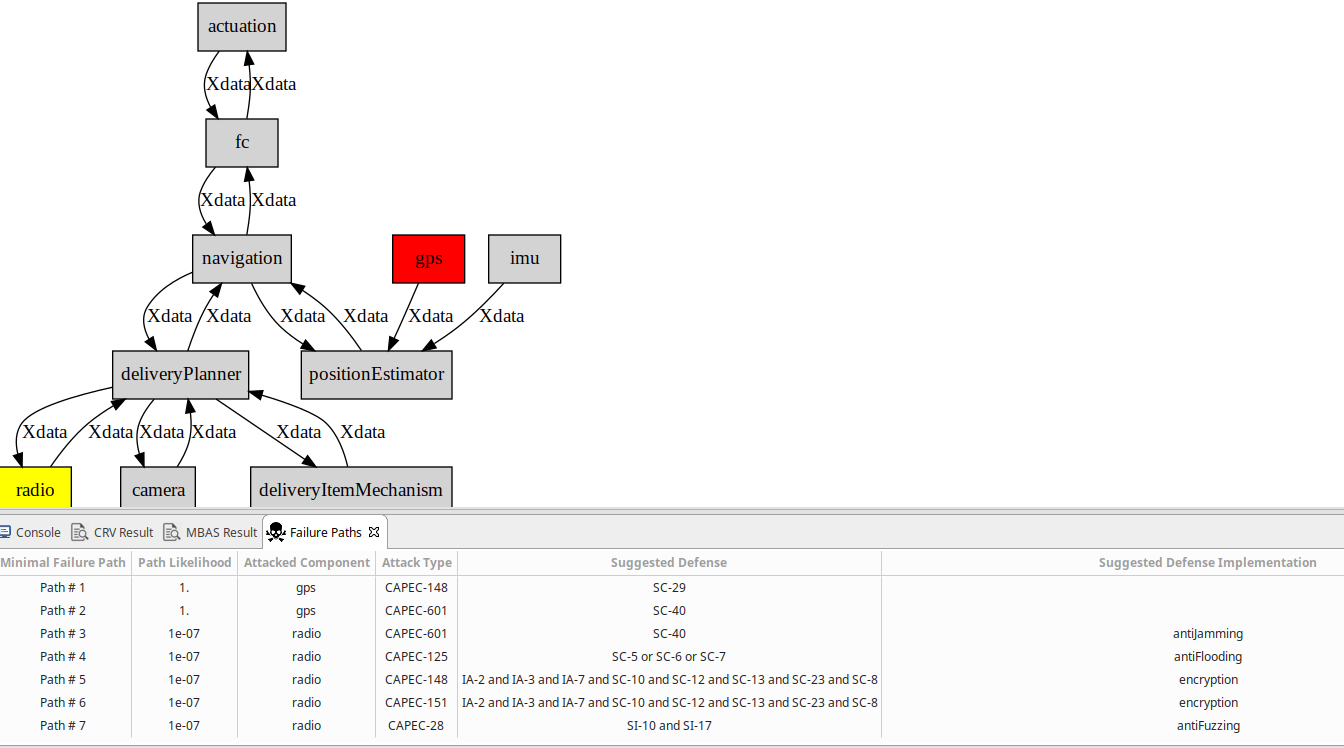
\* Cyber Resiliency Metric (Likelihood of Successful Attack)

The Acceptable and Calculated Likelihood of Successful Attack (Failure Likelihood) are shown in the figures below.The acceptable metric is based on the Mission Requirement "severity" parameter defined in the AADL model. VERDICT calculates the Calculated number. When the calculated number is equal to or below the acceptable number the design passes and the tool shows a green check mark. When the calculated value is higher than the acceptable number, the tool shows a red "X". If the user right clicks on the red "X", the tool shows the specific CAPEC's and NIST 800-53 suggested defenses.

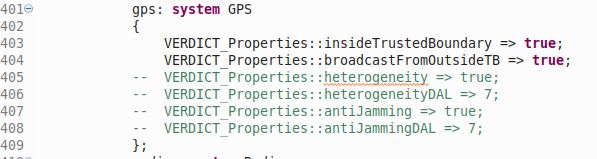
\* Identified Cyber Threats ([CAPEC's](https://capec.mitre.org/))

\* Recommended Cyber Defense Patterns ([NIST 800 - 53](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf))

