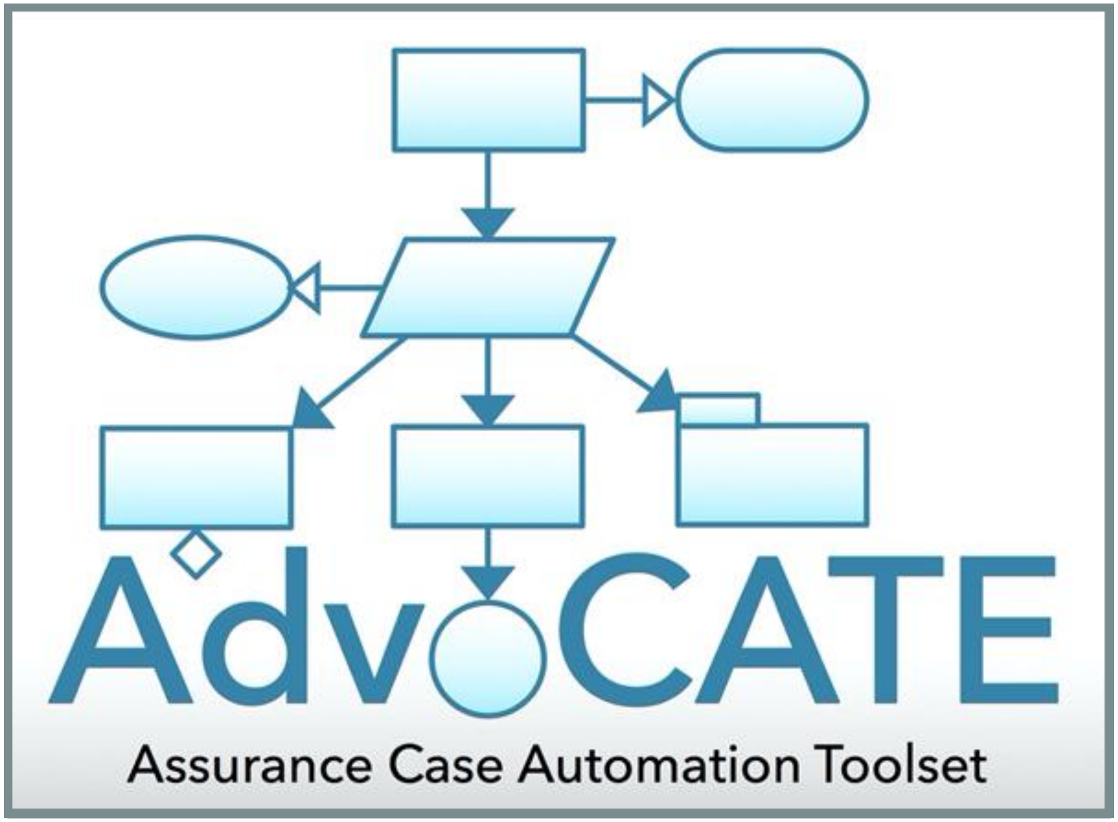
User Guide

AdvoCATE

Product Model 2.0



May 5, 2022
User Guide Version 1.4
Build Version 0.34.1-s

Welcome to AdvoCATE

Welcome to AdvoCATE, the Assurance Case Automation Toolset that streamlines and supports the development of assurance cases.

This user guide describes the functionality in **AdvoCATE 0.34.1-s**.

This document is divided into eight sections:

Who Should Use This Guide?

This document was developed for experienced assurance case developers who are new to AdvoCATE.

Scope of Guide

This user guide describes how to install and use the main features within AdvoCATE, while also explaining some of AdvoCATE's design aspects. This document assumes users are familiar with assurance case terminology, notation styles, and analysis methodology.

This user guide does not describe the methodology or the steps to develop assurance cases.

Guide References

PHA References

- FAA Air Traffic Organization. Safety Management System (SMS) Manual. April 2019.
  URL: *https://www.faa.gov/air_traffic/publications/media/ATO-SMS-Manual.pdf.*
  - Probability classes and their qualitative/quantitative characterization is in *Section 3* (document page 49, pdf page 56).
  - This information can also be found in *FAA Order 8040.4B Safety Risk Management Policy, Appendix C-2*, (pdf page 28).
- NASA. Range Flight Safety Requirements. NASA-STD-8719.25.
- NASA. Facility System Safety Guidebook. NASA-STD-8719.7.
- Department of Defense. Standard Practice System Safety. MIL-STD-882E. May 2012.

BHA, PhHA, FHA, & FMEA/FMECA Reference

- Department of Defense. Standard Practice System Safety. MIL-STD-882E. May 2012.

General Guidance on Risk-informed Safety Cases and Risk-based Safety

- H. Dezfuli, B. Allan, C. Everett, C. Christopher Smith, M. Stamatelatos, R. Youngblood. NASA System Safety Handbook, Volume 1, System Safety Framework and Concepts for Implementation. NASA/SP-2010-580, Version 1.0, Nov. 2011.
  URL: *https://ntrs.nasa.gov/citations/20120003291.*
- H. Dezfuli, A. Benjamin, C. Everett, M. Feather, P. Rutledge, D. Sen, R. Youngblood. NASA System Safety Handbook, Volume 2: System Safety Concepts, Guidelines, and Implementation Examples. NASA/SP-2014-612, Version 1.0, Nov. 2014.
  URL: *https://ntrs.nasa.gov/citations/20150015500.*

Aircraft Development and Safety Assessment Standards

- SAE Aerospace Council, S-18 Aircraft and Systems Development and Safety Assessment Committee. Guidelines for the Development of Civil Aircraft and Systems. SAE Aerospace Recommended Practice (ARP) 5754A. Dec. 2010.
- SAE Aerospace Council, S-18 Aircraft and Systems Development and Safety Assessment Committee. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. SAE ARP 4761. Dec 1996.

GSN References

- John Spriggs. GSN - The Goal Structuring Notation: A Structured Approach to Presenting Arguments. Springer. 2012.
- SCSC Assurance Case Working Group. Goal Structuring Notation Community Standard (Version 3). May 2021. URL: *https://scsc.uk/r141C:1.*

BTD References

- Bow Tie Diagrams. *UK Civil Aviation Authority Website on Bow Ties in Aviation*.

Need More Help?

For questions, comments, and additional help, please send an e-mail to: *advocate-feedback@lists.nasa.gov*.

If you are experiencing issues or encounter problems within the AdvoCATE application, in your email to us please attach a copy of your **AdvoCATE Configuration Log**. For more information about how to find your AdvoCATE Configuration Log, see the *Send AdvoCATE Configuration Log* section.

Table of Contents

Table of Contents

Get Started

What is AdvoCATE?

AdvoCATE is an Assurance Case Automation Toolset that allows you to create and edit assurance case documentation, link associated assurance artifacts, and perform risk analysis in one centralized location.

AdvoCATE's growing suite of automated features, such as pattern instantiation and argument verification, provide traceability and consistency across assurance hazards, requirements, arguments, safety architectures, and evidence.

System Requirements

In order to install AdvoCATE, you first need to install Java.

It is recommended to install Java version 11 (long term support) of the Java Runtime Environment (JRE) with the appropriate path variables set for your operating system (OS). However, any 64-bit Java version from 8 onwards will work.

NOTE:

- For Mac OS, versions up to 10.15 Catalina are currently supported.
- You may need to have administrative privileges to execute AdvoCATE if your OS is set up to prohibit execution of unsigned or third-party applications.
- If you do not have admin access on your Windows PC, you may not be allowed to run the application using AdvoCATE.exe, but you can still run the application using eclipsec.exe.

Install AdvoCATE

To install AdvoCATE

1. Unzip the **installer** into a folder of your choice.
2. Run the **executable application**.

AdvoCATE will install to your computer.

NOTE: For Mac and Linux computers, you must first make the AdvoCATE application executable.

Make AdvoCATE Executable on Mac & Linux

If you are a Mac or Linux user, to make the AdvoCATE application executable

1. Open a **command line terminal**.
2. Type the following command
   - Mac: **chmod u+x AdvoCATE.app/Contents/MacOS/AdvoCATE**
   - Linux: **chmod u+x AdvoCATE/advocate**

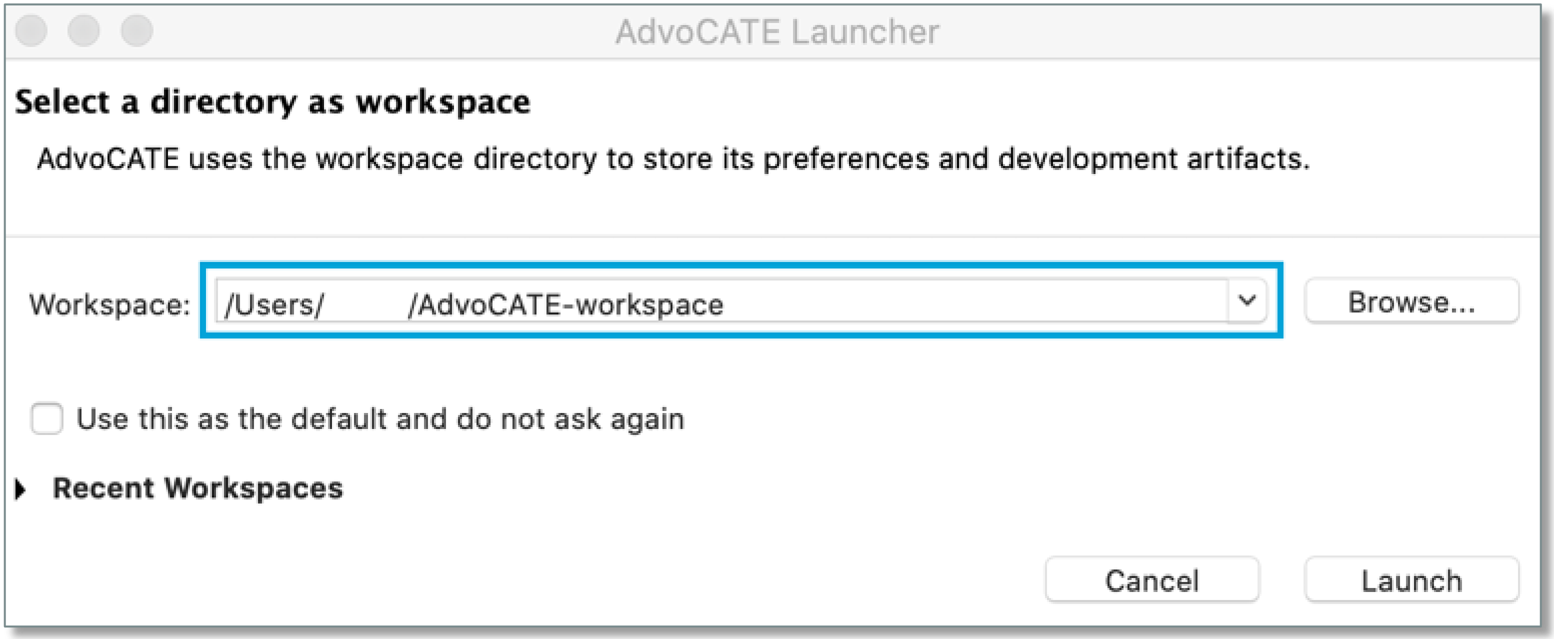
   NOTE: Your app file path should denote the exact file name and location where you saved your installation files.

The AdvoCATE application file will now be executable, and you can open AdvoCATE.

Workspaces

Upon launching AdvoCATE, a **Workspace** must be selected or created. The **Workspace** is a folder which stores various property files and metadata, including the **AdvoCATE Configuration Log**. For more information about how to find your AdvoCATE Configuration Log, see the *Send AdvoCATE Configuration Log* section.

By default, **AdvoCATE-workspace** will be created as a subdirectory in your user home directory and used as the Workspace, unless otherwise specified (*Figure 1*).



AdvoCATE Launcher

Select a directory as workspace

AdvoCATE uses the workspace directory to store its preferences and development artifacts.

Workspace: /Users/ /AdvoCATE-workspace

□ Use this as the default and do not ask again

Figure 1: When you launch AdvoCATE, you must select a Workspace to use.

Create a New Workspace

To create a new Workspace

1. Click **Browse…**. *The workspace can be set to the same folder where the AdvoCATE application is stored or elsewhere.*

   NOTE: If you create the Workspace directly within the AdvoCATE repository, you will need to use a command (like Git Ignore) to avoid committing all the non-project files (metadata, etc.).

2. Create a **new folder** in your file directory to select.
3. Click **Open**.
4. Click **Launch**.

Your new Workspace will open and load in AdvoCATE.

Select a Workspace

To select a Workspace path

1. Click **Browse…**.
2. Select a **directory location** in the pop-up window.
3. Click **Open**.
4. Click **Launch**.

Your selected Workspace will open and load in AdvoCATE.

Switch Workspace

To switch a Workspace

1. In the File menu, select **File** > **Switch Workspace**.
2. In the drop-down menu, either select an **existing Workspace** or click **Other…** to create a new Workspace. If you select Other, follow the steps to *Create a New Workspace*.
3. Click **Launch**.

Your selected Workspace will open and load in AdvoCATE.

Send AdvoCATE Configuration Log

To send your AdvoCATE Configuration Log to the AdvoCATE team

1. Locate your **AdvoCATE Configuration Folder** in your file system. *Your AdvoCATE Configuration Folder is usually where your saved Workspaces are located or where you made AdvoCATE executable.*
2. Inside your AdvoCATE Configuration Folder, locate your **Configuration Log**. *The Configuration Log ends in the extension **.log**.*
3. Send an email e-mail to *advocate-feedback@lists.nasa.gov* with the following information:
   a. A description of the **issue** or **problem** that occurred.
   b. A description of **what you were doing** when the issue or problem occurred.
   c. An attachment of your **Configuration Log** after the issue or problem occurred.

Depending on the severity of your problem, our team may also ask you for a copy of the Workspace or AdvoCATE Project that you were working on when the issue occurred.

AdvoCATE Perspective

After selecting a Workspace, AdvoCATE launches with the **AdvoCATE Perspective** open (*Figure 2*).

The Advocate Perspective is a visual container for the following set of Perspective Views and Editor:

- The **Toolbar** provides icons with functionality to create new AdvoCATE assurance artifacts and save them.
- The **Model Explorer View** lists assurance artifacts in each of your AdvoCATE Project folders.
- The **Outline View** provides a miniature view of your main diagram.
  - Outlines will only appear in the **Outline View** for diagrams and domain-specific languages (DSLs), and not tables.
- The **Editor** is used to interact with the primary assurance artifact you have open.
- The **Properties View** is used to edit the properties for AdvoCATE assurance artifacts in your editor.
- The **Problems View** is used to display the current errors, warnings, and other info in your Workspace, as well as to perform Quick Fixes for select Validation errors.

The views in the Advocate Perspective can be closed and rearranged.

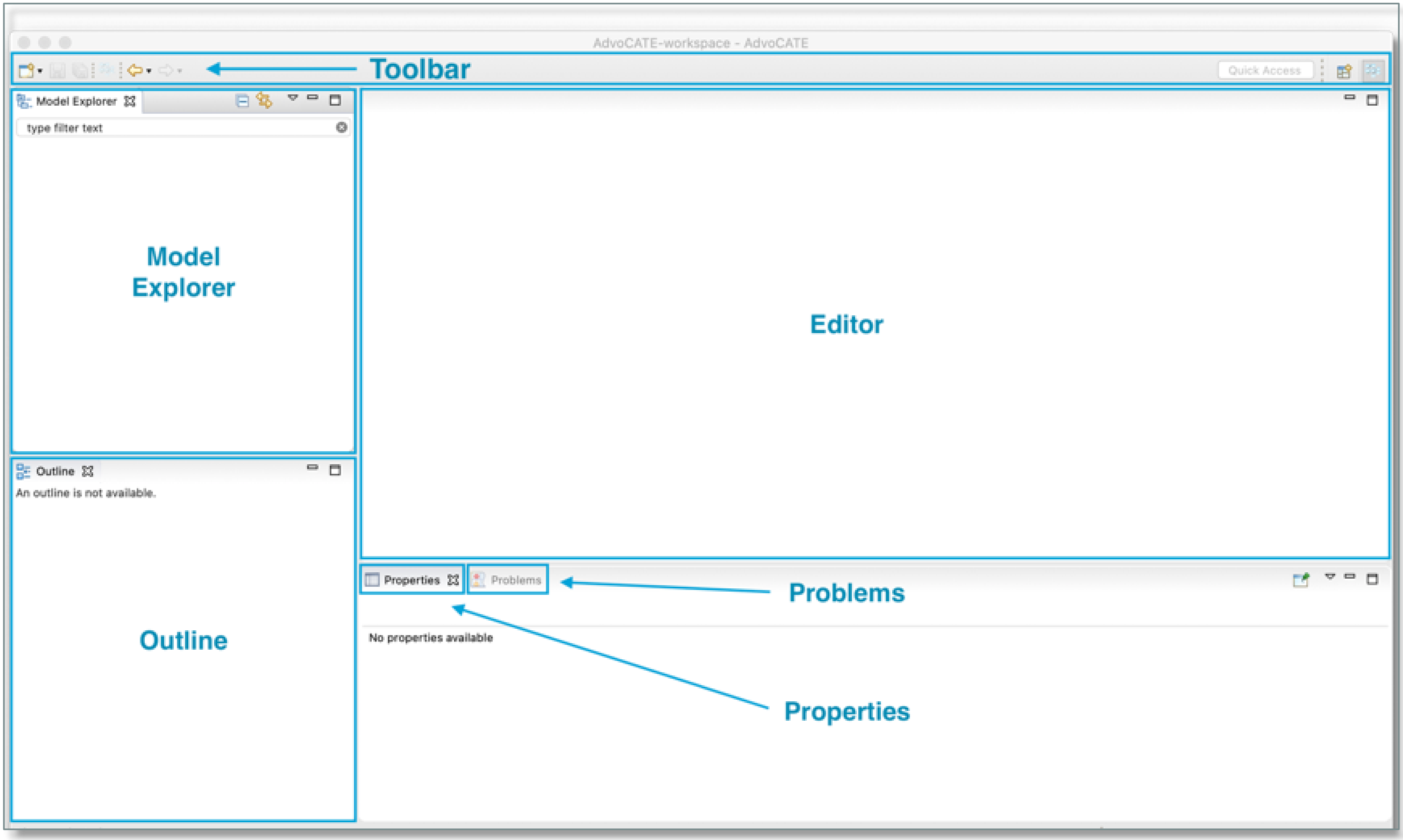


Figure 2: The AdvoCATE Perspective opens the Model Explorer, Outline, Properties, and Problems views, the Editor, and the Toolbar.

Open a Perspective View

To open a Perspective View

1. Select **Window** > **Show View** from the main menu.
2. Select the **view** you wish to open.

Your selected view will open.

Reset AdvoCATE Perspective

To reset the AdvoCATE Perspective to its default settings

1. Select **Window** > **Perspective** > **Reset Perspective…** from the main menu.
2. Confirm you want to **Reset Perspective**.

The AdvoCATE Perspective will reset to its default settings.

Quick Fix a Validation Error in the Problems View

To Quick Fix a Validation error in the Problems View

1. Open the **Problems View**.
2. Using the drop-down arrows, right-click the **Error**, **Warning**, or **Info** Validation error you wish to perform a Quick Fix for.
3. Select **Quick Fix** in the right-click menu.

   NOTE: Quick Fixes can only be performed for some Validation errors. As a result, the Quick Fix option may be grayed out and unable to be selected.

4. In the pop-up window, select a **fix** you wish to perform.
5. In the pop-up menu, select or deselect the **checkboxes** for the **problem/s** you wish to resolve using the selected fix.
6. Click **Finish**.

The Quick Fix will be performed for the selected problems. Use the **Save All** icon in the toolbar to update the Problems View list.

Create a New AdvoCATE Project

To create a new AdvoCATE Project

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **AdvoCATE Project**.

   By default, a project name is given and located in your active Workspace, which is convenient if you want to keep all your files in one place.

   When using the default location in the Workspace, the folder is set to the AdvoCATE Project name, but if it's stored elsewhere, it's possible to name the folder differently from the AdvoCATE Project (i.e., the assurance case). In that case, it's probably a good idea to give them the same name since the assurance case name will otherwise not be visible from the file system.

   If you created the Workspace directly within the AdvoCATE repository, you will need to use a command (like Git Ignore) to avoid committing all the non-project files (metadata, etc.).

3. Click **Next**.
4. Select the **Accept checkbox** for the AdvoCATE disclaimer.
5. Click **Next**.
6. Enter in a **new project name** as appropriate.
7. Specify your new AdvoCATE Project's **folder location**.
8. If you want to select a project location other than the workspace,
   - Uncheck the **Use Default Location** checkbox.
   - Type the path in the textbox field or click **Browse…** to select a folder location. *We recommend that the project name chosen be identical to its local folder in the file system.*
9. Click **Finish**.

The new AdvoCATE Project will open in the **Dashboard**.

Import an AdvoCATE Project

To import an AdvoCATE Project into a Workspace

1. Select **File** > **Import** from the main menu.
2. From the pop-up window, select **General** > **File System**.
3. Click **Next**.
4. In the **From Directory** textbox field, type in the textbox field or click **Browse…** to select your project's folder location.
5. Click **Open**.
6. Select the appropriate **checkboxes** to copy your selected AdvoCATE Projects and/or AdvoCATE Assurance Artifacts into the workspace.
7. In the **Into Folder** textbox field, type in the textbox field or click **Browse…** to select where to import your project.

   When importing an AdvoCATE Project, there is an option to "copy into the workspace". If you are just testing or reviewing and want to keep the original AdvoCATE Project pristine, then use this option. Otherwise, any saved edits will directly alter the original AdvoCATE Project.

   Alternatively, you can copy the AdvoCATE Project into the Workspace, edit it, and then use the Export menu to export those specific parts of the AdvoCATE Project back into the original location. For more information about exporting, see the File Export section.

8. Click **Open**.
9. Select the appropriate **checkbox options** to copy your projects into the Workspace.
10. Click **Finish**.

The imported AdvoCATE Project and/or AdvoCATE Assurance Artifacts will open in the **Dashboard**.

Delete an AdvoCATE Project or Assurance Artifact

To delete an AdvoCATE Project or Assurance Artifact

1. Right-click the **folder or file** in the **Model Explorer View** you wish to delete.
2. From the right-click menu, select ✖ **Delete**.
3. Click **Ok** to confirm your deletion request.

The selected AdvoCATE Project or Assurance Artifact will be deleted.

Main Features

AdvoCATE has fifteen main features:

1. *Dashboards*
2. *Physical Architecture (PhysArch)*
3. *Functional Specifications (FuncSpec)*
4. *Hazard Log*
5. *Requirements Log*
6. *Safety Architecture*
7. *Argument Structures*
8. *Argument Patterns*
9. *Data Trees*
10. *Data Mapping Files*
11. *Evidence Log*
12. *Tools Log*
13. *Allocations*
14. *Traceability & Navigation*
15. *File Export*

Dashboards

Advocate **Dashboards** are the main interface to interact with AdvoCATE Projects, Assurance Artifacts, and the tool.

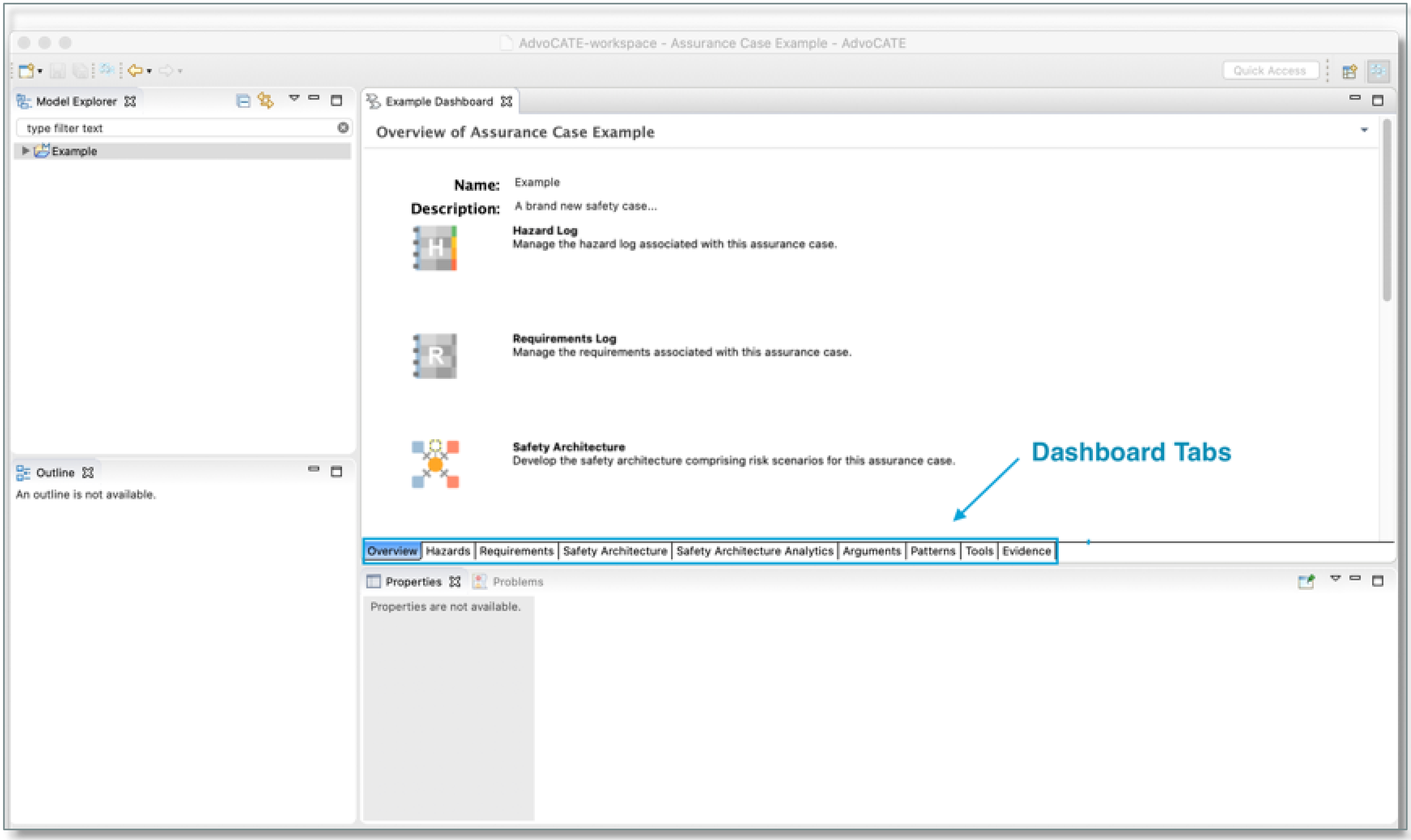
A newly created project opens the Dashboard by default.

Each Dashboard contains nine tabs to navigate through an AdvoCATE Project.

1. The **Overview** tab lists and describes all other dashboard tabs.
2. The **Hazards** tab manages the Hazard Log associated with an assurance case.
3. The **Requirements** tab manages the Requirements Log associated with an assurance case.
4. The **Safety Architecture** tab manages the risk scenarios associated with an assurance case.
5. The **Safety Architecture Analytics** tab shows a high-level overview of the associated risk levels of your Safety Architecture.
6. The **Arguments** tab manages Argument Structures associated with an assurance case.
7. The **Patterns** tab manages Argument Patterns and Data Trees associated with an assurance case.
8. The **Tools** tab manages the Tools Log associated with an assurance case.
9. The **Evidence** tab manages the Evidence Log associated with an assurance case.

Use the tabs to select a specific Dashboard to view (*Figure 3*).

NOTE: We'll refer to these Dashboard tabs as Dashboards in subsequent sections of this guide (e.g., Hazards Dashboard).



An outline is not available.

Overview of Assurance Case Example

Name: Example

Description: A brand new safety case...

Hazard Log
Manage the hazard log associated with this assurance case.

Requirements Log
Manage the requirements associated with this assurance case.

Safety Architecture
Develop the safety architecture comprising risk scenarios for this assurance case.

Properties are not available.

Figure 3: The Dashboard has nine tabs to navigate through an AdvoCATE Project.

Open Dashboard

To open the Dashboard

1. Click the **Open Dashboard** icon in toolbar. If you have multiple AdvoCATE Projects saved in your workspace, you can use the drop-down arrow next to the ***Open Dashboard*** icon to select which project dashboard to open.
    - Alternatively, you can right-click your project folder in **Model Explorer view** and select **Open Dashboard**.

When the Dashboard opens, the **Overview Dashboard** opens by default.

Overview Dashboard

In the Overview Dashboard, you can view and edit your Assurance Case name and description, import an Overview Image, and navigate to other Artifact Dashboards.

Import a New Overview Image

To import a new Overview Image

1. In the Model Explorer, select the **AdvoCATE Project** you wish to add a new Overview Image to.
2. In the Main Menu, select **File** > **Import….**.
3. In the pop-up window, toggle the **drop-down arrow** next to the **AdvoCATE** folder.
4. Select the **Dashboard Overview Image** import option.
5. Click **Next**.
6. Use the **Browse…** button to select the image file you wish to import.
7. Click **Finish**.

Your Overview Dashboard will update with the image.

NOTE: To change or remove your new Overview Image, right-click the Overview Image and pick your desired option.

Hazards Dashboard

In the Hazard Dashboard, there are ten windows you can use to view and manage your Hazard Log:

1. The **Hazard Tables** window lists the Hazard Log within organizational tables based on Hazardous Activity, System State, and Environmental Condition classifications.
2. The **Hazards Ordered by Risk** window lists the Risk Matrices for Hazards based on their risk levels.
3. The **Hazard Log Status** window lists the status of key Hazard Log entries: Hazards missing causes, Hazards missing effects, Causes missing mitigations, Mitigations missing verifications, Mitigations missing safety requirements, Mitigation count, Hazard count, and Unique causes.
4. The **Risk Distribution** window lists the distribution of initial and residual Hazards based on risk level classifications.
5. The **Action Type** window lists the distribution of residual Hazards based on action and residual risk level classifications.
6. The **Status Type** window lists the distribution of Open, Accepted, and Closed Hazard classifications.
7. The **Mitigations** window lists the distribution of new and existing Mitigations based on residual risk level classifications.
8. The **Threats Ordered by Likelihood** window lists the initial Risk Matrices for Threats based on their Likelihood levels.
9. The **Consequences Ordered by Severity** window lists the residual Risk Matrices for Consequences based on their Severity levels.
10. The **Mitigations Ordered by Importance** window lists the distribution of Mitigations based on a combination of Threat and Consequence levels.

Requirements Dashboard

In the Requirements Dashboard, there are five windows you can use to view and manage your Requirements Log:

1. The **Requirements Tables** window lists the Requirements Log as an organizational table.
2. The **Status** window lists the status of key Requirements Log entries: Missing source, Missing allocation, Missing verification method, and Complete.
3. The **External Sources** window lists the distribution of External Sources associated to the Requirements Log.
4. The **Allocations** window lists the distribution of Component and Function allocations associated to the Requirements Log.
5. The **Formalized Requirements** window lists the distribution of Not Formalized and Formalized Requirements.

Safety Architecture Dashboard

In the Safety Architecture Dashboard, there are four windows you can use to view and manage CESs and BTDs:

1. The **Hazardous Activities** window lists CESs based on Hazardous Activity, System State, and Environmental Condition classifications.
2. The **Events** window lists all Events, as well as Event Instances within their associated CESs.
3. The **Events Associated with Controlled Event Structures** window lists Event Instances within CESs. Orange-colored Events have at least one corresponding BTD associated to the Event Instance. Gray-colored Events have no corresponding BTDs associated to the Event Instance.
4. The **Barriers and Controls** window lists all Barriers and their associated Controls. If a Control is not associated to a Barrier, it will be listed separately.

Safety Architecture Analytics Dashboard

In the Safety Architecture Analytics Dashboard, there are six windows you can use to view and manage your Safety Architecture:

1. The **Hazards Ordered by Risk** window lists the Risk Matrices for Hazards based on their risk levels.
2. The **Safety Architecture Status** window lists the status of key Safety Architecture artifacts: Events missing bow ties, Consequences missing barriers, Escalations missing escalation barriers, Barriers missing escalations, and Threats missing barriers.
3. The **Threats Ordered by Likelihood** window lists the initial Risk Matrices for Threats based on their Likelihood levels.
4. The **Risk Distribution** window lists the distribution of initial and residual Hazards based on risk level classifications.
5. The **Consequences Ordered by Severity** window lists residual Risk Matrices for Consequences based on Severity levels.
6. The **Barriers Ordered by Importance** window lists the distribution of Barriers based on a combination of Threat and Consequence levels.

Arguments Dashboard

In the Arguments Dashboard, there are five windows you can use to view and manage your Argument Structures:

1. The **Arguments** window lists the Argument Structures within an AdvoCATE Project.
2. The **Issues** window lists any Validation errors for a selected Argument Structure in the Arguments window.
3. The **Metrics** window lists key artifacts for a selected Argument Structure in the Arguments window: Nodes, Undeveloped nodes, Maximum depth, Branching average, Goals, Goals fully supported, Goals partially supported, Solutions, and Solutions without evidence.
4. The **Status** window lists the percentage of developed nodes for a selected Argument Structure in the Arguments window.
5. The **Referenced Artifacts** window lists all the associated AdvoCATE Artifacts for a selected Argument Structure in the Arguments window based on Type, Referenced Artifact, Used For Solutions, and Arguments or Nodes.

Patterns Dashboard

In the Patterns Dashboard, there are five windows you can use to view and manage your Argument Patterns and Data Trees:

1. The **Patterns** window lists the Argument Patterns within an AdvoCATE Project.
2. The **Metrics** window key features for a selected Argument Pattern in the Patterns window: Maximum loop length, Maximum depth, Choices, Loops, Parameters, and Nodes.
3. The **Instantiation Data and Instances** window lists Argument Patterns, as well as their Source Data Trees and corresponding Instance Argument Structures.
4. The **Data Trees** window lists the Data Trees within an AdvoCATE Project.
5. The **Issues** window lists any Validation errors for a selected Argument Pattern in the Patterns window.

Tools Dashboard

In the Tools Dashboard, there are three windows you can use to view and manage your Tools Log:

1. The **Tools** window lists the Tools Log, its Tools, and the Tools' associated Tool Uses, as well as any associations to Requirements and Argument Nodes.
2. The **Tools Statistics** window lists Tools and their Invocations (or Tool Uses).
3. The **Toolsets** window lists Toolsets and their associated Tools.

Evidence Dashboard

In the Evidence Dashboard, there are five windows you can use to view and manage your Evidence Log:

1. The **Evidence Artifacts** window lists the Evidence Log and its Evidence Artifacts.
2. The **Summary** window lists the Identifier, Description, Purpose, Type, Version, Provenance, and Status fields for a selected Evidence Artifact in the Evidence Artifacts window.
3. The **Evidence Usage** window lists the AdvoCATE Artifact associations for a selected Evidence Artifact in the Evidence Artifacts window.
4. The **Evidence Types** window lists the distribution of Evidence Artifacts within the Evidence Type classifications.
5. The **Status** window lists the distribution of Pending, To be verified, and Verified classifications for Evidence Artifacts.

Physical Architecture (PhysArch)

The **Physical Architecture (PhysArch)** is where you list the physical decomposition of your AdvoCATE Project's system.

Create a New PhysArch File

To create a new PhysArch file

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Physical Decomposition**.
3. Click **Next**.
4. Click **Browse…** to select your AdvoCATE Project's folder location.
5. Select your **AdvoCATE Project's folder**.
6. Click **Ok**.
7. Name your **PhysArch file**.

   NOTE: PhysArch files must end in the file extension **.physarch**.

The new PhysArch file opens and contains example entries in the **PhysArch DSL**. The example entries in a newly generated PhysArch file are editable, and you can delete the unnecessary options.

PhysArch DSL

The **PhysArch domain-specific language (DSL)** is used to hierarchically list and define the Physical Components and Sub-Components of your system.

The PhysArch DSL starts with the **physical decomposition** keyword, followed by the syntax version number and name of the PhysArch. The PhysArch DSL's syntax uses Component, Sub-Component, Failure Mode, and Comment entries.

Hierarchal relationships between entries and sub-entries can be expressed by using brackets { } and indentations.

NOTE:

- For DSL Content Assist, users can use the CTRL + Space keyboard shortcut to see suggestions for DSL keyword entries.
- A DSL syntax error will occur if an entry is missing required details.
- For more information on the constraints between **Components** and **Failure Modes**, see the *Validations Appendix*.

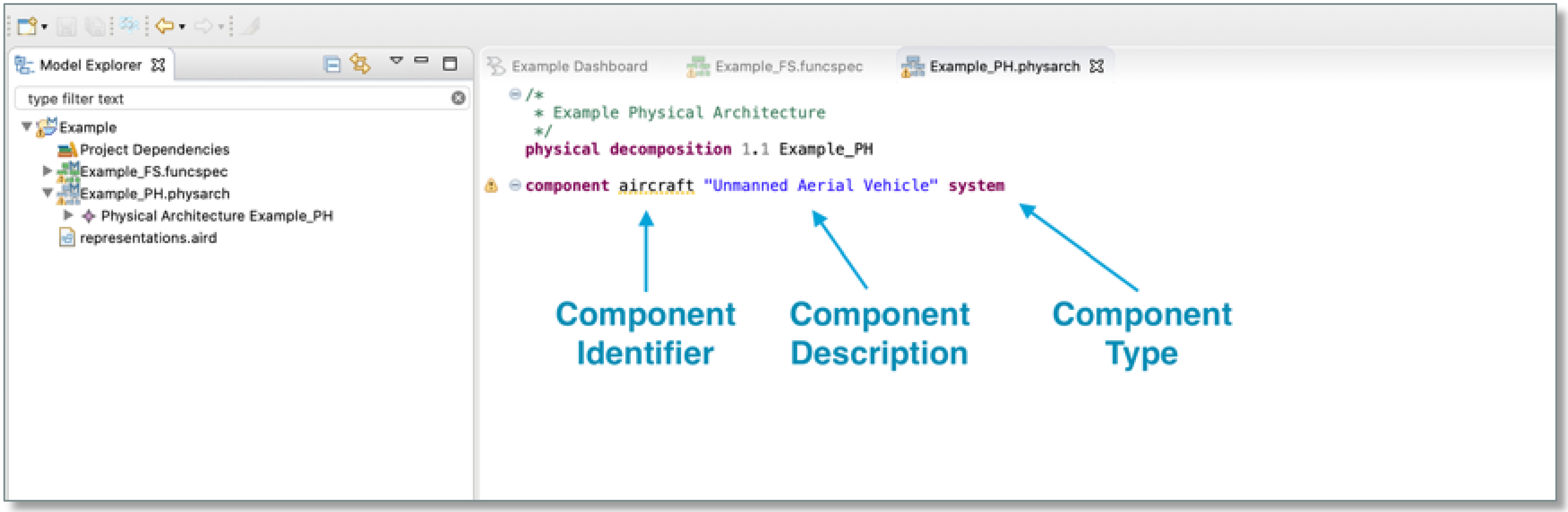
Components

Components are the physical items that form your overall system.

Component entries start with the **component** keyword and list the following details:

- **Identifier**: The component identifier is the abbreviated name of the Component. Identifiers cannot contain spaces.
- **Description**: The component description is enclosed with “” quotation marks and is the formal name of the Component.
- **Type**: The component can be classified as a **system** or **safety** type.
    - A **system** type constitutes part of the baseline system under assurance.
    - A **safety** type constitutes part of the safety system.

Below (*Figure 4*) is an example of a Component entry.



```
/*
 * Example Physical Architecture
 */
physical decomposition 1.1 Example_PH

component aircraft "Unmanned Aerial Vehicle" system
```

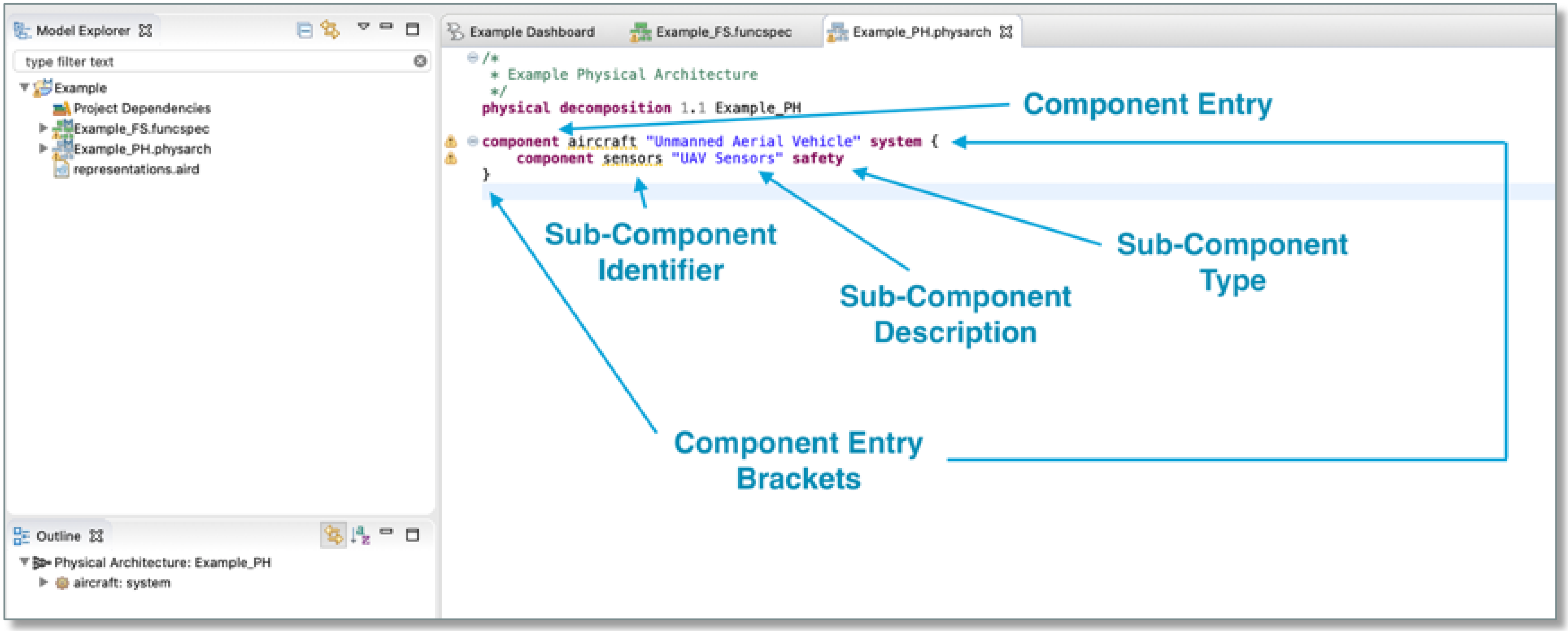
Figure 4: In the PhysArch, Component and Sub-Component DSL entries are comprised of a component element, an identifier, a description, and a type.

Sub-Components

Sub-Components are the physical items that form their parent Component.

Sub-Components entries are nested within their parent Component's {} brackets and follow the same format as Component entries. However, Sub-Components can have a different type from their parent Component.

Below (*Figure 5*) is an example of a parent Component entry and its child Sub-Component entry.



```
/*
 * Example Physical Architecture
 */
physical decomposition 1.1 Example_PH

component aircraft "Unmanned Aerial Vehicle" system {
    component sensors "UAV Sensors" safety
}
```

Figure 5: In the PhysArch, Sub-Component DSL entries are listed within the brackets of their parent Component entries.

Sub-Components can also be used to denote sub-systems and their physical decomposition. There are no restrictions on nesting levels.

Failure Modes

Failure Modes are the potential malfunctions of Components.

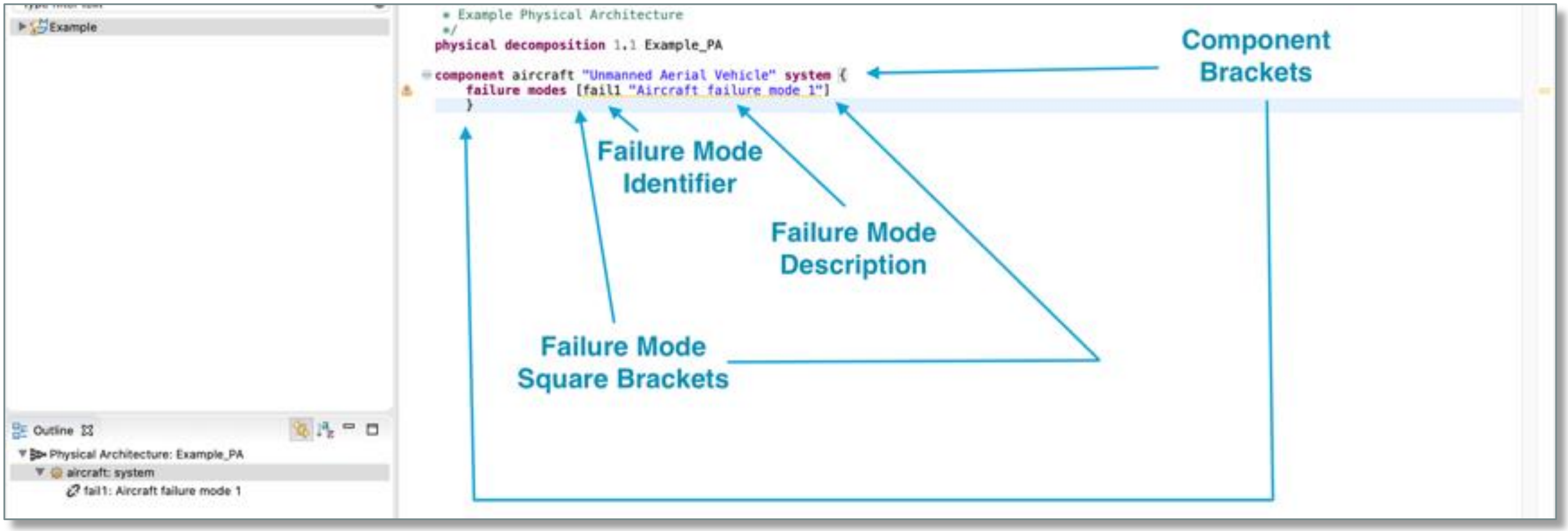
Failure Mode entries must be denoted under their corresponding Components. Failure Modes do not form a hierarchy and cannot be shared between Components.

Failure Mode entries start with the **failure modes** keyword and can list the following details in [ ] square brackets:

- **Identifier**: The failure mode identifier is the abbreviated name of the Failure Mode. Identifiers cannot contain spaces.
- **Description**: The failure mode description is enclosed with “” quotation marks and is the formal name of the Failure Mode.

If a Component has more than one Failure Mode, offset each identifier with a , comma.

Below (*Figure 6*) shows an example of a Component entry and its Failure Mode.



```
 * Example Physical Architecture
 */
physical decomposition 1.1 Example_PA

component aircraft "Unmanned Aerial Vehicle" system {
    failure modes [fail1 "Aircraft failure mode 1"]
}
```

```
Outline
▼ Physical Architecture: Example_PA
  ▼ aircraft: system
      fail1: Aircraft failure mode 1
```

Figure 6: In the PhysArch, Failure Mode DSL entries are listed under their corresponding parent Component.

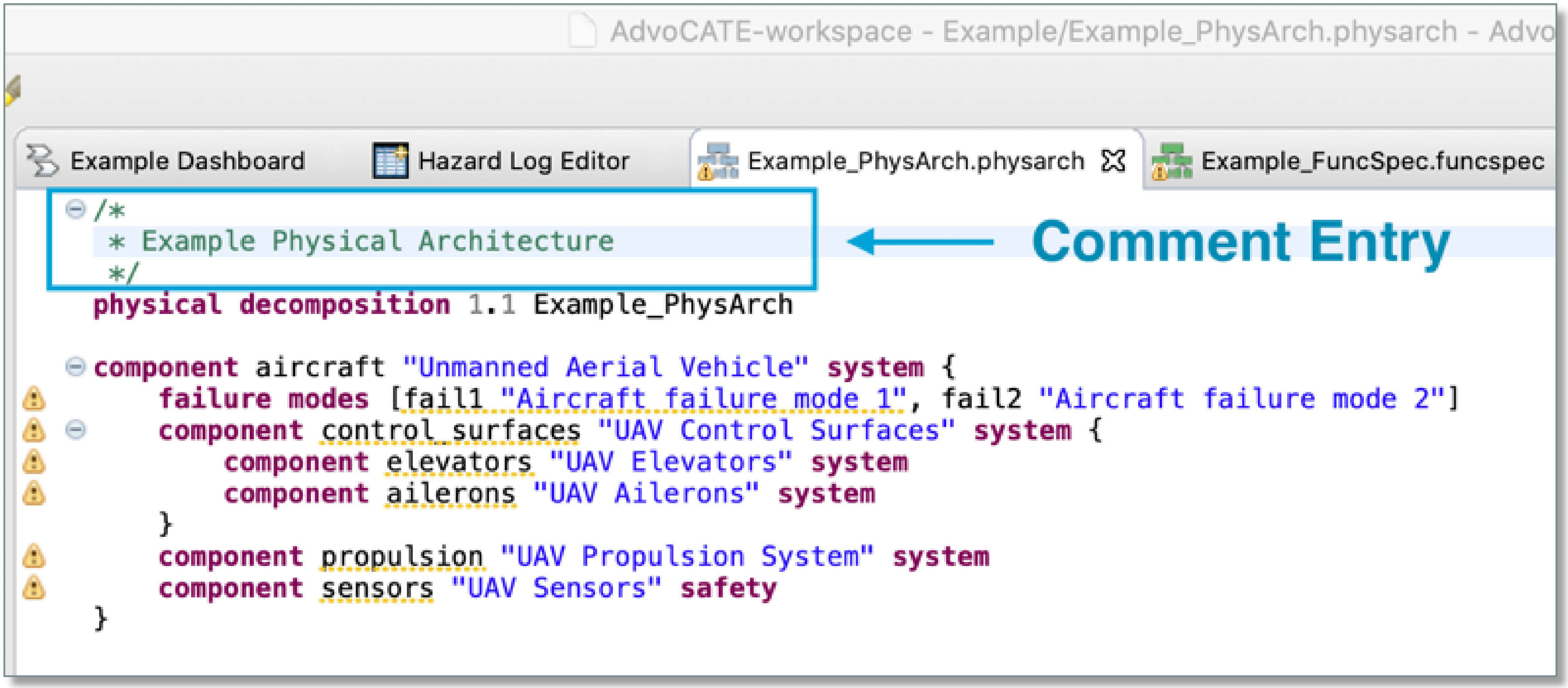
Comments

Comment entries are user notes in a DSL file. Comment entries do not affect other DSL entries.

Single-lined comments are enclosed within // // two-backslashes.

Multiple-lined comments are enclosed in /* */ a backslash and asterisk, with each comment line offset with an additional * asterisk (*Figure 7*).

You can use Comments in any DSL file within AdvoCATE.


AdvoCATE-workspace - Example/Example_PhysArch.physarch - Advo
Example Dashboard
Hazard Log Editor
Example_PhysArch.physarch
Example_FuncSpec.funcspec
Comment Entry

```
/*
 * Example Physical Architecture
 */
physical decomposition 1.1 Example_PhysArch

component aircraft "Unmanned Aerial Vehicle" system {
    failure modes [fail1 "Aircraft failure mode 1", fail2 "Aircraft failure mode 2"]
    component control_surfaces "UAV Control Surfaces" system {
        component elevators "UAV Elevators" system
        component ailerons "UAV Ailerons" system
    }
    component propulsion "UAV Propulsion System" system
    component sensors "UAV Sensors" safety
}
```

Figure 7: Comment DSL entries can be used in any AdvoCATE DSL file.

Functional Specifications (FuncSpec)

The **Functional Specifications (FuncSpec)** is where you list the functional decomposition of your AdvoCATE Project's system.

Create a New FuncSpec File

To create a new FuncSpec file

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Functional Decomposition**.
3. Click **Next**.
4. Click **Browse…** to select your AdvoCATE Project's folder location.
5. Select your **AdvoCATE Project's folder**.
6. Click **Ok**.
7. Name your **FuncSpec** file.

   NOTE: FuncSpec files must end in the file extension **.funcspec**.

The new FuncSpec file opens and contains example entries in the **FuncSpec DSL**. The example entries in a newly generated FuncSpec file are editable, and you can delete the unnecessary options.

FuncSpec DSL

The **FuncSpec domain-specific language (DSL)** starts with the function decomposition keyword, followed by the syntax version number and name of the FuncSpec. The FuncSpec contains two lists that define the Deviation and Function Hierarchies for your system.

The **Deviation Hierarchy** lists Deviation and Sub-Deviation Definition entries.

The **Function Hierarchy** lists Function, Sub-Function, Deviation Associations, and Component Allocation entries.

Hierarchal relationships between entries and sub-entries can be expressed by using { } brackets and indentations.

NOTE:

- For DSL Content Assist, users can use the CTRL + Space keyboard shortcut to see suggestions for DSL keyword entries.
- A DSL syntax error will occur if an entry is missing required details.
- For more information on the constraints between **Functions** and **Deviations**, see the *Validations Appendix*.

Deviation Hierarchy

The **Deviation Hierarchy** defines Deviations that you associate to corresponding Functions.

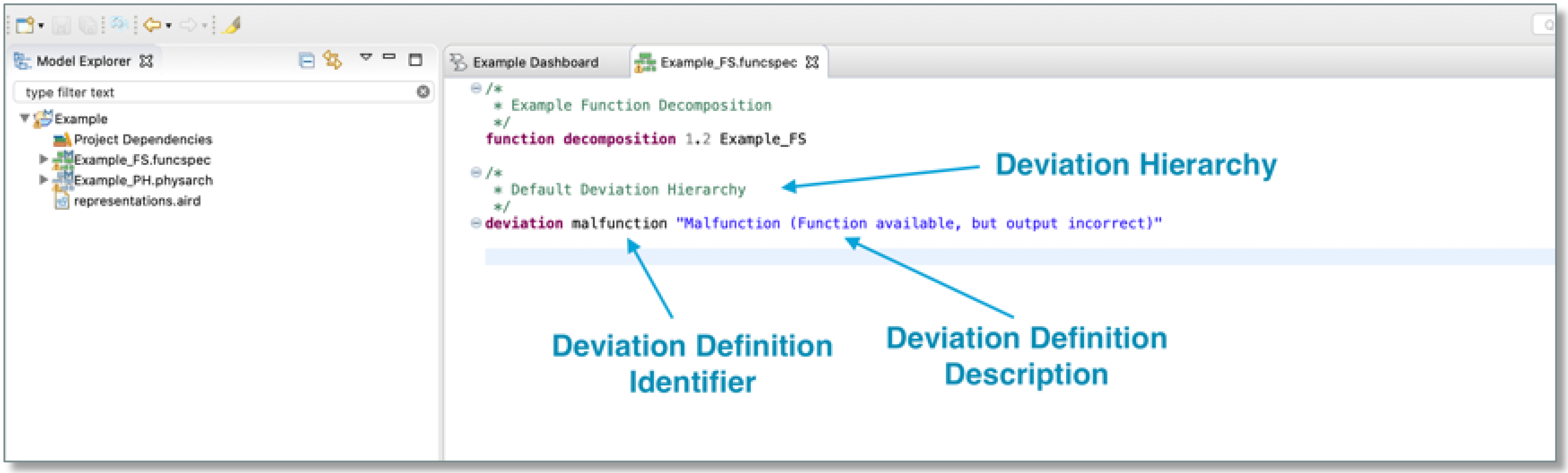
Deviation Definitions

Deviation Definitions declare the functional operations that violate the Function’s standard operating process.

Deviation Definition entries start with the **deviation** keyword and list the following details:

- **Identifier**: The deviation identifier is the abbreviated name of the Deviation Definition. Identifiers cannot contain spaces.
- **Description**: The deviation description is enclosed with “” quotation marks and is the formal name of the Deviation Definition.

Below (*Figure 8*) is an example of a Deviation Definition entry.



```
/*
 * Example Function Decomposition
 */
function decomposition 1.2 Example_FS

/*
 * Default Deviation Hierarchy
 */
deviation malfunction "Malfunction (Function available, but output incorrect)"
```

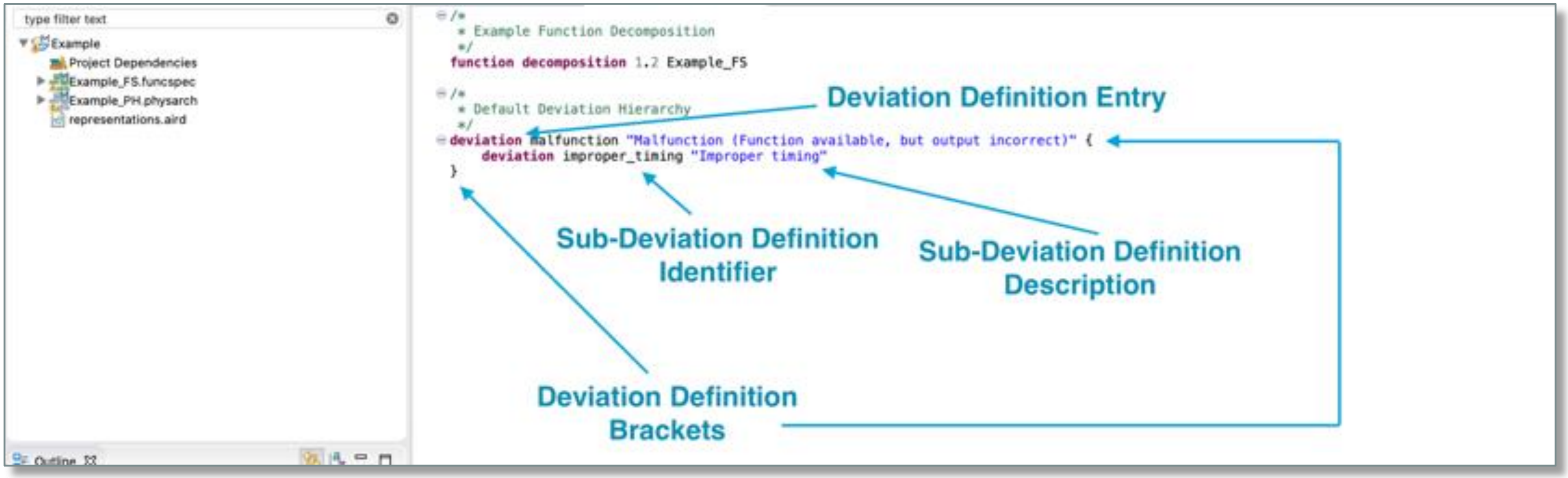
Figure 8: In the FuncSpec file, Deviation Definition DSL entries are listed in the Deviation Hierarchy.

Sub-Deviation Definitions

Sub-Deviation Definitions further define their parent Deviation.

Sub-Deviation Definition entries are within their parent Deviation Definition’s {} brackets and follow the same format as Deviation Definition entries.

Below (*Figure 9*) is an example of a parent Deviation Definition entry and its child Sub-Deviation Definition entry. There are no restrictions on nesting levels.



```
/*
 * Example Function Decomposition
 */
function decomposition 1.2 Example_FS

/*
 * Default Deviation Hierarchy
 */
deviation malfunction "Malfunction (Function available, but output incorrect)" {
    deviation improper_timing "Improper timing"
}
```

type filter text
Example
Project Dependencies
Example_FS.funcspec
Example_PH.physarch
representations.aird
Outline
Deviation Definition Entry
Sub-Deviation Definition Identifier
Sub-Deviation Definition Description
Deviation Definition Brackets

Figure 9: In the FuncSpec file, Sub-Deviation Definition DSL entries are listed within their parent Deviation Definition’s brackets.

Function Hierarchy

The **Function Hierarchy** defines Functions, Deviation Associations, and their corresponding Components Allocations from the PhysArch.

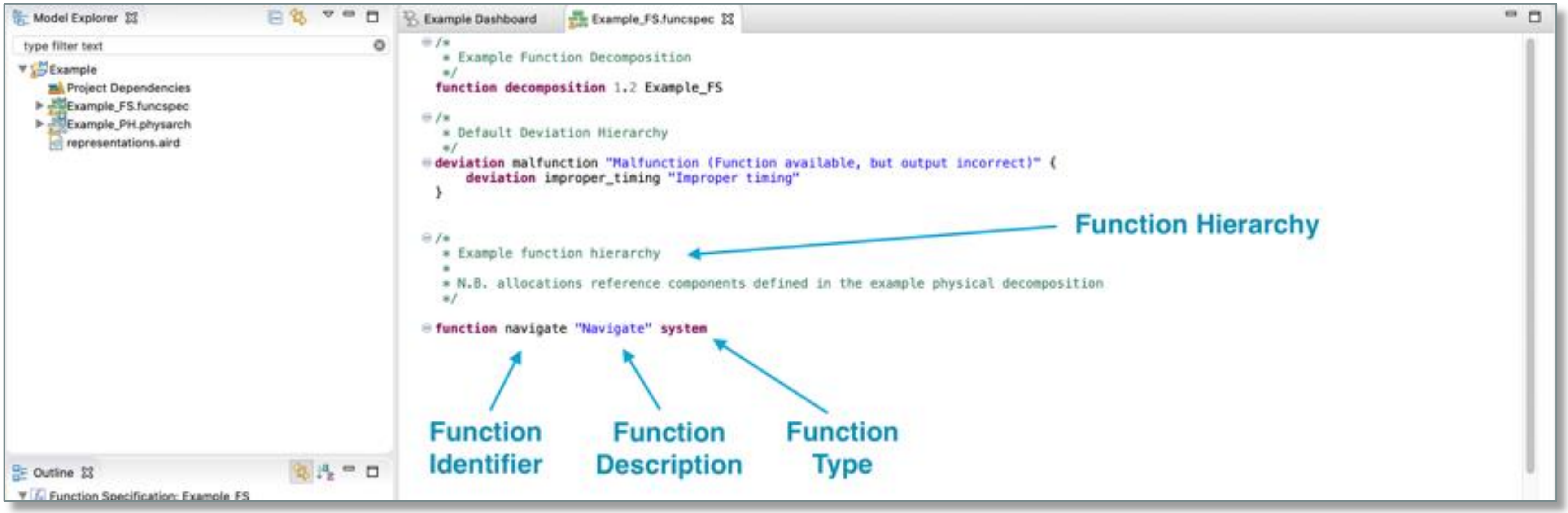
Functions

Function entries are various actions or processes that a system can perform.

Function entries start with the function keyword and list the following details:

- **Identifier**: The function identifier is the abbreviated name of the Function. Identifiers cannot contain spaces.
- **Description**: The function description is enclosed with “” quotation marks and is the formal name of the Function.
- **Type**: The function can be classified as a system or safety type.
  - A system type constitutes part of the baseline system under assurance.
  - A safety type constitutes part of the safety system.

Below (*Figure 10*) is an example of a Function entry.



```
/*
 * Example Function Decomposition
 */
function decomposition 1.2 Example_FS

/*
 * Default Deviation Hierarchy
 */
deviation malfunction "Malfunction (Function available, but output incorrect)" {
    deviation improper_timing "Improper timing"
}


/*
 * Example function hierarchy
 *
 * N.B. allocations reference components defined in the example physical decomposition
 */

function navigate "Navigate" system
```

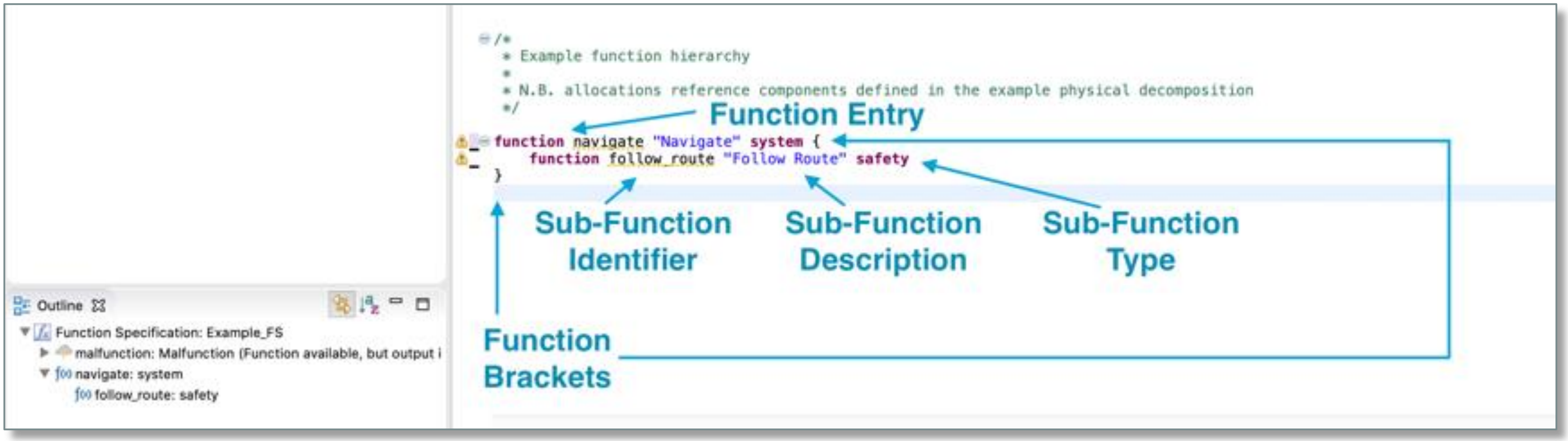
Figure 10: In the FuncSpec file, Function DSL entries are listed in the Function Hierarchy.

Sub-Functions

Sub-Functions are the processes or actions that form their parent Function.

Sub-Function entries follow the same format as Function entries. However, Sub-Functions can have a different type from their parent Function.

Below (*Figure 11*) is an example of a parent Function entry and its child Sub-Function entry. There are no restrictions on nesting levels.



```
/*
 * Example function hierarchy
 *
 * N.B. allocations reference components defined in the example physical decomposition
 */
function navigate "Navigate" system {
    function follow_route "Follow Route" safety
}
```

Outline
- Function Specification: Example_FS
  - malfunction: Malfunction (Function available, but output i
  - navigate: system
    - follow_route: safety

Figure 11: In the FuncSpec file, Sub-Function DSL entries are listed within their parent Function's brackets.

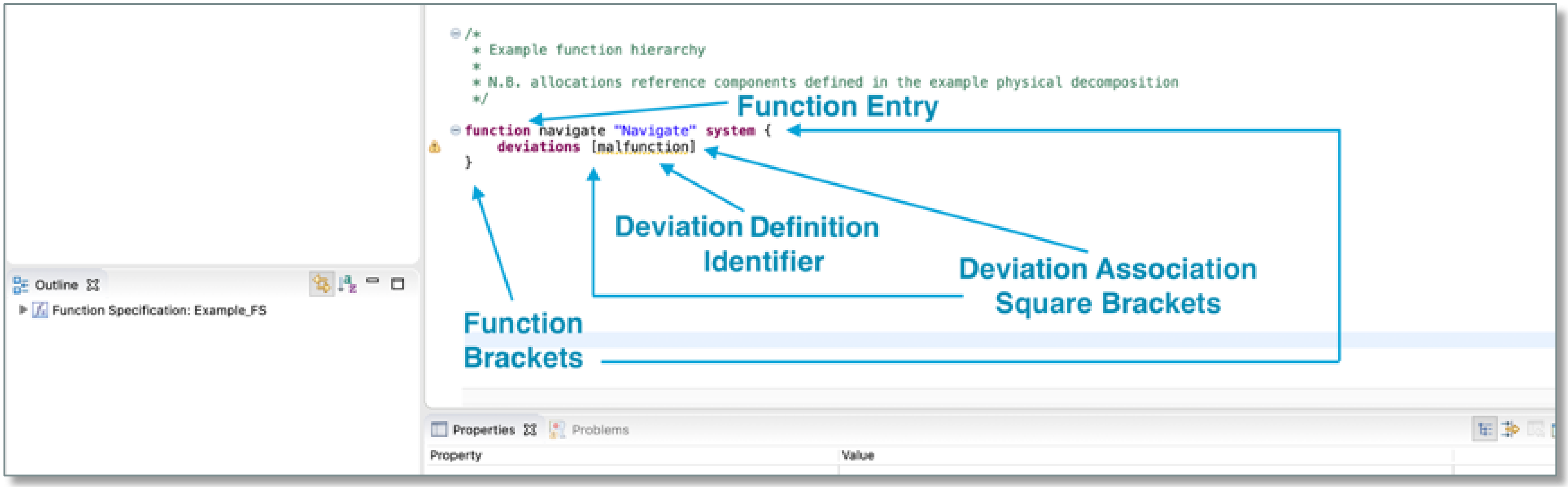
Deviation Associations

Deviation Associations are how Deviation Definitions (in the Deviation Hierarchy) are assigned to their corresponding Functions or Sub-Functions (in the Function Hierarchy).

Deviation Association entries must be denoted under their corresponding Functions. Deviation Associations are not inherited by Sub-Functions from their parent Function.

Deviation Association entries start with the **deviations** keyword and list their associated Deviation Definition identifier/s in [ ] square brackets. If a Function has more than one Deviation Association, offset each with a , comma.

Below (*Figure 12*) is an example of a Function entry and its Deviation Association and Sub-Function entries.



```
/*
 * Example function hierarchy
 *
 * N.B. allocations reference components defined in the example physical decomposition
 */
function navigate "Navigate" system {
    deviations [malfunction]
}
```

Function Entry
Deviation Definition Identifier
Deviation Association Square Brackets
Function Brackets
Outline
Function Specification: Example_FS
Properties
Problems
Property
Value

Figure 12: In the FuncSpec file, Deviation Association DSL entries are listed within their parent Function’s brackets.

Component Allocations

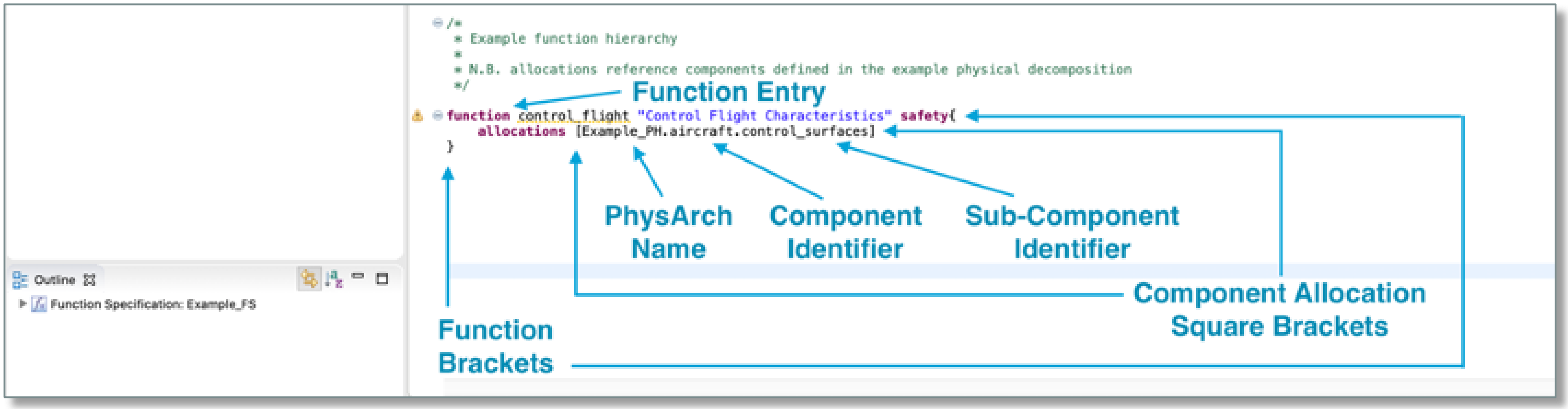
Component Allocations are how Functions reference the Physical Components they use.

Component Allocations entries must be denoted under their corresponding Functions.

Component Allocations entries start with the **allocations** keyword. In the Component Allocation's [ ] square brackets, list the PhysArch file name, followed by a . period. After the PhysArch name, list the Component identifier.

- If a Component Allocation is referencing a Sub-Component, the Sub-Component identifier needs to be referenced after its parent Component and a . period.
- If a Function has more than one Component Allocation, offset each with a , comma.

Below (*Figure 13*) is an example of a Function entry and its Component Allocation entry:



```
/*
 * Example function hierarchy
 *
 * N.B. allocations reference components defined in the example physical decomposition
 */
function control_flight "Control Flight Characteristics" safety{
    allocations [Example_PH.aircraft.control_surfaces]
}
```

Figure 13: In the FuncSpec file, Component Allocation DSL entries are listed within their parent Function's brackets.

Hazard Log

The **Hazard Log** contains a hierarchical list of Hazardous Activities and their System States, Environmental Conditions, Hazards, Causes, Mitigation Mechanisms, and Risk Assessments.

Each AdvoCATE project contains one **Hazard Log File**. However, the Hazard Log File can contain multiple **Hazard Logs** based on **Hazardous Activity**, **System State**, and **Environmental Condition** classifications.

Create a New Hazard Log

To create a new Hazard Log File

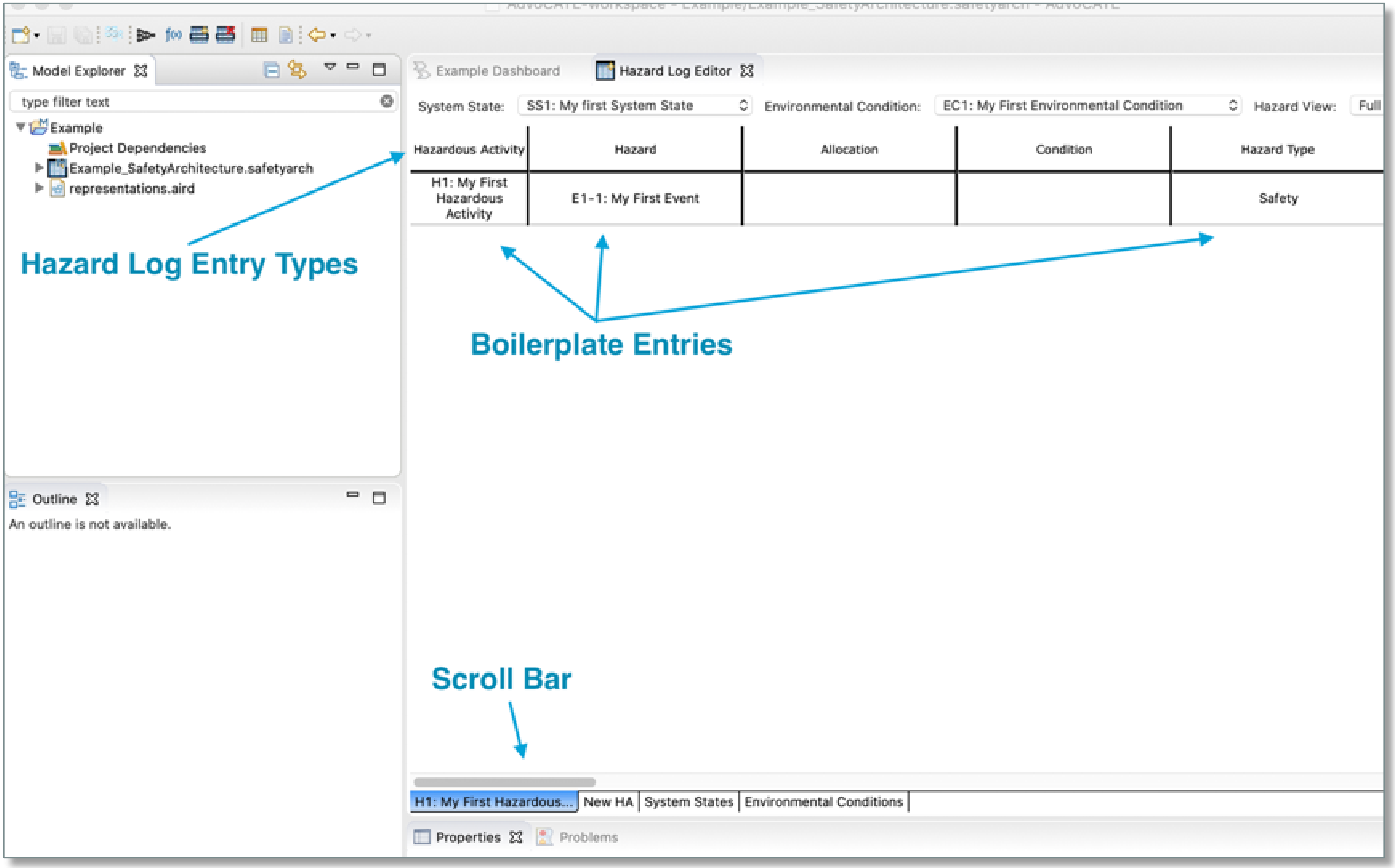
1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Hazard Log**.
3. Select the **AdvoCATE project folder** that will contain the Hazard Log.
4. Name the **Hazard Log**.

   NOTE: A Hazard Log must end in the extension ***.safetyarch*** since it is closely integrated with the Safety Architecture.

The **Hazard Log Editor** will open, and a new Hazard Log renders in tabular form.

A new **Hazard Log** contains editable boilerplate entries. Use the scroll bar to see the various entry types available in the **Full View** (*Figure 14*). For more information about the Full View, see the *Hazard Views* section.

We recommend editing the boilerplate entry with your first Hazardous Activity for your project. For more information about editing entries, see the *Hazard Log Entries* section.



AdvoCATE-Workspace - Example/Example_SafetyArchitecture.safetyarch - AdvoCATE

An outline is not available.

System State: SS1: My first System State

Environmental Condition: EC1: My First Environmental Condition

Hazard View: Full

Hazardous Activity	Hazard	Allocation	Condition	Hazard Type
H1: My First Hazardous Activity	E1-1: My First Event			Safety

H1: My First Hazardous... | New HA | System States | Environmental Conditions

Figure 14: When a new Hazard Log is created, the Hazard Table opens in Full View and boilerplate entries are created for your assurance case.

Preliminary Hazard Analysis (PHA)

A **Preliminary Hazard Analysis (PHA)** is the initial risk analysis performed to identify potential Hazardous Activities and Hazards associated with a system's operations and to create Mitigation Mechanisms to control and manage those Hazards.

As you perform a PHA, you may have to create new Hazard Logs based on based on emerging **Hazardous Activity**, **System State**, and **Environmental Condition** developments.

NOTE: This user guide assumes you know how to perform a PHA. For more information about how to perform a PHA, see the *Guide References* section.

Create a New Hazard Table

To create a new Hazard Table

1. Open the **Hazards Dashboard**.
2. In the **Hazard Tables** window, select the **Hazardous Activity** you want to create a Hazard Table under (*Figure 15*). *If you need to create a new **Hazardous Activity**, click the **New Hazardous Activity button**, and add your Hazardous Activity's description first.*
3. Once your Hazardous Activity is selected, click **New Hazard Table**.
4. Select your Hazardous Activity's **System State** from the drop-down menu. *If you need to create a new **System State** for your Hazardous Activity, see the* Create a New System State *section.*
5. Select your Hazardous Activity's **Environmental Condition** from the drop-down menu. *If you need to create a new **Environmental Condition** for your Hazardous Activity, see the* Create a New Environmental Condition *section.*
6. Click **Ok**.

The new **Hazard Table** will generate and be viewable in the **Hazard Tables** window.

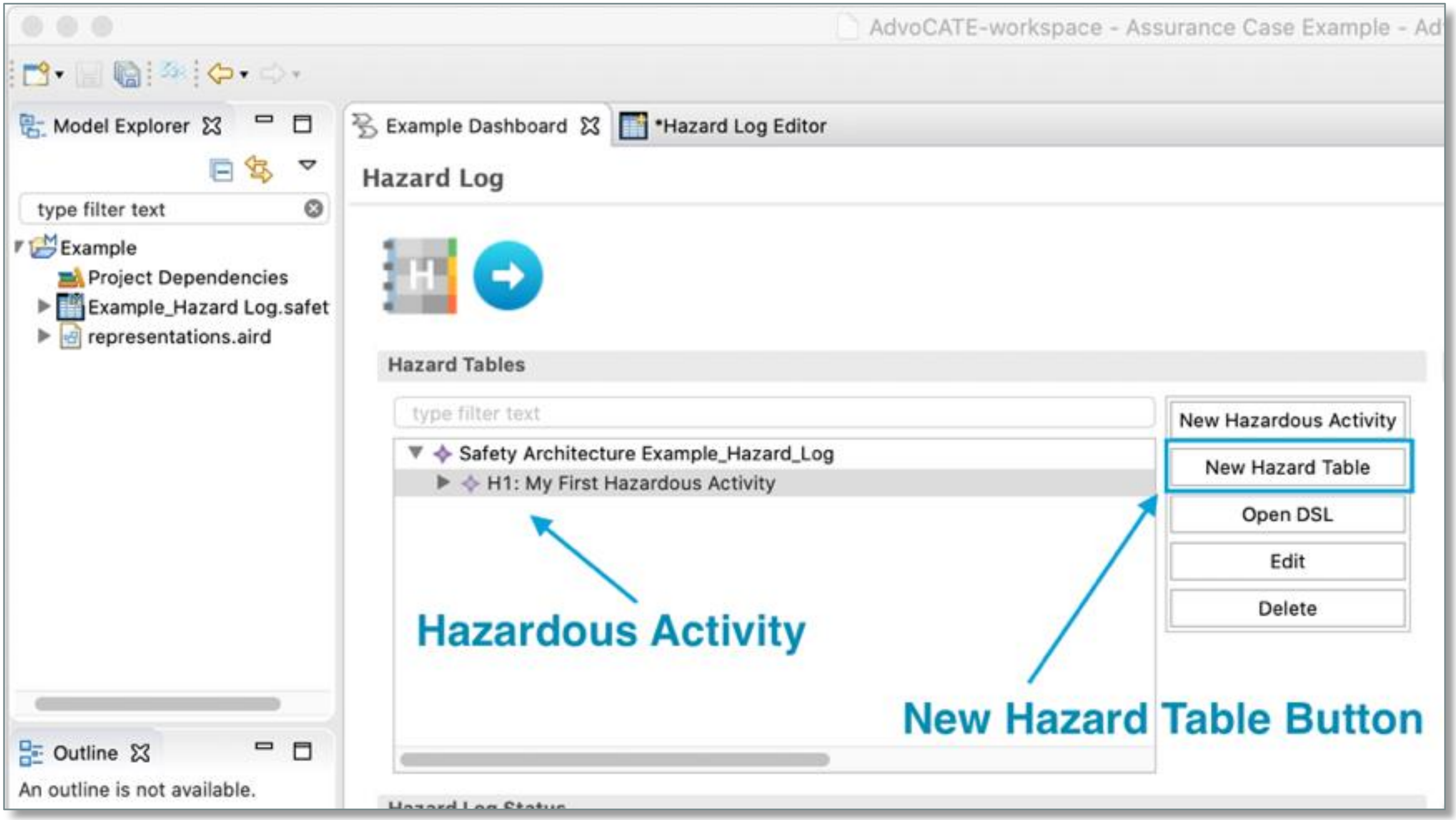


Figure 15: To create a new Hazard Table, select the Hazardous Activity you want to create it under before selecting the New Hazard Table button.

Open a Hazard Table

To open a Hazard Table

1. Open the **Hazards Dashboard**.
2. In the **Hazard Tables** window, select your **Hazard Table** the under the corresponding Hazardous Activity.
3. Click **Open**.

The Hazard Table will open in the **Hazard Log Editor**.

Delete a Hazard Table

To delete a Hazard Table

1. Open the **Hazards Dashboard**.
2. In the **Hazard Tables** window, select the **Hazard Table** you want to delete under the corresponding Hazardous Activity.
3. Click **Delete**.
4. Click **Yes** to confirm your deletion request.

The Hazard Table will be deleted from your AdvoCATE Project.

System States Table

If your AdvoCATE Project has various **System States**, you can enter those System States in the Hazard Log Editor.

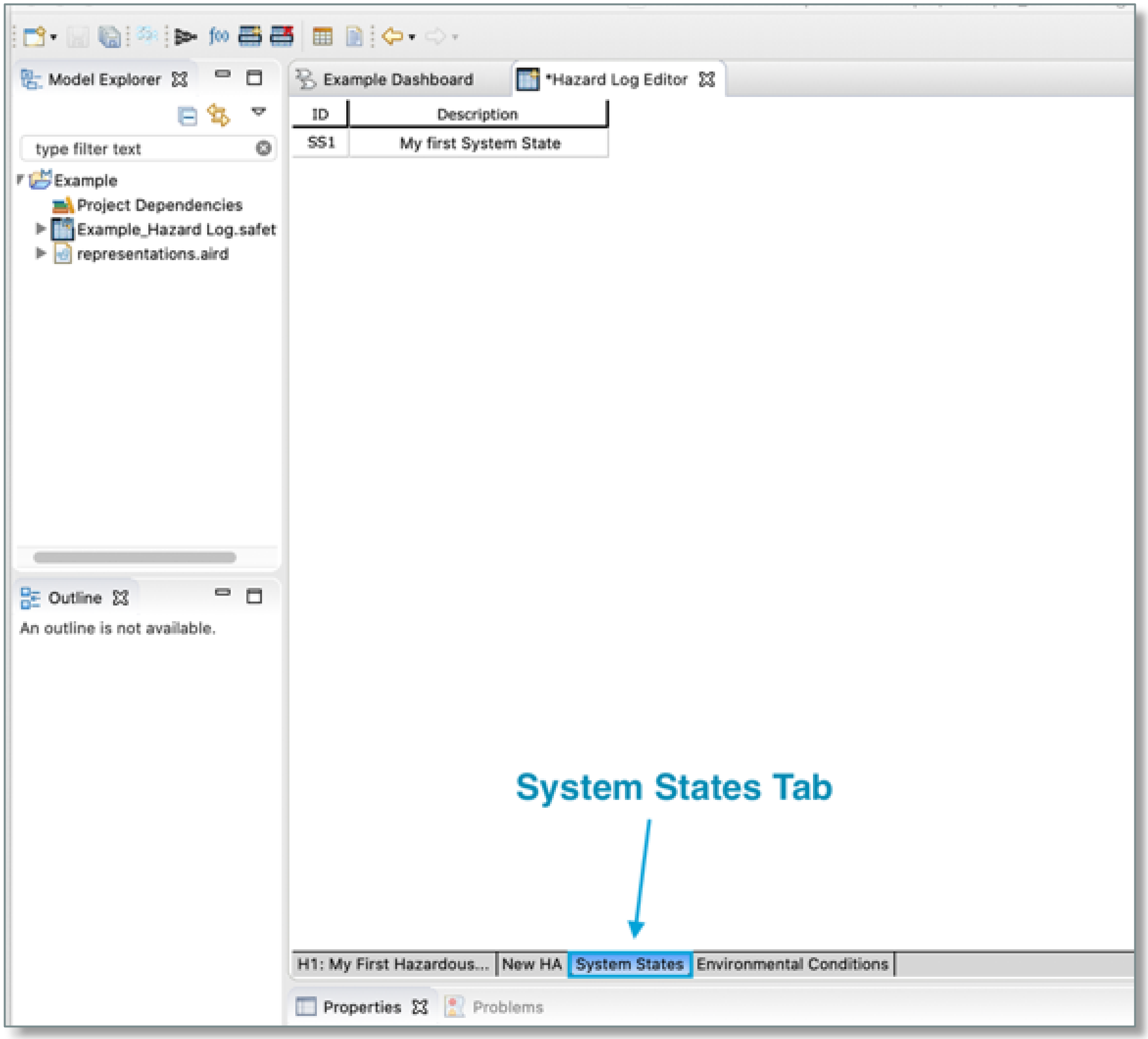
A **System State** is a condition your physical system may transform into for the purpose of a specific system function.

Open System States Table

To open your System States Table

1. Open your **Hazard Table**.
2. In the Hazard Log Editor, select the **System States** tab (*Figure 16*).

The System States Table will open.



ID	Description
SS1	My first System State

An outline is not available.

Figure 16: In the Hazard Log Editor, select the System States tab to open the System States Table.

Create a New System State

To create a new System State

1. Open your **System States Table**.
2. Click the **Add Element** icon in toolbar.

A new System State will generate in the **System States Table**.

Edit a System State

To edit a System State

1. Open your **System States Table**.
2. Double-click the **table cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click outside the **table cell** you edited to update your System State.

The text in your table cell will update to reflect the change.

Delete a System State

To delete a System State

1. Open your **System States Table**.
2. Select the **System State** you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected System State will be deleted from the **System States Table**.

Environmental Conditions Table

If your system will interact with various **Environmental Conditions**, you can enter those Environmental Conditions in the Hazard Log Editor.

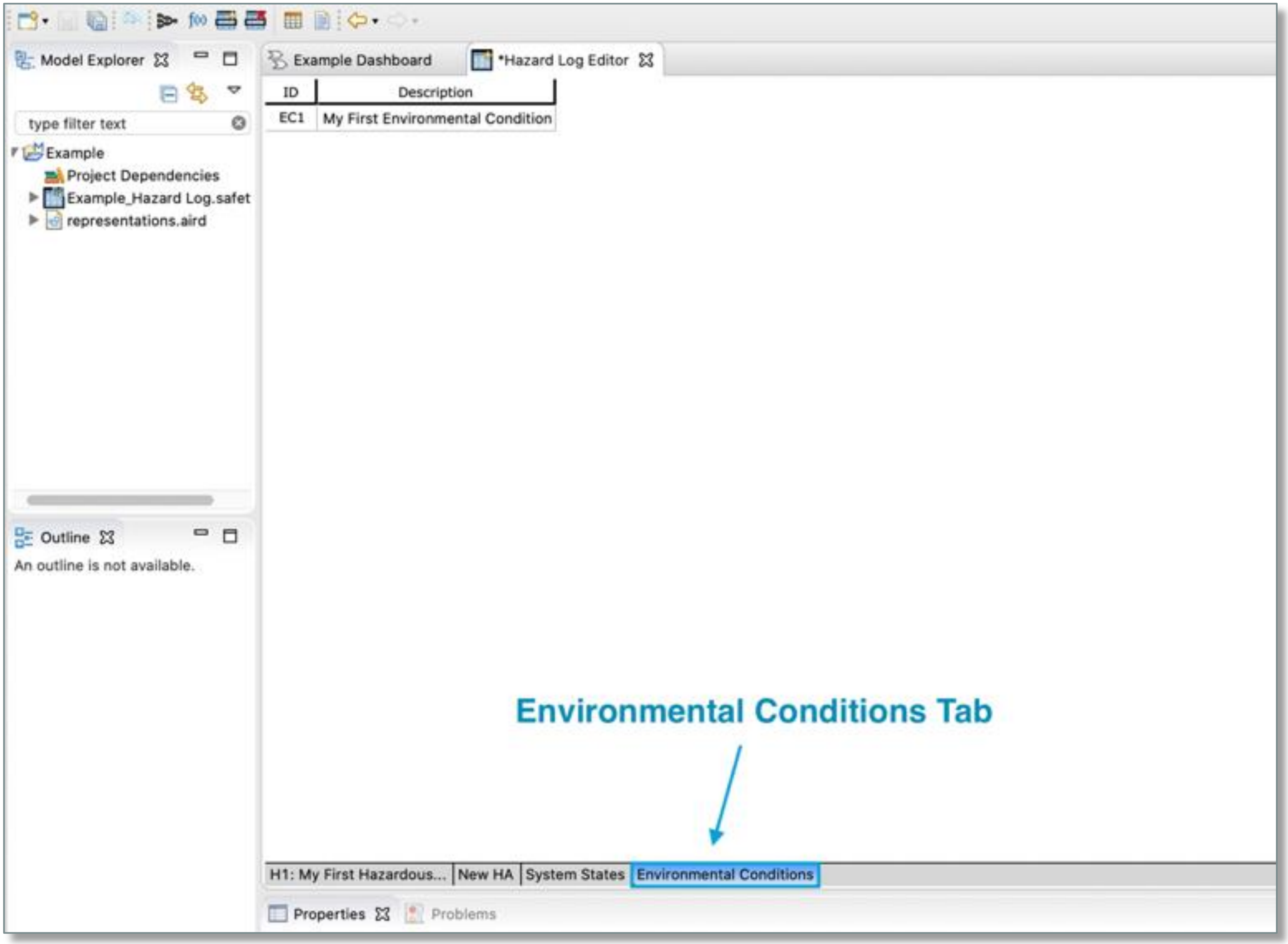
An **Environmental Condition** is the state of the surrounding environment your physical system operates in.

Open Environmental Conditions Table

To open your Environmental Conditions Table

1. Open your **Hazard Table**.
2. In the Hazard Log Editor, select the **Environmental Conditions** tab (*Figure 17*).

The Environmental Conditions Table will open.



ID	Description
EC1	My First Environmental Condition

Environmental Conditions Tab

Figure 17: In the Hazard Log Editor, select the Environmental Conditions tab to open the Environmental Conditions Table.

Create a New Environmental Condition

To create a new Environmental Condition

1. Open your **Environmental Conditions Table**.
2. Click the **Add Element** icon in toolbar.

A new Environmental Condition will generate in the **Environmental Conditions Table**.

Edit an Environmental Condition

To edit an Environmental Condition

1. Open your **Environmental Conditions Table**.
2. Double-click the **table cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click outside the **table cell** you edited to update your Environmental Condition.

The text in your table cell will update to reflect the change.

Delete an Environmental Condition

To delete an Environmental Condition

1. Open your **Environmental Conditions Table**.
2. Select the **Environmental Condition** you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected Environmental Condition will be deleted from the **Environmental Conditions Table**.

Hazard Table Entries

Each **Hazard Table** has the following table entries or sections in the **Full View**:

- *Hazardous Activity*
- *Hazard*
- *Hazard Allocation*
- *Hazard Condition*
- *Hazard Type*
- *Hazard Sources*
- *Hazard Causes*
- *Hazard Mitigations*
- *New?*
- *Mitigation Type*
- *Mitigation Requirements*
- *Mitigation Validations*
- *Mitigation Verifications*
- *Effects*
- *Hazard Initial Likelihood*
- *Hazard Initial Severity*
- *Hazard Initial Risk Level*
- *Hazard Initial Risk Rationale*
- *Hazard Residual Likelihood*
- *Hazard Residual Severity*
- *Hazard Residual Risk Level*
- *Hazard Residual Risk Rationale*
- *Hazard Action*
- *Hazard Status*
- *Hazard Relation*
- *Hazard Notes*

Hazardous Activity

Hazardous Activities are the triggering actions or tasks a system performs that lead to Hazards.

Since Hazardous Activities are performed in various **System States** and **Environmental Conditions**, instances of Hazardous Activities can be associated with different System States and Environmental Conditions.

Each **Hazard Log** can only reference one Hazardous Activity in one System State and one Environmental Condition.

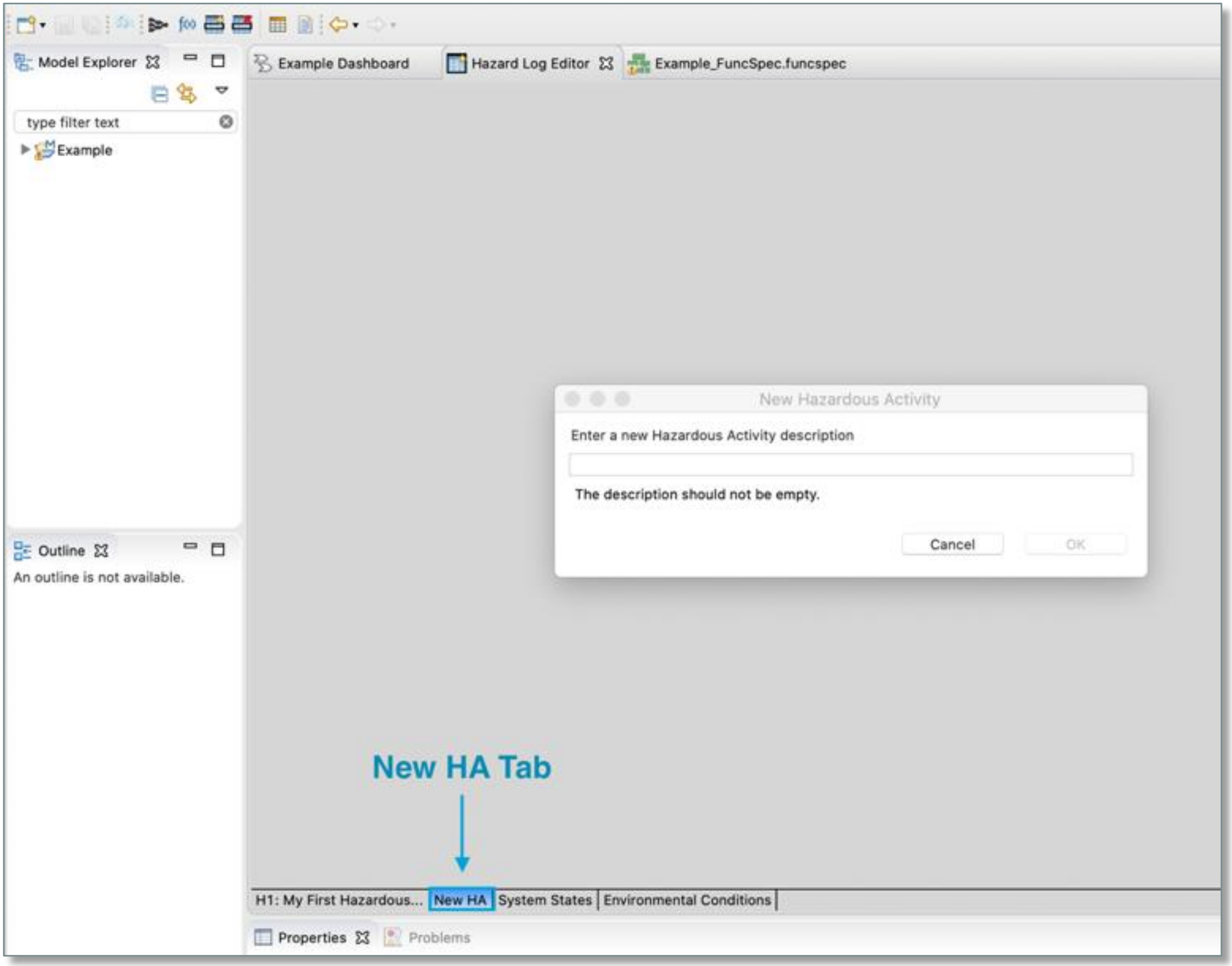
Create a New Hazardous Activity

To create a new Hazardous Activity

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, click the **New HA** tab at the bottom of the Hazard Log Editor located after the other Hazardous Activity tabs (*Figure 18*).
3. Enter your new Hazardous Activity's name.
4. Click **Ok**.

A new **Hazard Log** will generate and open in your Hazard Log Editor.

Alternatively, you can create a new Hazardous Activity from the **Hazard Dashboard**.



An outline is not available.

Enter a new Hazardous Activity description

The description should not be empty.

Figure 18: Select the New HA tab to create a new Hazardous Activity in the Hazard Log Editor.

Edit a Hazardous Activity

To edit a Hazardous Activity from the **Hazard Dashboard**

1. Open the **Hazard Tab**.
2. In the Hazard Log Editor, double-click the **Hazardous Activity cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click **outside the cell** you edited to update your Hazardous Activity.

The text in your table cell will update to reflect the change.

Alternatively, you can edit a Hazardous Activity from the **Hazard Dashboard**.

Delete a Hazardous Activity

To delete a Hazardous Activity from the **Hazard Dashboard**

1. Open the **Hazards Dashboard**.
2. In the **Hazard Tables** window, select the **Hazardous Activity** you want to delete.
3. Click the **Delete button** on the right side of the Hazards Table window.
4. Click **Yes** to confirm your deletion request.

Your selected Hazardous Activity will be deleted.

Hazard

Hazards are events (situations or scenarios) that represent a loss of control and have the potential for harm.

Hazards are potential results from performing a **Hazardous Activity** in relation to a **System State** and **Environmental Condition**.

Each **Hazard Table** can reference as many Hazards as needed.

Create a New Hazard

To create a new Hazard from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, click the **Add Element** icon in toolbar.

A new row and Hazard entry will generate in the **Hazard Log**.

Alternatively, you can create a new Hazard by selecting **Add Hazard Log Entry** from the right-click menu when you have a table cell selected.

Edit a Hazard

To edit a Hazard from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Hazard cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click **outside the cell** you edited to update your Hazard.

The text in your table cell will update to reflect the change.

Delete a Hazard

To delete a Hazard from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click a **cell in the Hazard row** you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected Hazard will be deleted.

Alternatively, you can delete a Hazard by selecting **Delete Hazard Log Entry** from the right-click menu when you have the desired Hazard row selected.

Physical Hazard Analysis (PhHA)

A **Physical Hazard Analysis (PhHA)** is the risk analysis performed to identify the Physical Hazards of Failure Modes and to create mitigation mechanisms to control and manage those Hazards.

NOTES:

- This user guide assumes you know how to perform a PhHA. For more information about how to perform a PhHA, see the *Guide References* section.
- In order to perform a PhHA, you must create and populate **PhysArch** and **FuncSpec** files. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

Add a Physical Hazard

To add a Physical Hazard

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, click the **Manage Physical Hazards** icon in toolbar.
3. Toggle the **drop-down arrows** for the corresponding **Physical Components**.
4. Select the **Failure Mode/s** you wish to add as Hazards.
5. Click **Ok**.

A new Hazard row will generate in the **Hazard Table** with corresponding **Hazard**, **Allocation**, **Condition**, and **Hazard Type** entries.

Functional Hazard Analysis (FHA)

A **Functional Hazard Analysis (FHA)** is the risk analysis performed to identify the Functional Hazards of Deviations and to create mitigation mechanisms to control and manage those Hazards.

NOTES:

- This user guide assumes you know how to perform a FHA. For more information about how to perform a FHA, see the *Guide References* section.
- In order to perform a FHA, you must create and populate **PhysArch** and **FuncSpec** files. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

Add a Functional Hazard

To add a Functional Hazard

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, click the f(x) **Manage Functional Hazards** icon in toolbar.
3. Toggle the ▶ **drop-down arrows** for the corresponding f(x) **Functions**.
4. Select the ⛈ **Deviation/s** you wish to add as Hazards.
5. Click **Ok**.

A new Hazard row will generate in the **Hazard Table** with corresponding **Hazard**, **Allocation**, **Condition**, and **Hazard Type** entries.

Hazard Allocation

A **Hazard Allocation** is the Component or Function where the Hazard is located.

In order to allocate a Hazard, you must create and populate **PhysArch** and **FuncSpec** files. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

When you perform a **PhHA**, **FHA**, or **BHA**, allocation entries will auto-populate based on your selections. For more information about PhHA, FHA, and BHA, see the *PhHA*, *FHA*, and *BHA* sections.

Each **Hazard** can reference one Allocation.

When you allocate a Hazard, the Hazard's corresponding **CES** and **BTDs** will automatically update.

Create a New Hazard Allocation

To create a new Hazard Allocation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Allocation cell** you wish to populate.
3. In the drop-down menu, toggle the **checkbox** for your corresponding **Component** or **Function**. *Unless the Hazard is known to be allocated to a Function, we recommend selecting a Component.*

The text in your table cell will update to reflect the change.

Edit a Hazard Allocation

To edit a Hazard Allocation, you must edit the corresponding **Component** or **Function** entry in your **PhysArch** or **FuncSpec** file. For more information on **PhysArch DSL** or **FuncSpec DSL**, visit the *PhysArch DSL* or *FuncSpec DSL* section.

Delete a Hazard Allocation

To delete a Hazard Allocation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Allocation cell** you wish to delete.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Component** or **Function** you wish to delete.

The text in your table cell will update to reflect the change.

Hazard Condition

Hazard Conditions are the Failure Modes, Deviations, or other situations that characterize the Hazard.

To associate a Hazard Condition, you must first populate the corresponding **Hazard Allocation** cell.

Each **Hazard** can reference one Condition.

When you associate a Hazard Condition, the Hazard’s corresponding **CES** and **BTDs** will automatically update.

Create a New Hazard Condition

To create a new Hazard Condition from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Condition cell** you wish to populate.
3. In the drop-down menu, toggle the **checkbox** for your corresponding **Failure Mode** or **Deviation**.