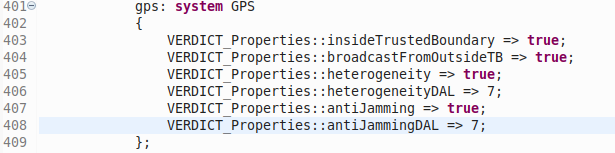




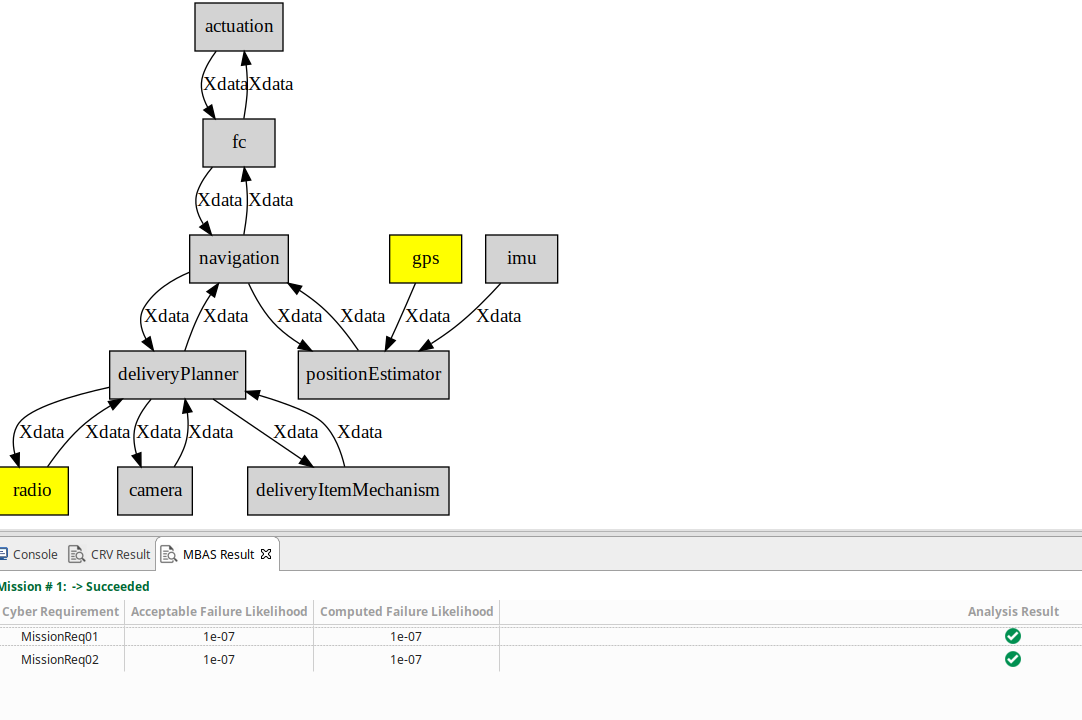
The following figure shows the Cyber Defense Properties commented out in the AADL model. This explains why the analysis failed.



The next figure shows the Cyber Defenses after they have been uncommented out.



The next figure shows the results passing now that the Cyber Defense Properties have been applied.



# VERDICT annex

We additionally define a new annex language – VERDICT for AADL to write mission requirements, cyber requirements, and cyber relations in AADL models. Cyber requirements and mission requirements may only be declared in a **verdict** annex within the top-level system type, and they describe the system as a whole. Cyber requirements can be aggregated and associated to a particular mission requirement. Cyber relations may only be declared within a subcomponent system type, and they describe the vulnerability flow between the inputs and outputs of an individual component within the system.

Cyber requirements, cyber relations, and mission requirements are declared with the CyberReq, CyberRel, and MissionReq keywords, respectively. The declarations are comprised of a list of field-value pairs within curly braces. Each field-value pair is denoted by the field name followed by = and then the value.

**CyberReq** {

-- required fields

id = "my\_cyber\_requirement"

condition = out1:A

severity = Hazardous

-- optional fields

targetLikelihood = 1e-07

cia = I

comment **=** "my comment"

description = "my description"

}

CyberReq fields are:

* id (in quotes), a unique identifier
* condition, an expression comprised of top-level system type output ports describing the condition under which the cyber requirement is active
* severity, the severity level (Catastrophic, Hazardous, Major, Minor, or None)
* targetLikelihood (optional), the probability that matches severity (e.g. 1e-05 for Major); has no additional semantic meaning but must match severity
* cia (optional), the security concern (C, I, or A)
* comment (optional, in quotes), an additional note with no semantic value
* description (optional, in quotes), additional information intended for display alongside the id

**CyberRel** {

-- required fields

id = "my\_cyber\_relation"

output = out1:A

-- optional fields

inputs = in1:A or in2:A

comment **=** "my comment"

description = "my description"

}

CyberRel fields are:

* id (in quotes), a unique identifier
* output, a single component output port security concern that becomes active under the input condition
* inputs (optional), an expression comprised of component input ports describing the condition under which the relation is active; if absent, signifies that the output condition is always active
* comment (optional, in quotes), an additional note with no semantic value
* description (optional, in quotes), additional information intended for display alongside the id

**MissionReq** {

-- required fields

id = "my\_mission\_requirement "

cyberReqs = "req\_a", "req\_b"

-- optional fields

comment **=** "my comment"

description = "my description"

}

MissionReq fields are:

* id (in quotes), a unique identifier
* cyberReqs, a comma-separated list of quoted cyber requirement ids
* comment (optional, in quotes), an additional note with no semantic value
* description (optional, in quotes), additional information intended for display alongside the id

Note that id, comment, and description are shared by all three, while phases and external are shared by CyberReq and CyberRel.

There is also a condensed syntax for cyber relations that is intended for ease of input and concision. The syntax is as follows:

**CyberRel** "[id]" = [input condition] => [output port]

If there is no input condition, i.e. the output is always active, then the input condition may be omitted as follows:

**CyberRel** "[id]" => [output port]

Note that there is no condensed syntax for cyber requirements or mission requirements.

The ports available to the inputs field of cyber relations are the input ports to the subcomponent to which they belong. Similarly, the ports available to the output field are the output ports of that subcomponent (although cyber relations only have one output port security concern). The ports available to the condition field of cyber requirements are the output ports of the top-level system type in which the requirement is declared.

The value of the cyber relation output field is a port security concern. A single port security concern is an input or output port following by a colon (:) and the security concern, either C, I, or A. For example, if out1 is an output port, out1:A is the port security concern describing the condition that out1 has an availability concern.

The value of the cyber requirement condition and cyber relation inputs fields are expressions. Expressions are composed of port security concerns linked by the logical connectives and, or, not, and parentheses. and and or are placed between two port security concerns while not is placed before a single port security concern. Expressions may be arbitrarily nested. For example:

out1:A and out2:I or not (out3:A and out4:I)

and has higher precedence than or. In other words, out1:A or out2:A and out3:A is parsed as out1:A or (out2:A and out3:A). Note that a single port security concern is a valid expression. **However, for the moment VERDICT tool chain only supports disjunctions of port security concerns.**