The text in your table cell will update to reflect the change.

Edit a Hazard Condition

To edit a Hazard Condition, you must edit the corresponding **Failure Mode** or **Deviation** entry in your **PhysArch** or **FuncSpec** file. For more information on **PhysArch DSL** or **FuncSpec DSL**, visit the *PhysArch DSL* or *FuncSpec DSL* section.

Delete a Hazard Condition

To delete a Hazard Condition from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Condition cell** you wish to delete.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Failure Mode** or **Deviation** you wish to delete.

The text in your table cell will update to reflect the change.

Hazard Type

A **Hazard Type** characterize states that increase risk since they can lead to undesired outcomes.

AdvoCATE supports the following Hazard Types:

- **Safety**: Hazard that can lead to loss or harm in the environment in the context of (worst-case) environmental conditions.
- **Functional**: Hazard that can lead to an undesired deviation in the expected system service.
- **Performance**: Hazard that can lead to an undesired deviation in the expected quality of a system service.
- **Security**: Hazard that can compromise availability, integrity, and confidentiality, in the context of an environmental condition and system usage.
- **Reliability**: Hazard that can diminish or eliminate the continuity of correct system service.
- **Availability**: Hazard that can diminish or eliminate the readiness for correct system service.
- **Maintainability**: Hazard that can diminish the capacity for modification and repair.

Hazard Sources

Hazard Sources represent an analysis, guideline, or regulation (which can be tied to a specific activity, item, location, or situation) that gives rise to a Hazard.

Internal Hazard Source correspond to activities or entities within AdvoCATE Artifacts and will auto-generate, such as when performing a BHA.

External Hazard Sources are simply strings describing the Source and can be entered manually into the table cell.

Create a New External Hazard Source

To create a new Hazard Source from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Source cell** you wish to populate.
3. Enter a new value in the **Create a New Source…** textbox field.
4. Click **outside the cell** you populated to create your Source.

The text in your table cell will update to reflect the change.

Add an Existing Barrier as an Internal Hazard Source

To add an Existing Barrier as an Internal Source from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Source cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Barrier** you wish to make a **Hazard Source**.
4. Click **outside the cell** you edited to add the Source to your Hazard.

The text in your table cell will update to reflect the change.

Edit a Hazard Source

To edit a Hazard Source from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Source cell** you wish to edit.
3. Enter your edits in the **Source cell** textbox field.
4. Click **outside the cell** you edited to update the Source.

The text in your table cell will update to reflect the change.

Remove a Source from a Hazard

To remove a Source from a Hazard in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Source cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **event** you wish to remove as a **Hazard Source**.
4. Click **outside the cell** you edited to remove the Source from the Hazard.

The text in your table cell will update to reflect the change.

Hazard Causes

Causes (or Threats) are the system or environmental events (depicted in AdvoCATE as an Event Instance) that create or lead to Hazards in a causal relationship.

Each **Hazard** can reference multiple Causes.

When you associate a Cause/s to a Hazard, the Hazard's corresponding **CES** and **BTDs** will automatically update.

Create a New Cause from the Hazard Log Editor

To create a new Hazard Cause from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Cause cell** you wish to populate.
3. Enter a new value in the **Create a New Cause…** textbox field.
4. Click **outside the cell** you populated to create your Cause.

The text in your table cell will update to reflect the change.

Add an Existing Event as a Cause from the Hazard Log Editor

To add an Existing Event as a Cause from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Cause cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **event** you wish to make a **Hazard Cause**.
4. Click **outside the cell** you edited to add the Cause to your Hazard.

The text in your table cell will update to reflect the change.

Edit a Cause from the Hazard Log Editor

To edit a Cause from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Cause cell** you wish to edit.
3. Enter your edits in the **Cause cell** textbox field.
4. Click **outside the cell** you edited to update the Cause.

The text in your table cell will update to reflect the change.

Show a Cause as a Hazard

To show a Cause as a Hazard from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Cause cell** you wish to show as a Hazard.
3. In the right-click menu, select **Show as Hazard**.

A new Hazard row will generate for the selected Cause.

Edit Initial Likelihood Value for a Cause from the Hazard Log Editor

To edit the Initial Likelihood Value for a Cause from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Cause cell** you wish to edit the Initial Likelihood Value for.
3. In the right-click menu, select **Edit Initial Likelihood**.
4. In the pop-up window, enter a value between 0 to 1 that is related to the Initial Likelihood the Event will occur.
5. Click Ok.

The Initial Likelihood Value for the Cause will update to reflect the change.

NOTE: When you update the Initial Likelihood Value, the Initial Likelihood classification will update as well.

Remove a Cause from a Hazard

To remove a Cause from a Hazard in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Cause cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **event** you wish to remove as a **Hazard Cause**.
4. Click **outside the cell** you edited to remove the Cause from the Hazard.

The text in your table cell will update to reflect the change.

Hazard Mitigations

Mitigations are mechanisms for risk reduction and hazard prevention for each **Cause** you associate to a **Hazard**.

Each **Cause** can reference multiple Mitigations.

When you associate a Mitigation to a Hazard, the Hazard's corresponding **CES** and **BTDs** will automatically update.

Create a New Hazard Mitigation

To create a new Mitigation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation cell** you wish to populate.
3. Enter a new value in the **Create a New Mitigation…** textbox field.
4. Click **outside the cell** you populated to create your Mitigation.

The text in your table cell will update to reflect the change.

NOTE: You must enter a Cause into the Hazard Log before entering a corresponding Mitigation.

Add an Existing Hazard Mitigation

To add an Existing Mitigation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Mitigation** you wish to add.
4. Click **outside the cell** you edited to add the Mitigation to your Hazard.

The text in your table cell will update to reflect the change.

Edit a Hazard Mitigation

To edit a Mitigation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation cell** you wish to edit.
3. Enter your edits in the **Mitigation cell** textbox field.
4. Click **outside the cell** you edited to update the Mitigation.

The text in your table cell will update to reflect the change.

Remove a Hazard Mitigation

To remove a Mitigation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Mitigation** you wish to remove.
4. Click **outside the cell** you edited to remove the Mitigation from the Hazard.

The text in your table cell will update to reflect the change.

Barrier Hazard Analysis (BHA)

A **Barrier Hazard Analysis (BHA)** is the risk analysis performed to identify the Functional or Physical Hazards of Barriers and creates mitigation mechanisms to control and manage those Escalations.

In order to perform a BHA, the Barrier (or Hazard Mitigation) must have an allocation/s.

NOTES:

- This user guide assumes you know how to perform a BHA. For more information about how to perform a BHA, see the *Guide References* section.
- In order to perform a BHA, you must create and populate **PhysArch** and **FuncSpec** files. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

Perform a BHA

To perform a BHA

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click a **Mitigation cell**.
3. In the right-click menu, select **Perform Barrier Hazard Analysis**.
4. In the pop-up menu, select where the hazard analysis type can be selected: **Via Function Decomposition** or **Via Physical Decomposition**.
5. Click **Next**. *Depending on the selected hazard analysis type in Step 4, the next window presents the available Functions and Deviations or Components and Failure Modes.*
6. Select the **checkboxes** next to the **decomposition elements** that you wish to add as Hazards.
7. Click **Finish**.

A new Hazard row/s will generate in the **Hazard Table** with corresponding entries based on your selections. The Causes and Mitigations created, added, or removed for these Hazards will additionally appear as Escalations in the CESs and BTDS where instances of the original Mitigation's Barrier appear.

New?

The **New?** entry allows you to classify a Mitigation as an **Existing Mitigation** or a **New Mitigation**.

Existing Mitigations are previously established Mitigations. When safety analysts analyze a Mitigation to determine its **Initial Risk Level** and those Mitigations are deemed unacceptable, you can propose **New Mitigations** to lower the risk.

By modeling both systems in one AdvoCATE Project, you can compute the **Initial** and **Residual Risk Levels** for a Mitigation. For more information about these Risk Levels, see the *Hazard Initial Risk Level* and *Hazard Residual Risk Level* sections.

Mitigation Type

Mitigation Types are the classifications corresponding to the order of precedence of a system's safety design.

AdvoCATE supports the following Mitigation Types:

- **Design Modification**: the elimination of hazards or risk reduction through system redesign.
- **Safety Feature**: the elimination of hazards or risk reduction through the incorporation of safety features.
- **Safety Device**: the elimination of hazards or risk reduction through the incorporation of safety devices.
- **Warning Device**: the elimination of hazards or risk reduction through the incorporation of warning devices.
- **Procedures and Training**: the elimination of hazards or risk reduction through the incorporation of procedures and training.

Mitigation Requirements

Mitigation Requirements are high-level requirement statements that describe how a specific **Mitigation** will reduce the risk of **Cause** events.

Each Mitigation can reference multiple Mitigation Requirements.

When you associate Mitigation Requirements in a Hazard Log, the **Requirements Log** and corresponding **BTDs** will automatically update.

Create a New Mitigation Requirement

To create a new Mitigation Requirement from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Requirement cell** you wish to populate.
3. Enter a new value in the **Create a New Mitigation Requirement …** textbox field.
4. Click **outside the cell** you populated to create your Mitigation Requirement.

The text in your table cell will update to reflect the change.

NOTES:

- You must enter a Mitigation into the Hazard Log before entering a corresponding Mitigation Requirement.
- If you have not already created a Requirement Log for your AdvoCATE Project, you will be prompted to create one.

Add an Existing Mitigation Requirement

To add an Existing Mitigation Requirement from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Requirement cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Mitigation Requirement** you wish to add.
4. Click **outside the cell** you edited to add the Mitigation Requirement to your Mitigation.

The text in your table cell will update to reflect the change.

Edit a Mitigation Requirement

To edit a Mitigation Requirement from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Requirement cell** you wish to edit.
3. Enter your edits in the **Mitigation Requirement cell** textbox field.
4. Click **outside the cell** you edited to update the Mitigation Requirement.

The text in your table cell will update to reflect the change.

Remove a Mitigation Requirement

To remove a Mitigation Requirement from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Requirement cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Mitigation Requirement** you wish to remove.
4. Click **outside the cell** you edited to remove the Mitigation Requirement from the Mitigation.

The text in your table cell will update to reflect the change.

Mitigation Validations

Mitigation Validations are rationale statements specifying why the corresponding **Mitigation Requirement** is acceptable.

Each **Mitigation Validation** can reference one Mitigation Requirement.

Create a New Mitigation Validation

To create a new Mitigation Validation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Validation cell** you wish to populate.
3. Enter a new value in the **Mitigation Validation cell** textbox field.
4. Click **outside the cell** you populated to create your Mitigation Validation.

The text in your table cell will update to reflect the change.

NOTE: You must enter a Mitigation Requirement into the Hazard Log before entering a corresponding Mitigation Validation.

Edit a Mitigation Validation

To edit a Mitigation Validation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Validation cell** you wish to edit.
3. Enter your edits in the **Mitigation Requirement cell** textbox field.
4. Click **outside the cell** you edited to update the Mitigation Validation.

The text in your table cell will update to reflect the change.

Delete a Mitigation Validation

To delete a Mitigation Validation from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Validation cell** you wish to edit.
3. Delete the text in the **Mitigation Requirement cell** textbox field using Backspace (PC) or Delete (Mac) on your keyboard.
4. Click **outside the cell** you edited to delete the Mitigation Validation from the Mitigation Requirement.

The text in your table cell will update to reflect the change.

Mitigation Verifications

Mitigation Verifications are the means of checking that the corresponding **Mitigation Requirement** meets its intent.

Each **Mitigation Requirement** can reference multiple Mitigation Verifications.

When you associate Mitigation Verifications in a Hazard Log, the **Requirements Log** will automatically update.

Create a New Mitigation Verification

To create a new Mitigation Verification from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Verification cell** you wish to populate.
3. Enter a new value in the **Create a New Verification Method…** textbox field.
4. Click **outside the cell** you populated to create your Mitigation Verification.

The text in your table cell will update to reflect the change.

NOTE: You must enter a Mitigation Requirement into the Hazard Log before entering a corresponding Mitigation Verification.

Add an Existing Mitigation Verification

To add an Existing Mitigation Verification from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Verification cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Mitigation Verification** you wish to add.
4. Click **outside the cell** you edited to add the Mitigation Verification to your Mitigation Requirement.

The text in your table cell will update to reflect the change.

Edit a Mitigation Verification

To edit a Mitigation Verification from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Verification cell** you wish to edit.
3. Enter your edits in the **Mitigation Verification cell** textbox field.
4. Click **outside the cell** you edited to update the Mitigation Verification.

The text in your table cell will update to reflect the change.

Remove a Mitigation Verification

To remove a Mitigation Verification from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Mitigation Verification cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Mitigation Verification** you wish to remove.
4. Click **outside the cell** you edited to remove the Mitigation Verification from the Mitigation Requirement.

The text in your table cell will update to reflect the change.

Effects

Effects (or Consequences) are the loss events to be avoided at the boundary of the system and its environment.

Each **Hazard** can reference multiple Effects, although there is usually one worst-case Effect.

When you associate Effects to a Hazard, the Hazard's corresponding **CES** and **BTDs** will automatically update.

Effects have corresponding Risk Analysis and Assessment entries, much like Hazards, and are located within the Effect's sub-columns:

- *Effect Initial Likelihood*
- *Effect Initial Severity*
- *Effect Initial Risk Level*
- *Effect Initial Rationale*
- *Effect Residual Likelihood*
- *Effect Residual Severity*
- *Effect Residual Risk Level*
- *Effect Residual Rationale*

Create a New Effect

To create a new Effect from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Effect Description cell** you wish to populate.
3. Enter a new value in the **Create a New Effect…** textbox field.
4. Click **outside the cell** you populated to create your Effect.

The text in your table cell will update to reflect the change.

Add an Existing Event as an Effect

To add an Existing Effect from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Effect Description cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Event** you wish to add as an Effect.
4. Click **outside the cell** you edited to add the Effect to your Hazard.

The text in your table cell will update to reflect the change.

Edit an Effect

To edit an Effect from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Effect Description cell** you wish to edit.
3. Enter your edits in the **Effect Description cell** textbox field.
4. Click **outside the cell** you edited to update the Effect.

The text in your table cell will update to reflect the change.

Show an Effect as a Hazard

To show an Effect as a Hazard from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Effect Description cell** you wish to show as a Hazard.
3. In the right-click menu, select **Show as Hazard**.

A new Hazard row will generate for the selected Effect.

Remove an Effect

To remove an Effect from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Effect cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Effect** you wish to remove.
4. Click **outside the cell** you edited to remove the Effect from your Hazard.

The text in your table cell will update to reflect the change.

Effect Initial Likelihood

The **Initial Likelihood** of an Effect is the initial level of probability that the Effect will occur. The Initial Likelihood provides a base metric of the Effect's probability given the current levels of integrity for your system's various **Barriers** and **Controls**.

For more information on Barriers and Controls, see the *CES Mitigation Node Types* and *BTDs Mitigation Node Types* sections.

AdvoCATE has five Likelihood classification levels based on the FAA's SMS manual:

1. **Frequent**: expected to occur routinely.
2. **Probable**: expected to occur often.
3. **Remote**: expected to occur infrequently.
4. **Extremely Remote**: expected to occur rarely.
5. **Extremely Improbable**: unlikely to occur, but not impossible.

NOTE: You can only edit the Initial Likelihood of Global Threats in the Hazard Log, that is, those Threat events that do not have any preceding events.

Effect Initial Severity

The **Initial Severity** of an Effect is the initial negative impact level the Effect can cause to its surrounding environment and other systems. The Initial Severity provides a base metric of the Effect's negative impact.

AdvoCATE has five Severity classification levels:

1. **Minimal**: Negligible safety effect.
2. **Minor**: Physical discomfort to other systems or environment. Slight damage to system.
3. **Major**: Physical distress or injuries other systems or environment. Substantial damage to system.
4. **Hazardous**: Multiple serious injuries; fatal injury to a relatively small number other systems; or a system loss without fatalities.
5. **Catastrophic**: Multiple fatalities, usually with the loss of the system.

NOTE: You can only edit the Initial Severity of Global Threats in the Hazard Log, that is, those Threat events that do not have any preceding events.

Effect Initial Risk Level

The **Initial Risk Level** of an Effect is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Effect. The Initial Risk Level provides a base metric of the extent of possible loss for a specific Effect.

AdvoCATE has three Risk classification levels:

1. **Low**: Severity and Likelihood map to the green cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without restriction or limitation; Effects are not required to be actively managed.
2. **Medium**: Severity and Likelihood map to the yellow cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without additional mitigation; however, tracking and monitoring are required.
3. **High**: Severity and Likelihood map to the red cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk requires mitigation, tracking, and monitoring.

Initial Risk Levels are automatically generated based on the Effect's **Initial Likelihood** and **Initial Severity** entries.

Effect Initial Risk Rationale

The **Initial Risk Rationale** of an Effect is the reasoning for the Initial Likelihood, Severity, and assigned Risk Level for the Effect.

Edit Initial Risk Rationale for an Effect in the Hazard Log Editor

To edit an Initial Risk Rationale for an Effect in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the Effect's **Initial Risk Rationale cell** that you wish to edit.
3. Enter your edit in the Effect's **Initial Risk Rationale cell** textbox field.
4. Click **outside the cell** you populated to update the Effect's **Initial Rationale**.

The text in your table cell will update to reflect the change.

Effect Residual Likelihood

The **Residual Likelihood** of an Effect is the level of probability that the Effect will occur after new mitigation mechanisms have been established. The Residual Likelihood provides a new metric of the Effect's probability given the new levels of integrity for your system's various **Barriers** and **Controls**.

Residual Likelihood entries are automatically derived using your Initial Risk Assessment entries and new mitigation mechanisms.

For more information on Barriers and Controls, see the *CES Mitigation Node Types* and *BTDs Mitigation Node Types* sections.

Effect Residual Severity

The **Residual Severity** of an Effect is the new negative impact level the Effect can cause to its surrounding environment and other systems once new mitigation mechanisms have been established. The Residual Severity provides a new metric of the Effect's negative impact.

Residual Severity entries are automatically derived using your Initial Risk Assessment entries and new mitigation mechanisms.

Effect Residual Risk Level

The **Residual Risk Level** of an Effect is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Effect. The Residual Risk Level provides a new metric of the extent of possible loss for a specific Effect once new mitigation mechanisms have been established.

Residual Risk Level entries are automatically derived using your Initial Risk Assessment entries and new mitigation mechanisms.

Effect Residual Risk Rationale

The **Residual Risk Rationale** of an Effect is the reasoning for the Residual Likelihood, Severity, and assigned Risk Level for the Effect.

Edit Residual Risk Rationale for an Effect in the Hazard Log Editor

To edit a Residual Risk Rationale for an Effect in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the Effect's **Residual Risk Rationale cell** that you wish to edit.
3. Enter your edit in the Effect's **Residual Risk Rationale cell** textbox field.
4. Click **outside the cell** you populated to update the Effect's **Residual Rationale**.

The text in your table cell will update to reflect the change.

Hazard Initial Likelihood

The **Initial Likelihood** of a Hazard is the initial level of probability that the Hazard will occur. The Initial Likelihood provides a base metric of the Hazard's probability given the current levels of integrity for your system's various **Barriers** and **Controls**.

For more information on Barriers and Controls, see the *CES Mitigation Node Types* and *BTDs Mitigation Node Types* sections.

AdvoCATE has five Likelihood classification levels based on the FAA's SMS manual:

1. **Frequent**: expected to occur routinely.
2. **Probable**: expected to occur often.
3. **Remote**: expected to occur infrequently.
4. **Extremely Remote**: expected to occur rarely.
5. **Extremely Improbable**: unlikely to occur, but not impossible.

Edit Initial Likelihood for a Hazard in the Hazard Log Editor

To edit an Initial Likelihood for a Hazard in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the Hazard's **Initial Likelihood cell** that you wish to edit.
3. From the drop-down menu, select the **Likelihood level** you wish associate to your Hazard.
4. Click **outside the cell** you populated to update the Hazard's **Initial Likelihood**.

The text in your table cell will update to reflect the change.

NOTE: You can only edit the Initial Likelihood of Global Threats in the Hazard Log, that is, those Threat events that do not have any preceding events.

Hazard Initial Severity

The **Initial Severity** of a Hazard is the initial negative impact level the Hazard can cause to its surrounding environment and other systems. The Initial Severity provides a base metric of the Hazard's negative impact.

AdvoCATE has five Severity classification levels:

1. **Minimal**: Negligible safety effect
2. **Minor**: Physical discomfort to other systems or environment. Slight damage to system.
3. **Major**: Physical distress or injuries other systems or environment. Substantial damage to system.
4. **Hazardous**: Multiple serious injuries; fatal injury to a relatively small number other systems; or a system loss without fatalities.
5. **Catastrophic**: Multiple fatalities, usually with the loss of the system.

Edit Initial Severity for a Hazard in the Hazard Log Editor

To edit an Initial Severity for a Hazard in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the Hazard's **Initial Severity cell** that you wish to edit.
3. From the drop-down menu, select the **Severity level** you wish associate to your Hazard.
4. Click **outside the cell** you populated to update the Hazard's **Initial Severity**.

The text in your table cell will update to reflect the change.

NOTE: You can only edit the Initial Severity of Global Threats in the Hazard Log, that is, those Threat events that do not have any preceding events.

Hazard Initial Risk Level

The **Initial Risk Level** of a Hazard is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Hazard. The Initial Risk Level provides a base metric of the extent of possible loss for a specific Hazard.

AdvoCATE has three Risk classification levels:

1. **Low**: Severity and Likelihood map to the green cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without restriction or limitation; Hazards are not required to be actively managed.
2. **Medium**: Severity and Likelihood map to the yellow cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without additional mitigation; however, tracking and monitoring are required.
3. **High**: Severity and Likelihood map to the red cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk requires mitigation, tracking, and monitoring.

Initial Risk Levels are automatically generated based on the Hazard's **Initial Likelihood** and **Initial Severity** entries.

Hazard Initial Risk Rationale

The **Initial Risk Rationale** of a Hazard is the reasoning for the Initial Likelihood, Severity, and assigned Risk Level for the Hazard.

Edit Initial Risk Rationale for a Hazard in the Hazard Log Editor

To edit an Initial Risk Rationale for a Hazard in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the Hazard's **Initial Risk Rationale cell** that you wish to edit.
3. Enter your edit in the Hazard's **Initial Risk Rationale cell** textbox field.
4. Click **outside the cell** you populated to update the Hazard's **Initial Rationale**.

The text in your table cell will update to reflect the change.

Hazard Residual Likelihood

The **Residual Likelihood** of a Hazard is the level of probability that the Hazard will occur after new mitigation mechanisms have been established. The Residual Likelihood provides a new metric of the Hazard's probability given the new levels of integrity for your system's various **Barriers** and **Controls**.

Residual Likelihood entries are automatically derived using your Initial Risk Assessment entries and new mitigation mechanisms.

For more information on Barriers and Controls, see the *CES Mitigation Node Types* and *BTDs Mitigation Node Types* sections.

Hazard Residual Severity

The **Residual Severity** of a Hazard is the new negative impact level the Hazard can cause to its surrounding environment and other systems once new mitigation mechanisms have been established. The Residual Severity provides a new metric of the Hazard's negative impact.

Residual Severity entries are automatically derived using your Initial Risk Assessment entries and new mitigation mechanisms.

Hazard Residual Risk Level

The **Residual Risk Level** of a Hazard is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Hazard. The Residual Risk Level provides a new metric of the extent of possible loss for a specific Hazard once new mitigation mechanisms have been established.

Residual Risk Level entries are automatically derived using your Initial Risk Assessment entries and new mitigation mechanisms.

Hazard Residual Risk Rationale

The **Residual Risk Rationale** of a Hazard is the reasoning for the Residual Likelihood, Severity, and assigned Risk Level for the Hazard.

Edit Residual Risk Rationale for a Hazard in the Hazard Log Editor

To edit a Residual Risk Rationale for a Hazard in the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the Hazard's **Residual Risk Rationale cell** that you wish to edit.
3. Enter your edit in the Hazard's **Residual Risk Rationale cell** textbox field.
4. Click **outside the cell** you populated to update the Hazard's **Residual Rationale**.

The text in your table cell will update to reflect the change.

Hazard Action

Action entries are decisions for the Hazard's management.

AdvoCATE's Hazard Log allows the following determinations of a Hazard to be recorded as Actions:

- **Track**: determined that the Hazard needs to be monitored to obtain additional data.
- **Mitigate**: determined that additional Mitigations are needed to lower the risk level of the Hazard.
- **Accept**: determined that the Hazard has an acceptable Risk Level.

By default, new **Hazards** are set to **Track**.

Hazard Status

Status entries are the resolutions for the Hazard's management.

AdvoCATE supports the following Statuses:

- **Open**: the Hazard is a threat.
- **Accepted**: the Hazard is within acceptable parameters.
- **Closed**: the Hazard is no longer a threat.

By default, new **Hazards** are set to **Open**.

Hazard Relation

Relation entries define the parent/child relationships of **Hazards** and **Subhazards.**

Relation entries will auto-populated based on **Subhazard** specifications.

Hazard Notes

Hazard Note entries are any comments you have about the corresponding Hazard for yourself, safety analysts, or review committees.

Add a Hazard Note

To add a Hazard Note

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Note cell** you wish to populate.
3. Enter a new value in the **Note cell** textbox field.
4. Click **outside the cell** you populated to add your Note.

The text in your table cell will update to reflect the change.

Edit a Hazard Note

To edit a Hazard Note

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Note cell** you wish to edit.
3. Enter your edits in the **Note cell** textbox field.
4. Click **outside the cell** you edited to update the Note.

The text in your table cell will update to reflect the change.

Delete a Hazard Note

To delete a Hazard Note

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click the **Note cell** you wish to edit.
3. Delete the text in the **Note cell** textbox field using Backspace (PC) or Delete (Mac) on your keyboard.
4. Click **outside the cell** you edited to delete the Note.

The text in your table cell will update to reflect the change.

Subhazards

Subhazards are child events of **Hazards** that have **Sub-Component** or **Sub-Function** allocations.

For more information about Sub-Components or Sub-Functions, see the *Sub-Component DSL* or *Sub-Function DSL* sections. For more information about Hazard Allocations, see the *Hazard Allocations* section.

Create a New Subhazard

To create a new Subhazard

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Hazard cell** you wish to make a Subhazard for.
3. From the menu, select **Create New Subhazard**.
4. Select a **Sub-Component** or **Sub-Function allocation** from the menu.
5. Click **Ok**.
6. Select the **Cause** effect/s associated to the Subhazard (if it has any). *These Cause effects will become* ***Effects*** *for the new Subhazard entry.*
7. Click **Ok**.

A new Subhazard row will generate in the **Hazard Log**.

NOTE: When you create a Subhazard, its parent Hazard must have a pre-existing Hazard Allocation with a Sub-Component or Sub-Function in your PhysArch or FuncSpec file.

Designate a Subhazard

To designate a pre-existing Hazard as a Subhazard

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Hazard cell** you wish to make a Subhazard.
3. From the menu, select **Designate as Subhazard of another Hazard**.
4. Select a **parent Hazard** for your new Subhazard. *Only Hazards with allocations will be available to choose as the parent in this dialog window.*
5. Click **Ok**.
6. If the new Subhazard has no pre-existing allocation, select a **Sub-Component** or **Sub-Function allocation** from the menu. *If the new Subhazard has a pre-existing allocation, skip to Step 8.*
7. Click **Ok**.
8. Select the **Cause** effect/s associated to the new Subhazard (if it has any). *These Cause effects will become **Effects** for the new Subhazard entry.*
9. Click **Ok**.

The Subhazard entry will generate in the Hazard Log.

Remove Subhazard Status

To revert a Subhazard to a Hazard

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Subhazard cell** you wish to make a Hazard.
3. From the menu, select **Remove Subhazard status**.

The Subhazard will become a Hazard entry, and its corresponding Relation entry will update to reflect the change.

Delete a Subhazard

To delete a Subhazard from the **Hazard Log Editor**

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, double-click a **cell in the Subhazard row** you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected Subhazard will be deleted.

Alternatively, you can delete a Subhazard by selecting **Delete Hazard Log Entry** from the right-click menu when you have the desired Subhazard row selected.

Generate Argument Structure from Hazard Log

To generate an **Argument Structure** from the Hazard Log

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Hazardous Activity cell** you wish to generate an **Argument Structure** from.
3. From the menu, select **Generate an argument from this hazard log**.
4. Select the **AdvoCATE Project folder** where you wish to save your Argument Structure. *We recommend selecting the same AdvoCATE Project folder where your Hazard Log is located.*
5. Enter the name you want to give your **Argument Structure** file.

   NOTE: Argument Structure files must end in the file extension **.argument**.

6. Click **Finish**.
7. If there are multiple roots options for new Argument Structure, select a **top Global Effect** as the root to generate the Argument Structure under.
8. Click **Ok**.

A new **Argument Structure** will generate and open. For more information about Argument Structures, see the *Argument Structures* section.

NOTE: The pattern that is instantiated with the Argument Structure—representing the assurance semantics of the Hazard Log—is encoded within AdvoCATE.

Generate Data Tree from Hazard Log

To generate a Data Tree from the Hazard Log

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click the **Hazardous Activity cell** you wish to generate a **Data Tree** from.
3. From the menu, select **Generate pattern data from this hazard log**.
4. Select the **AdvoCATE Project folder** where you wish to save your Data Tree. *We recommend selecting the same AdvoCATE Project folder where your Hazard Log is located.*
5. Enter the name you want to give your **Data Tree**.

   NOTE: Data Trees must end in the file extension **.patterndata**.

6. Click **Finish**.

A new Data Tree will generate and open in the **Data Tree Diagram Editor**.

NOTE: Auto-generated Data Trees are not currently laid out automatically. However, Data Tree layout does not affect the layout of the instance that it will be used to create.

For more information about Data Trees, see the *Data Trees* section.

Hazard Views

Hazard Views are the multiple view configurations of the Hazard Log. Hazard Views relate to certain steps in assurance case development.

By default, AdvoCATE provides the following Hazard Views:

- **Full View**: makes all columns in the Hazard Log visible and provides the optimal column view for a PHA.
- **Hazard Identification View**: shows only those columns in the Hazard Log that are most relevant to recording identified Hazards.
- **Risk Analysis View**: shows the columns required for establishing the Initial Risk Level posed by the Effects of the identified Hazards in the context of pre-existing Mitigations.
- **Risk Assessment View**: shows the columns needed to specify new Mitigations for those Hazards whose Initial Risk Level is unacceptable, and to determine the Residual Risk Level after the introduction of those new Mitigations.
- **Mitigation Requirements View**: shows those columns needed to specify the Mitigation Requirements corresponding to the new Mitigations defined in the Risk Assessment View.

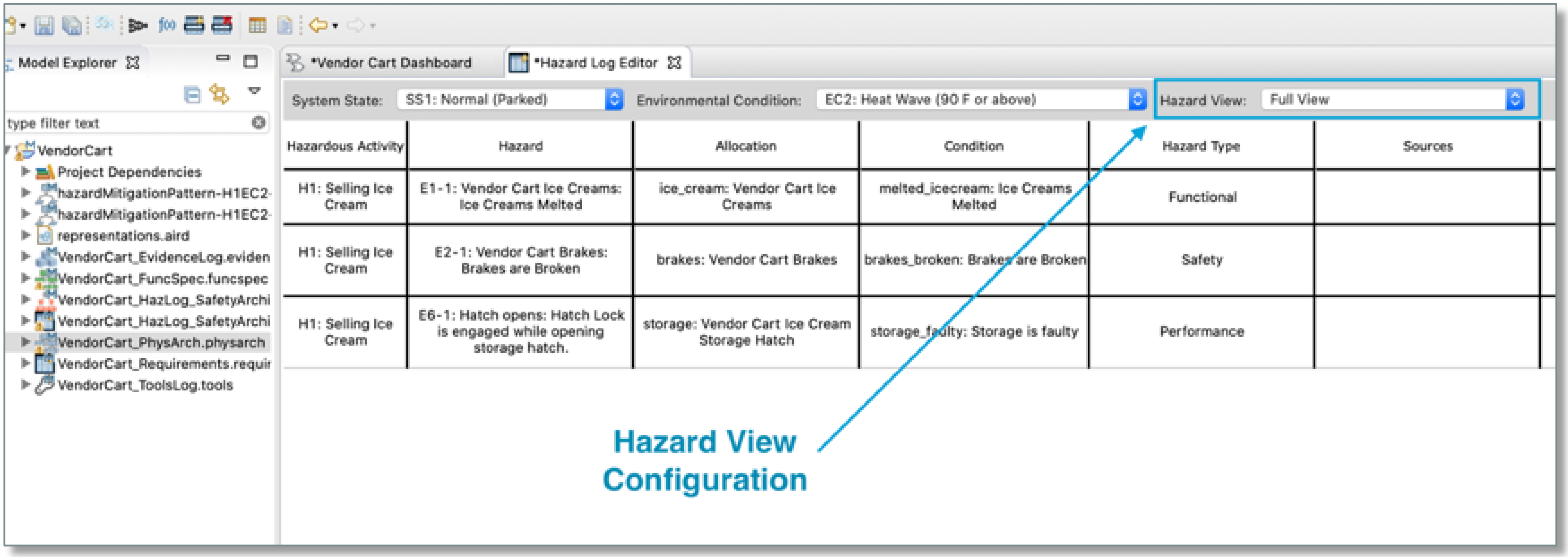
By default, the Hazard Log is set in the **Full View**.

Change Hazard View

To change the Hazard View

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, click the **blue arrows** next to the Hazard View configuration (*Figure 19*).
3. From the drop-down menu, select the **Hazard View** option you wish to view.

The Hazard Log will reconfigure to that view’s settings.



Model Explorer
type filter text
VendorCart
Project Dependencies
hazardMitigationPattern-H1EC2-
hazardMitigationPattern-H1EC2-
representations.aird
VendorCart_EvidenceLog.eviden
VendorCart_FuncSpec.funcspec
VendorCart_HazLog_SafetyArchi
VendorCart_HazLog_SafetyArchi
VendorCart_PhysArch.physarch
VendorCart_Requirements.requir
VendorCart_ToolsLog.tools
*Vendor Cart Dashboard
*Hazard Log Editor
Hazard View Configuration

System State: SS1: Normal (Parked)
Environmental Condition: EC2: Heat Wave (90 F or above)
Hazard View: Full View

Hazardous Activity	Hazard	Allocation	Condition	Hazard Type	Sources
H1: Selling Ice Cream	E1-1: Vendor Cart Ice Creams: Ice Creams Melted	ice_cream: Vendor Cart Ice Creams	melted_icecream: Ice Creams Melted	Functional	
H1: Selling Ice Cream	E2-1: Vendor Cart Brakes: Brakes are Broken	brakes: Vendor Cart Brakes	brakes_broken: Brakes are Broken	Safety	
H1: Selling Ice Cream	E6-1: Hatch opens: Hatch Lock is engaged while opening storage hatch.	storage: Vendor Cart Ice Cream Storage Hatch	storage_faulty: Storage is faulty	Performance	

Figure 19: To change the Hazard View configuration, click the blue arrows and select your desired view option.

Save a Custom Hazard View

To save a custom Hazard View

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, right-click a **column header cell**.
3. From the drop-down menu, select **Manage view configurations**.
4. Enter the name you want to give your **custom Hazard View**.
5. Click **Save**.

Your custom Hazard View will be saved and viewable in the **Manage view configurations** window.

NOTE: Before you can save a custom Hazard View, you must reorder or hide (using the right-click menu) the columns in the Hazard Log.

Load a Custom Hazard View

To load a custom Hazard View

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, select the **Hazard View** option that you saved your custom Hazard View under.
3. In the Hazard Log Editor, right-click a **column header cell**.
4. From the drop-down menu, select **Manage view configurations**.
5. Select the **custom Hazard View** you wish to load.
6. Click **Load**.

The Hazard Log will reconfigure to your custom view's settings.

NOTE: Custom Hazard Views are saved under the original Hazard View you customized. You must first select the name of the original Hazard View you customized before you open the **Manage view configurations**.

Reset a Hazard View

To reset a custom Hazard View

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, select the **Hazard View** option that you wish to reset.
3. In the Hazard Log Editor, right-click a **column header cell**.
4. From the drop-down menu, select ***Default View** option.
5. Click **Load**.

The Hazard Log will reconfigure to the Hazard View's default settings.

Delete a Custom Hazard View

To delete a custom Hazard View

1. Open the **Hazard Table**.
2. In the Hazard Log Editor, select the **Hazard View** option that you saved your custom Hazard View under.
3. In the Hazard Log Editor, right-click a **column header cell**.
4. From the drop-down menu, select **Manage view configurations**.
5. Select the **custom Hazard View** you wish to delete.
6. Click **Delete**.

Your custom Hazard View will be deleted.

NOTE: Even though you've deleted the custom Hazard View, the Hazard Log will still be configured in that custom view's settings. You must reset the Hazard View to restore its default configuration.

Requirements Log

The **Requirements Log** contains a hierarchical list of mitigation requirements and their allocations, sources, and verification methods.

Each AdvoCATE Project contains one Requirements Log.

Create a New Requirements Log

To create a new **Requirements Log**

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Requirements Log**.
3. Click **Next**.
4. Select the **AdvoCATE Project folder** that will contain the Requirements Log.
5. Name the Requirements Log file. *A Requirements Log file must end in the extension* **.requirements**.
6. Click **Finish**.

The **Requirements Log Editor** opens, and a new Requirements Log renders in tabular form with column headings.

By default, the new **Requirements Log** is empty and new entries must be added. Use the scroll bar to see the various entry types available. For more information about Requirements Log Entries, see the *Requirements Log Entries* section.

Open the Requirements Log

To open the Requirements Log

1. Open the **Requirements Dashboard**.
2. In the **Requirements Tables** window, select your **Requirements Log** (*Figure 20*).
3. Click **Open**.

The **Requirements Log** will open in the **Requirements Log Editor**.

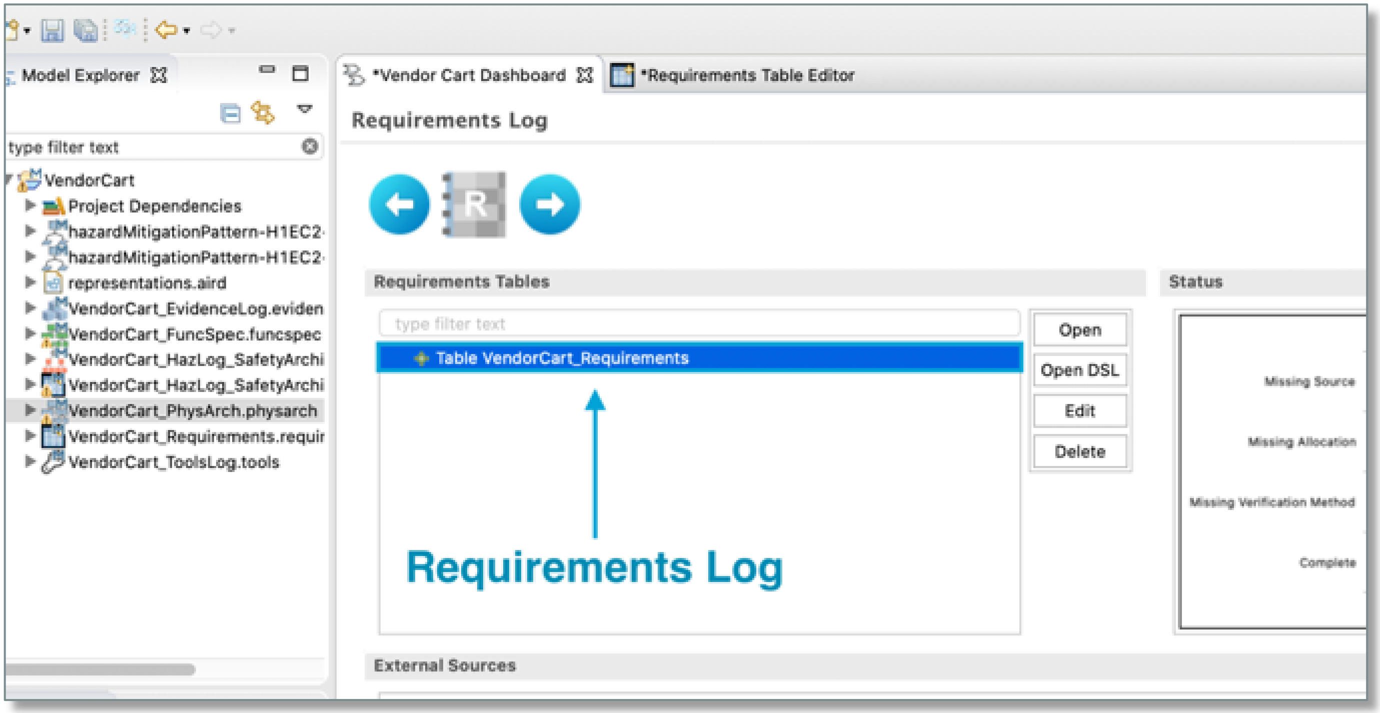

Model Explorer
type filter text
VendorCart
Project Dependencies
hazardMitigationPattern-H1EC2-
hazardMitigationPattern-H1EC2-
representations.aird
VendorCart_EvidenceLog.eviden
VendorCart_FuncSpec.funcspec
VendorCart_HazLog_SafetyArchi
VendorCart_HazLog_SafetyArchi
VendorCart_PhysArch.physarch
VendorCart_Requirements.requir
VendorCart_ToolsLog.tools
*Vendor Cart Dashboard
*Requirements Table Editor
Requirements Log
R
Requirements Tables
type filter text
Table VendorCart_Requirements
Open
Open DSL
Edit
Delete
Status
Missing Source
Missing Allocation
Missing Verification Method
Complete
Requirements Log
External Sources

Figure 20: In the Requirements Table window, select the Requirements Log and click the Open button.

External Sources Table

The **External Sources Table** lists external sources for Mitigation Requirements.

The External Sources listed in the External Sources Table will appear in the drop-down menu for the **Source** entry in your **Requirements Log**. For more information about the Requirements Log Source entry, see the *Requirement Source* section.

When an **External Source** entry has a ⚠ Validation, the entry has not been used in the **Requirements Log**. For more information about Validations, see the *Validations Appendix*.

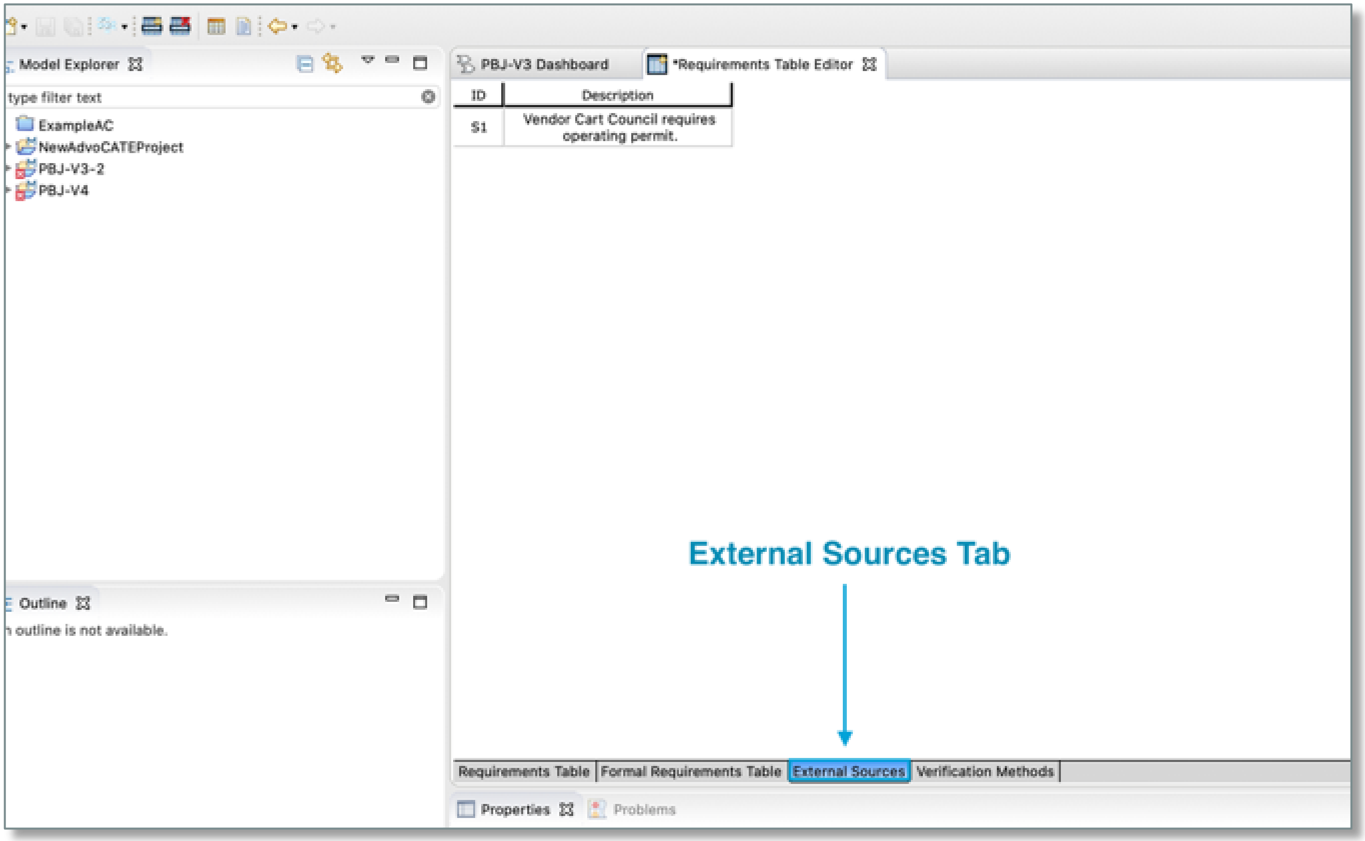
NOTE: **Internal Sources** will auto-populate in your Requirements Log based on your **Mitigation Requirement**'s corresponding **Mitigation** and **Hazard** entries in your Hazard Log.

Open External Sources Table

To open your External Sources Table

1. Open your **Requirements Log**.
2. In the Requirements Log Editor, select the **External Sources** tab (*Figure 21*).

The **External Sources Table** will open in the **Requirements Log Editor**.



ID	Description
S1	Vendor Cart Council requires operating permit.

Figure 21: In the Requirements Log Editor, select the External Sources tab to open the External Sources Table.

Create an External Source

To create a new External Source

1. Open your **External Sources Table**.
2. Click the **Add Element** icon in toolbar.

A new External Source entry will generate in the **External Sources Table**.

Edit an External Source

To edit an External Source

1. Open your **External Sources Table**.
2. Double-click the **table cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click outside the **table cell** you edited to update your External Source.

The text in your table cell will update to reflect the change.

Delete an External Source

To delete an External Source

1. Open your **External Sources Table**.
2. Select the **External Source** you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected External Source will be deleted.

Verification Methods Table

The **Verification Methods Table** lists the **Verification Methods/Mitigation Verifications** for your AdvoCATE Project. For more information about the Verification Methods, see the *Requirement Verification Method* section.

When you associate **Mitigation Verifications** in a Hazard Log, the **Verification Methods** in the **Requirements Log** and **Verification Methods Table** will automatically update.

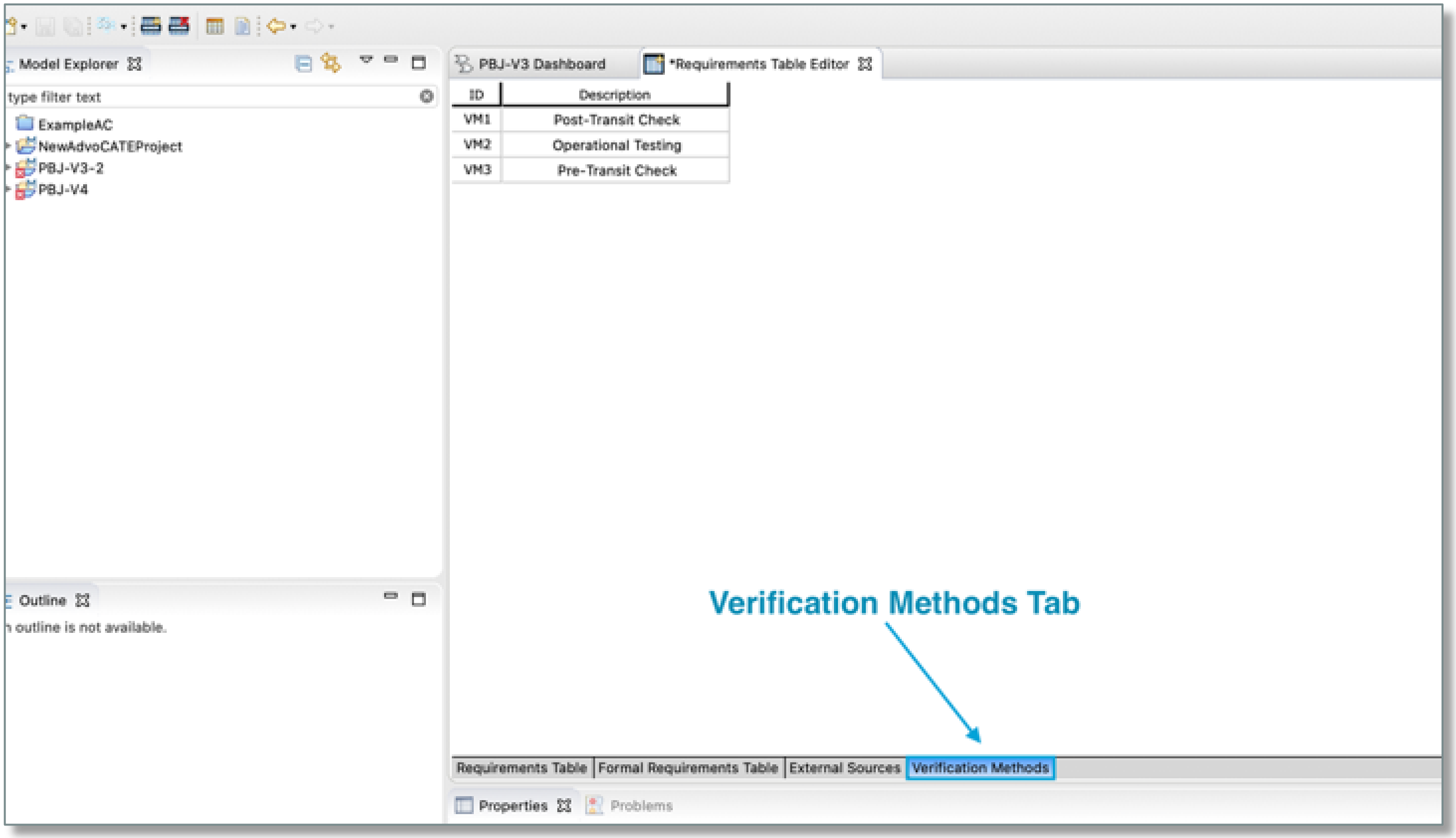
When a **Verification Method** entry has a ⚠ Validation, the entry has not been used in the **Requirements Log**. For more information about Validations, see the *Validations Appendix*.

Open Verification Methods Table

To open the Verification Methods Table

1. Open your **Requirements Log**.
2. In the Requirements Log Editor, select the **Verification Methods** tab (*Figure 22*).

The **Verification Methods Table** will open in the **Requirements Log Editor**.



ID	Description
VM1	Post-Transit Check
VM2	Operational Testing
VM3	Pre-Transit Check

Figure 22: In the Requirements Log Editor, select the Verification Methods tab to open the Verifications Methods Table.

Create a Verification Method

To create a new Verification Method

1. Open your **Verification Methods Table**.
2. Click the **Add Element** icon in toolbar.

A new Verification Method will generate in the **Verification Methods Table**.

NOTE: New Verification Methods can also be created in the **Hazard Log** and **Requirements Log**.

Edit a Verification Method

To edit a Verification Method

1. Open your **Verification Methods Table**.
2. Double-click the **table cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click outside the **table cell** you edited to update your Verification Method.

The text in your table cell will update to reflect the change.

Delete a Verification Method

To delete a Verification Method

1. Open your **Verification Methods Table**.
2. Select the **Verification Method** you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected Verification Method will be deleted from the **Verification Methods Table**.

Requirements Log Entries

Each **Requirements Log** has the following table entries:

- *Requirement ID*
- *Requirement Description*
- *Requirement Type*
- *Requirement Source*
- *Requirement Allocation*
- *Requirement Verification Method*
- *Requirement Verification Allocation*
- *Requirement Relations*
- *Requirement Notes*

Requirement ID

Requirement IDs are the identification marker for each **Mitigation Requirement**.

These IDs are auto-generated as new entries are created, and they allow for **Traceability** between the **Hazard Log**, **CESs**, and **BTDs**.

Create a New Requirement

To create a new Requirement in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, click the **Add Element** icon in toolbar.

A new Requirement row will generate in the **Requirements Log**.

Delete a Requirement

To delete a Requirement from the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click a **cell in the Requirement row** you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected Requirement will be deleted.

Requirement Description

Requirement Descriptions are high-level requirement statements that describe how a specific **Mitigation** will reduce the risk of **Cause** events.

When you associate Mitigation Requirements in a Hazard Log, the **Requirement Descriptions** will automatically update in the **Requirements Log**.

Edit a Requirement Description

To edit a Requirement Description in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Description cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click **outside the cell** you edited to update your Requirement Description.

The text in your table cell will update to reflect the change.

Requirement Type

Requirement Types classify the system dependability attribute/s that are impacted by a Mitigation Requirement.

AdvoCATE supports the following Requirement Types:

- **Safety**: Requirement describing the state of reduced risk of loss or harm to be attained by the system.
- **Functional**: Requirement describing the expected service to be delivered to its stakeholders.
- **Performance**: Requirement qualifying/quantifying the quality of system service to be delivered to its stakeholders.
- **Security**: Requirement describing the state of reduced security risk to be attained by the system.
- **Reliability**: Requirement qualifying/quantifying the (extent of) continuity of correct service to be delivered.
- **Availability**: Requirement qualifying/quantifying the (extent of) readiness for correct system service.
- **Maintainability**: Requirement describing the capacity of system modification and repair to be undergone.

Requirement Source

Requirement Sources represent an analysis, guideline, or regulation (which can be tied to a specific activity, item, location, or situation) that entails a Requirement.

Source entries list internal and external sources for Requirements.

Internal Sources correspond to activities or entities within AdvoCATE Artifacts and will auto-generate based on corresponding entries in your Hazard Log.

External Sources are simply strings describing the Source and can be entered manually into the **External Sources Table**. External Sources listed to the External Sources Table will appear in the drop-down menu for the **Source** entry in your **Requirements Log**. For more information about the External Source Table, see the *External Sources Table* section.

Create a New External Requirement Source

To create a new Source in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Source cell** you wish to populate.
3. Enter a new value in the **Create a New Source…** textbox field.
4. Click **outside the cell** you populated to create your Source.

The text in your table cell will update to reflect the change.

Add an Existing Requirement Source

To add an existing Source from the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Source cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Source** you wish to add.
4. Click **outside the cell** you edited to add the Source to your Requirement.

The text in your table cell will update to reflect the change.

Edit an Internal Requirement Source

To edit an Internal Requirement Source, you must change the **Requirement**'s corresponding **Mitigation** and **Hazard** entries in your **Hazard Log**.

For more information about Hazard Log Entries, see the *Hazard Log Entries* section.

Edit an External Requirement Source

To edit an External Source from the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Source cell** you wish to edit.
3. Enter your edits in the **Source cell** textbox field.
4. Click **outside the cell** you edited to update the Source.

The text in your table cell will update to reflect the change.

NOTE: Editing an External Requirement Source entry in the Requirements Log will also update the corresponding **External Source Table** entry.

Remove a Requirement Source

To remove a Requirement Source in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Source cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Source** you wish to remove.
4. Click **outside the cell** you edited to remove the Source from the Requirement.

The text in your table cell will update to reflect the change.

Requirement Allocation

A **Requirement Allocation** is the Component or Function that implements the **Requirement**.

In order to allocate a Requirement, you must create and populate **PhysArch** and **FuncSpec** files. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

Each **Requirement** can reference multiple Requirement Allocations.

When you allocate a Requirement, the Requirement’s corresponding **CESs** and **BTDs** will automatically update.

Create a New Requirement Allocation

To create a new Requirement Allocation in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Allocation cell** you wish to populate.
3. In the drop-down menu, toggle the **checkbox** for your corresponding **Component** or **Function**.

The text in your table cell will update to reflect the change.

Edit a Requirement Allocation

To edit a Requirement Allocation, you must edit the corresponding **Component** or **Function** entry in your **PhysArch** or **FuncSpec** file. For more information on **PhysArch DSL** or **FuncSpec DSL**, visit the *PhysArch DSL* or *FuncSpec DSL* section.

Delete a Requirement Allocation

To delete a Requirement Allocation from the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Allocation cell** you wish to delete.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Component** or **Function** you wish to delete.

The text in your table cell will update to reflect the change.

Requirement Verification Method

A **Verification Method** is the means of checking that the corresponding **Mitigation Requirement** meets its intent.

Each **Requirement** can reference multiple Verification Methods.

When you associate Mitigation Verifications in a Hazard Log, the **Verification Method** entries in the **Requirements Log** and **Verification Methods Table** will automatically update.

Create a New Requirement Verification Method

To create a new Verification Method in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Verification Method cell** you wish to populate.
3. Enter a new value in the **Create a New Verification Method…** textbox field.
4. Click **outside the cell** you populated to create your Verification Method.

The text in your table cell will update to reflect the change.

Add an Existing Requirement Verification Method

To add an existing Verification Method in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Verification Method cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Verification Method** you wish to add.
4. Click **outside the cell** you edited to add the Verification Method to your Requirement.

The text in your table cell will update to reflect the change.

Edit a Requirement Verification Method

To edit a Verification Method in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Verification Method cell** you wish to edit.
3. Enter your edits in the **Verification Method cell** textbox field.
4. Click **outside the cell** you edited to update the Verification Method.

The text in your table cell will update to reflect the change.

NOTE: Editing a Verification Method entry in the Requirements Log will also update the corresponding **Verification Methods Table** entry and the **Mitigation Verification** entry in the **Hazard Log**.

Remove a Verification Method

To remove a Verification Method from the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Verification Method cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Verification Method** you wish to remove.
4. Click **outside the cell** you edited to remove the Verification Method from the Requirement.

The text in your table cell will update to reflect the change.

Requirement Verification Allocation

A **Verification Allocation** is the **Evidence Artifact** that satisfies the **Verification Method**.

Each **Verification Method** can reference multiple Verification Allocations.

When you allocate a Verification Method, the **Evidence Log** will automatically update. For more information about the Evidence Log, see the *Evidence Log* section.

Create a New Requirement Verification Allocation

To create a new Verification Allocation in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Verification Allocation cell** you wish to populate.
3. Enter a new value in the **Create a New Verification Allocation…** textbox field.
4. Click **outside the cell** you populated to create your Verification Allocation.

The text in your table cell will update to reflect the change.

NOTES:

- You must enter a Verification Method into the Requirements Log before entering a corresponding Verification Allocation.
- If you are performing your first Verification Allocation and you have not already created an Evidence Log, a dialog window will open. For more information about the Evidence Log, see the *Evidence Log* section.

Edit a Requirement Verification Allocation

To edit a Verification Allocation in the **Requirements Log Editor**

1. Open your **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Verification Allocation cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click outside the **table cell** you edited to update your Verification Allocation.

The text in your table cell will update to reflect the change.

NOTE: Editing a Verification Allocation entry in the Requirements Log will also update the **Evidence Artifact Definition** entry in the **Evidence Log**.

Delete a Requirement Verification Allocation

To delete a Verification Allocation in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Verification Allocation cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Verification Allocation** you wish to edit.
4. Click **outside the cell** you edited to delete the Verification Allocation from the Verification Method.

The text in your table cell will update to reflect the change.

Requirement Relations

Requirement Relation entries define the relationships between parent **Requirements** and child **Sub-Requirements**.

Relation Types

- **Parent/Child**: A parent Requirement can be decomposed into two or more child Requirements if the composition of the child Requirements entails the parent.
- **Relate To…**: A requirement is related to another requirement when there exists a dependency between the two. Related-to thus subsumes the other relations.
- **Conflicts With…**: One Requirement conflicts with another when they are deemed to not be compatible, whether for reasons of functionality, performance, cost, or other attributes.
- **Assumed By/Assumes…**: One Requirement assumes another if the implementation of the former can exploit the latter being satisfied by the system, typically as a constraint of the environment or concept of operations. A Requirement that is assumed by another Requirement can reflect an otherwise implicit stakeholder expectation.
- **Derived From/Derives…**: A Requirement (RD) that is derived from another Requirement (R) captures the constraints emerging from the specific design and solution selected to realize R. RD can be allocated to the individual items used to implement R.
- **Implied By/Implies…**: A Requirement implies another if any implementation which implements the former also implements the latter. Typically, such Requirements are expressed at the same level of abstraction.
- **Refines/Refines To…**: A Requirement refines to another if the latter provides additional information that constrains possible implementations. Unlike with implied Requirements, refined Requirements are typically at a lower level of abstraction.

Create a New Parent Relation for a Requirement

To create a new Parent Relationship for a Requirement

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Relation cell** you wish to populate.
3. In the drop-down menu, select **Create New Parent**.

A new Requirement row will generate, and the corresponding Relation cells for the new parent and child will update to reflect the change.

Designate an Existing Parent Relation for a Requirement

To designate an existing Requirement as a Parent Relation

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **child Relation cell** you wish to designate a parent to.
3. In the drop-down menu, select **Select Parents**.
4. In the Select Parents window, select the **checkbox/es** for the corresponding Requirements you wish to add as parents.
5. Click **Ok**.

The corresponding Relation cells for the new parent/s and child will update to reflect the change.

Remove a Parent Relation for a Requirement

To remove a Parent Relationship for a Requirement

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Relation cell** you wish to edit.
3. In the drop-down menu, select **Remove Parent**.

The corresponding Relation cells for the former parent/s and child will update to reflect the change.

Edit Relations between Requirements

To edit relations between Requirements

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Relation cell** you wish to edit.
3. In the drop-down menu, select **Edit Relations**.
4. From the drop-down menu, select how you want to define the Relation for your selected Requirement.
5. In the Select Relations window, select the **checkbox/es** for the corresponding Requirements you wish to add this Relation to.
6. Click **Ok**.

The corresponding Relation cells will update to reflect the change.

Remove Relations between Requirements

To remove relations between Requirements

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Relation cell** you wish to edit.
3. In the drop-down menu, select **Edit Relations**.
4. From the drop-down menu, select the **Relation Type** you want to remove for your selected Requirement.
5. In the Select Relations window, deselect the **checkbox/es** for the corresponding Requirements you wish to remove.
6. Click **Ok**.

The corresponding Relation cells will update to reflect the change.

Requirement Notes

Note entries are any comments you have about the corresponding Requirement for yourself, safety analysts, or review committees.

Add a Requirement Note

To add a Requirement Note in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Note cell** you wish to populate.
3. Enter a new value in the **Note cell** textbox field.
4. Click **outside the cell** you populated to add your Note.

The text in your table cell will update to reflect the change.

Edit a Requirement Note

To edit a Requirement Note in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Note cell** you wish to edit.
3. Enter your edits in the **Note cell** textbox field.
4. Click **outside the cell** you edited to update the Note.

The text in your table cell will update to reflect the change.

Delete a Requirement Note

To delete a Requirement Note in the **Requirements Log Editor**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, double-click the **Note cell** you wish to edit.
3. Delete the text in the **Note cell** textbox field using Backspace (PC) or Delete (Mac) on your keyboard.
4. Click **outside the cell** you edited to delete the Note.

Sub-Requirements

Sub-Requirements are child requirements that fulfill a part of their parent Requirement's risk-reduction statements.

Create a New Sub-Requirement

To create a new Sub-Requirement

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, right-click the **Requirement** you wish to make a Sub-Requirement for.
3. From the drop-down menu, select **Create New Sub-Requirement**.

A new Sub-Requirement row will generate in the **Requirements Log**.

For more information about Requirement Relations, see the *Requirement Relations* section.

Associate Requirement to Control

To associate a Requirement to a Control

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, right-click the **Requirement** you wish to associate to a Control.
3. From the drop-down menu, select **Associate Requirement to Control**.
5. In the pop-up menu, select the **checkbox/es** for the corresponding **Controls** you wish to associate to your selected Requirement.
4. Click **Ok**.

Your Requirement Associations will update to reflect the change. For more information about Controls, see the *CES Mitigation Node Types* and *BTDs Mitigation Node Types* sections.

Edit Requirement’s Barrier/Mitigation Associations

To edit a Requirement’s Barrier/Mitigation Association

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, right-click the **Requirement** you wish to change Barrier/Mitigation associations for.
3. From the drop-down menu, select **Change Requirement Barrier/Mitigation Associations**.
6. In the pop-up menu, toggle the **checkbox/es** for the corresponding **Barriers** you wish to add or remove as an association from your selected Requirement.
4. Click **Ok**.

Your Barrier/Mitigation Associations for your Requirement will update to reflect the change.

For more information on Barriers, see the *CES Mitigation Node Types* and *BTDs Mitigation Node Types* sections.

Navigate to Associated Assurance Artifacts

To navigate to a Requirement’s associated Assurance Artifacts

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, right-click the **Requirement** you wish to see Assurance Artifacts for.
3. From the drop-down menu, select **Open associated artifact…**.
4. From the drop-down menu, select one of the following artifact options:
   - **Show Associated Argument Nodes**: the corresponding Argument Structure will open, and the GSN Nodes with the associated Requirement will be selected.
   - **Show Associated Solutions** the corresponding Argument Structure will open, and the Solution Nodes with the associated Evidence Artifacts will be selected.
   - **Show Associated Hazards in Hazard Log**: the corresponding Hazard Table will open, and the associated Hazard row will be highlighted.
   - **Show Associated Hazard in Bowtie**: the corresponding BTD will open, and the Event Node with the associated Hazard will be selected.
   - **Show Associated Control/Barrier Instances**: the corresponding CES or BTD will open, and the Control Node with the associated Controls/Barriers will be selected.

Based on your selection, the corresponding AdvoCATE file or diagram will open.

Generate Argument Structure from Requirements Log

To generate an Argument Structure from **the Requirements Log**

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, right-click an **ID cell** to generate an **Argument Structure**.
3. From the menu, select **Generate an argument from this requirements table**.
4. Select the **AdvoCATE Project folder** where you wish to save your Argument Structure. *We recommend selecting the same AdvoCATE Project folder where your Requirements Log is located.*
5. Enter the name you want to give your **Argument Structure** file.

   NOTE: Argument Structure files must end in the file extension **.argument**.

6. Click **Finish**.

A new **Argument Structure** will generate and open. For more information about Argument Structures, see the *Argument Structures* section.

NOTE: The Data Tree that is instantiated with the Argument Structure—representing the assurance semantics of the Requirements Log—is encoded within AdvoCATE.

Generate Data Tree from Requirements Table

To generate a new Data Tree from the Requirements Log

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, right-click the **ID cell** you wish to generate a **Data Tree** from.
3. From the menu, select **Generate data tree from this requirements table**.
4. Select the **AdvoCATE Project folder** where you wish to save your Data Tree. *We recommend selecting the same AdvoCATE Project folder where your Requirements Log is located.*
5. Enter the name you want to give your **Data Tree**.

   NOTE: Data Trees must end in the file extension **.patterndata**.

6. Click **Finish**.

A new Data Tree will generate and open in the **Data Tree Diagram Editor**. For more information about Data Trees, see the *Data Trees* section.

NOTE: Auto-generated Data Trees is not currently laid out automatically. However, Data Tree layout does not affect the layout of the Argument Instance that is created from the Data Tree.

Safety Architecture

The **Safety Architecture** is a collection of **Controlled Event Structures** (CESs) which consist of Events, Barriers, and Controls—the safety signature—and a collection of Hazards.

NOTE: AdvoCATE creates the Safety Architecture automatically from the semantics of the Hazard Log.

Create a New Safety Architecture

To create a new Safety Architecture

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Safety Architecture**.
3. Select the **AdvoCATE Project folder** that will contain the new Safety Architecture.
4. Name the **Safety Architecture**.

   NOTE: A Safety Architecture must end in the extension **.safetyarch**.

5. Click **Finish**.

The new Safety Architecture is created and opens the corresponding **Top Event** in the **BTD Diagram Editor**.

NOTE: If you already created a Hazard Log, a Safety Architecture already exists for your AdvoCATE Project.

Create a New Hazardous Activity from the Safety Architecture

To create a new Hazardous Activity from the **Safety Architecture**

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **New Hazardous Activity** button.
3. Enter your edits in the **Description** textbox field.
4. Click **Ok**.

The new **Hazardous Activity** will generate and be viewable in the **Hazardous Activities** window (*Figure 23*).

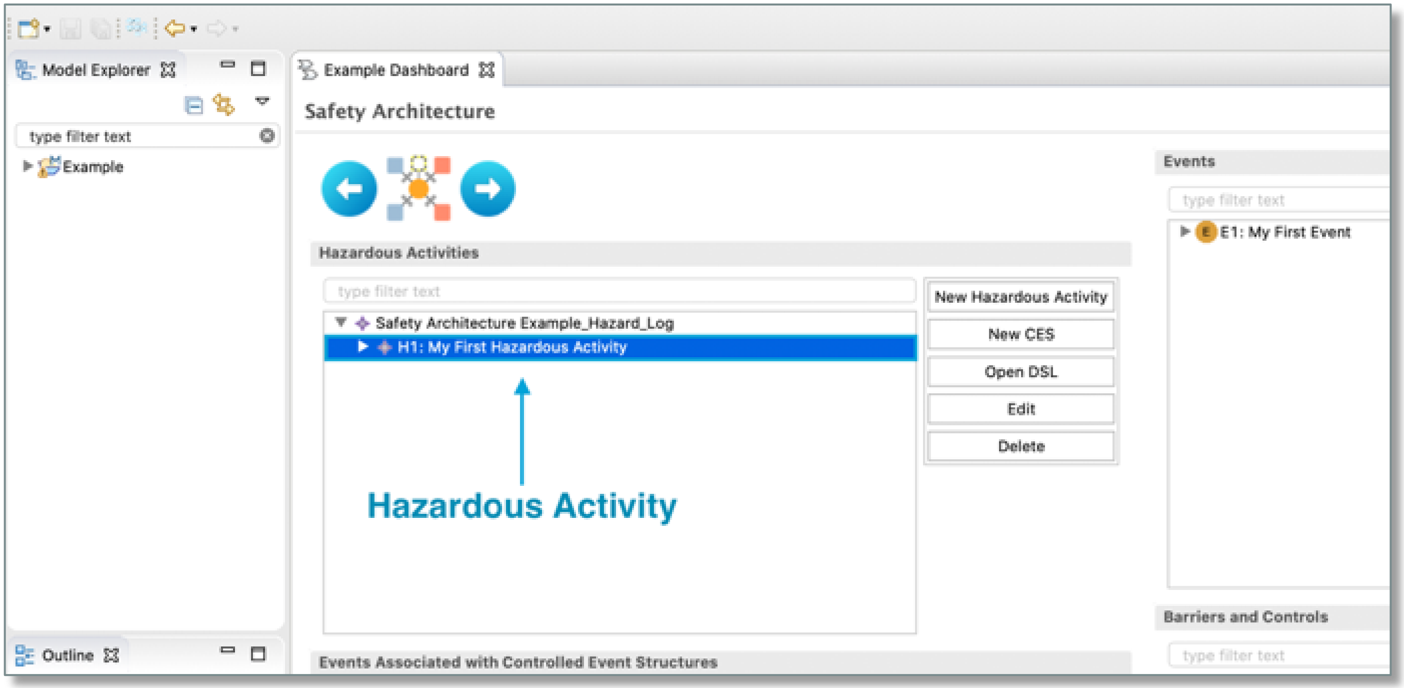


Figure 23: When you create a new Hazardous Activity from the Safety Architecture Dashboard, the new Hazardous Activity will generate and be viewable in the Hazardous Activates window.

Edit a Hazardous Activity from the Safety Architecture

To edit a Hazardous Activity from the **Safety Architecture**

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **Hazardous Activity** you want to edit.
3. Once your Hazardous Activity is selected, click **Edit**.
4. Enter your edits in the **Description** textbox field.
5. Click **Ok**.

The **Hazardous Activity** will update to reflect the change.

Delete a Hazardous Activity from the Safety Architecture

To delete a Hazardous Activity from the **Safety Architecture**

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **Hazardous Activity** you want to delete.
3. Click **Delete**.
4. Click **Yes** to confirm your deletion request.

The **Hazardous Activity** will be deleted.

Controlled Event Structures (CESs)

Controlled Event Structures (CESs) specify a chain of **Events** (or casual pathway) resulting in harm/loss, along with the system-level Mitigations, such as **Barriers** and **Controls**, that are invoked to manage harm/loss.

Just like the Hazard Log, each CES corresponds to a distinct combination of a **Hazardous Activity**, **System State**, and **Environmental Condition**.

NOTE: AdvoCATE creates CESs automatically from the semantics of the Hazard Log.

Create a New CES

To create a new CES

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **Hazardous Activity** you want to create a new CES under.
3. Once your Hazardous Activity is selected, click **New CES** (*Figure 24*).
4. Select your Hazardous Activity's **System State** from the drop-down menu. *If you need to create a new **System State** for your Hazardous Activity, see the* Create a New System State *section.*
5. Select your Hazardous Activity's **Environmental Condition** from the drop-down menu. *If you need to create a new **Environmental Condition** for your Hazardous Activity, see the Create a New Environmental Condition section.*
6. Click **Ok**.

The new **CES** will generate and be viewable in the **Hazardous Activities** window.

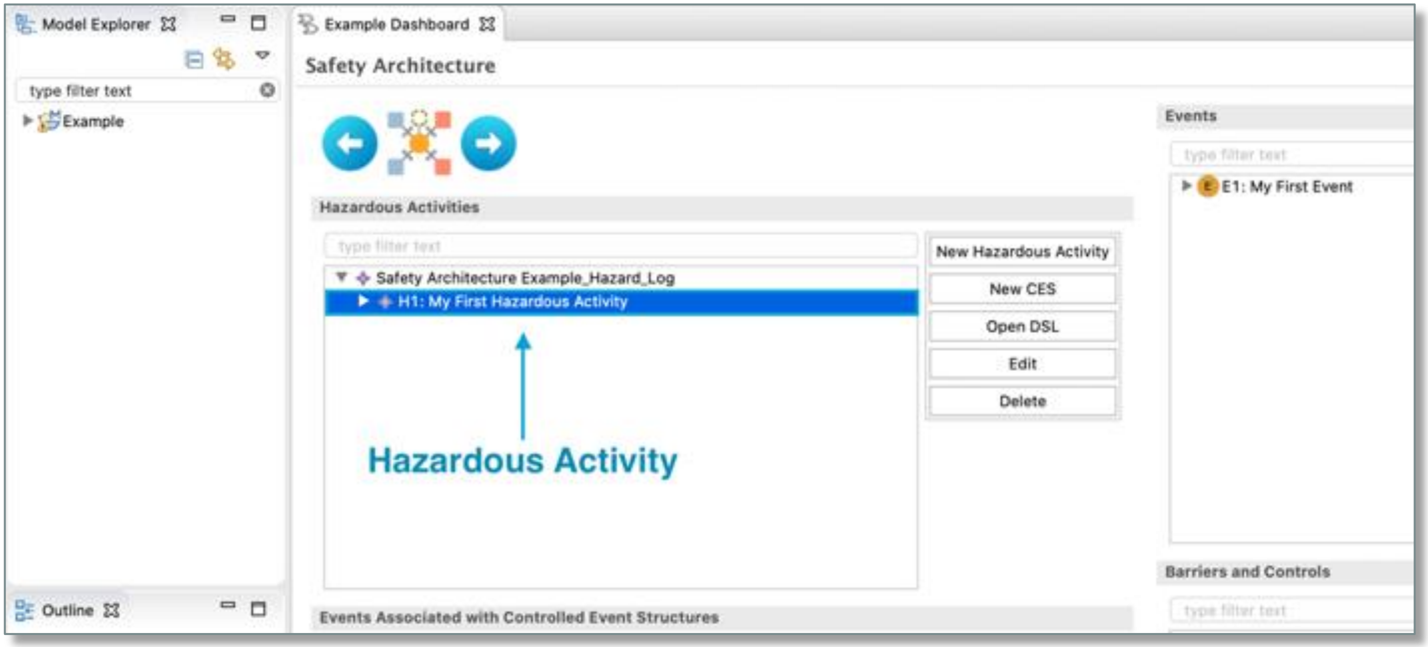


Figure 24: To create a new CES from the Safety Architecture Dashboard, you must select the Hazardous Activity you want to create the new CES under and then click the New CES button.

Open a CES

To open a CES

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. Select the **CES** you wish to open from the drop-down list (*Figure 25*).
4. Click **Open**.

The **CES** will open in the **CES Diagram Editor**.

NOTE: If the CES layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the CES Toolbar. The nodes and links will rearrange with equal spacing and alignment.

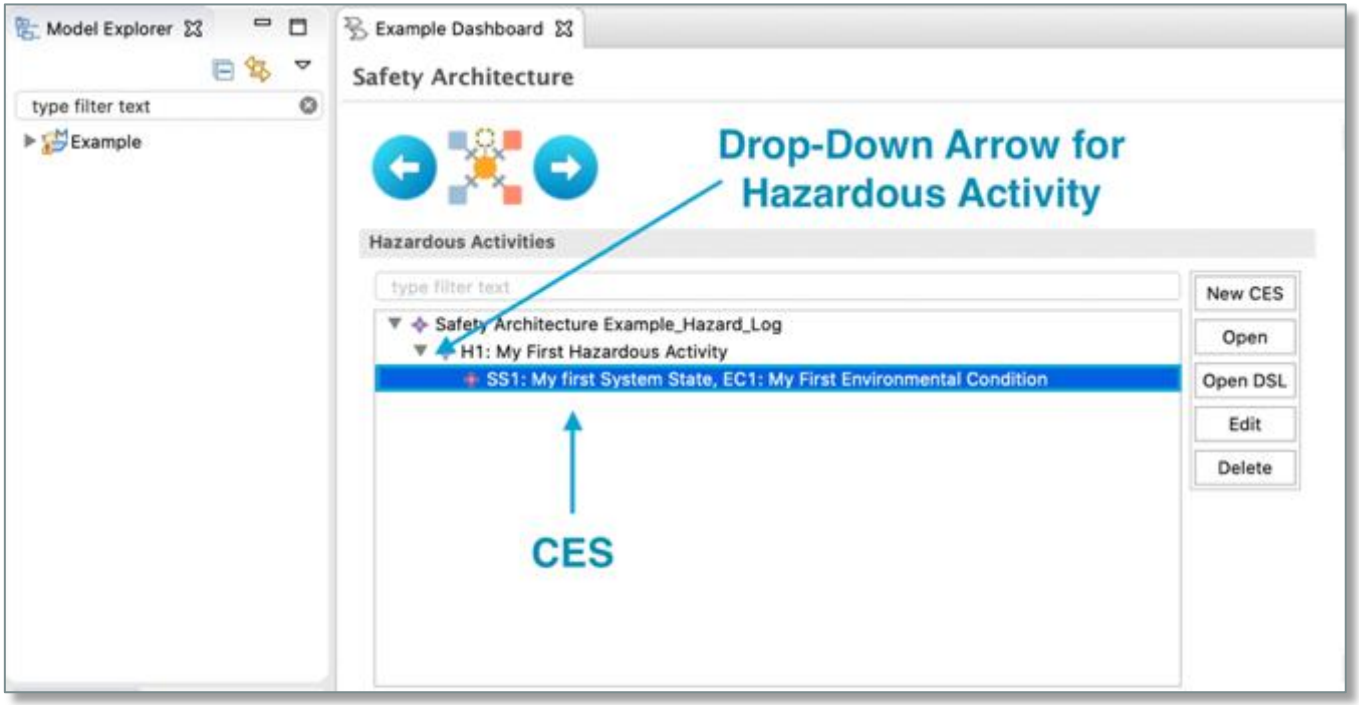

Model Explorer
type filter text
Example
Example Dashboard
Safety Architecture
Drop-Down Arrow for Hazardous Activity
Hazardous Activities
type filter text
Safety Architecture Example_Hazard_Log
H1: My First Hazardous Activity
SS1: My first System State, EC1: My First Environmental Condition
CES
New CES
Open
Open DSL
Edit
Delete

Figure 25: To open a CES from the Safety Architecture Dashboard, you must use the drop-down arrows next to the CES's corresponding Hazardous Activity to select the CES you wish to open.

Edit a CES's System State or Environmental Condition

To edit a CES's System State or Environmental Condition

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. Select the **CES** you wish to edit from the drop-down list.
4. Click **Edit**.
5. Use the **drop-down menus** to change the CES's **System State** or **Environmental Condition**.
6. Click **Ok**.

The **CES** will update to reflect the change.

Delete a CES

To delete a CES

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. Select the **CES** you wish to edit from the drop-down list.
4. Click **Delete**.
5. Click **Yes** to confirm your deletion request.

The **CES** will be deleted.

CES Diagram Editor

The **CES Diagram Editor** allows users to create various **Controls**, **Events**, and **Links** in an CES. The CES Diagram Editor has its own **Toolbar**, **Palettes**, **Links**, and **Association Links** icons.

CES Toolbar

The **CES Toolbar** is in the top-left corner of the CES Diagram Editor.

The CES Toolbar has the following tools:

- **Arrange All**: Arranges all Nodes with equal spacing and alignment.
- **Autosize all nodes to fit text**: Resizes all Nodes to display all their Description text.
- **Select All**: Selects everything in the CES Diagram.
  - **Select All Connectors**: Selects all Links in the CES Diagram.
  - **Select All Shapes**: Selects all Nodes in the CES Diagram.
- **Layers**: Shows/Hides different Layers in the CES Diagram.
- **Pin/Unpin**: Opens the Pinned Elements Wizard.
- **Zoom In**: Zooms in on the Diagram Canvas.
- **Zoom Out**: Zooms out on the Diagram Canvas.
- **Zoom Percentage**: Changes the visible percentage of the Diagram Canvas.
- **Export diagram as image**: Exports the Diagram Canvas as a JPG, PNG, SVG, or PDF.
- **Edit all node fonts**: Opens Font Selection Window to change the font for an entire CES.
- **Pin selected elements**: Pins selected element/s, so they cannot be moved.
- **Unpin selected elements**: Unpins selected element/s, so they can be moved.
- **Delete**: Deletes selected element/s.

CES Palette

The **CES Palette** is in the top-right corner of the CES Diagram Editor.

The CES Palette has the following tools:

- **Select**: Selects elements on Diagram Canvas.
- **Zoom In**: Zooms in on the selected portion of the Diagram Canvas.
- **Zoom Out**: Zooms out of the selected portion of the Diagram Canvas.
- **Note**: Creates a Note Node.
  - **Representation Link**: Creates a Representation Node of select AdvoCATE graphs, diagrams, tables, or views.
  - **Text**: Creates a Text Node.
  - **Note Attachment**: Creates a Note Node and Link to a selected Canvas element.
- **Generic Connection Creation Tool**: Draws Links between different Nodes. Links automatically format based on connected Node types.
- **Pin**: Pins selected element/s, so they cannot be moved.
- **Unpin**: Unpins selected element/s, so they can be moved.

CES Control Palette

The **CES Controls** are under the **CES Palette** in the CES Diagram Editor.

The CES Controls have the following tools:

- **New Control**: Creates a new Control and Control Node in your AdvoCATE Project.
- **Existing Control**: Creates a Control Node from an existing Control in your AdvoCATE Project.

CES Event Palette

The **CES Events** are under the **CES Palette** in the CES Diagram Editor.

The CES Events have the following tools:

- **New Event**: Creates a new Event and Event Node in your AdvoCATE Project.
- **Existing Event**: Creates an Event Node from an existing Event in your AdvoCATE Project.

CES Link Palette

The **CES Links** are under the **CES Palette** in the CES Diagram Editor.

The CES Links have the following tools:

- **Create Link**: Creates a Link between a source Event/ Control Node and a target Event/Control Node.
- **Create Escalation Link**: Creates an Escalation Link between an Escalation Factor and an Escalation Barrier/Control Node.

CES Association Link Palette

The **CES Association Links** are under the **CES Palette** in the CES Diagram Editor.

The CES Association Links have the following tools:

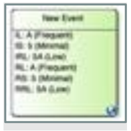
- **Argument Link**: Opens an Argument associated to an Event Node.
- **Barrier Requirement Associations**: Edits Requirements associated to a Barrier.

CES Event Node Types

Events are situations or scenarios that can impact your system.

CESs are made up of four **CES Event Node Types**:

- **Hazards** are events that represent a loss of control and have the potential for harm.
- **Causes** (or Threats) are the system or environmental events that create or lead to Hazards.
- **Escalation Factors** are second-level events that weaken or compromise the effectiveness of a Barrier or Control.
- **Effects** (or Consequences) are the loss events to be avoided at the boundary of the system and its environment.

In the CES Diagram Editor, all Event Nodes are depicted as  rounded, green nodes.

CES Event Nodes with a globe icon in the bottom-right corner are **Global Events**. **Global Causes** (or Global Threats) represent the beginning Causes/Threats of a Hazard. **Global Effects** (or Global Consequences) represent the final Effects/Consequences of a Hazard.

CES Event Nodes with a diagram icon in the bottom-right corner have diagram representations for the Event. To open these diagram representations, use the right-click menu to open the desired diagram representation.

When you update or change Event associations in the CES, the corresponding **Hazard Log** and **BTDs** will automatically update.

NOTE: While you can edit Event Nodes from an CES, **Bow Tie Diagrams (BTDs)** allow users to visually distinguish between Event Node Types. For more information about BTDs, see the *Bow Tie Diagrams (BTDs)* section.

CES Event Properties

In the **Properties View**, Event Nodes contain the following properties:

- **Name** is the identification marker for the Event Instance.
- **Description** is a statement that describes the Event.
- **Underlying Event** is the Event the Event Instance corresponds to.
- **Event Type** classifies the system dependability attribute/s that are impacted by the Event.
    - **Safety**: Event that can lead to loss or harm in the environment in the context of (worst-case) environmental conditions.
    - **Functional**: Event that can lead to an undesired deviation in the expected system service.
    - **Performance**: Event that can lead to an undesired deviation in the expected quality of a system service.
    - **Security**: Event that can compromise availability, integrity, and confidentiality, in the context of an environmental condition and system usage.
    - **Reliability**: Event that can diminish or eliminate the continuity of correct system service.
    - **Availability**: Event that can diminish or eliminate the readiness for correct system service.
    - **Maintainability**: Event that can diminish the capacity for modification and repair.
- **Associated Argument** is the Argument Structure that is associated to an Event.
- **Edit Hyperlinks** manages the hyperlinks for an Event Node.
    - **Add…**: adds a hyperlink to an Event Node.
    - **Edit…**:edits the location path for the selected hyperlink.
    - **Remove**: deletes the selected hyperlink.
    - **View…**: opens the folder location for the hyperlinked file.
- **Initial Likelihood Value** is a value between 0 to 1 that is related to the Initial Likelihood the Event will occur.
- **Initial Likelihood** is the initial level of probability that the Event will occur. AdvoCATE has five Likelihood classification levels based on the FAA's SMS manual.
    - **Frequent**: expected to occur routinely.
    - **Probable**: expected to occur often.
    - **Remote**: expected to occur infrequently.
    - **Extremely Remote**: expected to occur rarely.
    - **Extremely Improbable**: unlikely to occur, but not impossible.
- **Initial Severity** is the initial negative impact level the Event can cause to its surrounding environment and other systems. The Initial Severity provides a base metric of the Event's negative impact. AdvoCATE has five Severity classification levels.
    - **Minimal**: Negligible safety effect.
    - **Minor**: Physical discomfort to other systems or environment. Slight damage to system.
    - **Major**: Physical distress or injuries other systems or environment. Substantial damage to system.
    - **Hazardous**: Multiple serious injuries; fatal injury to a relatively small number other systems; or a system loss without fatalities.

  - **Catastrophic**: Multiple fatalities, usually with the loss of the system.
- **Initial Risk Level** is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Event. The Initial Risk Level provides a base metric of the extent of possible loss for a specific Event. AdvoCATE has three Risk classification levels.
  - **Low**: Severity and Likelihood map to the green cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without restriction or limitation; Event is not required to be actively managed.
  - **Medium**: Severity and Likelihood map to the yellow cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without additional mitigation; however, tracking and monitoring are required.
  - **High**: Severity and Likelihood map to the red cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk requires mitigation, tracking, and monitoring.
- **Residual Likelihood Value** is a value between 0 to 1 that is related to the Residual Likelihood the Event will occur.
- **Residual Likelihood** is the level of probability that the Event will occur after new mitigation mechanisms have been established. The Residual Likelihood provides a new metric of the Event's probability given the new levels of integrity for your system's various **Barriers** and **Controls**.
- **Residual Severity** is the new negative impact level the Event can cause to its surrounding environment and other systems once new mitigation mechanisms have been established. The Residual Severity provides a new metric of the Event's negative impact.
- **Residual Risk Level** is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Event. The Residual Risk Level provides a new metric of the extent of possible loss for a specific Event once new mitigation mechanisms have been established.
- **Depth** is the maximum width (i.e., length of a path in from the top), in any (Depth) BTD for which that Event Instance is the Top Event.

Create a Node for a New CES Event

To create a node for a new CES Event

1. Open the **CES Diagram Editor**.
2. Select the E **New Event** icon in the **CES Event Palette**.
3. Click the **Canvas** where you want to create new node for a New Event.

The new CES Event Node will generate with placeholder text and be viewable in the **CES Diagram Editor**.

Create a Node for an Existing CES Event

To create a node for an existing CES Event

1. Open the **CES Diagram Editor**.
2. Select the E **Existing Event** icon in the **CES Event Palette**.
4. Click the **Canvas** where you want to create your new node for an existing Event.
3. From the pop-up window, select the **Existing Event** you wish to create a node for.
4. Click **Ok**.

The new CES Event Node will generate with the existing Event's details and be viewable in the **CES Diagram Editor**.

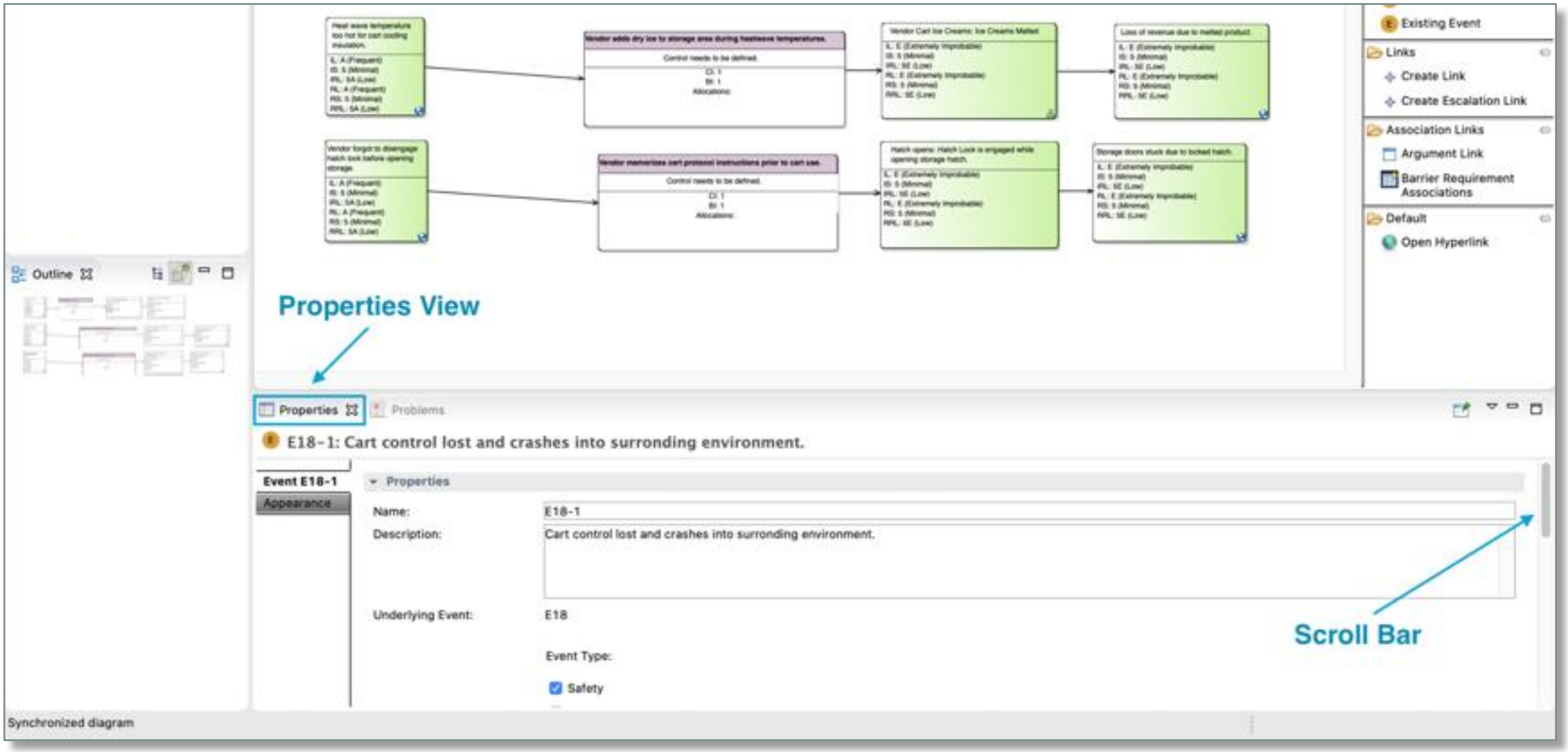
Edit a CES Event Node

To edit a CES Event Node

1. Open the **CES Diagram Editor**.
2. Select the **Event Node** you wish to edit.
3. In the **Properties View**, enter your edit/s into the corresponding fields (*Figure 26*).
4. Click outside the **Event Node** you edited to update your CES Event Node.

The Event Node in the **CES Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node's displayed contents will not update until after the Description field has been deselected.


Heat wave temperature too hot for cart cooling insulation.
IL: A (Frequent)
IS: 5 (Minimal)
IRL: 5A (Low)
RL: A (Frequent)
RS: 5 (Minimal)
RRL: 5A (Low)
Vendor adds dry ice to storage area during heatwave temperatures.
Control needs to be defined.
CI: 1
BI: 1
Allocations:
Vendor Cart Ice Creams: Ice Creams Melted
IL: E (Extremely Improbable)
IS: 5 (Minimal)
IRL: 5E (Low)
RL: E (Extremely Improbable)
RS: 5 (Minimal)
RRL: 5E (Low)
Loss of revenue due to melted product.
Vendor forgot to disengage hatch lock before opening storage.
Vendor memorizes cart protocol instructions prior to cart use.
Hatch opens: Hatch Lock is engaged while opening storage hatch.
Storage doors stuck due to locked hatch.
Outline
Existing Event
Links
Create Link
Create Escalation Link
Association Links
Argument Link
Barrier Requirement Associations
Default
Open Hyperlink
Properties View
Properties
Problems
Event E18-1
Appearance
Properties
Scroll Bar
Synchronized diagram

E18-1: Cart control lost and crashes into surronding environment.

Name: E18-1

Description: Cart control lost and crashes into surronding environment.

Underlying Event: E18

Event Type:

☑ Safety

Figure 26: To edit a CES Event Node, select the Event Node you wish to edit and then enter your edits into the corresponding fields in the Properties View.

Delete an CES Event Node

To delete a CES Event Node

1. Open the **CES Diagram Editor**.
2. Select the **Event Node** you wish to delete.
3. Click the **Delete** icon in CES Toolbar.

Your selected Event Node will be deleted from the **CES Diagram Editor**.

CES Mitigation Node Types

Mitigations are mechanisms for risk reduction and Hazard prevention.

CESs are made up of four **CES Mitigation Node Types**:

- **Barriers** are high-level or generic mitigations for Hazards, Causes, and Effects.
  - **Escalation Factor Barriers** are high-level or generic mitigations for Escalation Factors.
- **Controls** are low-level or specific mitigations that implement or form a Barrier.
  - **Escalation Factor Controls** are low-level or specific mitigations that implement or form an Escalation Factor Barrier.

In the CES Diagram Editor, all Mitigation Nodes are depicted as rounded, white nodes with a pink label. The pink label refers to the associated Barrier or Escalation Factor Barrier, while the white node lists the Control or Escalation Factor Control (*Figure 27*).

When you update or change Mitigation associations in the CES, the corresponding **Hazard Log** and **BTDs** will automatically update.

NOTE: While you can edit Mitigation Nodes from an CES, **Bow Tie Diagrams (BTDs)** allow users to visually distinguish between Mitigation Node Types. For more information about BTDs, see the *Bow Tie Diagrams (BTDs)* section.

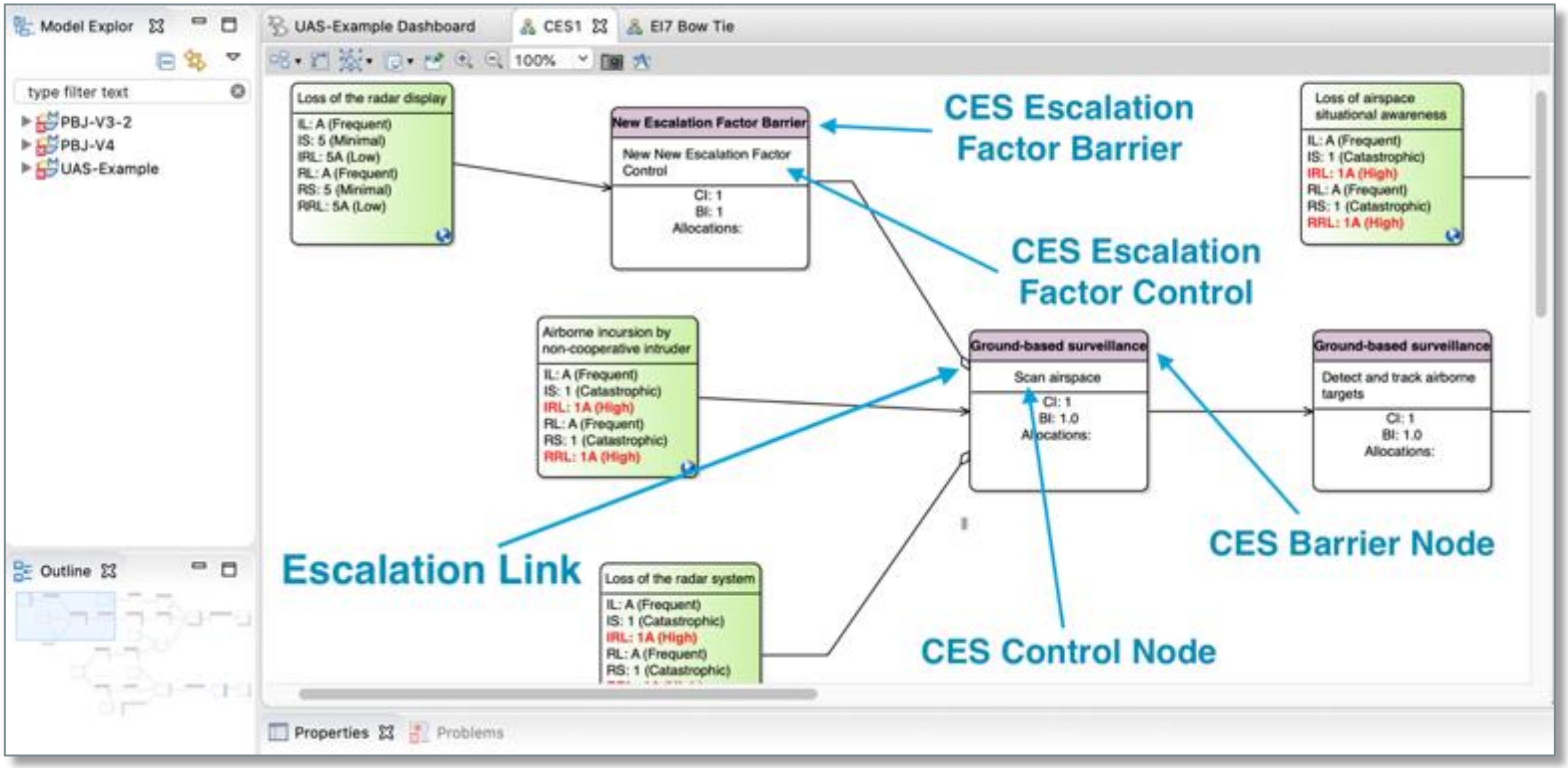


Figure 27: In CES Mitigation Nodes, the pink label refers to the associated Barrier or Escalation Factor Barrier, while the white node lists the Control or Escalation Factor Control. Escalation Links are depicted as a solid black line with a diamond.

CES Barrier Properties

In the **Properties View**, Barrier Nodes contain the following properties:

- **Barrier** is the underlying Barrier the Barrier Instance corresponds to.
- **Name** is the identification marker for the Barrier Instance.
- **Description** is a statement that describes the Barrier.
- **Edit Hyperlinks** manages the hyperlinks for a Barrier Node.
    - **Add…**: adds a hyperlink to a Barrier Node.
    - **Edit…**: edits the location path for the selected hyperlink.
    - **Remove**: deletes the selected hyperlink.
    - **View…**: opens the folder location for the hyperlinked file.
- **Barrier Integrity** is a value between 0 to 1 that is related to the likelihood of a dangerous breach of the Barrier.
- **New Mitigation** classifies a Mitigation as a New Mitigation instead of an Existing Mitigation. Existing Mitigations are previously established Mitigations. When safety analysts analyze a Mitigation to determine its Initial Risk Level and those Mitigations are deemed unacceptable, you can propose New Mitigations to lower the risk. By modeling both systems in one AdvoCATE Project, you can compute the **Initial** and **Residual Risk** levels for a Mitigation.
- **Mitigation Type** is the classification corresponding to the order of precedence of a system's safety design. AdvoCATE supports five Mitigation Types.
    - **Design Modification**: risk reduction through system redesign.
    - **Safety Feature**: risk reduction through the incorporation of safety features.
    - **Safety Device**: reduction through the incorporation of safety devices.
    - **Warning Device**: risk reduction through the incorporation of warning devices.
    - **Procedures and Training**: risk reduction through the incorporation of procedures and training.
- **Barrier Associated Requirements** are the Mitigation Requirements that describe how the Barrier will reduce the risk of CES Events. To see a full list of Mitigation Requirements for your AdvoCATE project, see the *Requirements Log* section.
- **Barrier Functional Allocations** are the Functions performed or used by the Barrier. To allocate a Function to a Barrier, you must first create and populate the **FuncSpec** file. For more information about the FuncSpec file, see the *Functional Specifications* section.
- **Barrier Physical Allocations** are the Components used by the Barrier. To allocate a Component to a Barrier, you must first create and populate the **PhysArch** file. For more information about PhysArch file, see the *Physical Architecture* section.

When you change CES Barrier Properties, the corresponding **BTDs** will automatically update.

CES Control Properties

In the **Properties View**, Control Nodes contain the following properties:

- **Name** is the identification marker for the Control Instance.
- **Description** is a statement that describes the Control.
- **Underlying Control** is the Control the Control Instance corresponds to.
- **Control Associated Argument** is the Argument Structure that justifies a dependability claim about the Control.
- **Edit Hyperlinks** manages the hyperlinks for a Control Node.
  - **Add…**: adds a hyperlink to a Control Node.
  - **Edit…**: edits the location path for the selected hyperlink.
  - **Remove**: deletes the selected hyperlink.
  - **View…**: opens the folder location for the hyperlinked file.
- **Control Integrity** is a value between 0 to 1 that is related to the likelihood of a dangerous breach of the Control.
- **Control Associated Requirements** are the Mitigation Requirements that describe how the Control will reduce the risk of CES Events. To see a full list of Mitigation Requirements for your AdvoCATE Project, see the *Requirements Log* section.
- **Control Functional Allocations** are the Functions performed or used by the Control. To allocate a Function to a Control, you must first create and populate the **FuncSpec** file. For more information about the FuncSpec file, see the *Functional Specifications* section.
- **Control Physical Allocations** are the Components used by the Control. To allocate a Component to a Control, you must first create and populate the **PhysArch** file. For more information about PhysArch file, see the *Physical Architecture* section.

When you change CES Control Properties, the corresponding **BTDs** will automatically update.

Create a Node for a New CES Control

To create a node for a new CES Control

1. Open the **CES Diagram Editor**.
2. Select the **New Control** icon in the **CES Control Palette**.
3. In the **Canvas**, select the **CES Link** where you want to create new node for a New Control.

The new CES Control Node will generate with placeholder text and be viewable in the **CES Diagram Editor**.

Create a Node for an Existing CES Control

To create a node for an existing CES Control

1. Open the **CES Diagram Editor**.
2. Select the **Existing Control** icon in the **CES Control Palette**.
3. In the **Canvas**, select the **CES Link** where you want to create new node for an Existing Control.
4. From the pop-up window, select the **Existing Control** you wish to create a node for.
5. Click **Ok**.

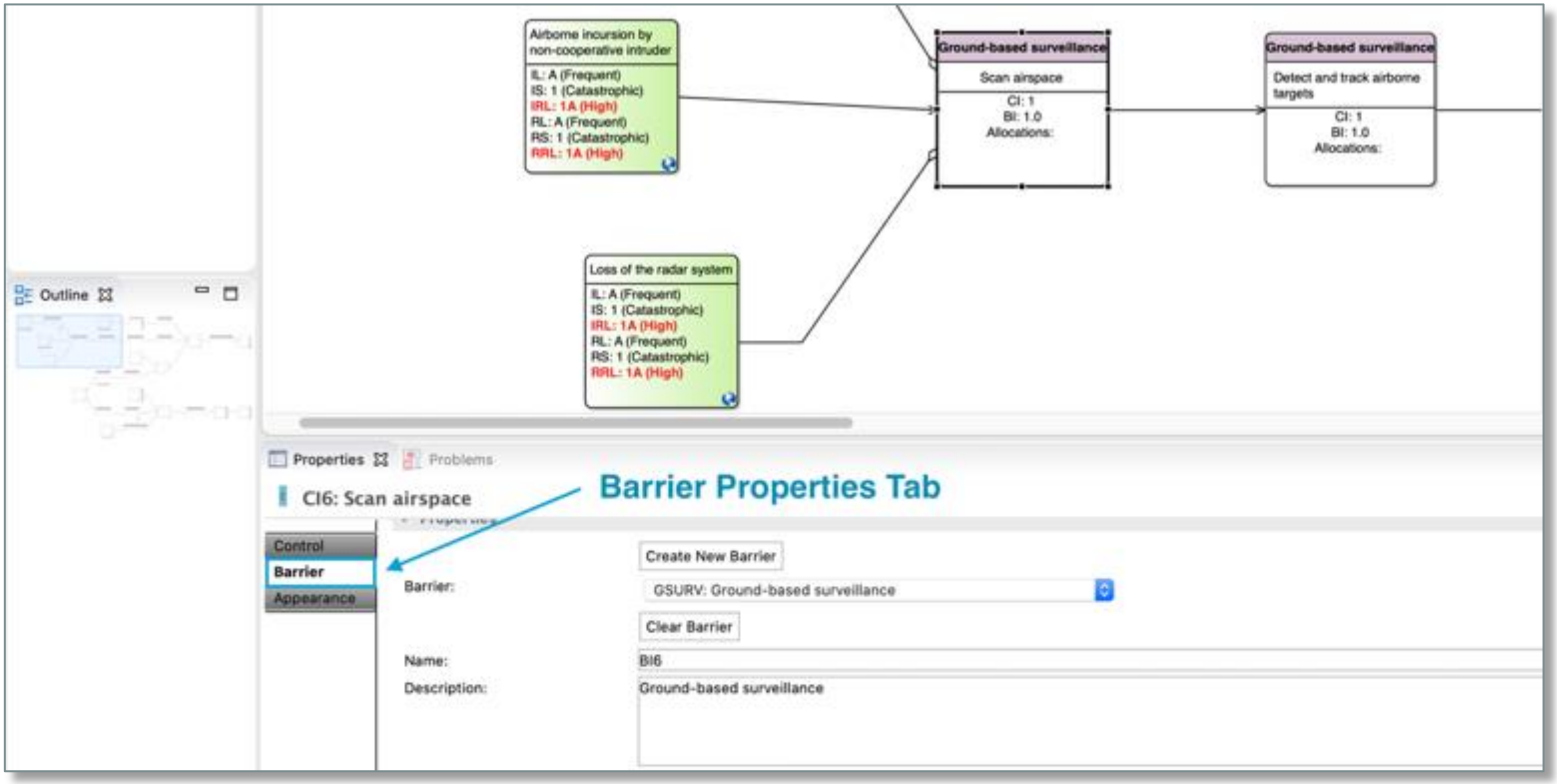
The new CES Control Node will generate with the existing Control's details and be viewable in the **CES Diagram Editor**.

Add a New CES Barrier to a Control Node

To add a new CES Barrier to a Control Node

1. Open the **CES Diagram Editor**.
2. Select the **Control Node** you wish to add a new **Barrier Node** to.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 28*).
4. Click the **Create New Barrier** button.

The new CES Barrier Node label will generate above the Control with placeholder text and will be viewable in the **CES Diagram Editor**.



CI6: Scan airspace

Barrier: GSURV: Ground-based surveillance

Name: BI6

Description: Ground-based surveillance

Figure 28: In the CES Diagram Editor, the Barrier Properties Tab is located within the Properties View.

Add an Existing CES Barrier to a Control Node

To add an existing CES Barrier to a Control Node

1. Open the **CES Diagram Editor**.
2. Select the **Control Node** you wish to add an existing **Barrier Node** to.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 28*).
4. Select the **drop-down arrows** next to the **Barrier** field.
5. Select the **existing Barrier** you wish to add to the Control.

The existing CES Barrier Node label will generate above the Control and will be viewable in the **CES Diagram Editor**.

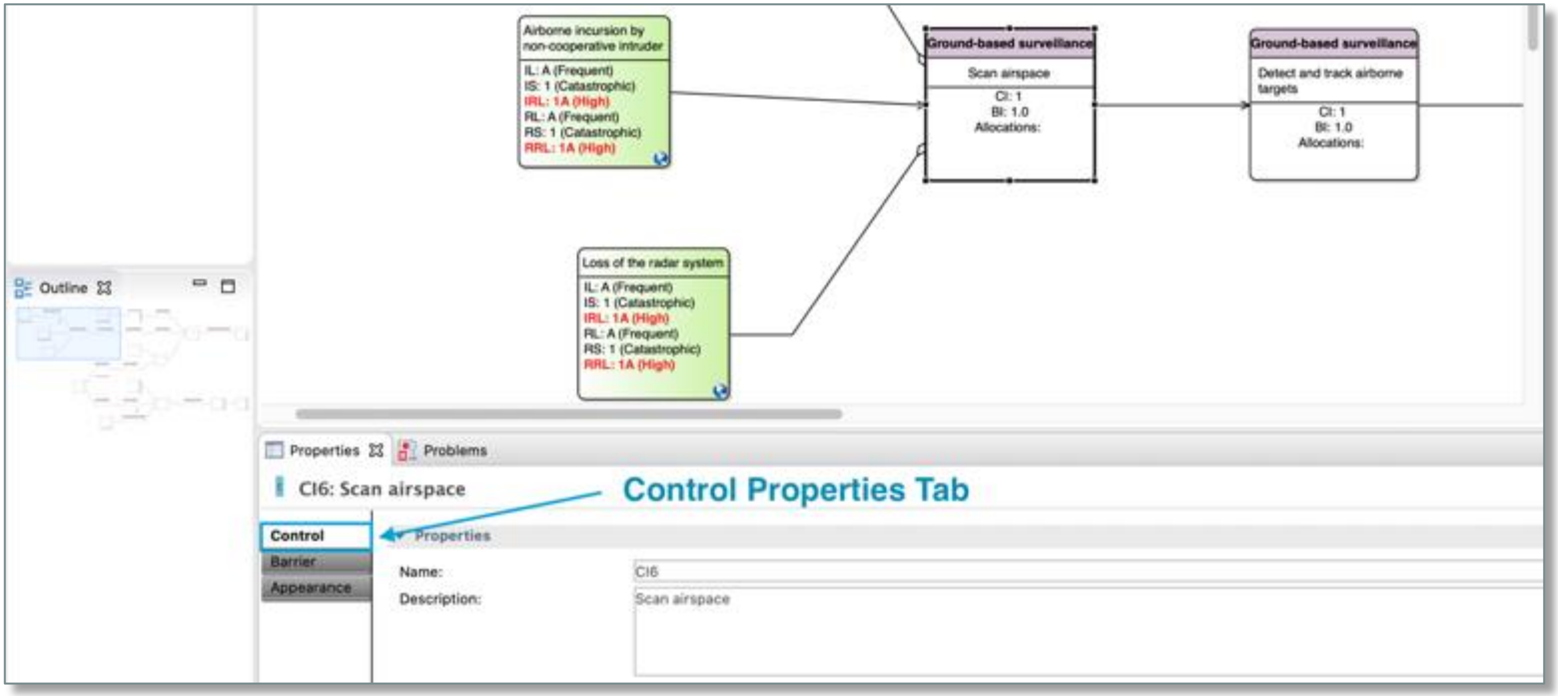
Edit a CES Control Node

To edit a CES Control Node

1. Open the **CES Diagram Editor**.
2. Select the **Control Node** you wish to edit.
3. In the **Properties View**, select the **Control Properties** tab (*Figure 29*).
4. Enter your edit/s into the corresponding fields.
5. Click outside the **Control Node** you edited to update your CES Control Node.

The Control Node in the **CES Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node's displayed contents will not update until after the Description field has been deselected.



Name: CI6

Description: Scan airspace

Figure 29: In CES Diagram Editor, the Control Properties Tab is located within the Properties View.

Edit a CES Barrier Node

To edit a CES Barrier Node

1. Open the **CES Diagram Editor**.
2. Select the **Control Node** with the **Barrier Node label** you wish to edit.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 28*).
4. Enter your edit/s into the corresponding fields.
5. Click outside the **Control Node** you edited to update your CES Barrier Node.

The Barrier Node label in the **CES Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node's displayed contents will not update until after the Description field has been deselected.

Remove a CES Barrier Node from a Control

To remove a CES Barrier Node from a Control

1. Open the **CES Diagram Editor**.
2. Select the **Control Node** you wish to remove a Barrier from.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 28*).
4. Click the **Clear Barrier** button.

The Barrier Node label will be removed from the selected Control.

Delete an CES Control Node

To delete a CES Control Node

1. Open the **CES Diagram Editor**.
2. Select the **Control Node** you wish to delete.
3. Click the **Delete** icon in CES Toolbar.

Your selected Control Node will be deleted from the **CES Diagram Editor**.

CES Link Types

Links create references between source and target nodes. In the CES Diagram Editor, Links are generally depicted as a → solid black line with an arrow between nodes.

However, when Escalation Factors are present, **Escalation Links** are depicted as a —◇ solid black line with a diamond.

Create a New CES Link

To create a new CES Link

1. Open the **CES Diagram Editor**.
2. Select the ✧ **Create Link** icon in the **CES Link Palette**.
3. In the **Canvas**, select the **source node** where you want to start the new Link.
4. In the **Canvas**, select the **target node** where you want the new Link to end.

The new CES Link will generate in between the source and target nodes and will be viewable in the **CES Diagram Editor**.

Create a New Escalation Link

To create a new CES Escalation Link

1. Open the **CES Diagram Editor**.
2. Select the ✧ **Create Escalation Link** icon in the **CES Link Palette**.
3. In the **Canvas**, select the **source node** where you want to start the new Escalation Link.
4. In the **Canvas**, select the **target node** where you want the new Escalation Link to end.

The new CES Escalation Link will generate in between the source and target nodes and will be viewable in the **CES Diagram Editor**.

Delete an CES Link or Escalation Link

To delete a CES Link or Escalation Link

1. Open the **CES Diagram Editor**.
2. Select the **Link** or **Escalation Link** you wish to delete.
3. Click the ✖ **Delete** icon in CES Toolbar.

Your selected Link or Escalation Link will be deleted from the **CES Diagram Editor**.

Bow Tie Diagrams (BTDs)

Each Event in a CES can be opened and modeled as a **Top Event** in a **Bow Tie Diagram** (BTD). BTDs display the immediate **Threats** and **Consequences** of the **Top Event**. The width of a BTD is usually three Events, and it includes all the Barriers and Controls on the path between those Events. The width of a BTD also requires Escalation Links to be one Event deep.

Depth Bow Tie Diagrams (Depth BTDs) are BTDs without width restrictions.

Create a New BTD from the Safety Architecture Dashboard

To create a new BTD from the **Safety Architecture Dashboard**

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. From the drop-down list, select the **CES** you wish to create a BTD for.
4. In the **Events Associated with Controlled Event Structures** window, select the **Hazardous Activity** you want to create a new BTD under (*Figure 30*).
5. Once your Hazardous Activity is selected, click **New Bow Tie**.
6. From the pop-up window, select the **Event** you wish to create as a **Top Event** in a new BTD.
7. Click **Ok**.

The new **BTD** will generate and open in the **BTD Diagram Editor**.

NOTE: If the BTD layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the BTD Toolbar. The nodes and links will rearrange with equal spacing and alignment.

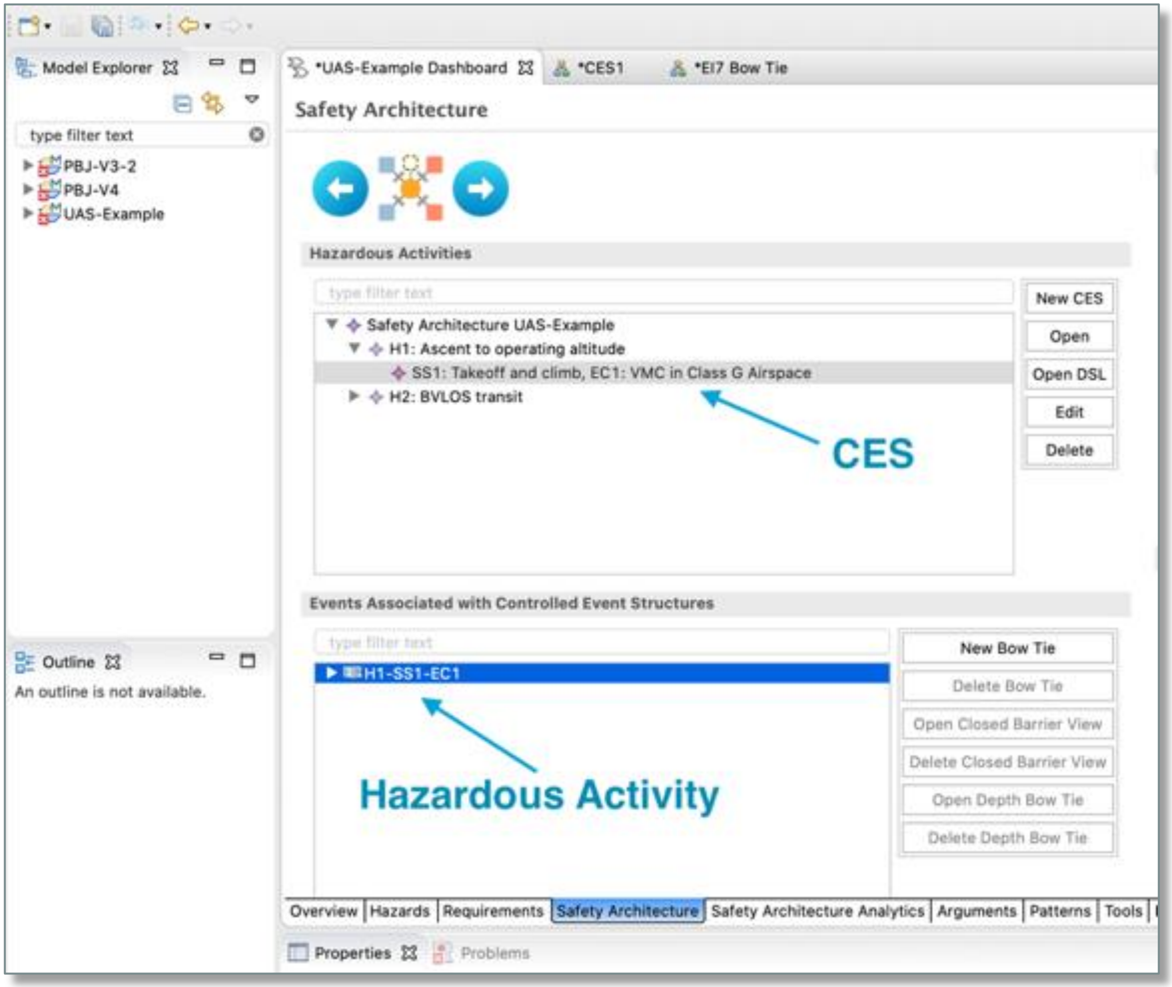


Figure 30: To create a BTD from the Safety Architecture Dashboard, select the corresponding CES in the Hazardous Activity window and the Hazardous Activity in the Events Associated with Controlled Event Structures window before selecting the New Bow Tie button.

Create a New BTD from an CES Event

To create a new BTD from an **CES Event**

1. Open a **CES**.
2. In the CES Diagram Editor, right-click the **CES Event Node** you wish to create a BTD for.
3. In the right-click menu, select **New Bowtie Diagram**.

   NOTE: If the **New Bowtie Diagram** option is not available in the right-click menu, a BTD for the Event already exists.

The new **BTD** will generate and open in the **BTD Diagram Editor**.

NOTE: If the BTD layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the BTD Toolbar. The nodes and links will rearrange with equal spacing and alignment.

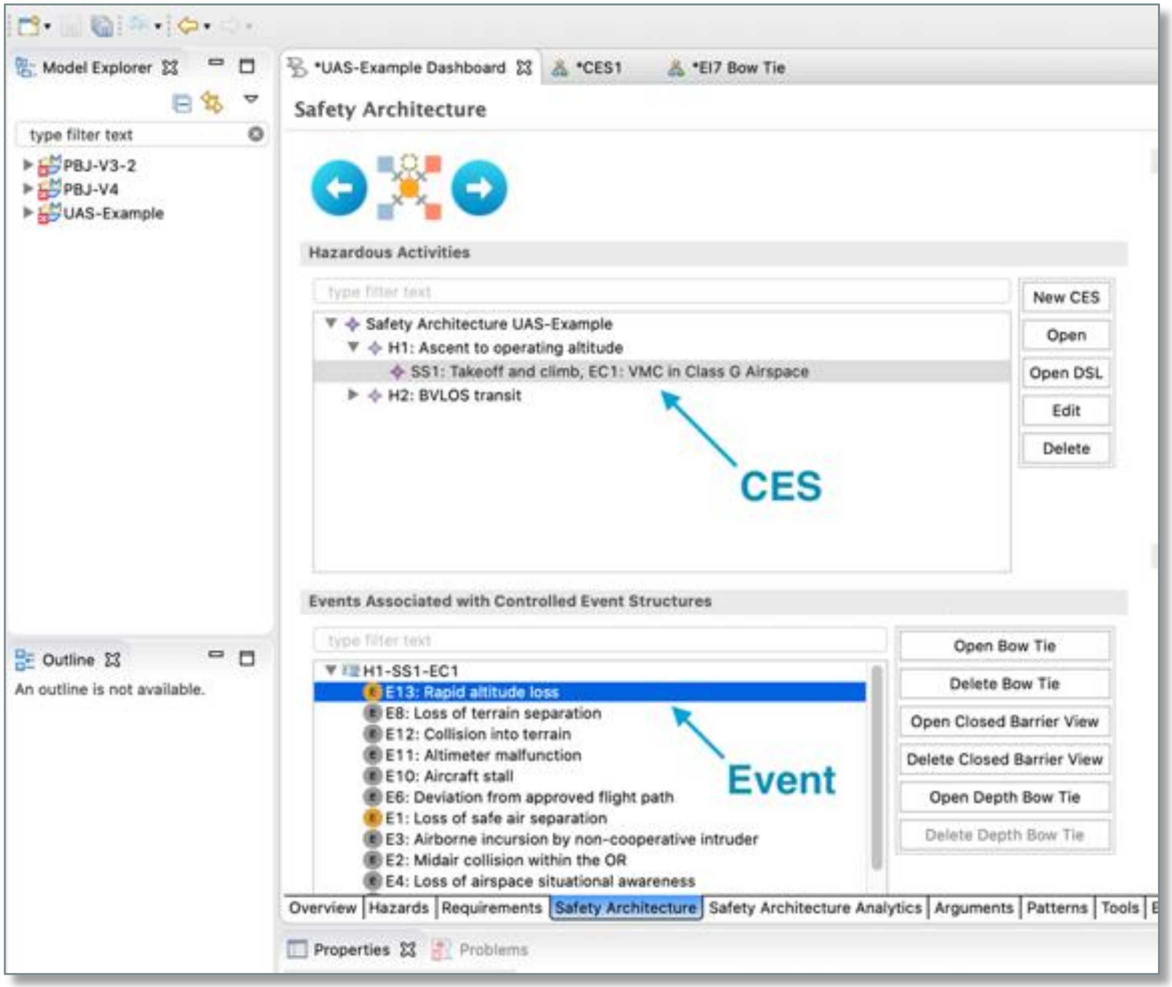
Open a BTD from the Safety Architecture Dashboard

To open a BTD from the **Safety Architecture Dashboard**

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. From the drop-down list, select the **CES** you wish to open a BTD for.
4. In the **Events Associated with Controlled Event Structures** window, select the **drop-down arrow** next to the **Hazardous Activity** you want to open a BTD under.
5. Under the Hazardous Activity, select the **Event** you wish to open as a BTD (*Figure 31*).
6. Once your Event is selected, click **Open Bow Tie**.

The **BTD** will open in the **BTD Diagram Editor**.

NOTE: If the BTD layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the BTD Toolbar. The nodes and links will rearrange with equal spacing and alignment.



- Safety Architecture UAS-Example
  - H1: Ascent to operating altitude
    - SS1: Takeoff and climb, EC1: VMC in Class G Airspace
  - H2: BVLOS transit

- H1-SS1-EC1
  - E13: Rapid altitude loss
  - E8: Loss of terrain separation
  - E12: Collision into terrain
  - E11: Altimeter malfunction
  - E10: Aircraft stall
  - E6: Deviation from approved flight path
  - E1: Loss of safe air separation
  - E3: Airborne incursion by non-cooperative intruder
  - E2: Midair collision within the OR
  - E4: Loss of airspace situational awareness

Figure 31: To open or delete a BTD, Depth BTD, or Closed Barrier View from the Safety Architecture Dashboard, you must select the corresponding CES in the Hazardous Activity window and Event in the Events Associated with Controlled Event Structures window first.

Open a BTD from an CES Event

To open a BTD from an **CES Event**

1. Open a **CES**.
2. In the CES Diagram Editor, right-click the **CES Event Node** you wish to open a BTD for.
3. In the right-click menu, select **Open** > **Bow Tie**.

   NOTE: If the **Open** > **Bow Tie** option is not available in the right-click menu, a BTD for the Event does not exists.

The **BTD** will open in the **BTD Diagram Editor**.

NOTE: If the BTD layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the BTD Toolbar. The nodes and links will rearrange with equal spacing and alignment.

Open a Depth BTD from the Safety Architecture Dashboard

To open a Depth BTD from the **Safety Architecture Dashboard**

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. From the drop-down list, select the **CES** you wish to open a Depth BTD for.
4. In the **Events Associated with Controlled Event Structures** window, select the **drop-down arrow** next to the **Hazardous Activity** you want to open a Depth BTD under.
5. Under the Hazardous Activity, select the **Event** you wish to open as a Depth BTD (*Figure 31*).
6. Once your Event is selected, click **Open Depth Bow Tie**.

The **Depth BTD** will open in the **BTD Diagram Editor**.

NOTE: If the Depth BTD layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the BTD Toolbar. The nodes and links will rearrange with equal spacing and alignment.

Delete a BTD from the Safety Architecture

To delete a BTD from the **Safety Architecture Dashboard**

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. From the drop-down list, select the **CES** you wish to delete a BTD from.
4. In the **Events Associated with Controlled Event Structures** window, select the **drop-down arrow** next to the **Hazardous Activity** you want to delete a BTD under.
5. Under the Hazardous Activity, select the **Event** you wish to delete the BTD from (*Figure 31*).
6. Once your Event is selected, click **Delete Bow Tie**.
7. Click **Yes** to confirm your deletion request.

The **BTD** will be deleted.

BTD Diagram Editor

The **BTD Diagram Editor** allows users to create various **Links**, **Events**, and **Controls** in a BTD. The BTD Diagram Editor has its own **Toolbar**, **Palettes**, and **Association Links**.

BTD Toolbar

The **BTD Toolbar** is in the top-left corner of the BTD Diagram Editor.

The BTD Toolbar has the following tools:

- **Arrange All**: Arranges all Nodes with equal spacing and alignment.
- **Autosize all nodes to fit text**: Resizes all Nodes to display all their Description text.
- **Select All**: Selects everything in the BTD.
  - **Select All Connectors**: Selects all Links in the BTD.
  - **Select All Shapes**: Selects all Nodes in the BTD.
- **Layers**: Shows/Hides different Layers in the BTD.
- **Pin/Unpin**: Opens the Pinned Elements Wizard.
- **Zoom In**: Zooms in on the Diagram Canvas.
- **Zoom Out**: Zooms out on the Diagram Canvas.
- **Zoom Percentage**: Changes the visible percentage of the Diagram Canvas.
- **Export diagram as image**: Exports the Diagram Canvas as a JPG, PNG, SVG, or PDF.
- **Edit all node fonts**: Opens Font Selection Window to change the font for an entire BTD.
- **Pin selected elements**: Pins selected element/s, so they cannot be moved.
- **Unpin selected elements**: Unpins selected element/s, so they can be moved.
- **Delete**: Deletes selected element/s.

BTD Palette

The **BTD Palette** is in the top-right corner of the BTD Diagram Editor.

The BTD Palette has the following tools:

- **Select**: Selects elements on Diagram Canvas.
- **Zoom In**: Zooms in on the selected portion of the Diagram Canvas.
- **Zoom Out**: Zooms out of the selected portion of the Diagram Canvas.
- **Note**: Creates a Note Node.
    - **Representation Link**: Creates a Representation Node of select AdvoCATE graphs, diagrams, tables, or views.
    - **Text**: Creates a Text Node.
    - **Note Attachment**: Creates a Note Node and Link to a selected Canvas element.
- **Generic Connection Creation Tool**: Draws Links between different Nodes. Links automatically format based on connected Node types.
- **Pin**: Pins selected element/s, so they cannot be moved.
- **Unpin**: Unpins selected element/s, so they can be moved.

BTD Association Link Palette

The **BTD Association Links** are under the **BTD Palette** in the BTD Diagram Editor.

The BTD Association Links have the following tools:

- **Barrier Requirement Associations**: Edits Requirements associated to a Barrier.
- **Control Requirement Associations**: Edits Requirements associated to a Control.
- **Argument Link**: Opens an Argument associated to an Event Node.

BTD Event Palette

The **BTD Events** are under the **BTD Palette** in the BTD Diagram Editor.

The BTD Events have the following tools:

- E **New Event**: Creates a new Event and Event Node in your AdvoCATE Project.
- E **Existing Event**: Creates an Event Node from an existing Event in your AdvoCATE Project.

BTD Control Palette

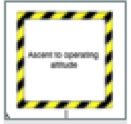
The **BTD Controls** are under the **BTD Palette** in the BTD Diagram Editor.

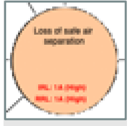
The BTD Controls have the following tools:

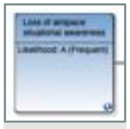
- C **New Control**: Creates a new Control and Control Node in your AdvoCATE Project.
- C **Existing Control**: Creates a Control Node from an existing Control in your AdvoCATE Project.

BTD Event Node Types

BTDs are made up of five **BTD Event Node Types**:

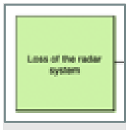
- **Hazardous Activities** render as a  black-and-yellow-striped square node and are the triggering actions or tasks a system performs that led to Top Events. Hazardous Activity Nodes are created when a new BTD is created.

- **Top Events** render as an  orange circle node. Top Event Nodes are created when a new BTD is created.

    - Top Events that represent Hazards are a loss of control and have the potential for harm.
    - If an Event appears as a Hazard in another BTD, the Event Node will have a "H" in the bottom-right corner of the node.
- **Threats** (or Causes) render as a  blue-gradient square. Threats are the system or environmental events that create or lead to Top Events.

  - If an Event appears as a Threat in another BTD, the Event Node will have a “T” in the bottom-right corner of the node.

- **Consequences** (or Effects) render as a
Midair collision within t...
IL: E (Extremely improbable)
IS: 1 (Catastrophic)
RL: E (Extremely improbable)
red-gradient square. Consequences are the loss events to be avoided at the boundary of the system and its environment and are the negative impacts of Top Events.
  - If an Event appears as a Consequence in another BTD, the Event Node will have a “C” in the bottom-right corner of the node.

- **Escalation Factors** render as a 

green square node. Escalation Factors are second-level events that weaken or compromise the effectiveness of a Barrier or Control.

BTD Event Nodes with a globe icon in the bottom-right corner are **Global Events**. **Global Threats** (or Global Causes) represent the beginning Threats/Causes of a Hazard. **Global Consequences** (or Global Effects) represent the final Consequences/Effects of a Hazard.

BTD Event Nodes with a diagram icon in the bottom-right corner have diagram representations for the Event. To open these diagram representations, use the right-click menu to open the desired diagram representation.

When you update or change Event associations in a BTD, the corresponding **Hazard Log** and **CES** will automatically update.

BTD Event Properties

In the **Properties View**, Event Nodes contain the following properties:

- **Name** is the identification marker for the Event Instance.
- **Description** is a statement that describes the Event.
- **Underlying Event** is the Event the Event Instance corresponds to.
- **Event Type** classifies the system dependability attribute/s that are impacted by the Event.
  - **Safety**: Event that can lead to loss or harm in the environment in the context of (worst-case) environmental conditions.
  - **Functional**: Event that can lead to an undesired deviation in the expected system service.
  - **Performance**: Event that can lead to an undesired deviation in the expected quality of a system service.
  - **Security**: Event that can compromise availability, integrity, and confidentiality, in the context of an environmental condition and system usage.
  - **Reliability**: Event that can diminish or eliminate the continuity of correct system service.
  - **Availability**: Event that can diminish or eliminate the readiness for correct system service.
  - **Maintainability**: Event that can diminish the capacity for modification and repair.
- **Associated Argument** is the Argument Structure that is associated to the Event.
- **Edit Hyperlinks** manages the hyperlinks for an Event Node.
  - **Add…**: adds a hyperlink to an Event Node.
  - **Edit…**:edits the location path for the selected hyperlink.
  - **Remove**: deletes the selected hyperlink.
  - **View…**: opens the folder location for the hyperlinked file.
- **Initial Likelihood Value** is a value between 0 to 1 that is related to the Initial Likelihood the Event will occur.
- **Initial Likelihood** is the initial level of probability that the Event will occur. AdvoCATE has five Likelihood classification levels based on the FAA's SMS manual.
  - **Frequent**: expected to occur routinely.
  - **Probable**: expected to occur often.
  - **Remote**: expected to occur infrequently.
  - **Extremely Remote**: expected to occur rarely.
  - **Extremely Improbable**: unlikely to occur, but not impossible.
- **Initial Severity** is the initial negative impact level the Event can cause to its surrounding environment and other systems. The Initial Severity provides a base metric of the Event's negative impact. AdvoCATE has five Severity classification levels.
  - **Minimal**: Negligible safety effect.
  - **Minor**: Physical discomfort to other systems or environment. Slight damage to system.
  - **Major**: Physical distress or injuries other systems or environment. Substantial damage to system.
  - **Hazardous**: Multiple serious injuries; fatal injury to a relatively small number other systems; or a system loss without fatalities.

  - **Catastrophic**: Multiple fatalities, usually with the loss of the system.
- **Initial Risk Level** is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Event. The Initial Risk Level provides a base metric of the extent of possible loss for a specific Event. AdvoCATE has three Risk classification levels.
  - **Low**: Severity and Likelihood map to the green cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without restriction or limitation; Event is not required to be actively managed.
  - **Medium**: Severity and Likelihood map to the yellow cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk is acceptable without additional mitigation; however, tracking and monitoring are required.
  - **High**: Severity and Likelihood map to the red cells in the Risk Matrices in the Safety Architecture Analytics Dashboard. This safety risk requires mitigation, tracking, and monitoring.
- **Residual Likelihood Value** is a value between 0 to 1 that is related to the Residual Likelihood the Event will occur.
- **Residual Likelihood** is the level of probability that the Event will occur after new mitigation mechanisms have been established. The Residual Likelihood provides a new metric of the Event's probability given the new levels of integrity for your system's various **Barriers** and **Controls**.
- **Residual Severity** is the new negative impact level the Event can cause to its surrounding environment and other systems once new mitigation mechanisms have been established. The Residual Severity provides a new metric of the Event's negative impact.
- **Residual Risk Level** is the expression of possible loss in terms of Likelihood (probability) and Severity (impact) of the Event. The Residual Risk Level provides a new metric of the extent of possible loss for a specific Event once new mitigation mechanisms have been established.
- **Depth** is the maximum width (i.e., length of a path in from the top), in any (Depth) BTD for which that Event Instance is the Top Event.

Create a Node for a New BTD Threat

To create a node for a new BTD Threat

1. Open the **BTD Diagram Editor**.
2. Select the E **New Event** icon in the **BTD Event Palette**.
3. Click the **Canvas** on the **left side** of the **Top Event** to create new Threat Node.

The new BTD Threat Node will generate with placeholder text and be viewable in the **BTD Diagram Editor**.

Create a Node for an Existing BTD Threat

To create a node for an existing BTD Threat

1. Open the **CES Diagram Editor**.
2. Select the E **Existing Event** icon in the **BTD Event Palette**.
4. Click the **Canvas** on the **left side** of the **Top Event** to create a Threat Node from an existing Event.
3. From the pop-up window, select the **Existing Event** you wish to create a Threat Node for.
4. Click **Ok**.

The new BTD Threat Node will generate with the existing Event's details and be viewable in the **BTD Diagram Editor**.

Create a Node for a New BTD Consequence

To create a node for a new BTD Consequence

1. Open the **BTD Diagram Editor**.
2. Select the E **New Event** icon in the **BTD Event Palette**.
3. Click the **Canvas** on the **right side** of the **Top Event** to create new Consequence Node.

The new BTD Consequence Node will generate with placeholder text and be viewable in the **BTD Diagram Editor**.

Create a Node for an Existing BTD Consequence

To create a node for an existing BTD Consequence

1. Open the **CES Diagram Editor**.
2. Select the E **Existing Event** icon in the **BTD Event Palette**.
5. Click the **Canvas** on the **right side** of the **Top Event** to create a Consequence Node from an existing Event.
3. From the pop-up window, select the **Existing Event** you wish to create a Consequence Node for.
4. Click **Ok**.

The new BTD Consequence Node will generate with the existing Event's details and be viewable in the **BTD Diagram Editor**.

Create a Node a New BTD Escalation Factor

To create a node for a new BTD Escalation Factor

1. Open the **BTD Diagram Editor**.
2. Select the E **New Event** icon in the **BTD Event Palette**.
3. In the **Canvas**, click the **Control Node** that the Escalation Factor will weaken or compromise.

The new BTD Escalation Factor Node will generate with placeholder text and be viewable in the **BTD Diagram Editor**.

Create a Node for an Existing BTD Escalation Factor

To create a node for an existing BTD Escalation Factor

1. Open the **CES Diagram Editor**.
2. Select the E **Existing Event** icon in the **BTD Event Palette**.
3. In the **Canvas**, click the **Control Node** that the Escalation Factor will weaken or compromise.
4. From the pop-up window, select the **Existing Event** you wish to create an Escalation Factor Node for.
5. Click **Ok**.

The new BTD Escalation Factor Node will generate with the existing Event's details and be viewable in the **BTD Diagram Editor**.

Edit a BTD Event Node

To edit a BTD Event Node

1. Open the **BTD Diagram Editor**.
2. Select the **Event Node** you wish to edit.
3. In the **Properties View**, enter your edit/s into the corresponding fields.
4. Click outside the **Event Node** you edited to update your BTD Event Node.

The Event Node in the **BTD Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node's displayed contents will not update until after the Description field has been deselected.

Delete a BTD Event Node

To delete a BTD Event Node

1. Open the **BTD Diagram Editor**.
2. Select the **Event Node** you wish to delete.
3. Click the ✖ **Delete** icon in BTD Toolbar.

Your selected Event Node will be deleted from the **BTD Diagram Editor**.

BTD Mitigation Node Types

BTDs are made up of four **BTD Mitigation Node Types** (*Figure 32*):

- **Barriers** render as the yellow label for rounded, white Control Nodes and are the high-level or generic Mitigations for Top Events, Threats, and Consequences.
  - **Escalation Factor Barriers** render as the yellow label for rounded, yellow Escalation Factor Control Nodes and are high-level or generic Mitigations for Escalation Factors.
- **Controls** render as rounded, white nodes with yellow Barrier labels and are low-level or specific Mitigations that implement or form a Barrier.
  - **Escalation Factor Controls** render as rounded, yellow nodes with yellow Escalation Factor Barrier labels and are low-level or specific Mitigations that implement or form an Escalation Factor Barrier.

When you update or change Mitigation associations in the BTD, the corresponding **Hazard Log** and **CES** will automatically update.

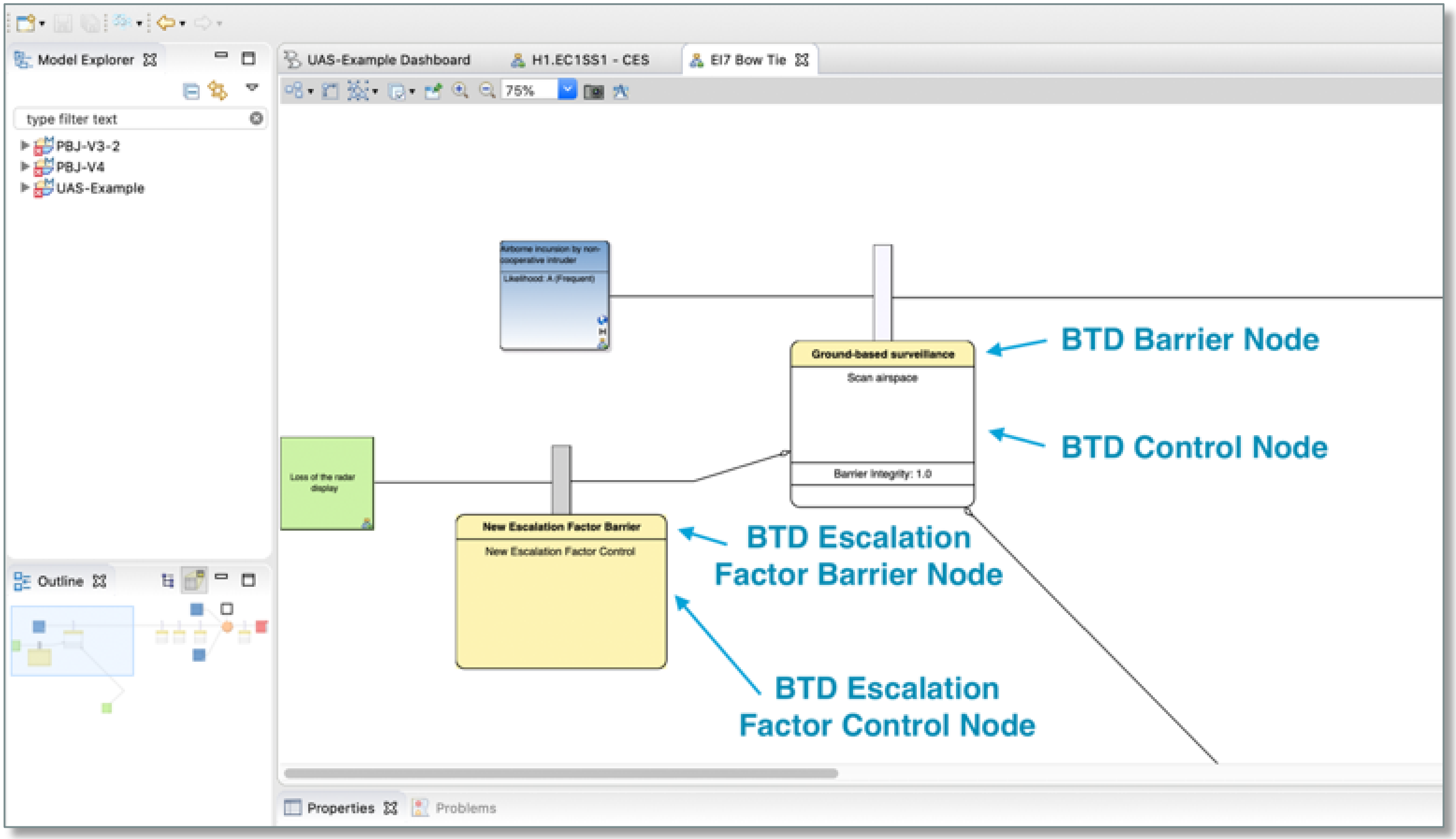

Model Explorer
type filter text
PBJ-V3-2
PBJ-V4
UAS-Example
Outline
UAS-Example Dashboard
H1.EC1SS1 - CES
EI7 Bow Tie
75%
Airborne incursion by non-cooperative intruder
Likelihood: A (Frequent)
Ground-based surveillance
Scan airspace
Barrier Integrity: 1.0
Loss of the radar display
New Escalation Factor Barrier
New Escalation Factor Control
BTD Barrier Node
BTD Control Node
BTD Escalation Factor Barrier Node
BTD Escalation Factor Control Node
Properties
Problems

Figure 32: BTD Barriers render as the yellow label for rounded, white BTD Control nodes. BTD Escalation Factor Barriers render as the yellow label for rounded, yellow Escalation Factor Control nodes.

BTD Barrier Properties

In the **Properties View**, Barrier Nodes contain the following properties:

- **Barrier** is the underlying Barrier the Barrier Instance corresponds to.
- **Name** is the identification marker for the Barrier Instance.
- **Description** is a statement that describes the Barrier.
- **Edit Hyperlinks** manages the hyperlinks for a Barrier Node.
  - **Add…**: adds a hyperlink to a Barrier Node.
  - **Edit…**: edits the location path for the selected hyperlink.
  - **Remove**: deletes the selected hyperlink.
  - **View…**: opens the folder location for the hyperlinked file.
- **Barrier Integrity** is a value between 0 to 1 that is related to the likelihood of a dangerous breach of the Barrier.
- **New Mitigation** classifies a Mitigation as a New Mitigation instead of an Existing Mitigation. Existing Mitigations are previously established mitigations. When safety analysts analyze a Mitigation to determine the Initial Risk Level and that Mitigation is deemed unacceptable, you can propose New Mitigations to lower the risk. By modeling both systems in one AdvoCATE Project, you can compute the **Initial** and **Residual Risk Levels** for a Mitigation.
- **Mitigation Type** is the classification corresponding to the order of precedence of a system's safety design. AdvoCATE supports five Mitigation Types.
  - **Design Modification**: risk reduction through system redesign.
  - **Safety Feature**: risk reduction through the incorporation of safety features.
  - **Safety Device**: reduction through the incorporation of safety devices.
  - **Warning Device**: risk reduction through the incorporation of warning devices.
  - **Procedures and Training**: risk reduction through the incorporation of procedures and training.
- **Barrier Associated Requirements** are the Mitigation Requirements that describe how the Barrier will reduce the risk of BTD Events. To see a full list of Mitigation Requirements for your AdvoCATE project, see the *Requirements Log* section.
- **Barrier Functional Allocations** are the Functions performed or used by the Barrier. To allocate a Function to a Barrier, you must first create and populate the **FuncSpec** file. For more information about the FuncSpec file, see the *Functional Specifications* section.
- **Barrier Physical Allocations** are the Components used by the Barrier. To allocate a Component to a Barrier, you must first create and populate the **PhysArch** file. For more information about PhysArch file, see the *Physical Architecture* section.

When you change BTD Barrier Properties, the corresponding **CESs** will automatically update.

BTD Control Properties

In the **Properties View**, Control Nodes contain the following properties:

- **Name** is the identification marker for Control Instance.
- **Description** is a statement that describes the Control.
- **Control Associated Argument** is the Argument Structure that justifies a dependability claim about the Control.
- **Edit Hyperlinks** manages the hyperlinks for a Control Node.
    - **Add…**: adds a hyperlink to a Control Node.
    - **Edit…**: edits the location path for the selected hyperlink.
    - **Remove**: deletes the selected hyperlink.
    - **View…**: opens the folder location for the hyperlinked file.
- **Control Integrity** is a value between 0 to 1 that is related to the likelihood of a dangerous breach of the Control.
- **Control Associated Requirements** are the Mitigation Requirements that describe how the Control will reduce the risk of CES Events. To see a full list of Mitigation Requirements for your AdvoCATE Project, see the *Requirements Log* section.
- **Control Functional Allocations** are the Functions performed or used by the Control. To allocate a Function to a Control, you must first create and populate the **FuncSpec** file. For more information about the FuncSpec file, see the *Functional Specifications* section.
- **Control Physical Allocations** are the Components used by the Control. To allocate a Component to a Control, you must first create and populate the **PhysArch** file. For more information about PhysArch file, see the *Physical Architecture* section.

When you change BTD Control Properties, the corresponding **CESs** will automatically update.

Create a Node for a New BTD Control

To create a node for a new BTD Control

1. Open the **BTD Diagram Editor**.
2. Select the **New Control** icon in the **BTD Control Palette**.
3. In the **Canvas**, select the **BTD Link** between two Event Nodes where you want to create a node for a New Control.

The new BTD Control Node will generate with placeholder text and be viewable in the **BTD Diagram Editor**.

Create a Node for an Existing BTD Control

To create a node for an existing BTD Control

1. Open the **BTD Diagram Editor**.
2. Select the **Existing Control** icon in the **BTD Control Palette**.
3. In the **Canvas**, select the **BTD Link** between two Event Nodes where you want to create a node for an Existing Control.
4. From the pop-up window, select the **Existing Control** you wish to create a node for.
5. Click **Ok**.

The new BTD Control Node will generate with the existing Control's details and be viewable in the **BTD Diagram Editor**.

Create a Node for a New BTD Escalation Factor Control

To create a node for a new BTD Escalation Factor Control

1. Open the **BTD Diagram Editor**.
2. Select the **New Control** icon in the **BTD Control Palette**.
3. In the **Canvas**, select the **BTD Link** between an Escalation Factor Node and a Control Node where you want to create a node for a New Escalation Factor Control.

The new BTD Escalation Factor Control Node will generate with placeholder text and be viewable in the **BTD Diagram Editor**.

Create a Node for an Existing BTD Escalation Factor Control

To create a node for an existing BTD Escalation Factor Control

1. Open the **BTD Diagram Editor**.
2. Select the **Existing Control** icon in the **BTD Control Palette**.
3. In the **Canvas**, select the **BTD Link** between an Escalation Factor Node and a Control Node where you want to create a node for an Existing Control.
4. From the pop-up window, select the **Existing Control** you wish to create a node for.
5. Click **Ok**.

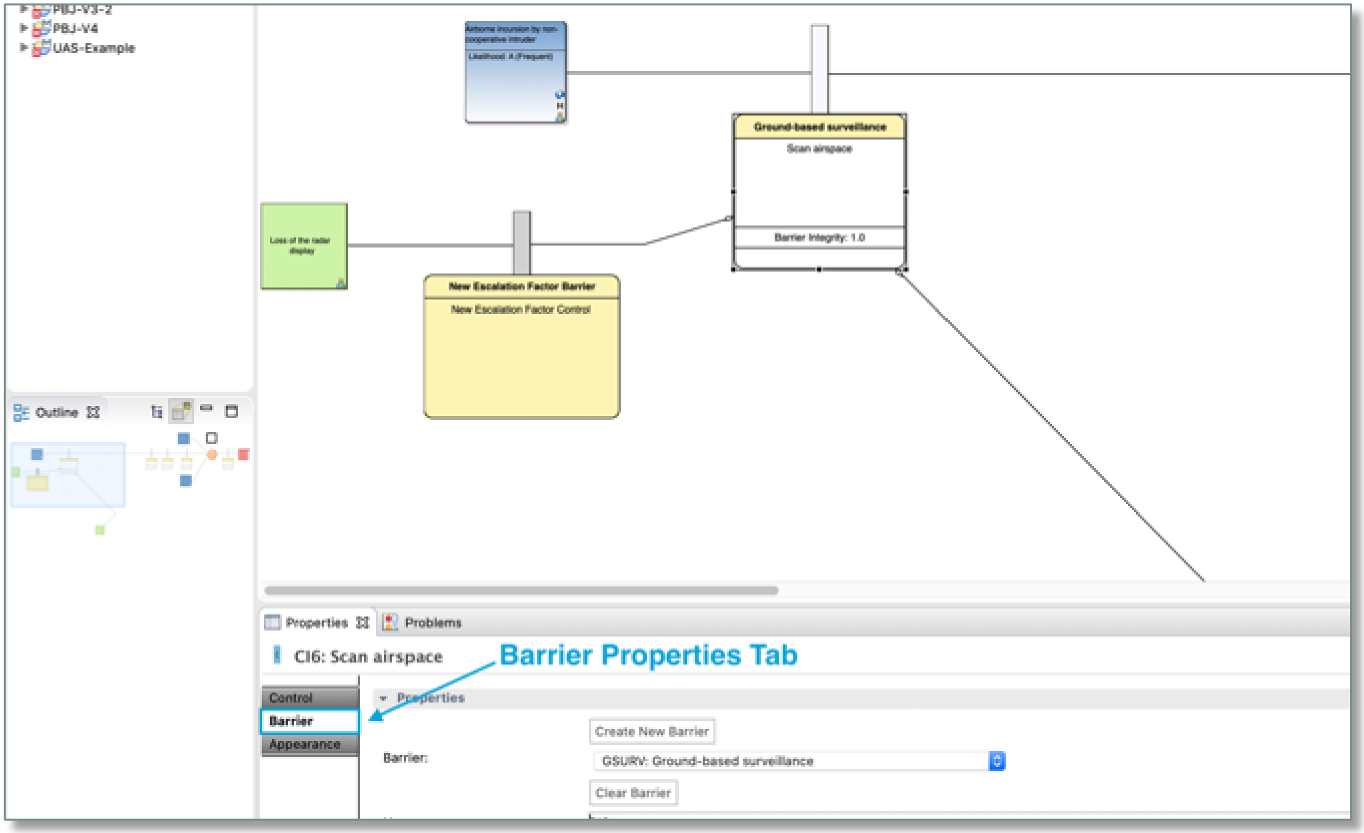
The new BTD Escalation Factor Control Node will generate with the existing Control's details and be viewable in the **BTD Diagram Editor**.

Add a New BTD Barrier to a Control Node

To add a new BTD Barrier to a Control Node

1. Open the **BTD Diagram Editor**.
2. Select the **Control Node** you wish to add a new **Barrier Node** to.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 33*).
4. Click the **Create New Barrier** button.

The new BTD Barrier Node label will generate above the Control with placeholder text and will be viewable in the **BTD Diagram Editor**.



Barrier: GSURV: Ground-based surveillance

Figure 33: In the BTD Diagram Editor, the Barrier Properties Tab is located within the Properties View.

Add an Existing BTD Barrier to a Control Node

To add an existing BTD Barrier to a Control Node

1. Open the **BTD Diagram Editor**.
2. Select the **Control Node** you wish to add an existing **Barrier Node** to.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 33*).
4. Select the **drop-down arrows** next to the **Barrier** field.
5. Select the **existing Barrier** you wish to add to the Control.

The existing BTD Barrier Node label will generate above the Control and will be viewable in the **BTD Diagram Editor**.

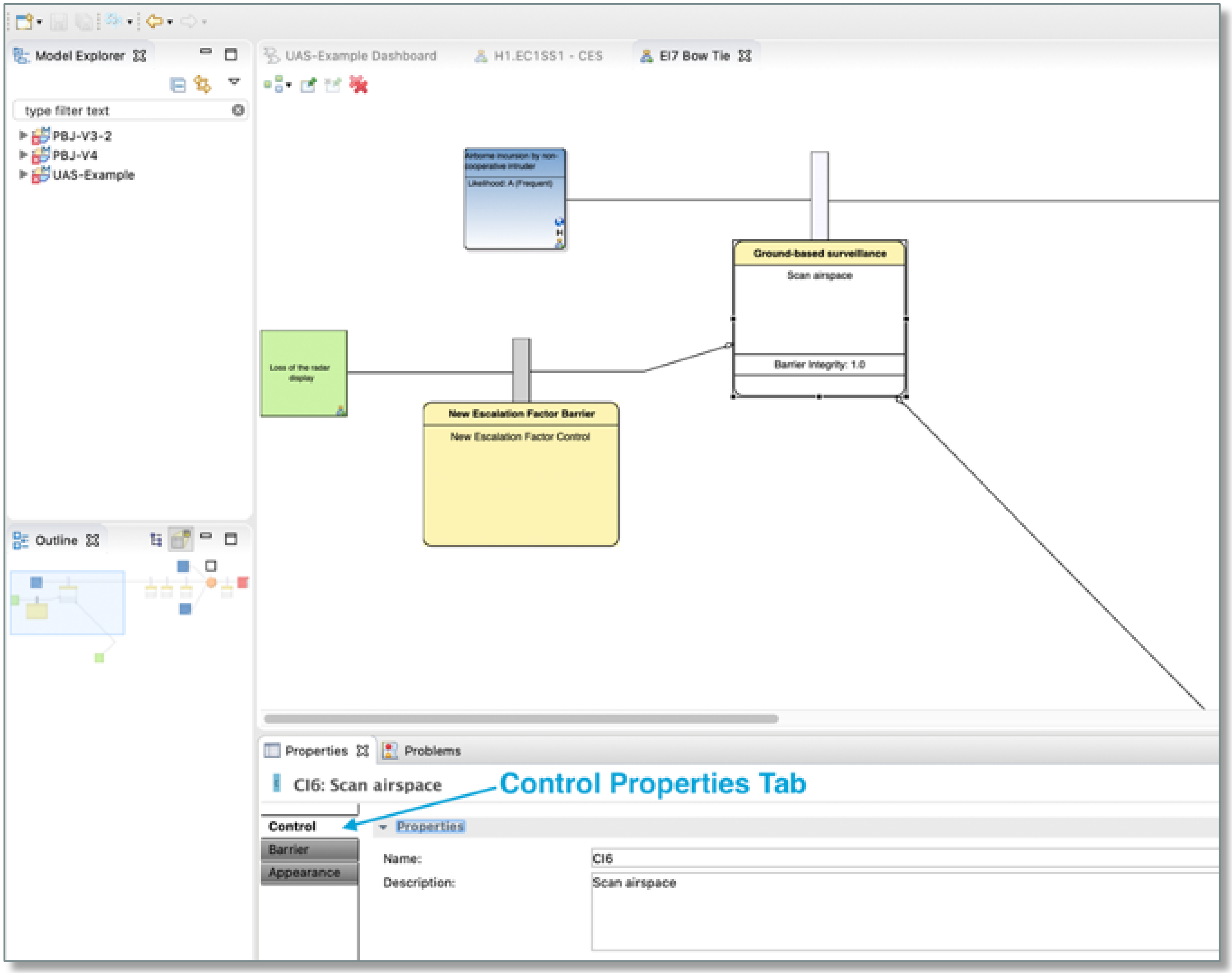
Edit a BTD Control Node

To edit a BTD Control Node

1. Open the **BTD Diagram Editor**.
2. Select the **Control Node** you wish to edit.
3. In the **Properties View**, select the **Control Properties** tab (*Figure 34*).
4. Enter your edit/s into the corresponding fields.
5. Click outside the **Control Node** you edited to update your BTD Control Node.

The Control Node in the **BTD Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node’s displayed contents will not update until after the Description field has been deselected.



Control Properties Tab

CI6: Scan airspace

Properties

Name: CI6

Description: Scan airspace

Figure 34: In the BTD Diagram Editor, the Control Properties Tab is located within the Properties View.

Edit a BTD Barrier Node

To edit a BTD Barrier Node

1. Open the **BTD Diagram Editor**.
2. Select the **Control Node** with the **Barrier Node label** you wish to edit.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 33*).
4. Enter your edit/s into the corresponding fields.
5. Click outside the **Control Node** you edited to update your BTD Barrier Node.

The Barrier Node label in the **BTD Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node’s displayed contents will not update until after the Description field has been deselected.

Remove a BTD Barrier Node from a Control

To remove a BTD Barrier Node from a Control

1. Open the **BTD Diagram Editor**.
2. Select the **Control Node** you wish to remove a Barrier from.
3. In the **Properties View**, select the **Barrier Properties** tab (*Figure 33*).
4. Click the **Clear Barrier** button.

The Barrier Node label will be removed from the selected Control.

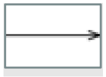
Delete an BTD Control Node

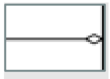
To delete a BTD Control Node

1. Open the **BTD Diagram Editor**.
2. Select the **Control Node** you wish to delete.
3. Click the ✖ **Delete** icon in BTD Toolbar.

Your selected Control Node will be deleted from the **BTD Diagram Editor**.

BTD Link Types

Links create references between source and target nodes. In the BTD Diagram Editor, Links are generally depicted as a  solid black line with an arrow between nodes.

However, when Escalation Factors are present, **Escalation Links** are depicted as a  solid black line with a diamond.

Safety Architecture Views

Safety Architecture Views are the multiple view configurations within the Safety Architecture. Safety Architecture Views relate to certain steps or review processes in assurance case development.

AdvoCATE provides the following Safety Architecture Views:

- **Barrier Slice Diagram**: This diagram presents the Controls associated with a given Barrier alongside the Threats and Consequences that they mitigate. It can be considered as a high-level specification of the functionality to be delivered by a given Barrier, and it is relevant for those stakeholders who are responsible for (implementing or executing) the Barrier and its respective Controls.
- **Event Slice Diagram**: This diagram presents all the Hazards pertinent to a given Event, e.g., where there is a Hazard release, along with all the Threats and Consequences of that Event within the relevant situation. This diagram can be useful during the development and assessment of the overall risk analysis in focusing on a specific, high-priority/critical Event that occurs within different operating situations.
- **Closed Barrier View**: This view abstracts the details of the Controls and only presents the different Barriers applicable for risk management in a specific scenario. This view is useful in concisely communicating the risk reduction measures that were undertaken, thus providing assurance of operational safety in the presence of new surveillance and avoidance capabilities. This view also provides the basis for a rapid risk assessment.
- **Barrier Table**: This table lists all the Barriers within an AdvoCATE Project. This table is operationally useful and can help derive standard operating procedures, checklists of tasks, etc.
- **Control Table**: This table lists all the Controls within an AdvoCATE Project. This table is operationally useful and can help derive standard operating procedures, checklists of tasks, etc.

Open a Barrier Slice Diagram

To open a Barrier Slice Diagram

1. Open the **Safety Architecture Dashboard**.
2. In the **Barriers And Controls** window, right-click the **Barrier** you wish to open as a Barrier Slice Diagram.
3. From the right-click menu, select **Open Barrier Slice Diagram**.

The **Barrier Slice Diagram** will open for the selected Barrier.

NOTE: If the diagram layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the Toolbar. The nodes and links within the diagram will rearrange with equal spacing and alignment.

Open an Event Slice Diagram

To open an Event Slice Diagram

1. Open the **Safety Architecture Dashboard**.
2. In the **Events** window, right-click the **Event** you wish to open as an Event Slice Diagram.
3. From the right-click menu, select **Open Event Slice Diagram**.

The **Event Slice Diagram** will open for the selected Event.

NOTE: If the diagram layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the Toolbar. The nodes and links within the diagram will rearrange with equal spacing and alignment.

Open the Closed Barrier View

To open the Closed Barrier View

1. Open the **Safety Architecture Dashboard**.
2. In the **Hazardous Activities** window, select the **drop-down arrow** next to the CES's corresponding **Hazardous Activity**.
3. From the drop-down list, select the **CES** you wish to open the Closed Barrier View for.
4. In the **Events Associated with Controlled Event Structures** window, select the **drop-down arrow** next to the **Hazardous Activity** you want to open the Closed Barrier View for (*Figure 31*).
5. Under the Hazardous Activity, select the **Event** you wish to open the Closed Barrier View for.
6. Once your Event is selected, click **Open Closed Barrier View**.

The **Closed Barrier View** will open for the selected Event.

NOTE: If the diagram layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the Toolbar. The nodes and links within the diagram will rearrange with equal spacing and alignment.

Open the Barrier Table

To open the Barrier Table

1. Open the **Safety Architecture Dashboard**.
2. In the **Barriers And Controls** window, right-click a **Barrier** or **Control**.
3. From the right-click menu, select **Open Barrier Table**.

The **Barrier Table** will open.

Open the Control Table

To open the Control Table

1. Open the **Safety Architecture Dashboard**.
2. In the **Barriers And Controls** window, right-click a **Barrier** or **Control**.
3. From the right-click menu, select **Open Control Table**.

The **Control Table** will open.

Argument Structures

Argument Structures use the Goal Structuring Notation (GSN) to graphically communicate how a system is acceptably safe to operate.

Each AdvoCATE Project can contain multiple Argument Structures.

NOTE: This user guide assumes you know how to use GSN. For more information about GSN, see the *Guide References* section.

Create a New Argument Structure

To create a new Argument Structure

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Argument Structure**.
3. Select the **AdvoCATE Project folder** that will contain the new Argument Structure.
4. Name the Argument Structure.

   NOTE: An Argument Structure must end in the extension **.argument**.

5. Click **Finish**.

The new Argument Structure opens in the **Argument Diagram Editor**.

Import an External Argument Structure

To import an existing Argument Structure into an AdvoCATE Project

1. In the **Model Explorer View**, right-click your **AdvoCATE Project** file.
2. From the right-click menu, select **Import…**.
3. In the Import window, toggle the **drop-down arrow** for the AdvoCATE folder.
4. In the drop-down menu for the AdvoCATE folder, select **Argument**.
5. Click **Next**.
6. Select the **Browse…** button to locate your existing Argument Structure.
7. Click **Open**.
8. Click **Finish**.

Your corresponding Argument Structure will import and open in the **Argument Diagram Editor**.

Argument Diagram Editor

The **Argument Diagram Editor** allows users to create various **GSN Nodes** in a logical progression. The Argument Diagram Editor has its own **Toolbar**, **Palette**, and **Core GSN** icons.

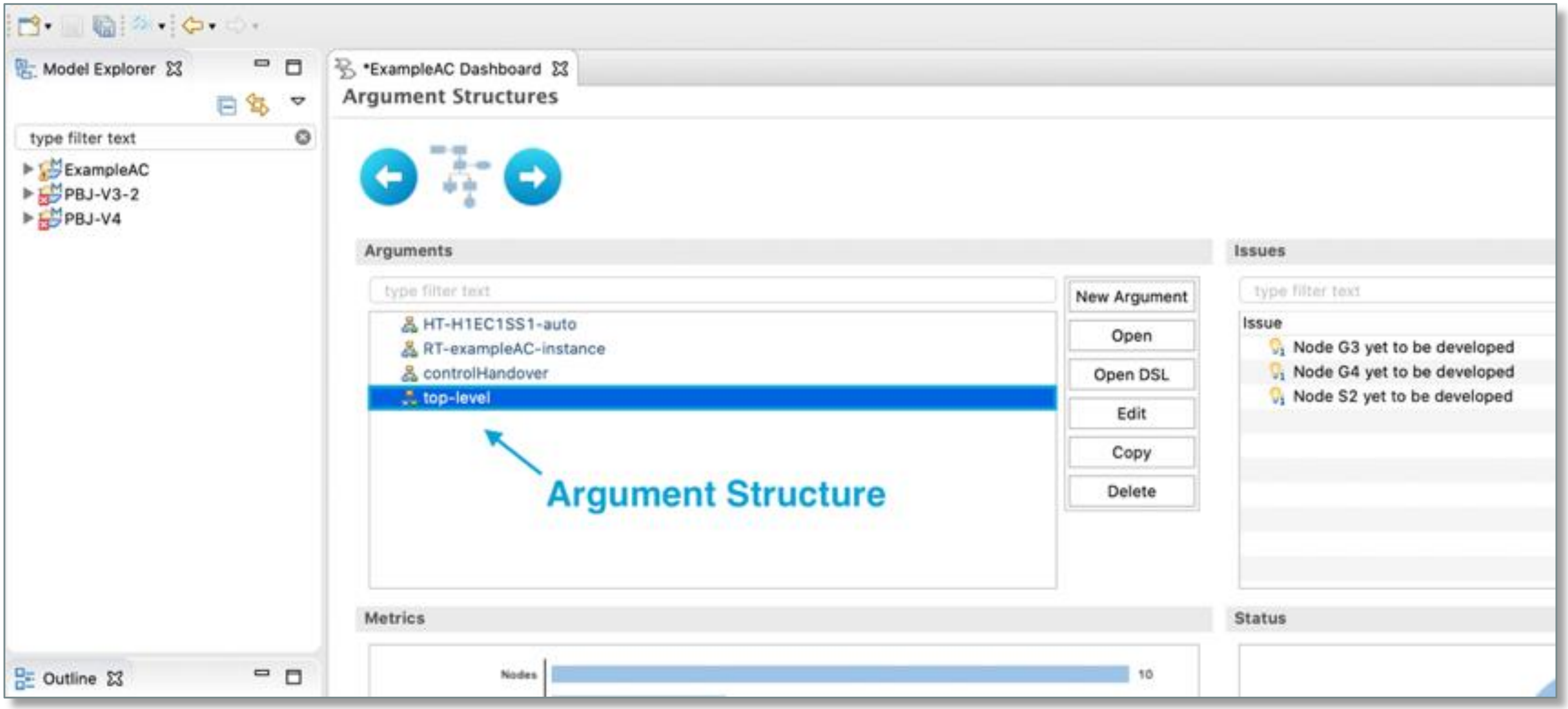
Open an Argument Structure

To open an Argument Structure

1. Open the **Arguments Dashboard**.
2. In the **Arguments** window, select the **Argument Structure** you wish to open (*Figure 35*).
3. Click **Open**.

The Argument Structure will open in the **Argument Diagram Editor**.

NOTE: If the diagram layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the Toolbar. The nodes and links within the diagram will rearrange with equal spacing and alignment.



- HT-H1EC1SS1-auto
- RT-exampleAC-instance
- controlHandover
- top-level

Argument Structure

Issue
Node G3 yet to be developed
Node G4 yet to be developed
Node S2 yet to be developed

Figure 35: In the Argument Dashboard, Argument Structures for an AdvoCATE Project are listed in the Arguments window.

Argument Toolbar

The **Argument Toolbar** is in the top-left corner of the Argument Diagram Editor.

The Argument Toolbar has the following tools:

- **Arrange All**: Arranges all Nodes with equal spacing and alignment.
- **Autosize all nodes to fit text**: Resizes all Nodes to display all their Description text.
- **Select All**: Selects everything in the Argument Diagram.
  - **Select All Connectors**: Selects all Links in the Argument Diagram.
  - **Select All Shapes**: Selects all Nodes in the Argument Diagram.
- **Layers**: Shows/Hides different Layers in the Argument Diagram.
- **Pin/Unpin**: Opens the Pinned Elements Wizard.
- **Zoom In**: Zooms in on the Diagram Canvas.
- **Zoom Out**: Zooms out on the Diagram Canvas.
- **Zoom Percentage**: Changes the visible percentage of the Diagram Canvas.
- **Export diagram as image**: Exports the Diagram Canvas as a JPG, PNG, SVG, or PDF.
- **Edit all node fonts**: Opens Font Selection Window to change the font for an entire Argument Structure.
- **Pin selected elements**: Pins selected element/s, so they cannot be moved.
- **Unpin selected elements**: Unpins selected element/s, so they can be moved.
- **Delete**: Deletes selected element/s.

Argument Palette

The **Argument Palette** is in the top-right corner of the Argument Diagram Editor.

The Argument Palette has the following tools:

- **Select**: Selects elements on Diagram Canvas.
- **Zoom In**: Zooms in on the selected portion of the Diagram Canvas.
- **Zoom Out**: Zooms out of the selected portion of the Diagram Canvas.
- **Note**: Creates a Note Node.
    - **Representation Link**: Creates a Representation Node of select AdvoCATE graphs, diagrams, tables, or views.
    - **Text**: Creates a Text Node.
    - **Note Attachment**: Creates a Note Node and Link to a selected Canvas element.
- **Generic Connection Creation Tool**: Draws Links between different Nodes. Links automatically format based on connected Node types.
- **Pin**: Pins selected element/s, so they cannot be moved.
- **Unpin**: Unpins selected element/s, so they can be moved.

Argument Core GSN

The **Argument Core GSN** is under the **Argument Palette** and Default tools in the Argument Diagram Editor.

The Argument Core GSN has the following tools:

- **Assumption**: Creates an Assumption Node.
- **Context**: Creates a Context Node.
- **Solution**: Creates a Solution Node.
- **Goal**: Creates a Goal Node.
- **Justification**: Creates a Justification Node.
- **Strategy**: Creates a Strategy Node.
- **Is Supported By**: Creates an Is-Supported-By Link between Nodes.
- **In Context Of**: Creates an In-Context-Of Link between Nodes.

GSN Node Types

There are six **GSN Node Types**:

- **Assumption Node**: Renders as an oval with the letter “A” and presents an unsubstantiated statement.
- **Context Node**: Renders as a rounded rectangle and references contextual information and Evidence Artifacts.
- **Solution Node**: Renders as a circle and references Evidence Artifacts.
- **Goal Node**: Renders as a rectangle and presents a claim forming part of the argument. If the pathway below a Goal Node is not finished, users can indicate the “To Be Developed” status in the node’s Properties View.
- **Justification Node**: Renders as an oval with the letter “J” and presents a rationale statement.
- **Strategy Node**: Renders as a parallelogram and describes the inference between Goal and Sub-Goal/s. If the pathway below a Strategy Node is not finished, users can indicate the “To Be Developed” status in the node’s Properties View.

Create a GSN Node

To create a new GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **Node type** you want to create from the **Argument Core GSN**.
3. Click the **Canvas** where you want to create your new GSN Node.

The new GSN Node will generate and be viewable in the **Argument Diagram Editor**.

NOTE: Alternatively, you can use *GSN Node Keyboard Shortcuts* to create the Argument Structure rapidly.

Use GSN Node Keyboard Shortcuts

To use GSN Node Keyboard Shortcuts

1. Open the **Argument Diagram Editor**.
2. Select the **Canvas**.
3. Use the following **shortcuts** to place node/s on the Canvas:

Node	Mac	Windows / Linux
Assumption	Option A	Alt A
Context	Option C	Alt C
Goal	Option G	Alt G
Justification	Option J	Alt J
Solution	Option E	Alt E
Strategy	Option S	Alt S

Each new node created with a shortcut will be automatically linked to the currently selected node. To select a different Node, use the mouse pointer or the **Tab**/**Shift Tab** keys to cycle between the Nodes.

If no node is selected while creating a new node using a shortcut, the new node will be created without any links.

If the currently selected node cannot be connected to the node being added using a shortcut (e.g., Context to Context), then the new node will be created without any links.

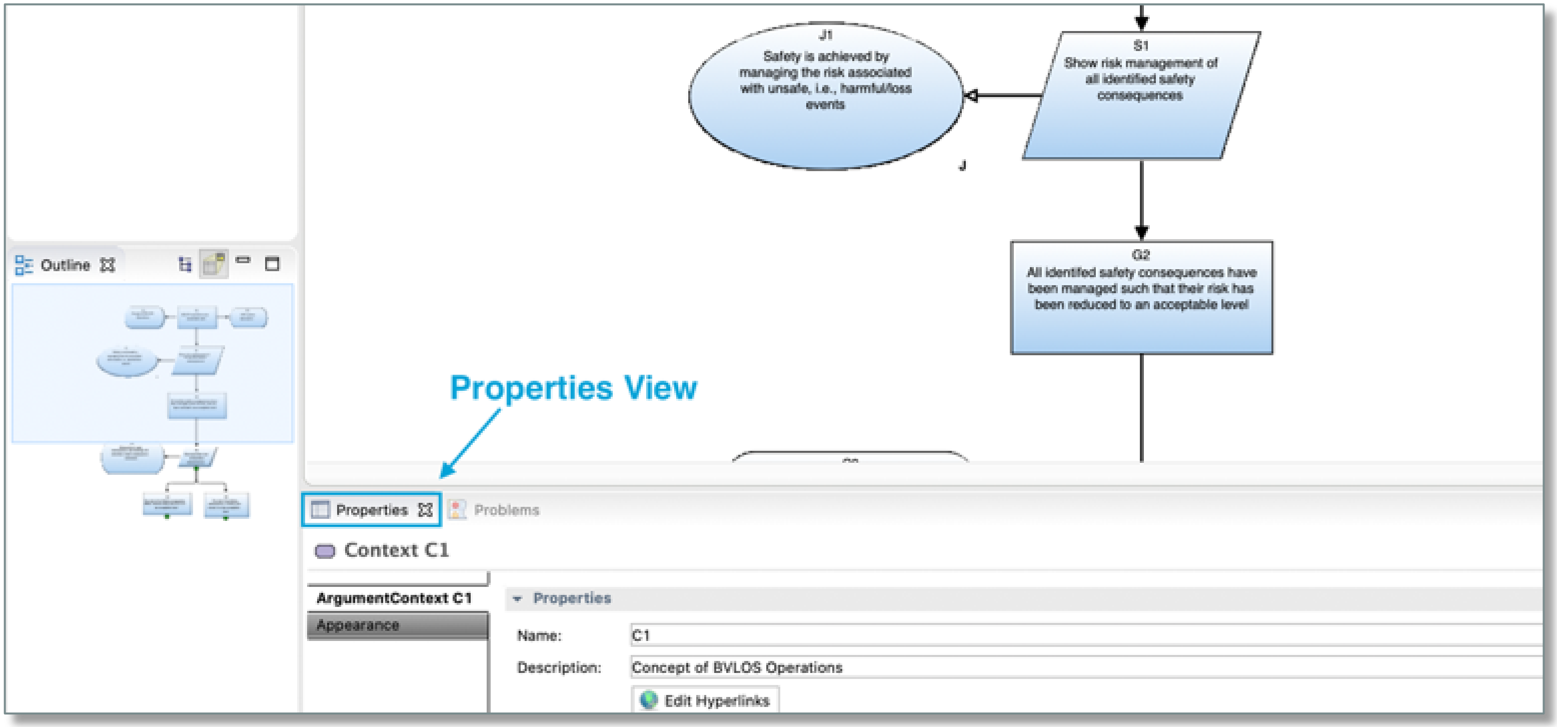
Edit a GSN Node

To edit a GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **Node** you wish to edit.
3. In the **Properties View**, enter your edit/s into the corresponding fields (*Figure 36*).
4. Click outside the **Node** you edited to update your GSN Node.

The GSN Node in the **Argument Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node's displayed contents will not update until after the Description field has been deselected.


J1
Safety is achieved by managing the risk associated with unsafe, i.e., harmful/loss events
J
S1
Show risk management of all identified safety consequences
G2
All identified safety consequences have been managed such that their risk has been reduced to an acceptable level
Outline
Properties View
Properties
Problems
ArgumentContext C1
Appearance
Properties
Edit Hyperlinks

Context C1

Name: C1

Description: Concept of BVLOS Operations

Figure 36: In the Argument Diagram Editor, use the Properties View to view and edit a selected GSN node's properties.

Delete a GSN Node

To delete a GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **Node** you wish to delete.
3. Click the **Delete** icon in Argument Toolbar.

Your selected GSN Node will be deleted from the **Argument Diagram Editor**.

GSN Nodes and Evidence

The following GSN Nodes can have associated Evidence:

- **Assumption Node**
- **Context Node**
- **Solution Node**
- **Justification Node**

Manage Associated Evidence

To manage associated Evidence to a GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **Node** you wish to manage.
3. In the **Properties View**, select the **Manage Associated Evidence** button.
4. In the pop-up window, toggle the **checkbox** for the corresponding **Evidence Artifact** you wish to associate or disassociate from the selected Node.
5. Click **Ok**.

The GSN Node’s Properties will update to reflect the change.

Create a New Evidence Artifact for a GSN Node

To create a new Evidence Artifact for a GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **Node** you wish to edit.
3. In the **Properties View**, select the **Create New Evidence Artifact** button.
4. In the pop-up window, enter the new **Evidence Artifact Description**.
5. Click **Ok**.

A new Evidence Artifact entry will be created and associated to your selected GSN node.

Open Selected Evidence in DSL

To open an Evidence Artifact DSL entry associated to a GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **Node** you wish to edit.
3. In the **Properties View** under Evidence, select the associated **Evidence Artifact** you wish to open.
4. Select the **Open Selected Evidence in DSL** button.

The **Evidence Log** will open, and the selected **Evidence Artifact entry** will be highlighted.

NOTE: For more information about the **Evidence Log**, see the *Evidence Log* section.

GSN Link Types

There are two **GSN Link Types**:

- **Is-Supported-By Link**: Renders as a line with a solid arrowhead and allows support relationships between nodes. Permitted between Goal-to-Goal, Goal-to-Strategy, Goal-to-Solution, Strategy-to-Goal, Strategy-to-Solution, and Strategy-to-Strategy nodes.
- **In-Context-Of Link**: Renders as a line with a hollow arrowhead and declares a contextual relationship between nodes. Permitted between Goal-to-Context, Goal-to-Assumption, Goal-to-Justification, Strategy-to-Context, Strategy-to-Assumption, Strategy-to-Justification, Solution-to-Context, Solution-to-Assumption, and Solution-to-Justification nodes.

Create a GSN Link

To create a new GSN Link

1. Open the **Argument Diagram Editor**.
2. Select the **Generic Connection Creation Tool**.
3. Select the **parent node** on the **Canvas** where you want the link to start.
4. Select the **child node** on the **Canvas** where you want the link to end.

A new GSN Link will generate based on the node types selected.

Delete a GSN Link

To delete a GSN Link

1. Open the **Argument Diagram Editor**.
2. Select the **Link** you wish to delete.
3. Click the **Delete** icon in Argument Toolbar.

Your selected GSN Link will be deleted from the **Argument Diagram Editor**.

Manage Hyperlinks for a GSN Node

To manage a hyperlink for a GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **Node** you wish to edit.
3. In the **Properties View**, click the **Edit Hyperlinks** button.
4. Based on your task, select the appropriate **button**:
    - **Add…**: adds a hyperlink to an **AdvoCATE Project**, **Workspace**, or **File System** file.
    - **Edit…**: edits the location path for the selected hyperlink.
    - **Remove**: deletes the selected hyperlink.
    - **View…**: opens the folder location for the hyperlinked file.
5. Click **Ok**.

The hyperlink for your selected GSN Node will update accordingly.

NOTE: Node hyperlinks are useful to store external references for an AdvoCATE Project.

Insert a Pattern Instance into an Argument Structure

To insert a Pattern Instance into an **Argument Structure**

1. Open the **Argument Diagram Editor**.
2. Right-click the **Node** you wish to edit.
3. From the right-click menu, select **Insert Pattern Instance** and where you would like the **root** created:
   - **at root of target argument**
   - **at root of instance argument**
4. From the pop-up window, select the **Argument Pattern** you wish to instantiate.
5. Click **Next**.
6. Select the **Data Tree** you wish to use to instantiate the Argument Pattern.
7. Click **Finish**.
8. If multiple data match the root of the Pattern, select the appropriate **data** to generate the Argument root.

The Pattern Instance will generate according to the Instance option selected.

NOTES:

- The **Insert Pattern Instance** option appears in the right-click menu for Goal, Strategy, and Solution nodes.
- The new Pattern Instance may generate on top of other nodes. Click the Canvas to deselect the Pattern Instance, and then select the **Arrange All** icon from the Argument Toolbar. The Argument Nodes will rearrange with equal spacing and alignment.

Insert an Argument Structure into an Argument Structure

To insert an Argument Structure into an **Argument Structure**

1. Open the **Argument Diagram Editor**.
2. Right-click the **Node** you wish to edit.
3. From the right-click menu, select **Insert Argument**.
4. From the pop-up window, select the **Argument Structure** you wish to add.
5. Click **Finish**.

The Argument Structure will generate below the selected Node.

NOTE: The new Argument Structure may generate on top of other nodes. Click the Canvas to deselect the new nodes, and then select the **Arrange All** icon from the Argument Toolbar. The Argument Nodes will rearrange with equal spacing and alignment.

Edit Associations for a GSN Node

To edit associations for a GSN Node

1. Open the **Argument Diagram Editor**.
2. Right-click the **Node** you wish to edit.
3. From the right-click menu, select **Edit Associations**.
4. From the pop-up window, toggle the **checkbox** for the corresponding **Safety Architecture Element**/s you wish to associate or disassociate from the selected Node.
5. Click **Ok**.

The GSN Node’s associations will update to reflect the change/s.

Argument Splitting

Argument Splitting is when you divide a full Argument Structure into smaller chunks at the node level.

When an Argument Structure is split, the original Argument Structure is left unchanged. Instead, a new **Split Argument Diagram** is created that copies the **Split Region**. The Split Region consists of the **Split Node**, the node at which the split was done, and all nodes either above or below that Split Node.

As you update Split Argument Diagrams, the original Argument Structure will update to reflect the change/s.

Split Nodes in the Split Argument Diagram will be annotated with green rectangular **Continued Elsewhere** and a **Developed From** annotation (*Figure 37*).

Argument Structures cannot use the **Split Below** option on **Context Nodes** or **Solution Nodes**, but they can be used on **Goal Nodes** and **Strategy Nodes** to develop an argument further.

NOTE: Split Arguments do not currently appear on the Argument Dashboard.

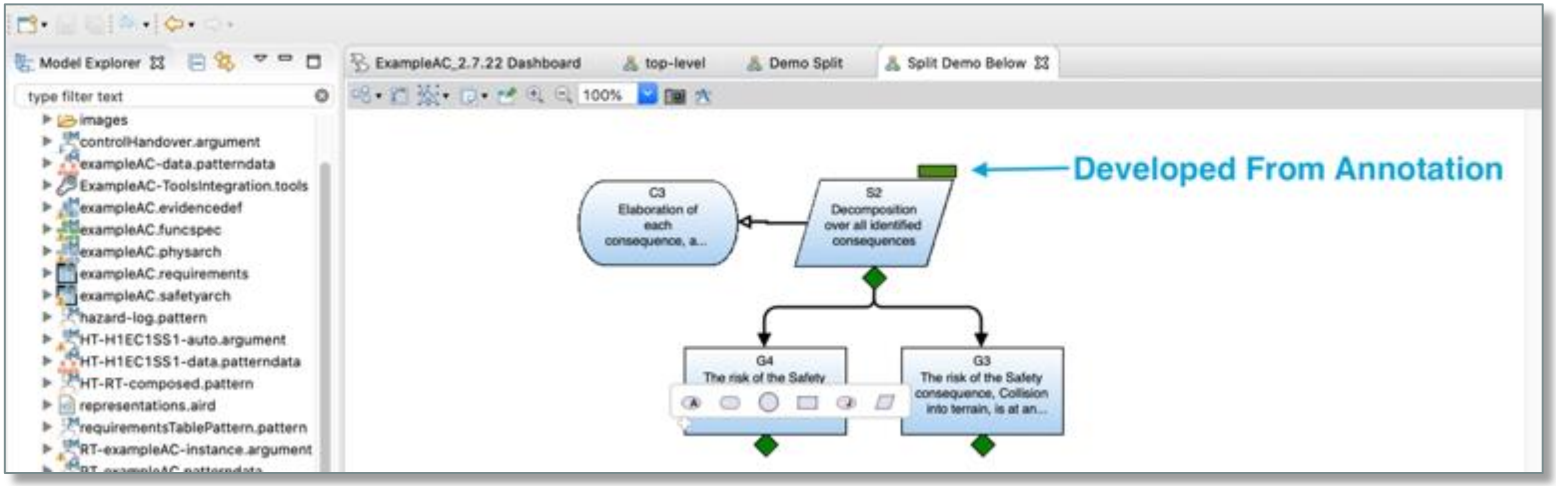

Model Explorer
type filter text
images
controlHandover.argument
exampleAC-data.patterndata
ExampleAC-ToolsIntegration.tools
exampleAC.evidencedef
exampleAC.funcspec
exampleAC.physarch
exampleAC.requirements
exampleAC.safetyarch
hazard-log.pattern
HT-H1EC1SS1-auto.argument
HT-H1EC1SS1-data.patterndata
HT-RT-composed.pattern
representations.aird
requirementsTablePattern.pattern
RT-exampleAC-instance.argument
ExampleAC_2.7.22 Dashboard
top-level
Demo Split
Split Demo Below
100%
Developed From Annotation
C3
Elaboration of each consequence, a...
S2
Decomposition over all identified consequences
G4
The risk of the Safety
G3
The risk of the Safety consequence, Collision into terrain, is at an...

Figure 37: Split Nodes in the Split Argument Diagram will be annotated with green rectangular Continued Elsewhere and a Developed From annotations.

Create a New Split Argument

To create a new Split Argument

1. Open the **Argument Diagram Editor**.
2. Right-click the **Node** you wish to make a Split Node.
3. From the right-click menu, select **Split Argument Diagram**.
4. From the sub-menu, select one of the following **options**:
   - **Split Argument Above**: The Split Region consists of the Split Node and all Nodes above the Split Node.
   - **Split Argument Below**: The Split Region consists of the Split Node and all Nodes below the Split Node.

A new **Split Argument Diagram** is created and opens in the **Diagram Editor**.

NOTES:

- Context Nodes that are Split Nodes will appear in all Split Regions that include that Split Node, whether added before or after splitting.
- New nodes created within a Split Region will automatically belong to all Split Regions that include their parent node, except if that parent node is a "split-below" Split Node for that Split Region. In that case, new supporting nodes added to a Split Node are not added to the currently edited Split Region but are instead added to the lower Split Argument.
- In each case, any nodes added to a Split Argument are also added to the full Argument Structure.

Delete a Split Argument Diagram

To delete a Split Argument Diagram

1. Right-click the **Split Argument Diagram** in the **Model Explorer View** you wish to delete.
2. From the right-click menu, select ✖ **Delete**.
3. Click **Ok** to confirm your deletion request.

The selected Split Argument Diagram will be deleted.

NOTE: Deleting a Split Argument Diagram does not delete that Split Region in the full Argument Structure.

Open a Split Argument Diagram

To open a Split Argument Diagram

1. Open the **Argument Diagram Editor**.
2. Right-click a **Node** in the Canvas.
3. In the drop-down menu, select **Split Argument Diagram** > **Show Split Region**.
4. From the pop-up window, select the **Split Region** you wish to open.

The **Split Argument Diagram** for the selected Split Region will open in the **Diagram Editor**.

Splits Views

The **Splits View** allows users to view the overall structure of an **Argument Structure** and see how individual **Split Argument Diagrams** within an Argument Structure relate to each other. Users can also use the Splits View to navigate to **Split Argument Diagrams** within an **Argument Structure**.

Once a Splits View is created, you can view and open it in the **Model Explorer**.

Create a New Splits View

To create a new Splits View

1. In the **Model Explorer**, use the **drop-down arrows** next to your **Argument Structure** that you wish to create and open in the Splits View.
2. Right-click the **Argument Diagram model object**.
3. In the right-click menu, select **New Representation** > **Argument Splits View**.
4. From the pop-up window, name the new **Splits View**.

The new **Splits View** will open in the **Splits View Diagram Editor**.

Argument Patterns

Argument Patterns are repeatedly used abstractions of successful Argument Structures within an assurance case. These Argument Patterns capture the reasoning that link certain (types of) claims to the available (types of) evidence and is accompanied by clear **Pattern Parameters**.

AdvoCATE provides two types of Pattern abstractions: **Structural Abstraction** and **Element Abstraction**.

Structural Abstraction applies to the GSN **Is-Supported-By** and **In-Context-Of** Links and is supported by the concepts of **Multiplicity** and **Choice**.

Multiplicity indicates an n-ary relation between GSN nodes.

- A multiplicity of many means 0 or more, and it is indicated as a ● filled circle on a Link and an annotation indicating the lower and upper Bounds of the multiplicity.
- Optional multiplicity means 0 or 1, and it is indicated as a ○ hollow circle on a Link.

Choice indicates alternatives in the relations between GSN nodes indicating a k-of-m choice. A Choice is indicated as a ◆ filled diamond with one source node and at least two or more target nodes. Choices are also annotated with Bounds to indicate how a Choice will be resolved during instantiation.

Multiplicity can be combined with Choices by placing a multiplicity symbol prior to the Choice. This action is equivalent to placing that multiplicity symbol on all the alternatives after the Choice.

Element Abstraction applies to the **Uninstantiated** and **Uninstantiated and Undeveloped** GSN notions.

Uninstantiated (UI) refers to abstract elements whose Pattern Parameters are replaced with concrete node instances upon Pattern Instantiation. A UI instance is indicated as a ▲ small green triangle attached to the base of a GSN node, which contains a Pattern Parameter.

Pattern Parameters are given in the form of {variable :: type} in the node description.

Uninstantiated and Undeveloped (UU) refers to UI entities that are also undeveloped. Thus, upon Pattern Instantiation, an abstract UU entity is replaced with a concrete, but undeveloped, node instance. A UU instance is indicated as a ◆ green diamond with a horizontal line, attached to the base of a Goal or Strategy Node, which contains a Pattern Parameter.

Each AdvoCATE Project can contain multiple Argument Patterns.

NOTE: This user guide assumes you know how to use GSN. For more information about GSN, see the *Guide References* section.

Create a New Argument Pattern

To create a new Argument Pattern

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Argument Pattern**.
3. Select the **AdvoCATE Project folder** that will contain the new Argument Pattern.
4. Name the Argument Pattern.

   NOTE: An Argument Pattern must end in the extension **.pattern**.

5. Click **Finish**.

The new Argument Pattern opens in the **Pattern Diagram Editor**.

Import an External Argument Pattern

To import an existing Argument Pattern into an AdvoCATE Project

1. In the **Model Explorer View**, right-click your **AdvoCATE Project** file.
2. From the right-click menu, select **Import…**.
3. In the Import window, toggle the **drop-down arrow** for the General folder.
4. In the drop-down menu for the General folder, select **File System**.
5. Click **Next**.
6. Select the **Browse…** button to locate the existing folder with your desired Argument Pattern.
7. Select the **folder**.
8. Click **Open**.
9. In the selected folder, toggle the **checkbox** for the Argument Pattern you wish to import.
10. Click **Finish**.

Your corresponding Argument Pattern will import.

NOTE: If the Argument Pattern layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the Toolbar. The nodes and links within the diagram will rearrange with equal spacing and alignment.

Pattern Diagram Editor

The **Pattern Diagram Editor** allows users to manually create Argument Patterns. The Pattern Diagram Editor has its own **Toolbar**, **Palette**, and **Core GSN** icons.

Open an Argument Pattern

To open an Argument Pattern

1. Open the **Patterns Dashboard**.
2. In the **Patterns** window, select the **Argument Pattern** you wish to open (*Figure 38*).
3. Click **Open**.

The Argument Pattern will open in the **Pattern Diagram Editor**.

NOTE: If the Argument Pattern layout is not optimized, click on the diagram canvas to deselect any nodes. Then, select the **Arrange All** icon from the Toolbar. The nodes and links within the diagram will rearrange with equal spacing and alignment.

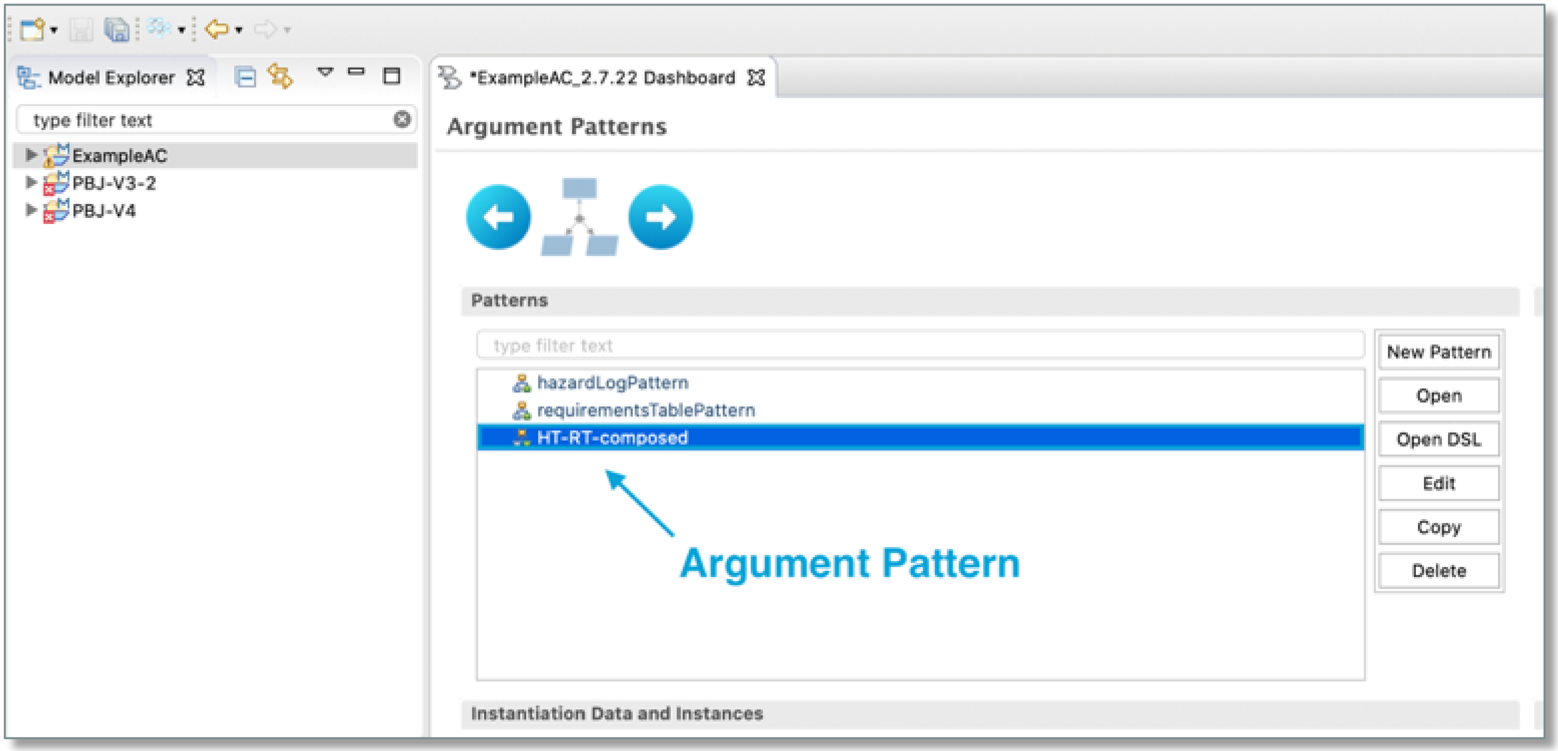


Figure 38: In the Patterns Dashboard, Argument Patterns for an AdvoCATE Project are listed in the Patterns window.

Pattern Toolbar

The **Pattern Toolbar** is in the top-left corner of the Pattern Diagram Editor.

The Pattern Toolbar has the following tools:

- **Arrange All**: Arranges all Nodes with equal spacing and alignment.
- **Autosize all nodes to fit text**: Resizes all Nodes to display all their Description text.
- **Select All**: Selects everything in the Pattern Diagram.
    - **Select All Connectors**: Selects all Links in the Pattern Diagram.
    - **Select All Shapes**: Selects all Nodes in the Pattern Diagram.
- **Layers**: Shows/Hides different Layers in the Pattern Diagram.
- **Pin/Unpin**: Opens the Pinned Elements Wizard.
- **Zoom In**: Zooms in on the Diagram Canvas.
- **Zoom Out**: Zooms out on the Diagram Canvas.
- 100% **Zoom Percentage**: Changes the visible percentage of the Diagram Canvas.
- **Export diagram as image**: Exports the Diagram Canvas as a JPG, PNG, SVG, or PDF.
- **Edit all node fonts**: Opens Font Selection Window to change the font for an entire Pattern Structure.
- **Pin selected elements**: Pins selected element/s, so they cannot be moved.
- **Unpin selected elements**: Unpins selected element/s, so they can be moved.
- **Delete**: Deletes selected element/s.

Pattern Palette

The **Pattern Palette** is in the top-right corner of the Pattern Diagram Editor.

The Pattern Palette has the following tools:

- **Select**: Selects elements on Diagram Canvas.
- **Zoom In**: Zooms in on the selected portion of the Diagram Canvas.
- **Zoom Out**: Zooms out of the selected portion of the Diagram Canvas.
- **Note**: Creates a Note Node.
    - **Representation Link**: Creates a Representation Node of select AdvoCATE graphs, diagrams, tables, or views.
    - **Text**: Creates a Text Node.
    - **Note Attachment**: Creates a Note Node and Link to a selected Canvas element.
- **Generic Connection Creation Tool**: Draws Links between different Nodes. Links automatically format based on connected Node types.
- **Pin**: Pins selected element/s, so they cannot be moved.
- **Unpin**: Unpins selected element/s, so they can be moved.

Pattern Core GSN

The **Pattern Core GSN** is under the **Pattern Palette** and Default tools in the Pattern Diagram Editor.

The Pattern Core GSN has the following tools:

- **Assumption**: Creates an Assumption Node.
- **Context**: Creates a Context Node.
- **Solution**: Creates a Solution Node.
- **Goal**: Creates a Goal Node.
- **Justification**: Creates a Justification Node.
- **Strategy**: Creates a Strategy Node.
- **Is Supported By**: Creates an Is-Supported-By Link between Nodes.
- **In Context Of**: Creates an In-Context-Of Link between Nodes.
- **Choice**: Creates a Choice element between Nodes.

Pattern Node Types

There are six **Pattern Node Types**:

- **Assumption Node**: Renders as an oval with the letter “A” and presents an unsubstantiated statement.
- **Context Node**: Renders as a rounded rectangle and references contextual information and Evidence Artifacts.
- **Solution Node**: Renders as a circle and references Evidence Artifacts.
- **Goal Node**: Renders as a rectangle and presents a claim forming part of the argument.
- **Justification Node**: Renders as an oval with the letter “J” and presents a rationale statement.
- **Strategy Node**: Renders as a parallelogram and describes the inference between Goal and Sub-Goal/s.

Create a New Pattern Node

To create a new Pattern Node

1. Open the **Pattern Diagram Editor**.
2. Select the **Node type** you want to create from the **Pattern Core GSN**.
3. Click the **Canvas** where you want to create your new Pattern Node.

The new Pattern Node will generate and be viewable in the **Pattern Diagram Editor**.

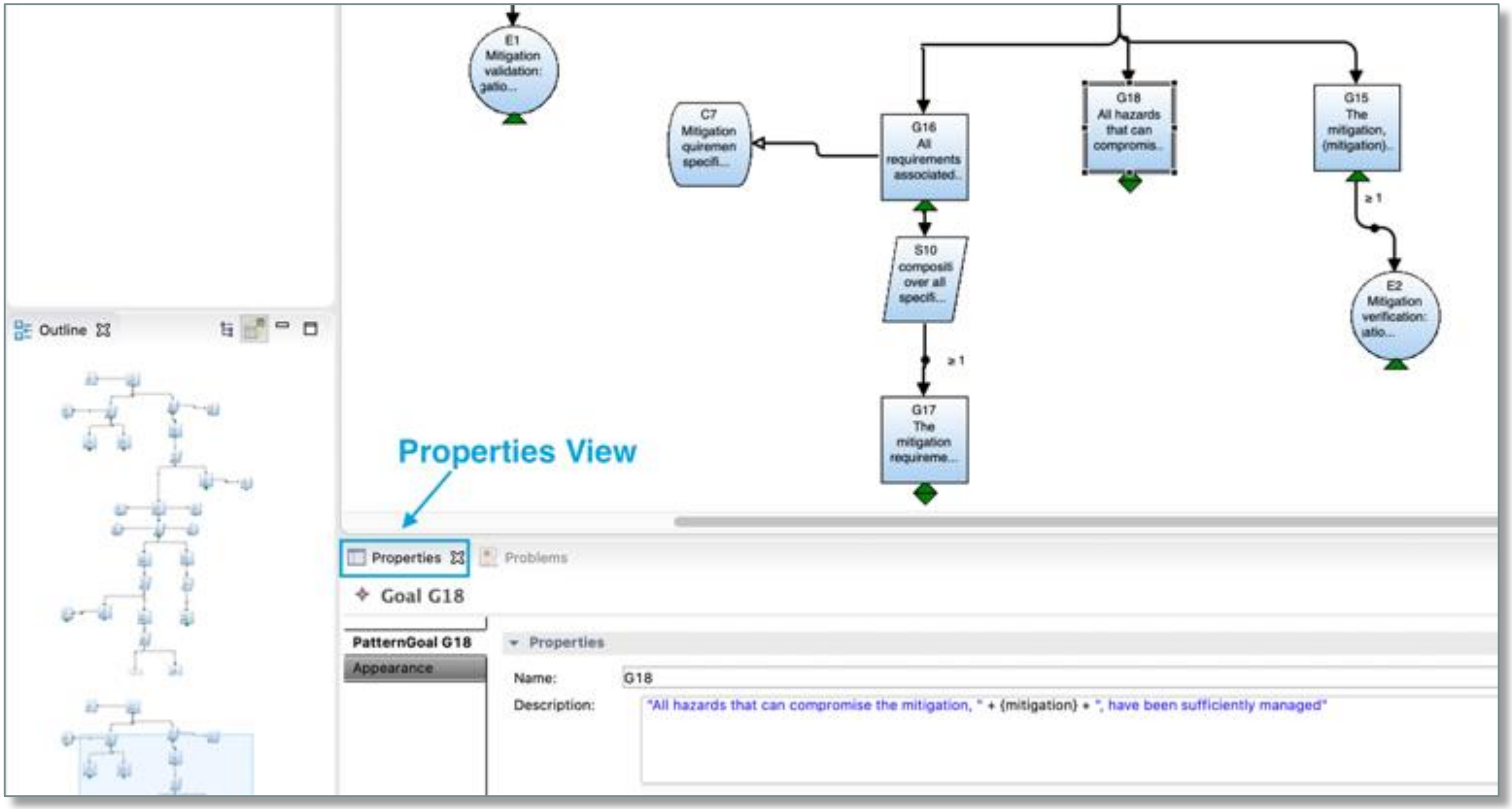
Edit a Pattern Node

To edit a Pattern Node

1. Open the **Pattern Diagram Editor**.
2. Select the **Node** you wish to edit.
3. In the **Properties View**, enter your edit/s into the corresponding fields (*Figure 39*).
4. Click outside the **Node** you edited to update your Pattern Node.

The Pattern Node in the **Pattern Diagram Editor** will update to reflect the change.

NOTE: When the Description field is being edited via the Properties View, the node’s displayed contents will not update until after the Description field has been deselected.



Properties | Problems

Goal G18

PatternGoal G18

Appearance

Properties

Name: G18

Description: "All hazards that can compromise the mitigation, " + {mitigation} + ", have been sufficiently managed"

Figure 39: In the Pattern Diagram Editor, use the Properties View to view and edit a selected node's properties.

Delete a Pattern Node

To delete a Pattern Node

1. Open the **Pattern Diagram Editor**.
2. Select the **Node** you wish to delete.
3. Click the **Delete** icon in Pattern Toolbar.

Your selected Pattern Node will be deleted from the **Pattern Diagram Editor**.

Pattern Link Types

There are four **Pattern Link Types**:

- **Is-Supported-By Link**: Renders as a line with a solid arrowhead and allows support relationships between nodes. Permitted between Goal-to-Goal, Goal-to-Strategy, Goal-to-Solution, Strategy-to-Goal, Strategy-to-Solution, and Strategy-to-Strategy nodes.
- **In-Context-Of Link**: Renders as a line with a hollow arrowhead and declares a contextual relationship between nodes. Permitted between Goal-to-Context, Goal-to-Assumption, Goal-to-Justification, Strategy-to-Context, Strategy-to-Assumption, Strategy-to-Justification, Solution-to-Context, Solution-to-Assumption, and Solution-to-Justification nodes.
- **Choice Link**: Renders as a filled diamond and adds Choice branches (in the form of an Is-Supported-By or In-Context-Of-Link) between one source node and at least two or more target nodes.
- **Pattern Loop**: Renders as an Is-Supported-By or In-Context-Of Link that connects a source node to a target node that is the source node's ancestor. Pattern Loops allow Argument Patterns to specify recursive reasoning (*Figure 40*).

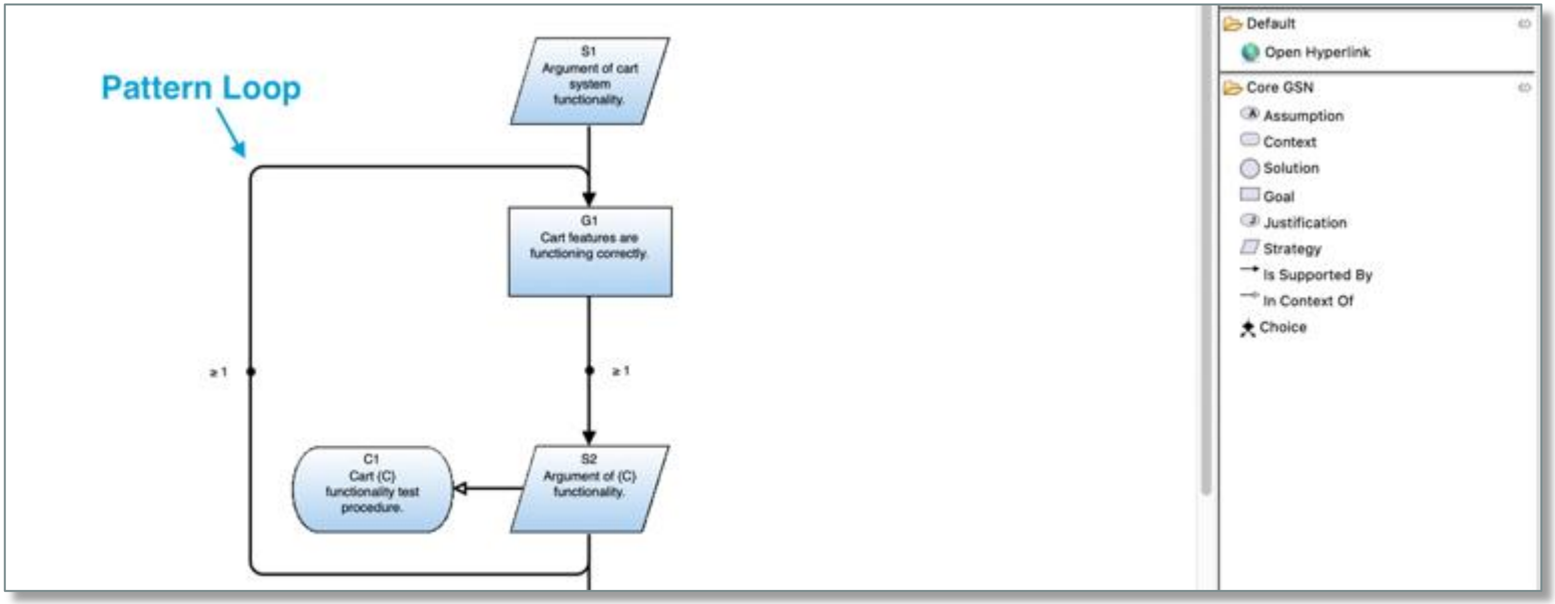

Pattern Loop
S1
Argument of cart system functionality.
G1
Cart features are functioning correctly.
≥1
≥1
C1
Cart (C) functionality test procedure.
S2
Argument of (C) functionality.
Default
Open Hyperlink
Core GSN
Assumption
Context
Solution
Goal
Justification
Strategy
Is Supported By
In Context Of
Choice

Figure 40: In the Pattern Diagram Editor, you can use Pattern Loops to specify recursive reasoning in Argument Patterns.

Create a Pattern Link

To create a new Pattern Link

1. Open the **Pattern Diagram Editor**.
2. Select the **Generic Connection Creation Tool** in the **Pattern Palette**.
3. Select the **parent node** on the **Canvas** where you want the connection to start.
4. Select the **child node** on the **Canvas** where you want the connection to end.

A new Pattern Link will generate based on the node types selected.

Create a Choice Link

To create a new Choice Link

1. Open the **Pattern Diagram Editor**.
2. Select the **Choice** icon in the **Pattern Core GSN**.
3. Select the **source node** on the **Canvas** where you want the link to start.
4. Select the **target node** on the **Canvas** where you want the link to end. If you want to create another Choice link from the same Choice node, see the *Create a Choice Branch* section.

A new Choice Link will generate between the selected nodes.

Create a Choice Branch

To create a new Choice Branch from an existing Choice node

1. Open the **Pattern Diagram Editor**.
2. Select the **Choice** icon in the **Pattern Core GSN**.
3. Select the existing **Choice node** on the Canvas where you want the branch to start.
4. Select the **target node** on the **Canvas** where you want the branch to end.

A new Choice Branch will generate between the selected nodes.

Create a Pattern Loop

To create a Pattern Loop

1. Open the **Pattern Diagram Editor**.
2. Select the **Generic Connection Creation Tool** in the **Pattern Palette**.
3. Select the **source node** on the **Canvas** where you want the link to start.
4. Select the **target node** on the **Canvas** where you want the link to end.

   NOTE: The target node must be an ancestor of the selected source node.

5. Use the **Arrange All** icon twice to properly display your Pattern Loop.

A new Pattern Loop will generate and be properly displayed between the selected nodes.

Delete a Pattern Link

To delete a Pattern Link

1. Open the **Pattern Diagram Editor**.
2. Select the **Link** you wish to delete.
3. Click the **Delete** icon in Pattern Toolbar.

Your selected Pattern Link will be deleted from the **Pattern Diagram Editor**.

Pattern Parameters

Pattern Nodes can contain **Pattern Parameters**, which are variables that are replaced with data upon **Pattern Instantiation**.

Create a New Pattern Parameter

To create a new Pattern Parameter

1. Open the **Pattern Diagram Editor**.
2. Right-click the **Canvas.**
3. In the right-click menu, select **Edit Parameters**.
4. In the pop-up window, click the **Add…** button.
5. Enter the **name** for your new Pattern Parameter.
6. Click **Ok**.

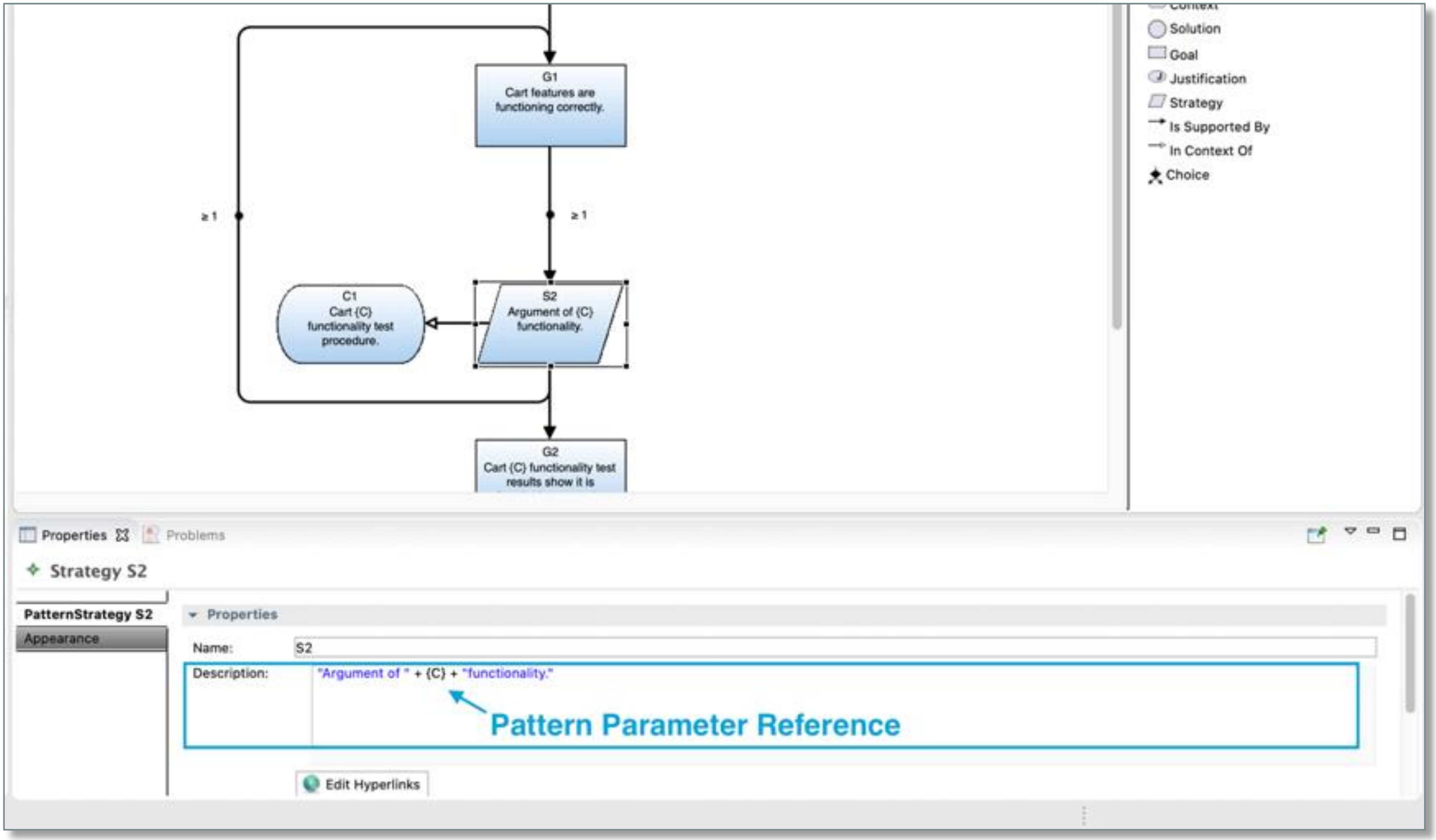
The new Pattern Parameter will generate.

Reference a Pattern Parameter in a Pattern Node

To reference a Pattern Parameter in a Pattern Node

1. Open the **Pattern Diagram Editor**.
2. Select the **Node** you wish to edit.
3. In the **Properties View**, enter a **Description** for your selected Node using the format below:
   - “boilerplate text” + {parameter} + “boilerplate text” (*Figure 41*).
4. Click outside the **Node** you edited to update your Pattern Node.

The Pattern Parameter will be referenced in the selected node.


G1
Cart features are functioning correctly.
≥ 1
≥ 1
C1
Cart {C} functionality test procedure.
S2
Argument of {C} functionality.
G2
Cart {C} functionality test results show it is
Context
Solution
Goal
Justification
Strategy
Is Supported By
In Context Of
Choice
Properties
Problems
Strategy S2
PatternStrategy S2
Appearance
Properties
Pattern Parameter Reference
Edit Hyperlinks

Name: S2

Description: "Argument of " + {C} + "functionality."

Figure 41: In the Pattern Diagram Editor, you can reference Pattern Parameters in a Node's Description field.

Edit a Pattern Parameter

To edit a Pattern Parameter

1. Open the **Pattern Diagram Editor**.
2. Right-click the **Canvas.**
3. In the right-click menu, select **Edit Parameters**.
4. In the pop-up window, select the **Pattern Parameter** you wish to edit.
5. Click the **Edit…** button.
6. Enter your **edits** for the selected Pattern Parameter.
7. Click **Ok**.

The Pattern Parameter will update.

Delete a Pattern Parameter

To delete a Pattern Parameter

1. Open the **Pattern Diagram Editor**.
2. Right-click the **Canvas.**
3. In the right-click menu, select **Edit Parameters**.
4. In the pop-up window, select the **Pattern Parameter** you wish to deleted.
5. Click the **Remove…** button.

The Pattern Parameter will be deleted.

Choice Multiplicity

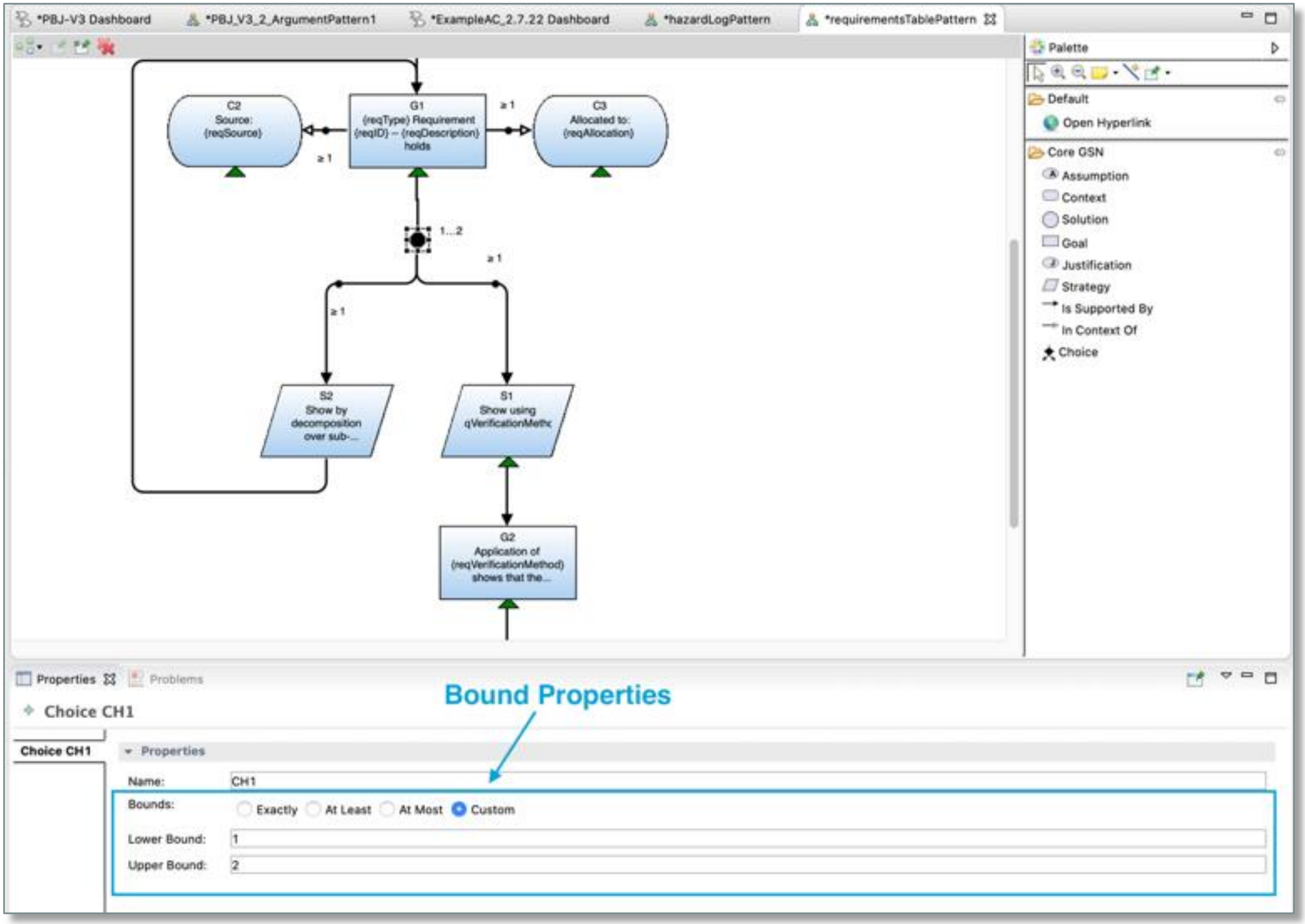
Choice Multiplicity allows users to indicate the multiplicity Bounds for a selected Choice node or link.

Edit Choice Multiplicity

To edit Choice Multiplicity

1. Open the **Pattern Diagram Editor**.
2. Select the **Choice** node or link you wish to edit.
3. In the **Properties View**, select one of the following **Bounds** options for your selected Choice node or link (*Figure 42*):
   - Exactly
   - At Least
   - At Most
   - Custom
4. Depending on your selection, enter the n-ary limits of the Bound/s.
5. Click outside the **textbox field** you edited to update your Choice node or link.

The Bounds for the Choice node or link will update to reflect the change/s.



Bound Properties

Name: CH1

Bounds: ○ Exactly ○ At Least ○ At Most ● Custom

Lower Bound: 1

Upper Bound: 2

Figure 42: In the Pattern Diagram Editor, you can edit Choice Multiplicity in a Choice Node's Properties View.

Link Multiplicity

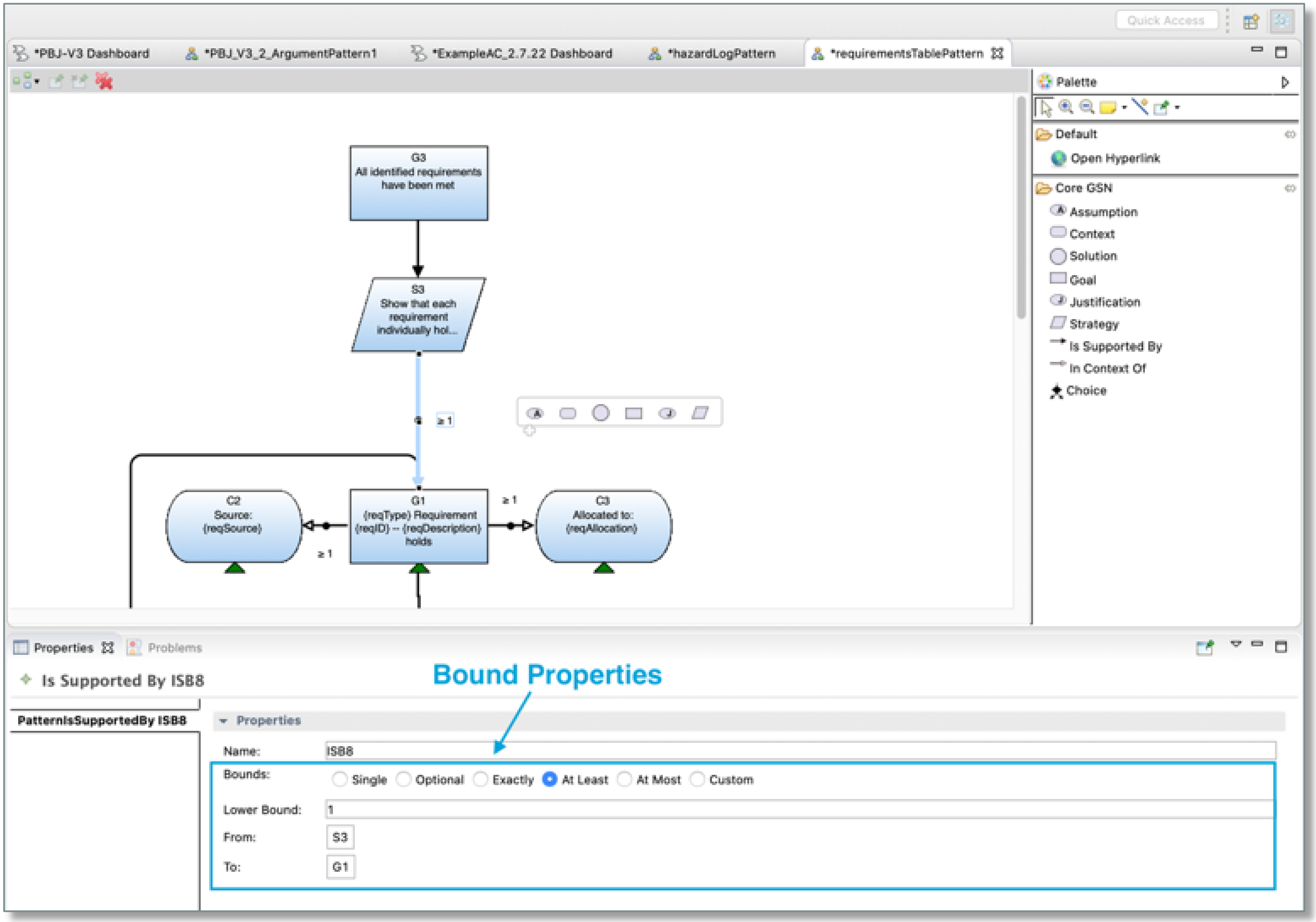
Link Multiplicity allows users to indicate the multiplicity Bounds for a selected Pattern Link.

Edit Link Multiplicity

To edit Link Multiplicity

1. Open the **Pattern Diagram Editor**.
2. Select the **Pattern Link** you wish to edit.
3. In the **Properties View**, select one of the following **Bounds** options for your selected Pattern Link (*Figure 43*):
    - Single
    - Optional
    - Exactly
    - At Least
    - At Most
    - Custom
4. Depending on your selection, enter the n-ary limits of the Bound/s.
5. Click outside the **textbox field** you edited to update your Pattern Link.

The Bounds for the Pattern Link will update to reflect the change/s.


Quick Access
*PBJ-V3 Dashboard
*PBJ_V3_2_ArgumentPattern1
*ExampleAC_2.7.22 Dashboard
*hazardLogPattern
*requirementsTablePattern
G3
All identified requirements have been met
S3
Show that each requirement individually hol...
≥ 1
C2
Source: {reqSource}
G1
{reqType} Requirement {reqID} -- {reqDescription} holds
C3
Allocated to: {reqAllocation}
≥ 1
≥ 1
Palette
Default
Open Hyperlink
Core GSN
Assumption
Context
Solution
Goal
Justification
Strategy
Is Supported By
In Context Of
Choice
Properties
Problems
PatternIsSupportedBy ISB8
Properties

Is Supported By ISB8

Bound Properties

Name: ISB8

Bounds: ○ Single ○ Optional ○ Exactly ● At Least ○ At Most ○ Custom

Lower Bound: 1

From: S3

To: G1

Figure 43: In the Pattern Diagram Editor, you can edit Link Multiplicity in a Link's Properties View.

Pattern Instantiation

Once an **Argument Pattern** and its corresponding **Data Tree** are created, you can instantiate an Argument Pattern.

Pattern Instantiation is when an Argument Pattern and Data Trees merge to generate GSN Nodes and Links in an **Argument Structure**.

When an **Argument Pattern** is instantiated, the **Pattern Parameters** referenced in an Argument Pattern will generate according to the data values assigned to said Pattern Parameters in the corresponding **Data Tree**.

For more information about Data Trees or Argument Structures, see the *Data Trees* or *Argument Structures* sections.

Instantiate an Argument Pattern

To instantiate an Argument Pattern

1. Open the **Argument Pattern** in the Pattern Diagram Editor.
2. Right-click the **canvas** in the Pattern Diagram Editor.
3. From the right-click menu, select **Instantiate Pattern**.
4. Select the appropriate **Data Tree** to use.
5. Click **Next**.
6. Select the **AdvoCATE Project** that will contain the Instance Argument Structure.
7. Enter the **name** for the Instance Argument Structure.

   NOTE: An Argument Structure File must end in the extension **.argument**.

8. Click **Finish**.

A new Argument Structure File will be created, and the **Instance Argument Structure** will open in the **Argument Diagram Editor.**

Data Trees

Pattern data for instantiating patterns can manually specified in a **Data Tree**, generated from internal artifacts (like the Hazard Log and Requirements Log), or generated from external data using **Data Mapping Files**.

For more information on Data Mapping Files, see the *Data Mapping Files* section.

Data Trees list Pattern Parameters in relation to various data values to use during the **Pattern Instantiation** process. For more information on Pattern Instantiation, see the *Pattern Instantiation* section.

Create a New Data Tree Manually

To create a new Data Tree manually

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Data Tree**.
3. Select the **AdvoCATE Project folder** that will contain the new Data Tree.
4. Name the **Data Tree**.

   NOTE: A Data Tree must end in the extension **.patterndata**.

5. Click **Finish**.

The new Data Tree will open in either the **Data Tree Diagram Editor** or **Data Tree DSL Editor** (based on the last Editor type you have open).

NOTE: If you need to open a different Editor type, use the right-click menu to **Open In** or **Open With** a different Editor type.

Data Tree Diagram Editor

The **Data Tree Diagram Editor** allows users to manually create Data Trees. The Data Tree Diagram Editor has its own **Toolbar**, **Palette**, and **Nodes** icons.

Open Data Tree Diagram Editor

To open the Data Tree Diagram Editor

1. In the **Model Explorer View**, right-click the **Data Tree** you wish to open.
2. From the right-click menu, select **Open With** > **Diagram Editor**.

Your selected Data tree will open in the **Data Tree Diagram Editor**.

Data Tree Toolbar

The **Data Tree Toolbar** is in the top-left corner of the Data Tree Diagram Editor.

The Data Tree Toolbar has the following tools:

- **Arrange All**: Arranges all Nodes with equal spacing and alignment.
- **Autosize all nodes to fit text**: Resizes all Nodes to display all their Description text.
- **Select All**: Selects everything in the Data Tree Diagram.
  - **Select All Connectors**: Selects all Link in the Data Tree Diagram.
  - **Select All Shapes**: Selects all Nodes in the Data Tree Diagram.
- **Layers**: Shows/Hides different Layers in the Data Tree Diagram.
- **Pin/Unpin**: Opens the Pinned Elements Wizard.
- **Zoom In**: Zooms in on the Diagram Canvas.
- **Zoom Out**: Zooms out on the Diagram Canvas.
- **Zoom Percentage**: Changes the visible percentage of the Diagram Canvas.
- **Export diagram as image**: Exports the Diagram Canvas as a JPG, PNG, SVG, or PDF.
- **Edit all node fonts**: Opens Font Selection Window to change the font for an entire Data Tree.
- **Pin selected elements**: Pins selected element/s, so they cannot be moved.
- **Unpin selected elements**: Unpins selected element/s, so they can be moved.
- **Delete**: Deletes selected element/s.

Data Tree Palette

The **Data Tree Palette** is in the top-right corner of the Data Tree Diagram Editor.

The Data Tree Palette has the following tools:

- **Select**: Selects elements on Diagram Canvas.
- **Zoom In**: Zooms in on the selected portion of the Diagram Canvas.
- **Zoom Out**: Zooms out of the selected portion of the Diagram Canvas.
- **Note**: Creates a Note Node.
  - **Representation Link**: Creates a Representation Node of select AdvoCATE graphs, diagrams, tables, or views.
  - **Text**: Creates a Text Node.
  - **Note Attachment**: Creates a Note Node and Link to a selected Canvas element.
- **Generic Connection Creation Tool**: Draws Links between different Nodes. Links automatically format based on connected Node types.
- **Pin**: Pins selected element/s, so they cannot be moved.
- **Unpin**: Unpins selected element/s, so they can be moved.

Data Tree Nodes

The **Data Tree Nodes** is under the **Data Tree Palette** and Default tools in the Data Tree Diagram Editor.

The Data Tree Nodes has the following tool:

- **Add Child Node**: Creates a Child Node.

Data Tree Node Types

There are two **Data Tree Node Types**:

- **Root Node**: Renders as a grey oval when the Data Tree is created. The Root Node is the foundation for a Data Tree and will align with an Argument Pattern's Pattern Parameters once the pattern is instantiated.
- **Child Node**: Renders as a red rectangle, and it references a Pattern Parameter and a corresponding data value.

Create a New Child Node in a Data Tree File

To create a new Child Node in a Data Tree File

1. Open the **Data Tree Diagram Editor**.
2. Select the **Add Child Node** icon from the **Data Tree Nodes**.
3. Click the **Root Node** or **Child Node** that you want to connect your new Child Node to.

A new Child Node will generate in the **Data Tree Diagram Editor**.

Edit a Child Node in a Data Tree

To edit a Child Node in a Data Tree

1. Open the **Data Tree Diagram Editor**.
2. Slow-click the **Node Description** you wish to edit. *The Node Description will automatically open in a textbox field.*
3. Enter your edits. *A Child Node needs to list a Pattern Parameter, followed by a colon and a corresponding data value (Parameter: Value).*
4. Click outside the **Node** you edited to update your selected Node.

The Child Node in the **Data Tree Diagram Editor** will update to reflect the change.

Delete a Child Node in a Data Tree

To delete a Child Node in a Data Tree

1. Open the **Data Tree Diagram Editor**.
2. Select the **Child Node** you wish to delete.
3. Click the ✖ **Delete** icon in Data Tree Toolbar.

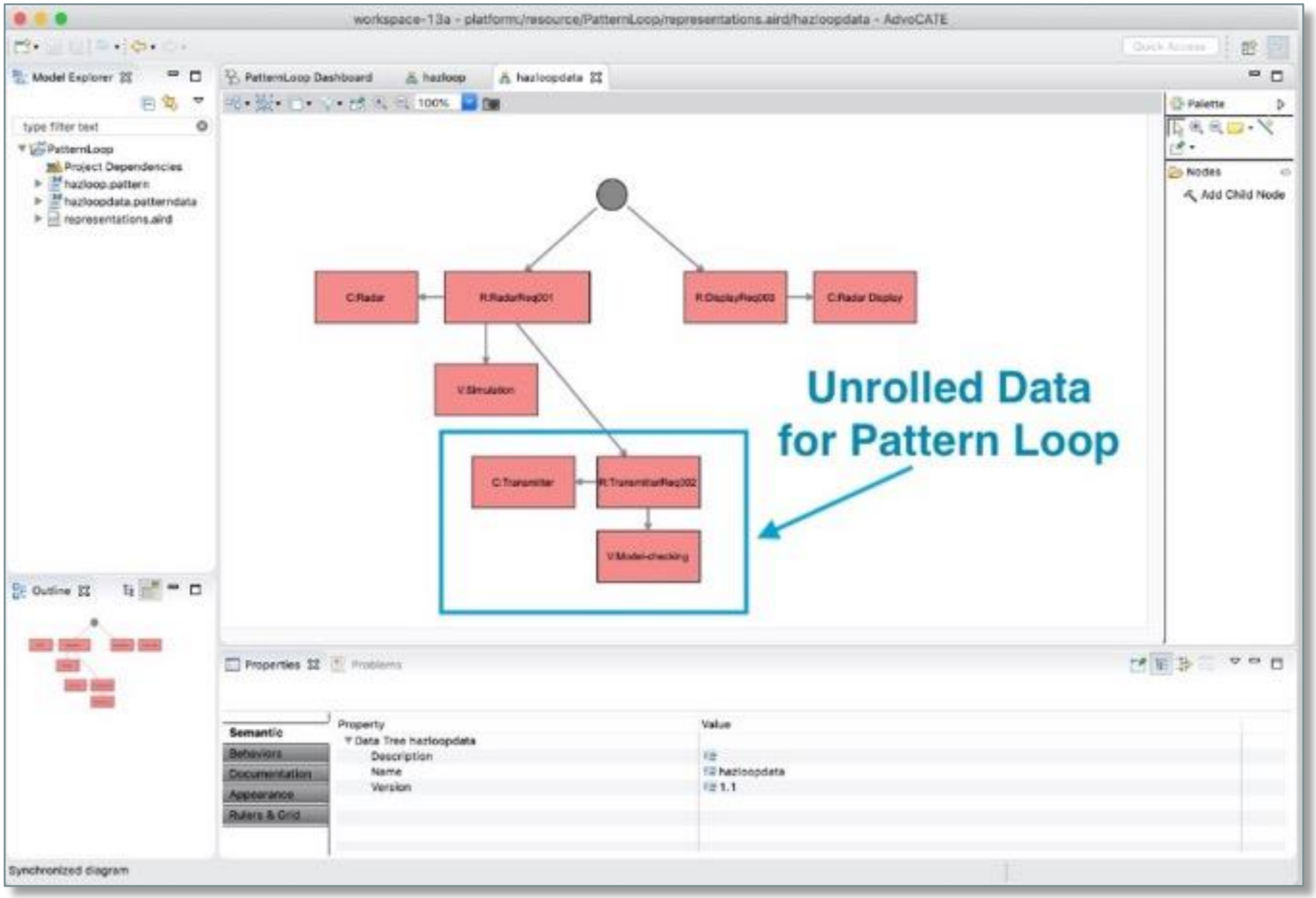
Your selected Child Node will be deleted from the **Data Tree Diagram Editor**.

Create Data for Pattern Loops in a Data Tree

To create data for Pattern Loops in a Data Tree

1. Open the **Data Tree Diagram Editor**.
2. Select the 🔨 **Add Child Node** icon from the **Data Tree Nodes**.
3. Click the **Child Node** where the recursive reasoning starts. *A new Child Node will be created and connected to the Node you selected.*
4. Edit the new Child Node’s **Pattern Parameter** so that it matches the Child Node it is connected to.
5. Edit the new Child Node’s **data value** with a new data value.
6. If necessary for your Pattern Loop, create subsequent **Child Nodes** for which the recursion occurs. *The data for the pattern fragment over which recursion occurs is explicitly specified as though the loop were unrolled (Figure 44).*

When your Argument Pattern is instantiated, the Pattern Loop will generate with the data values in these Child Nodes.



Property	Value
Data Tree hazloopdata	
Description	
Name	hazloopdata
Version	1.1

Figure 44: If necessary for your Pattern Loop, create subsequent Child Nodes for which the recursion occurs. The data for the pattern fragment over which recursion occurs is explicitly specified as though the loop were unrolled.

Generate a New Data Tree from the Hazard Log

To generate a new Data Tree from the Hazard Log

1. Open the **Hazard Log**.
2. In the Hazard Log Editor, right-click the **Hazardous Activity cell** you wish to generate a **Data Tree** from.
3. From the menu, select **Generate data tree from this hazard log**.
4. Select the **AdvoCATE Project folder** where you wish to save your Data Tree. *We recommend selecting the same AdvoCATE Project folder where your Hazard Log is located.*
5. Enter the name you want to give your **Data Tree**.

   NOTE: Data Trees must end in the file extension **.patterndata**.

6. Click **Finish**.

A new Data Tree will generate and open in the **Data Tree Diagram Editor**.

NOTE: Auto-generated Data Trees are not currently laid out automatically. However, Data Tree layout does not affect the layout of the instance that it will be used to create.

Generate a New Data Tree from the Requirements Log

To generate a new Data Tree from the Requirements Log

1. Open the **Requirements Log**.
2. In the Requirements Log Editor, right-click the **ID cell** you wish to generate a **Data Tree** from.
3. From the menu, select **Generate data tree from this requirements table**.
4. Select the **AdvoCATE Project folder** where you wish to save your Data Tree. *We recommend selecting the same AdvoCATE Project folder where your Requirements Log is located.*
5. Enter the name you want to give your **Data Tree**.

   NOTE: Data Trees must end in the file extension **.patterndata**.

6. Click **Finish**.

A new Data Tree will generate and open in the **Data Tree Diagram Editor**.

NOTE: Auto-generated Data Trees are not currently laid out automatically. However, Data Tree layout does not affect the layout of the instance that it will be used to create.

Generate a New Data Tree from a Data Mapping File

To generate a new Data Tree from a Data Mapping File

1. In the **Model Explorer View**, right-click the **Data Mapping File** that you wish to generate a new Data Tree from.
2. In the right-click menu, select **Generate Data Tree**.
3. If an external data file is not specified for the Data Mapping File, click **Browse…** to select the XML file. *If your Data Mapping File already has an external data file specified, skip to step 5.*

   NOTE: XML external data files must end in the extension **.xml**.

4. Click **Next**.
5. Select the **AdvoCATE Project folder** where you wish to save your new Data Tree.
6. Click **Finish**.

A new Data Tree will generate and open in the **Data Tree Diagram Editor.**

Data Mapping Files

Data Mapping Files allow users to instantiate patterns from external data. Data Mapping Files take external data from XML files and translates that external data into a **Data Tree** using the **Data Mapping DSL**.

For more information about generating a Data Tree from a Data Mapping File, see the *Generate a New Data Tree from a Data Mapping File* section.

Create a New Data Mapping File

To create a new Data Mapping File

1. In the **Model Explorer View**, select the **AdvoCATE Project folder** that will contain the new Data Mapping File.
2. Click the **New** icon in toolbar.
3. From the drop-down menu, select **Data Mapping File**.
4. Name the **Data Mapping File**.

   NOTE: A Data Mapping File must end in the extension **.datamap**.

5. If you have an XML file ready with your external data specified, click **Browse…** to select the XML file. *If the XML file not specified now, the Data Mapping File cannot be validated against the external data until the Data Tree generation step. If your XML is not ready, skip to step 7.*

   NOTE: XML external data files must end in the extension **.xml**.

6. Click **Open**.
7. Click the **Pattern dropdown arrows** to specify an Argument Pattern.
8. Select the **Argument Pattern** you wish to associate to the new Data Mapping File.

   NOTE: The Argument Pattern must already exist in AdvoCATE to be available to select.

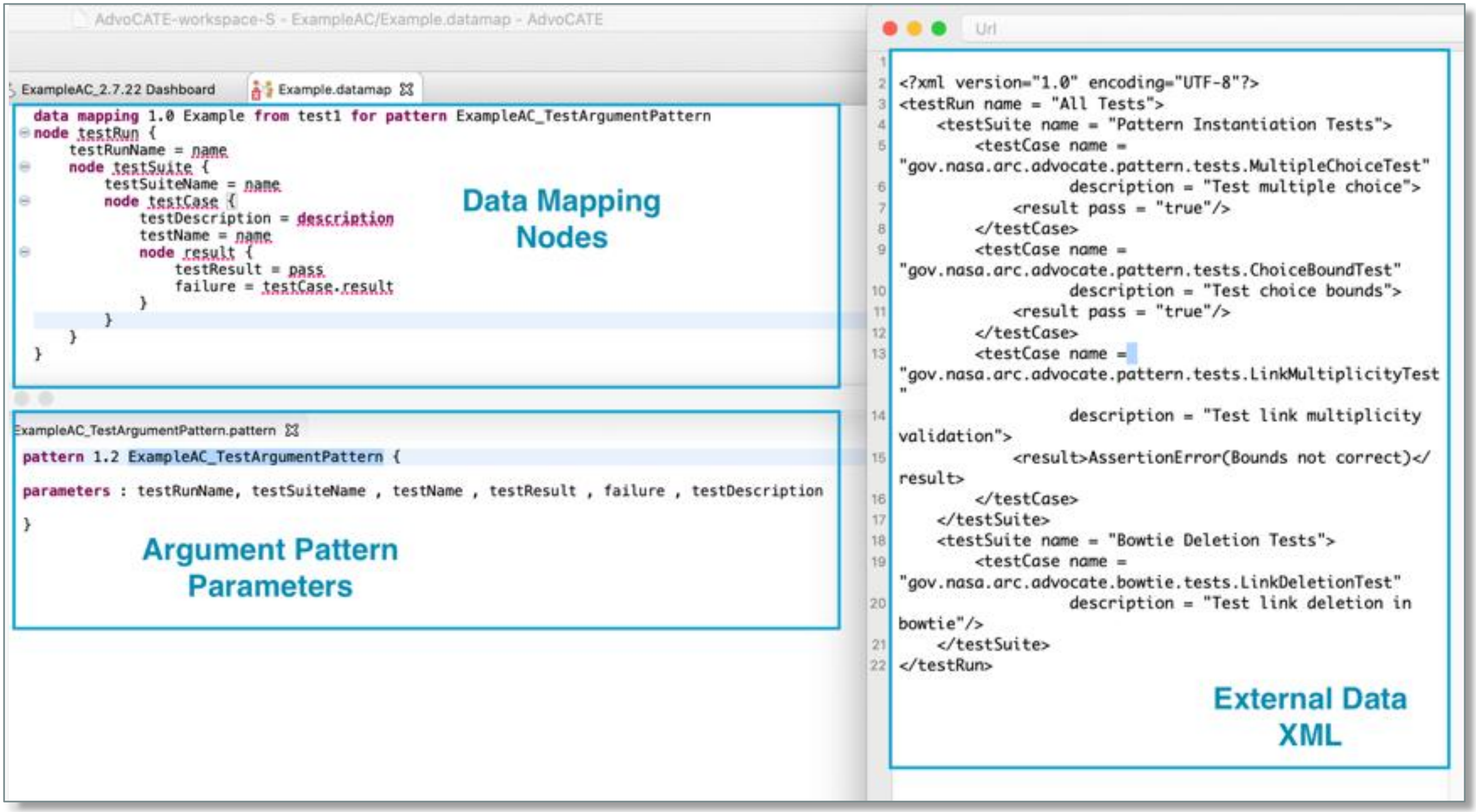
9. Click **Finish**.

The new Data Mapping File opens in the **Data Mapping DSL Editor**.

Data Mapping DSL

The **Data Mapping domain-specific language (DSL)** is used to declare the external data source (an XML file), the target **Argument Pattern** for which to map the external data, and **Mapping Nodes** that correspond to **Pattern Parameters** and **XML tags**.

Below (*Figure 45*) are examples of a corresponding Data Mapping File, XML file, and Argument Pattern Parameters.


AdvoCATE-workspace-S - ExampleAC/Example.datamap - AdvoCATE
ExampleAC_2.7.22 Dashboard
Example.datamap
Data Mapping Nodes
ExampleAC_TestArgumentPattern.pattern
Argument Pattern Parameters
Url
External Data XML

```
data mapping 1.0 Example from test1 for pattern ExampleAC_TestArgumentPattern
node testRun {
    testRunName = name
    node testSuite {
        testSuiteName = name
        node testCase {
            testDescription = description
            testName = name
            node result {
                testResult = pass
                failure = testCase.result
            }
        }
    }
}
```

```
pattern 1.2 ExampleAC_TestArgumentPattern {

parameters : testRunName, testSuiteName , testName , testResult , failure , testDescription

}
```

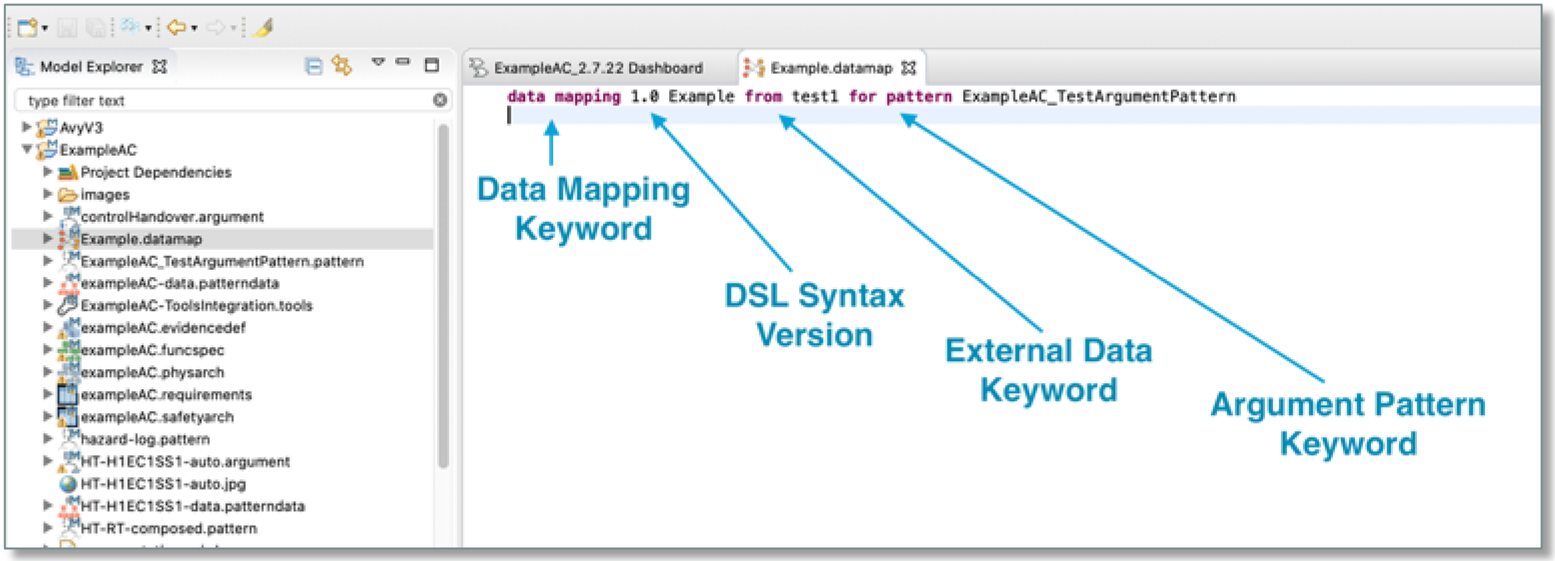
```
<?xml version="1.0" encoding="UTF-8"?>
<testRun name = "All Tests">
    <testSuite name = "Pattern Instantiation Tests">
        <testCase name =
"gov.nasa.arc.advocate.pattern.tests.MultipleChoiceTest"
                    description = "Test multiple choice">
            <result pass = "true"/>
        </testCase>
        <testCase name =
"gov.nasa.arc.advocate.pattern.tests.ChoiceBoundTest"
                    description = "Test choice bounds">
            <result pass = "true"/>
        </testCase>
        <testCase name =
"gov.nasa.arc.advocate.pattern.tests.LinkMultiplicityTest
"
                    description = "Test link multiplicity
validation">
            <result>AssertionError(Bounds not correct)</
result>
        </testCase>
    </testSuite>
    <testSuite name = "Bowtie Deletion Tests">
        <testCase name =
"gov.nasa.arc.advocate.bowtie.tests.LinkDeletionTest"
                    description = "Test link deletion in
bowtie"/>
    </testSuite>
</testRun>
```

Figure 45: When you generate a new Data Tree from a Data Mapping File, you'll need corresponding XML and Argument Pattern files.

When you create a new Data Mapping File, the Data Mapping DSL syntax for the specified external data source and Argument Pattern will generate based on your selections.

- Data Mapping Files start with the **data mapping** keyword, followed by the syntax version number and name of the Data Mapping File.
- External Data entries start with the **from** keyword, followed by the name of specified external data XML file.
- Argument Pattern entries start with the **for pattern** keyword, followed by the name of the specified Argument Pattern.

Below (*Figure 46*) is an example of a specified Argument Pattern and external data source in a Data Mapping File:



```
data mapping 1.0 Example from test1 for pattern ExampleAC_TestArgumentPattern
```

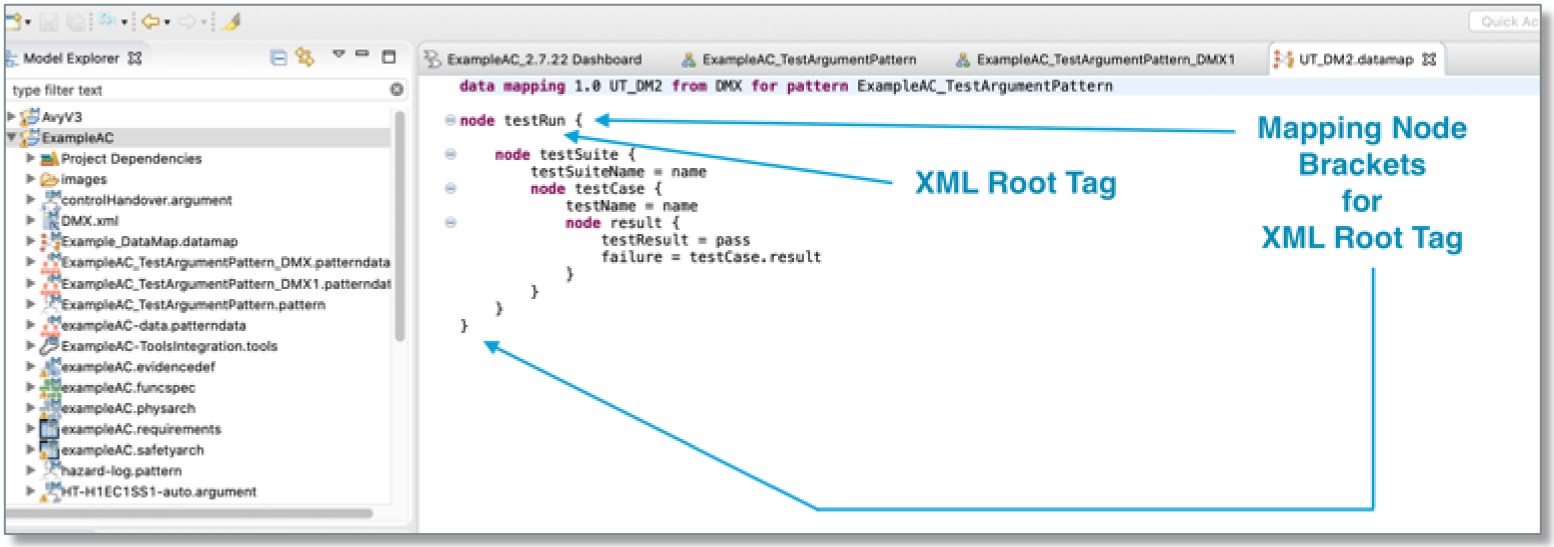
Figure 46: When you create a new Data Mapping File, the Data Mapping DSL syntax for the specified external data source and Argument Pattern will generate based on your selections.

Mapping Nodes

Mapping Nodes refer to the **XML tags** from the external data source and can assign **Pattern Parameters** to data values from within the corresponding XML tags.

NOTE: The Data Mapping DSL is case sensitive. Mapping Nodes, Pattern Parameters, and XML tags entries in a Data Mapping File must match the corresponding information in their associated Argument Patterns and XML files.

The first Mapping Node entry must start with a unique **root XML tag** that is the parent of all the other Mapping Nodes (*Figure 47*). However, it's children Mapping Nodes can use common XML tags found in your specified XML file.



Model Explorer
type filter text
AvyV3
ExampleAC
Project Dependencies
images
controlHandover.argument
DMX.xml
Example_DataMap.datamap
ExampleAC_TestArgumentPattern_DMX.patterndata
ExampleAC_TestArgumentPattern_DMX1.patterndat
ExampleAC_TestArgumentPattern.pattern
exampleAC-data.patterndata
ExampleAC-ToolsIntegration.tools
exampleAC.evidencedef
exampleAC.funcspec
exampleAC.physarch
exampleAC.requirements
exampleAC.safetyarch
hazard-log.pattern
HT-H1EC1SS1-auto.argument
ExampleAC_2.7.22 Dashboard
ExampleAC_TestArgumentPattern
ExampleAC_TestArgumentPattern_DMX1
UT_DM2.datamap
Quick Ac
Mapping Node Brackets for XML Root Tag
XML Root Tag

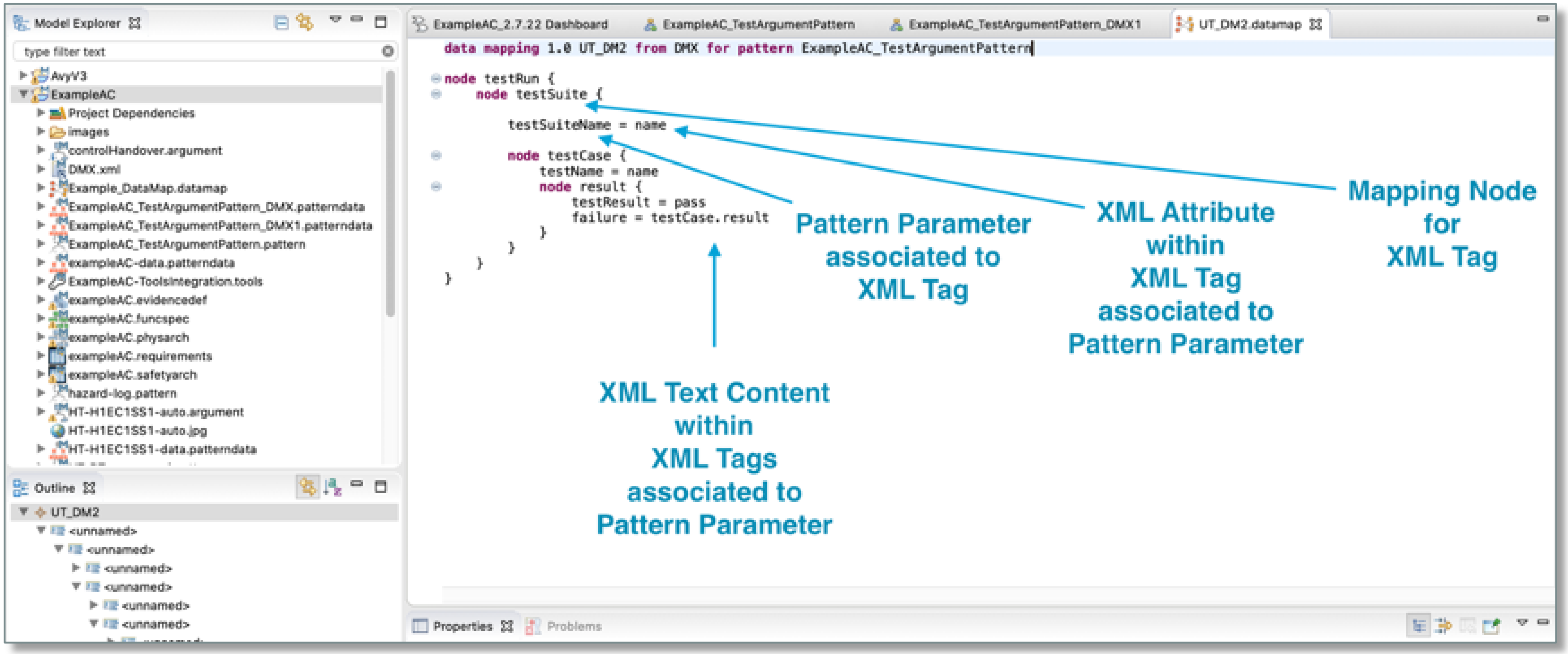
```
data mapping 1.0 UT_DM2 from DMX for pattern ExampleAC_TestArgumentPattern

node testRun {
    node testSuite {
        testSuiteName = name
        node testCase {
            testName = name
            node result {
                testResult = pass
                failure = testCase.result
            }
        }
    }
}
```

Figure 47: The first Mapping Node DSL entry in a Data Mapping File must be associated to a unique root XML tag.

Mapping Node entries start with the **node** keyword, followed by an XML tag.

- Within each Mapping Node's { } brackets, you can list Pattern Parameters, followed by an = equal sign to assigned data values found within the corresponding XML tag, called **Parameter Mappings**.

Parameter Mappings can associate XML attributes and text context found within XML tags (*Figure 48*).



```
data mapping 1.0 UT_DM2 from DMX for pattern ExampleAC_TestArgumentPattern

node testRun {
    node testSuite {

        testSuiteName = name

        node testCase {
            testName = name
            node result {
                testResult = pass
                failure = testCase.result
            }
        }
    }
}
```

Model Explorer
type filter text
AvyV3
ExampleAC
Project Dependencies
images
controlHandover.argument
DMX.xml
Example_DataMap.datamap
ExampleAC_TestArgumentPattern_DMX.patterndata
ExampleAC_TestArgumentPattern_DMX1.patterndata
ExampleAC_TestArgumentPattern.pattern
exampleAC-data.patterndata
ExampleAC-ToolsIntegration.tools
exampleAC.evidencedef
exampleAC.funcspec
exampleAC.physarch
exampleAC.requirements
exampleAC.safetyarch
hazard-log.pattern
HT-H1EC1SS1-auto.argument
HT-H1EC1SS1-auto.jpg
HT-H1EC1SS1-data.patterndata
Outline
UT_DM2
<unnamed>
<unnamed>
<unnamed>
<unnamed>
<unnamed>
<unnamed>
ExampleAC_2.7.22 Dashboard
ExampleAC_TestArgumentPattern
ExampleAC_TestArgumentPattern_DMX1
UT_DM2.datamap
Properties
Problems
Mapping Node for XML Tag
XML Attribute within XML Tag associated to Pattern Parameter
Pattern Parameter associated to XML Tag
XML Text Content within XML Tags associated to Pattern Parameter

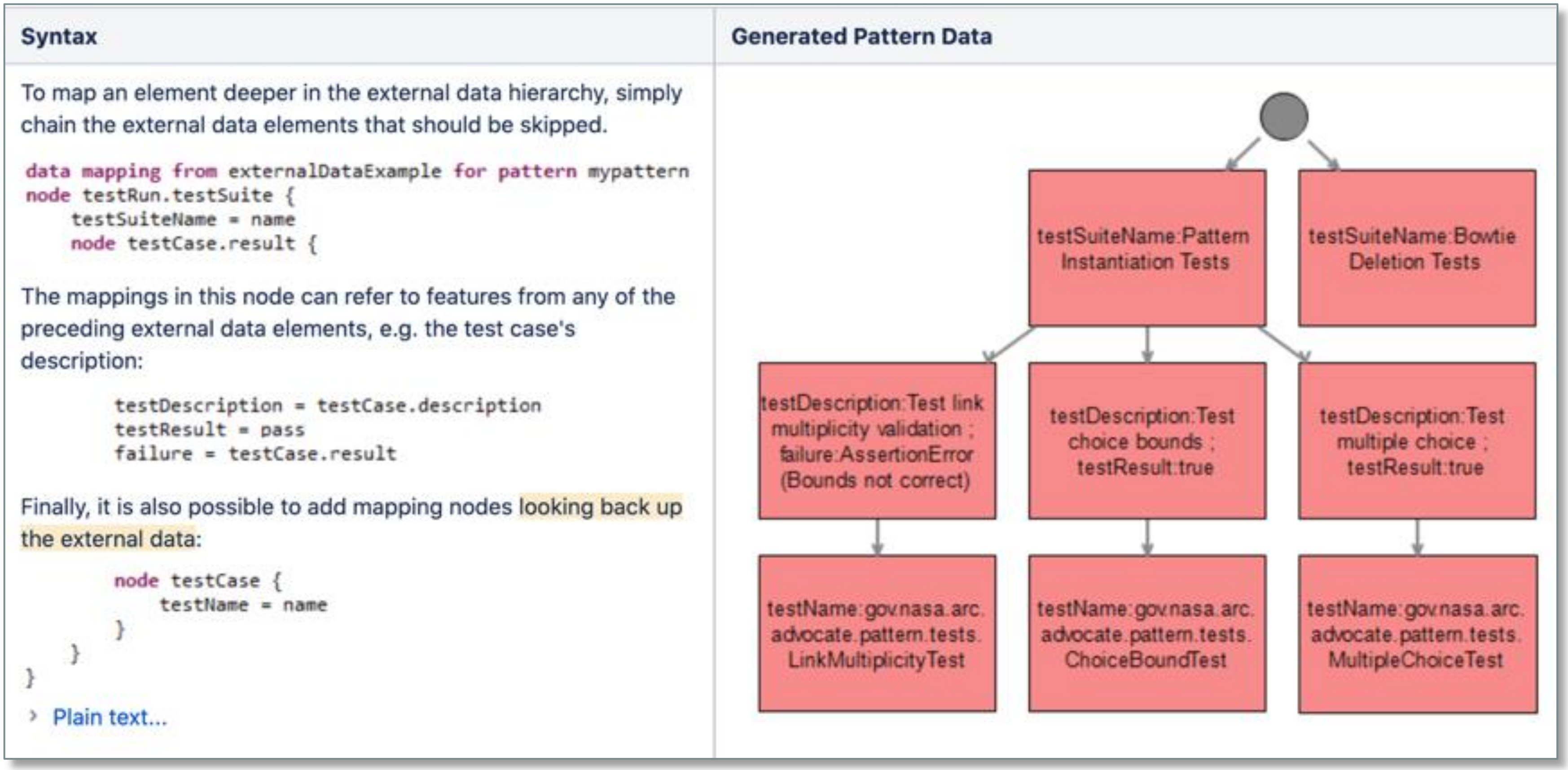
Figure 48: Within a Mapping Node's brackets, you can associate Pattern Parameters to XML attributes and text content found within the corresponding XML tags, called Parameter Mappings.

Advanced Mapping Node Techniques

Navigating Through a Data Tree

Navigating through a Data Tree allows users to refer to Mapping Nodes and Parameter Mappings above or below other external data in the data hierarchy.

To add Mapping Nodes that “fast-forward” or look deeper in the external data hierarchy, simply chain the XML tags with a . period that should be skipped (*Figure 49*). The Parameter Mappings in this “fast-forwarded” Mapping Node can refer to data values from any of the preceding XML tags.



Syntax

To map an element deeper in the external data hierarchy, simply chain the external data elements that should be skipped.

```
data mapping from externalDataExample for pattern mypattern
node testRun.testSuite {
    testSuiteName = name
    node testCase.result {
```

The mappings in this node can refer to features from any of the preceding external data elements, e.g. the test case's description:

```
        testDescription = testCase.description
        testResult = pass
        failure = testCase.result
```

Finally, it is also possible to add mapping nodes looking back up the external data:

```
        node testCase {
            testName = name
        }
    }
}
```

› Plain text...

Generated Pattern Data

testSuiteName:Pattern Instantiation Tests
testSuiteName:Bowtie Deletion Tests
testDescription:Test link multiplicity validation ; failure:AssertionError (Bounds not correct)
testDescription:Test choice bounds ; testResult:true
testDescription:Test multiple choice ; testResult:true
testName:gov.nasa.arc.advocate.pattern.tests.LinkMultiplicityTest
testName:gov.nasa.arc.advocate.pattern.tests.ChoiceBoundTest
testName:gov.nasa.arc.advocate.pattern.tests.MultipleChoiceTest

Figure 49: When navigate through a Data Tree, you can “fast-forward” and “rewind” to external data.

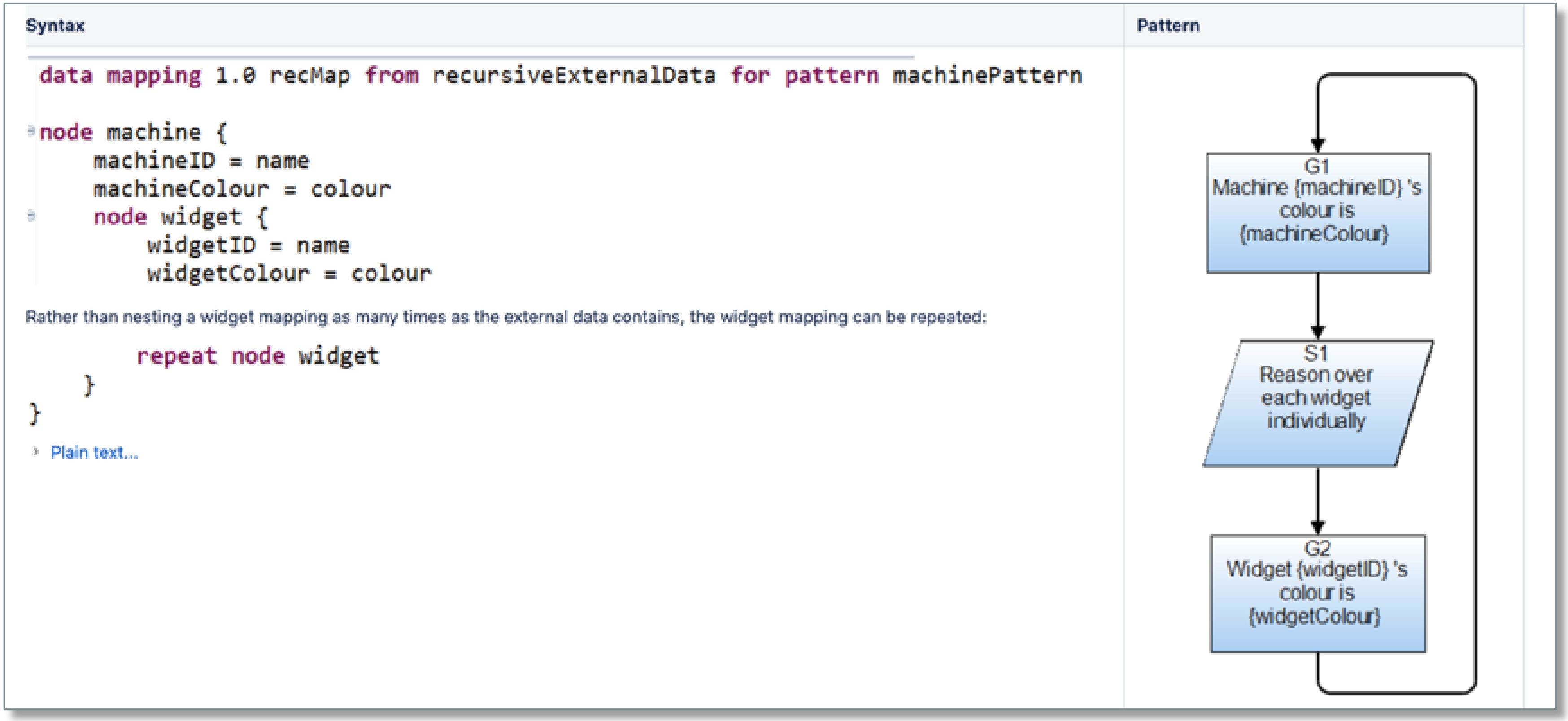
To add Mapping Nodes that “rewind” or look back to previous external data in the data hierarchy, simply add the XML tags that were skipped.

Recursive Mapping

Some Argument Patterns use recursive reasoning or Pattern Loops, which correspond to deeply nested external data structures.

Rather than nesting a Mapping Node as many times as the external data needs, Mapping Node entries can be repeated with the **repeat node** keyword (*Figure 50*).

The repeated Mapping Nodes will produce different **Data Trees** based on different external data sources or XML files.



```
data mapping 1.0 recMap from recursiveExternalData for pattern machinePattern

node machine {
    machineID = name
    machineColour = colour
    node widget {
        widgetID = name
        widgetColour = colour
```

Rather than nesting a widget mapping as many times as the external data contains, the widget mapping can be repeated:

```
        repeat node widget
    }
}
```

› Plain text...

G1
Machine {machineID} 's colour is {machineColour}
S1
Reason over each widget individually
G2
Widget {widgetID} 's colour is {widgetColour}

Figure 50: You can use the repeat node in a Data Mapping File for recursive reasoning in Argument Patterns (called Pattern Loops).

Evidence Log

Evidence constitutes all artifacts that substantiate data relevant to an assurance case (e.g., models, test results, and documentation).

Individual **Evidence Artifacts** (also called Evidence Items) are declared in an **Evidence Log**.

Once an **Evidence Log** is created, Evidence can be declared using the **Evidence Log DSL** and **Evidence Dashboard**.

Creating a New Evidence Log

To create a new Evidence Log

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Evidence Log**.
3. Click **Next**.
4. Click **Browse…** to select your AdvoCATE Project's folder location.
5. Select your **AdvoCATE Project's folder**.
6. Enter the **name** you want to give your Evidence Log.

   NOTE: The Evidence Log must end in the file extension **.evidencedef**.

7. Click **Finish**.

A new **Evidence Log** opens and contains example entries in the **Evidence Log DSL**.

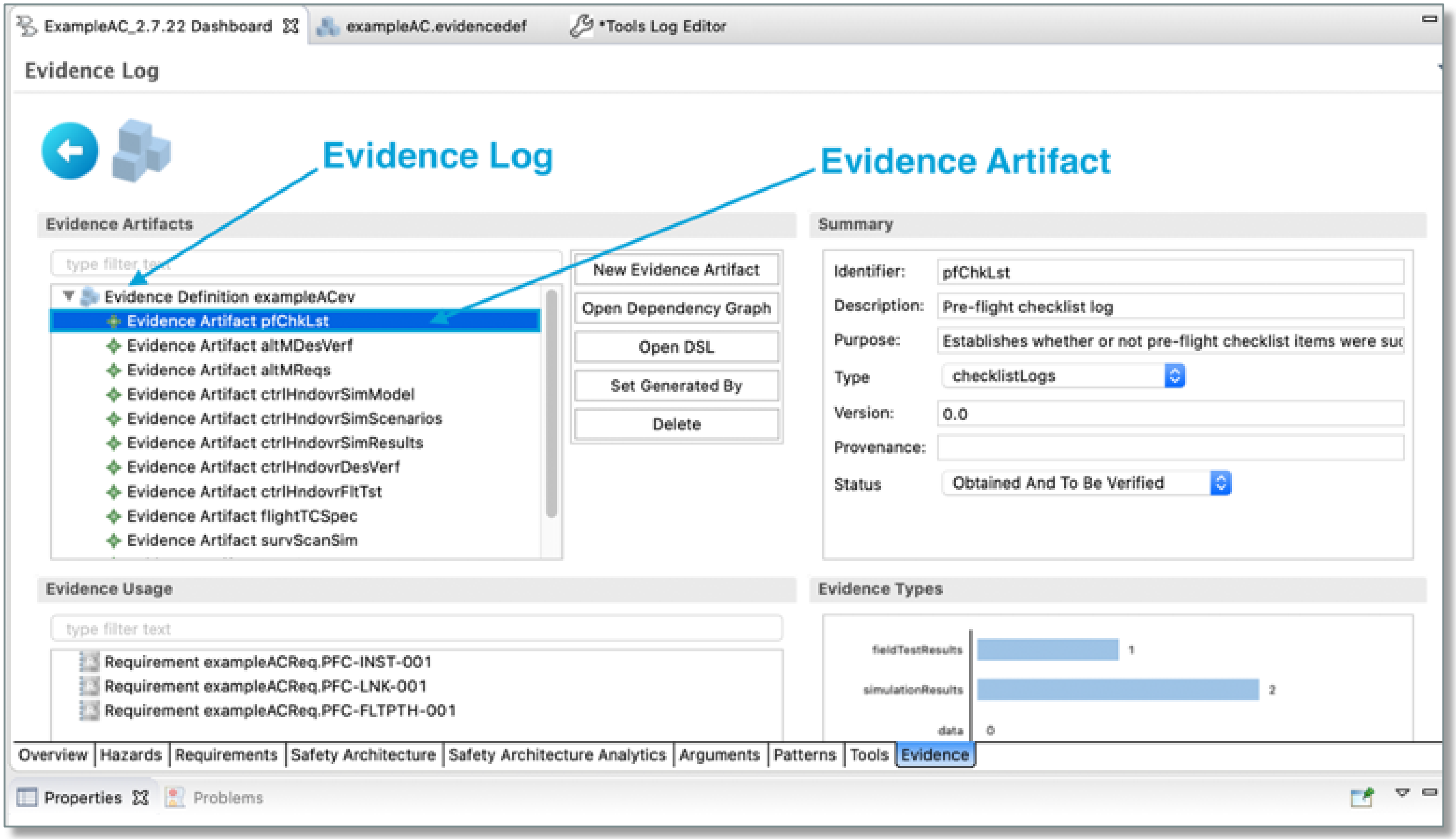
NOTE: If Evidence Artifacts are created in an **Argument Structure**, the **Requirements Log**, or the **Tools Use Table**, a dialog window will open to create an Evidence Log if one does not already exist.

Open the Evidence Log DSL Editor

To open the Evidence Log DSL Editor

1. Open the **Evidence Dashboard**.
2. In the **Evidence Artifacts** window, select the **Evidence Log** or an **Evidence Artifact** (*Figure 51*).
3. Click **Open DSL**.

The Evidence Log will open in the **Evidence Log DSL Editor**.



- Evidence Definition exampleACev
  - Evidence Artifact pfChkLst
  - Evidence Artifact altMDesVerf
  - Evidence Artifact altMReqs
  - Evidence Artifact ctrlHndovrSimModel
  - Evidence Artifact ctrlHndovrSimScenarios
  - Evidence Artifact ctrlHndovrSimResults
  - Evidence Artifact ctrlHndovrDesVerf
  - Evidence Artifact ctrlHndovrFltTst
  - Evidence Artifact flightTCSpec
  - Evidence Artifact survScanSim

Identifier: pfChkLst
Description: Pre-flight checklist log
Purpose: Establishes whether or not pre-flight checklist items were suc
Type checklistLogs
Version: 0.0
Provenance:
Status Obtained And To Be Verified

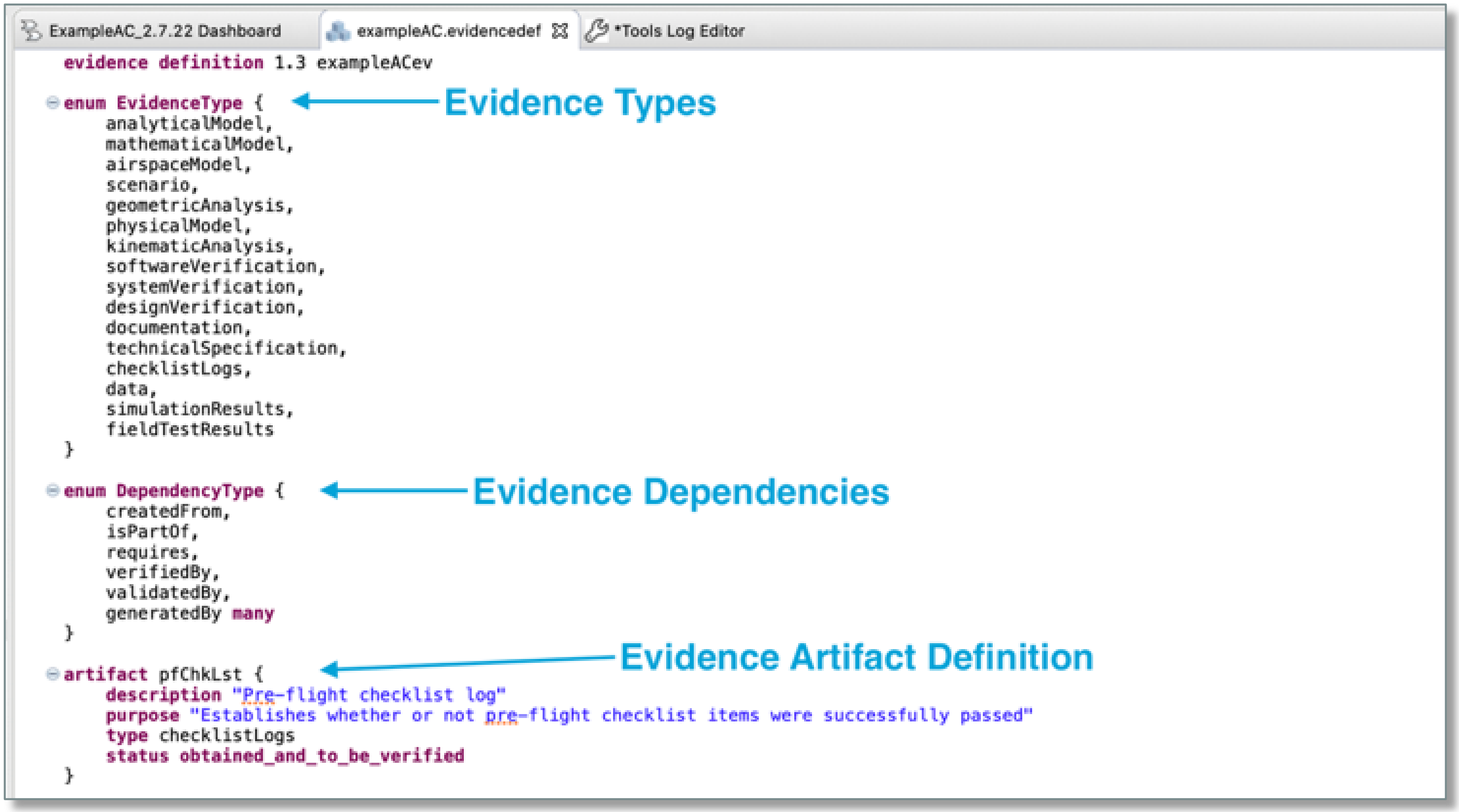
Requirement exampleACReq.PFC-INST-001
Requirement exampleACReq.PFC-LNK-001
Requirement exampleACReq.PFC-FLTPTH-001

Figure 51: In the Evidence Artifacts window, you can select the Evidence Log or a specific Evidence Artifact.

Evidence Log DSL

The **Evidence Log domain-specific language (DSL)** is used to declare **Evidence Types** and **Evidence Dependencies**, and to define **Evidence Artifact Definitions** (*Figure 52*).

The Evidence Log DSL starts with the evidence definition keyword, followed by the syntax version number and name of the Evidence Log.



```
evidence definition 1.3 exampleACev

enum EvidenceType {
    analyticalModel,
    mathematicalModel,
    airspaceModel,
    scenario,
    geometricAnalysis,
    physicalModel,
    kinematicAnalysis,
    softwareVerification,
    systemVerification,
    designVerification,
    documentation,
    technicalSpecification,
    checklistLogs,
    data,
    simulationResults,
    fieldTestResults
}

enum DependencyType {
    createdFrom,
    isPartOf,
    requires,
    verifiedBy,
    validatedBy,
    generatedBy many
}

artifact pfChkLst {
    description "Pre-flight checklist log"
    purpose "Establishes whether or not pre-flight checklist items were successfully passed"
    type checklistLogs
    status obtained_and_to_be_verified
}
```

Figure 52: In the Evidence Log DSL Editor, you can view and edit Evidence Types, Evidence Dependencies, and Evidence Artifact Definitions.

Evidence Types

Evidence Types are the classifications of **Evidence Artifacts** for your assurance case.

Evidence Types are user-extensible, so you can create Evidence Types specific to your needs.

The example Evidence Types in a newly generated Evidence Log are editable, and you can delete the unnecessary options.

Evidence Type entries start with the **enum EvidenceType** keyword and use , commas to distinguish individual Evidence Types. The list of Evidence Types is enclosed in brackets { }.

Evidence Dependencies

An **Evidence Dependency** is a relation between an **Evidence Artifact**/s.

Evidence Dependencies are user-extensible, so you can create Evidence Dependencies specific to your needs.

By default, Evidence Dependencies have an arity 1-1 (or one-to-one relations between Evidence Artifacts). However, you can use the **many** keyword to reclassify Evidence Dependencies with an arity 1-n (or to have multiple relations between Evidence Artifacts).

The example Evidence Dependencies in a newly generated Evidence Log are editable, and you can delete the unnecessary options.

Evidence Dependency entries start with the **enum DependencyType** keyword and use , commas to distinguish individual Evidence Dependencies. The list of Evidence Dependencies is enclosed in brackets { }.

Evidence Artifact Definitions

Evidence Artifact Definition entries define your various **Evidence Artifacts** for your AdvoCATE Project (*Figure 53*).

Evidence Artifacts can be used in several places in an AdvoCATE Project:

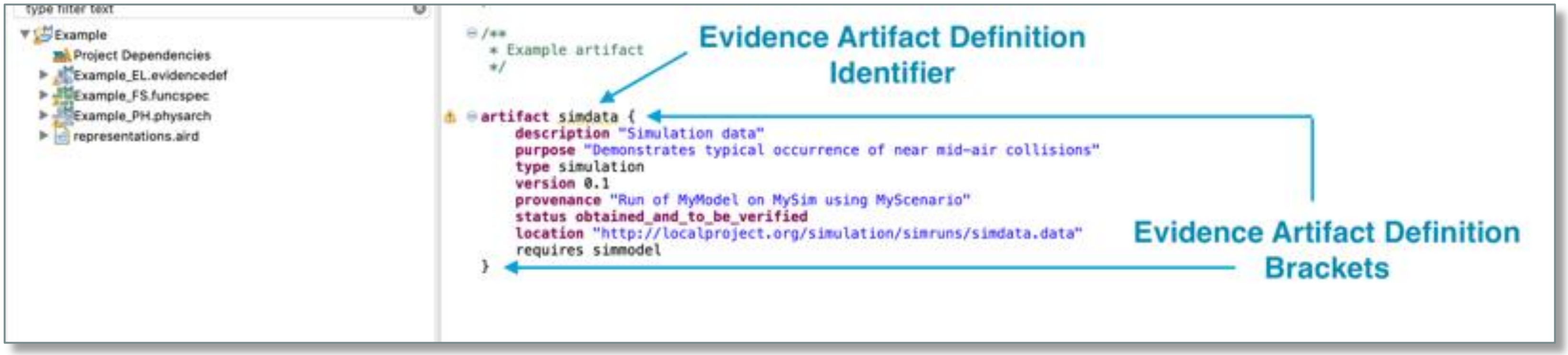
- In an **Argument Structure**: Evidence Artifacts can be linked to **Solution**, **Assumption**, **Justification**, or **Context Nodes**.
- In the **Requirements Log**: Evidence Artifacts can be listed as **Verification Allocations**.
- In the **Tools Use Table**: Evidence Artifacts can be listed as a **Tool Use Input** or **Tool Use Output**.

Evidence Artifact Definition entries start with the artifact keyword, followed by an identifier. Each Evidence Artifact Definition is then enclosed in {} brackets.

Evidence Artifact Definitions have multiple fields and can lists the following details:

- **Identifier**: uses the identifier to name each unique Evidence Artifact.
- **Description**: uses the description keyword to describe the Evidence Artifact and is enclosed with "" quotation marks.
- **Purpose**: uses the purpose keyword to describe why your Evidence Artifact is needed and is enclosed with "" quotation marks.
- **Type**: uses the type keyword to classify your Evidence Artifact as one of the types from your **Evidence Types** list. For more information about the Evidence Types list, see the *Evidence Types* section.
- **Version**: uses the version keyword to list the version number of an Evidence Artifact.
- **Provenance**: uses the provenance keyword to list the origin of the Evidence Artifact and is enclosed with "" quotation marks.
- **Status**: uses the status keyword to explain the current state of the Evidence Artifact.
  - AdvoCATE currently supports the following three statuses for Evidence Artifacts:
    - obtained_and_to_be_verified: the Evidence Artifact has been obtained but needs to be verified.
    - obtained_and_verified: the Evidence Artifact has been obtained and has been verified.
    - pending: the Evidence Artifact has not been obtained.
- **Location**: uses the location keyword to hyperlink the digital Evidence Artifact file and is enclosed with "" quotation marks.
- **Dependencies**: uses one of the dependencies from your **Evidence Dependencies** list as a keyword and is followed by the identifier of the dependent Evidence Artifact. For more information about the Evidence Dependencies list, see the *Evidence Dependencies* section.
  - Evidence Artifacts that are a result of a **Tool Use Output** will list automatically list the generatedBy dependency.

NOTE: Evidence Artifact Definition entries must use the artifact keyword, an identifier, and a type. All other entry details are optional.



```
/**
 * Example artifact
 */

artifact simdata {
    description "Simulation data"
    purpose "Demonstrates typical occurrence of near mid-air collisions"
    type simulation
    version 0.1
    provenance "Run of MyModel on MySim using MyScenario"
    status obtained_and_to_be_verified
    location "http://localproject.org/simulation/simruns/simdata.data"
    requires simmodel
}
```

Figure 53: In the Evidence Log DSL Editor, you can edit Evidence Artifact Definition entries with additional keyword fields within the Evidence Artifacts Definition entry's brackets.

Create a New Evidence Artifact Definition with the Evidence Dashboard

To create a new Evidence Artifact Definition using the **Evidence Dashboard**

1. Open the **Evidence Dashboard**.
2. In the Evidence Artifacts window, select the **Create New Evidence Artifact** button.

A new Evidence Artifact Definition entry will be created and listed in the Evidence Artifacts window.

Edit an Evidence Artifact Definition with the Evidence Dashboard

To edit an Evidence Artifact Definition using the **Evidence Dashboard**

1. Open the **Evidence Dashboard**.
2. In the Evidence Artifacts window, select the **Evidence Artifact Definition** you wish to edit. *A Summary window will appear in the right-hand corner.*
3. In the Summary window, edit your **Evidence Artifact Definition's** information using the textbox fields and drop-down menus.
4. Click outside of the Summary window to update your **Evidence Artifact Definition** entry.

The Evidence Artifact Definition will update to reflect the changes.

NOTE: You cannot create new Evidence Types or establish Evidence Dependencies from the Evidence Dashboard. To create new Evidence Types or establish Evidence Dependencies, see the *Evidence Log DSL* section.

Delete an Evidence Artifact Definition with the Evidence Dashboard

To delete an Evidence Artifact Definition using the **Evidence Dashboard**

1. Open the **Evidence Dashboard**.
2. In the Evidence Artifacts window, double-click **Evidence Artifact Definition** you wish to delete.
3. Click the **Delete** button.
4. Click **Yes** to confirm your deletion request.

Your selected Evidence Artifact Definition will be deleted.

Open an Evidence Dependency Graph

To open an Evidence Dependency Graph

1. Open the **Evidence Dashboard**.
2. In the Evidence Artifacts window, select the **Evidence Artifact Definition** you wish to open as an Evidence Dependency Graph.
3. Click the **Open Dependency Graph** button.

Your selected Evidence Artifact Definition will open as an Evidence Dependency Graph.

Import an Existing Evidence Artifact

To import an existing Evidence Artifact file into an AdvoCATE Project

1. In the **Model Explorer View**, right-click your **Evidence Log** file.
2. From the right-click menu, select **Import…**.
3. In the Import window, toggle the **drop-down arrow** for the AdvoCATE folder.
4. In the drop-down menu for the AdvoCATE folder, select **Evidence Item**.
5. Click **Next**.
6. Select the **Evidence Log** you wish to import your Evidence Artifact file into. *If you haven't created an Evidence Log yet, you can type a new file name into the textbox field.*
7. Select the **Browse…** button to locate your existing Evidence Artifact file.
8. If you want to copy the Evidence Artifact file into your AdvoCATE Project, select the **Copy** checkbox. *If you do not want to make a copy of your Evidence Artifact file in AdvoCATE, a hyperlink to your external Evidence Artifact file will be created.*
9. Enter a **Name** for your Evidence Artifact into the corresponding textbox field.
10. Enter a **Description** for your Evidence Artifact into the corresponding textbox field.
11. From the Type drop-down menu, select the **Type** classification for your Evidence Artifact file.
12. From the Status drop-down menu, select the **Status** of your Evidence Artifact file.
13. Click **Finish**.

Your corresponding Evidence Log will open in the Evidence Log DSL Editor, and a new Evidence Artifact Definition entry will be created for your imported Evidence Artifact file.

NOTE: You cannot create new Evidence Types or establish Evidence Dependencies from the Import Window. To create new Evidence Types or establish Evidence Dependencies, see the *Evidence Log DSL* section.

Tools Log

Tools represent external techniques for the creation of **Evidence Artifacts**, such as verification tools. A Tool takes one or more Evidence Artifacts as an input and produces one or more Evidence Artifacts as an output.

Once a **Tools Log** is created, Tools can be declared in the **Tools Table** or using the **Tools Log DSL**.

Create a New Tools Log

To create a new Tools Log

1. Click the **New** icon in toolbar.
2. From the drop-down menu, select **Tool Log**.
3. Click **Next**.
4. Select your **AdvoCATE Project's folder**.
5. Enter the **name** you want to give your **Tools Log**.

   NOTE: The Tools Log must end in the file extension **.tools**.

6. Click **Finish**.

A new **Tools Table** opens in the **Tools Log Editor**.

Tools Table

The **Tools Table** defines Tools and their Toolset integrations. The Tools Table can be sorted by the Description and Toolset columns. For more information about the Toolsets, see the *Toolset Table* section.

The definitions made in the Tools Table then correlate to **Tool entries** in the **Tools Use Table**. For more information about the Tools Use Table, see the *Tools Use Table* section.

Open the Tools Table

To open the Tools Table

1. Open the **Tools Dashboard**.
2. In the Tools window, select the **Tools Log** (*Figure 54*).
3. Select the **Open** button.

The **Tools Table** will open in the **Tools Log Editor**.

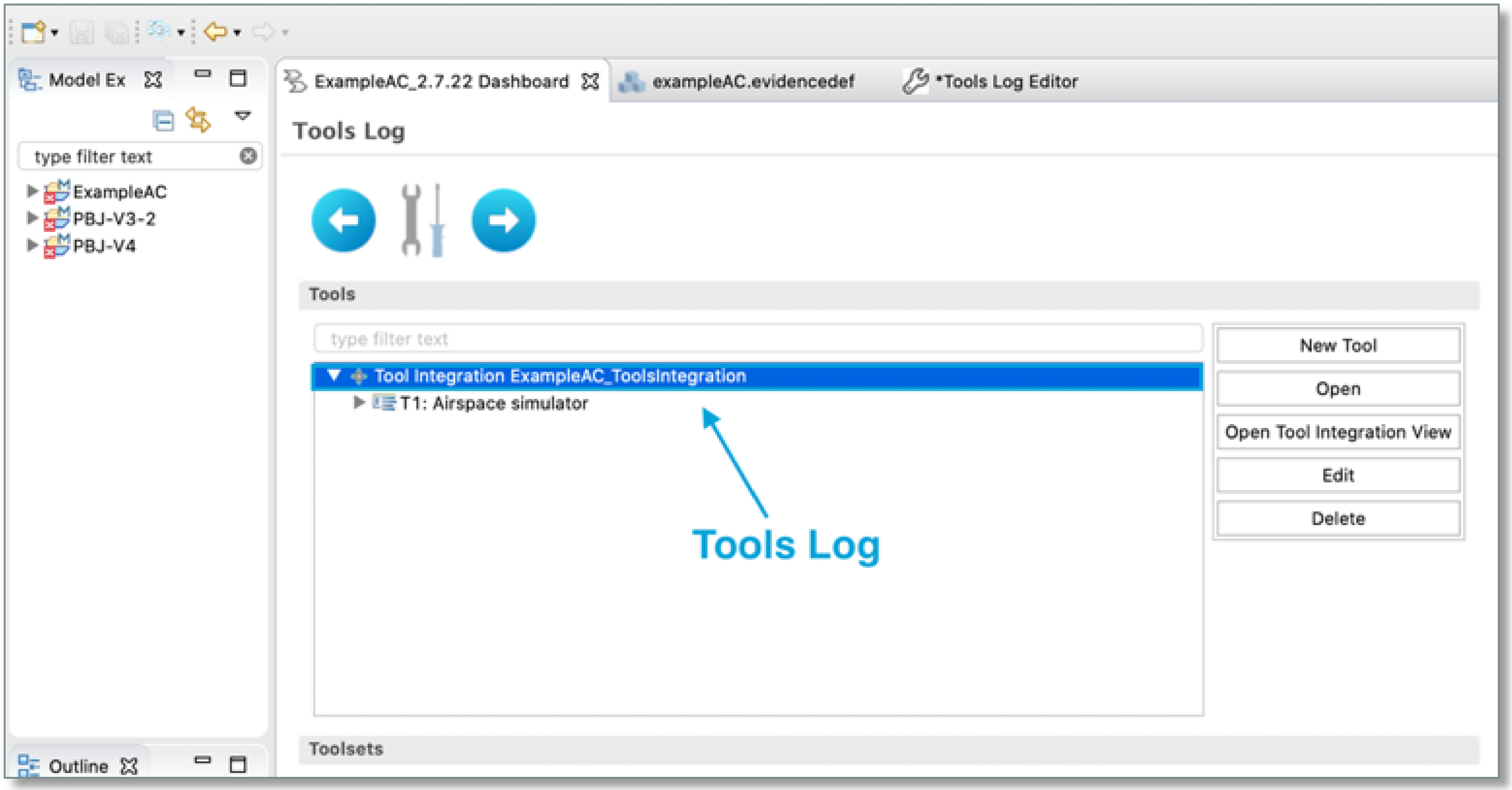


Figure 54: In the Tools Dashboard, you can select the Tools Log in the Tools window.

Tools Table Entries

The Tools Table has the following entries:

- *Tool ID*
- *Tool Description*
- *Tool Toolset*
- *Tool Inputs*
- *Tool Outputs*

Tool ID

Tool IDs are the identification marker for each **Tool**.

These IDs are auto-generated as new Tool entries are created, and they allow for **Traceability** between the **Tools Table**, **Tools Use Table**, and **Toolset Table**.

Create a New Tool

To create a new Tool

1. Open your **Tools Table**.
2. Click the **Add Element** icon in toolbar.

A new Tool entry will generate in the Tools Table.

Alternatively, you can create a new Tool from the **Tools Dashboard**.

Delete a Tool

To delete a Tool

1. Open your **Tools Table**.
2. In the Tools Log Editor, select the **Tool** entry you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected Tool entry will be deleted from the Tools Table.

Alternatively, you can delete a Tool from the **Tools Dashboard**.

Tool Description

Tool Descriptions define the purpose of the Tool.

Edit a Tool Description

To edit a Tool Description from the **Tools Log Editor**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Description cell** you wish to edit.
3. Enter your edit/s into the textbox.
4. Click **outside the cell** you edited to update your Tool Description.

The text in your table cell will update to reflect the change.

Alternatively, you can edit a Tool from the **Tools Dashboard**.

Tool Toolset

The **Tool Toolset** is the Toolset associated to the Tool in the Tools Table. The **Toolset ID** interconnects the **Tools Table**, the **Tools Use Table**, and the **Toolset Table**.

Create a New Toolset in the Tools Table

To create a new Toolset in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Toolset cell** you wish to populate.
3. Enter a new value in the **Create a New Toolset…** textbox field.
4. Click **outside the cell** you populated to create your Toolset.

The text in your table cell will update to reflect the change.

Alternatively, you can create a new Toolset from the **Tools Dashboard**.

Add an Existing Toolset in the Tools Table

To add an existing Toolset in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Toolset cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding **Toolset entry** wish to add.
4. Click **outside the cell** you edited to add the Toolset to your Tool.

The text in your table cell will update to reflect the change.

Remove a Toolset in the Tools Table

To remove a Toolset in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Toolset cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding **Toolset entry** you wish to remove.
4. Click **outside the cell** you edited to remove the Toolset from the Tool.

The text in your table cell will update to reflect the change.

Alternatively, you can delete a Toolset from the **Tools Dashboard**.

Tool Input Types

Tool Input Types are the types of **Evidence Artifacts** a Tool needs to implement the Tool's Tool Use.

Create a New Tool Input Type

To create a new Tool Input Type in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Input Types cell** you wish to populate.
3. Enter a new value in the **Create a New Input Type…** textbox field.
4. Click **outside the cell** you populated to create your Input Type.

The text in your table cell will update to reflect the change.

NOTE: If you have not already created an **Evidence Log**, a dialog window will open. For more information about the Evidence Log, see the *Evidence Log* section.

Add Existing Tool Input Types

To add existing Tool Input Types in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Input Types cell** you wish to populate.
3. In the drop-down menu, select the **checkbox/es** for the corresponding Input **Evidence Type/s** you wish to add.
4. Click **outside the cell** you edited to add the Input Type/s to your Tool.

The text in your table cell will update to reflect the change.

Alternatively, you can add existing Tool Input Types from the **Tools Dashboard** by using the **Edit Tool Input Types** button.

Edit Tool Input Types

To edit Tool Input Types in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Input Types cell** you wish to edit.
3. Enter your edits in the **Input Types cell** textbox field.
4. Click **outside the cell** you edited to update the Input Type.

The text in your table cell will update to reflect the change.

Remove Tool Input Types

To remove Tool Input Types in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Input Types cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox/es** for the corresponding Input **Evidence Type/s** you wish to remove.
4. Click **outside the cell** you edited to remove the Input Type/s from the Tool.

The text in your table cell will update to reflect the change.

Alternatively, you can remove Tool Input Types from the **Tools Dashboard** by using the **Edit Tool Input Types** button.

Tool Output Types

Tool Outputs are a list of **Evidence Types** that dispense from the Tool's Tool Use.

Create a New Tool Output Type

To create a new Tool Output Type in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Output Types cell** you wish to populate.
3. Enter a new value in the **Create a New Output Type…** textbox field.
4. Click **outside the cell** you populated to create your Output Type.

The text in your table cell will update to reflect the change.

NOTE: If you have not already created an **Evidence Log**, a dialog window will open. For more information about the Evidence Log, see the *Evidence Log* section.

Add Existing Tool Output Types

To add existing Tool Output Types in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Output Types cell** you wish to populate.
3. In the drop-down menu, select the **checkbox/es** for the corresponding Output **Evidence Type/s** you wish to add.
4. Click **outside the cell** you edited to add the Output Type/s to your Tool.

The text in your table cell will update to reflect the change.

Alternatively, you can add existing Tool Output Types from the **Tools Dashboard** by using the **Edit Tool Output Types** button.

Edit Tool Output Types

To edit Tool Output Types in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Output Types cell** you wish to edit.
3. Enter your edits in the **Output Types cell** textbox field.
4. Click **outside the cell** you edited to update the Output Type.

The text in your table cell will update to reflect the change.

Remove Tool Output Types

To remove Tool Output Types in the **Tools Table**

1. Open your **Tools Table**.
2. In the Tools Log Editor, double-click the **Output Types cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox/es** for the corresponding Output **Evidence Type/s** you wish to remove.
4. Click **outside the cell** you edited to remove the Output Type/s from the Tool.

The text in your table cell will update to reflect the change.

Alternatively, you can remove Tool Output Types from the **Tools Dashboard** by using the **Edit Tool Output Types** button.

Tools Use Table

The **Tools Use Table** declares the application of a Tool to a specific **Evidence Artifact** in your AdvoCATE Project called a **Tool Use**. The Tools Use Table can be sorted by the Tool and Toolset columns.

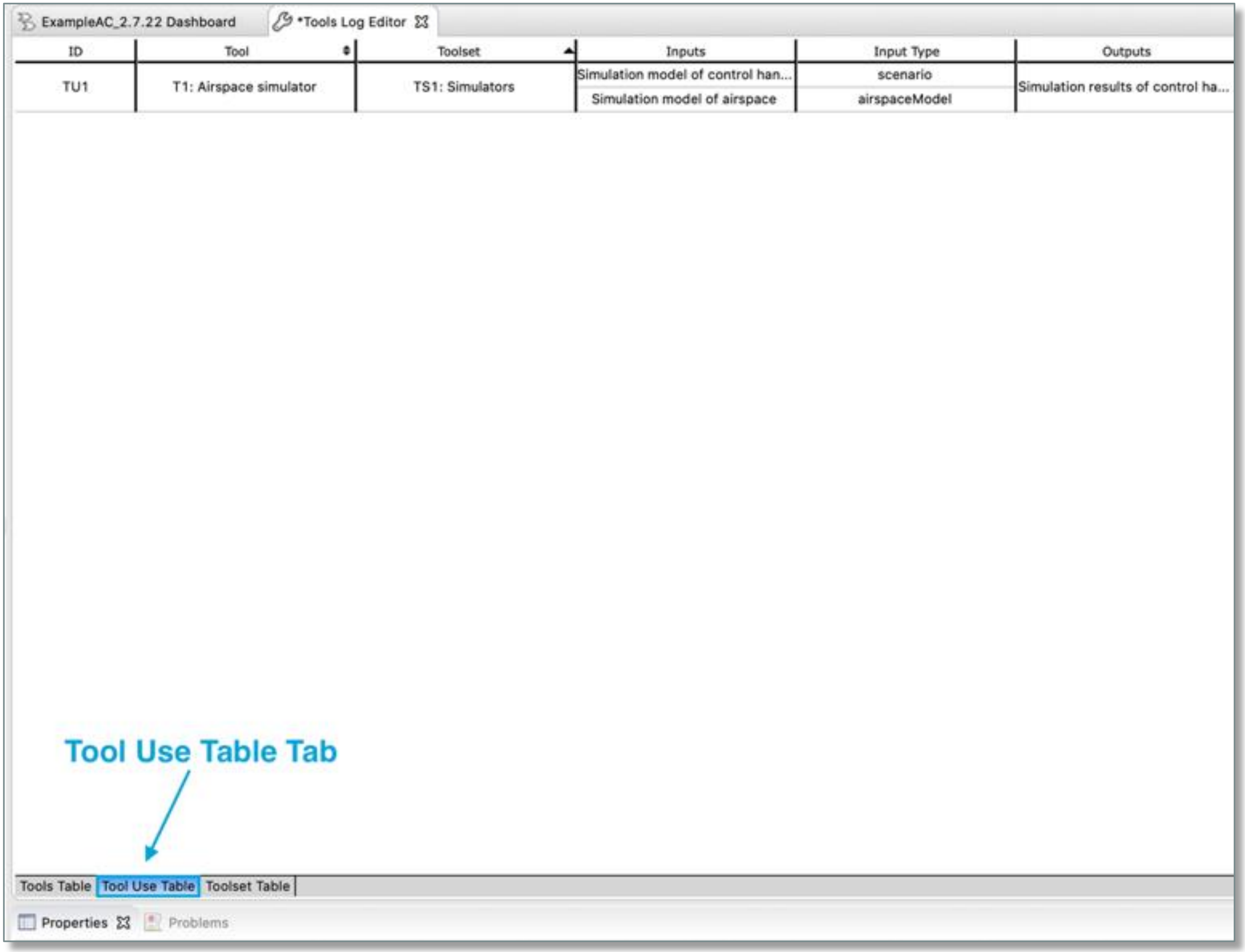
Open the Tools Use Table

To open the Tools Use Table

1. Open the **Tools Table**.
2. In the Tools Log Editor, select the **Tools Use Table** tab (*Figure 55*).

The Tools Use Table will open.

Alternatively, you can open the Tool Use Table from the **Tools Dashboard**.



ID	Tool	Toolset	Inputs	Input Type	Outputs
TU1	T1: Airspace simulator	TS1: Simulators	Simulation model of control han...	scenario	Simulation results of control ha...
			Simulation model of airspace	airspaceModel	

Figure 55: In the Tools Log Editor, select the Tools Use Table tab to open the Tools Use Table.

Tools Use Table Entries

The Tools Use Table has the following entries:

- *Tool Use ID*
- *Tool Use Tool*
- *Tool Use Toolset*
- *Tool Use Inputs*
- *Tool Use Input Type*
- *Tool Use Outputs*
- *Tool Use Output Type*

Tool Use ID

Tool Use IDs are the identification marker for each **Tool Use**.

Tool Use IDs are auto-generated as new entries are created, and they allow for **Traceability** between the **Tools Table**, **Tools Use Table**, and **Toolset Table**.

Create a New Tool Use

To create a new Tool Use

1. Open your **Tool Use Table**.
2. Click the **Add Element** icon in toolbar.
3. In the pop-up window, select the **Tool** you wish to create a Tool Use for.
4. Click **Ok**.

A new Tool Use entry will generate in the Tools Use Table.

NOTE: You must have a Tool entry in the Tools Table to create a Tool Use.

Alternatively, you can create a new Tool Use from the **Tools Dashboard**.

Delete a Tool Use

To delete a Tool Use

1. Open your **Tool Use Table**.
2. In the Tools Log Editor, select the **Tool Use** entry you wish to delete.
3. Click the **Remove Element** icon in toolbar.
4. Click **Yes** to confirm your deletion request.

Your selected Tool Use entry will be deleted from the Tools Use Table.

Alternatively, you can delete a Tool Use from the **Tools Dashboard**.

Tool Use Tool

Tool Use Tool entries reference the **Tool** that is being used from the **Tools Table** and will auto-populate based off the Tool entry in the Tools Table.

Tool Use Toolset

The **Tool Use Toolset** is the Toolset associated to the **Tool Use** in the **Tools Use Table** and will auto-populate based off the Tool's Toolset entry in the Tools Table.

Tool Use Inputs

Tool Use Inputs are the **Evidence Artifacts** needed to use a Tool.

Create a New Tool Use Input

To create a new Tool Use Input in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Input cell** you wish to populate.
3. Enter a new value in the **Create a New Input Use…** textbox field.
4. Click **outside the cell** you populated to create your Tool Use Input.

The text in your table cell will update to reflect the change.

Add an Existing Tool Use Input

To add an existing Tool Use Input in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Input cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding Input **Evidence Artifact** you wish to add.
4. Click **outside the cell** you edited to add the Tool Use Input.

The text in your table cell will update to reflect the change.

Alternatively, you can add an existing Tool Use Input from the **Tools Dashboard** by using the **Edit Tool Use Inputs** button.

Edit a Tool Use Input

To edit a Tool Use Input in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Input cell** you wish to edit.
3. Enter your edits in the **Input cell** textbox field.
4. Click **outside the cell** you edited to update the Tool Use Input.

The text in your table cell will update to reflect the change.

Remove a Tool Use Input

To remove a Tool Use Input in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Input cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding Input **Evidence** you wish to remove.
4. Click **outside the cell** you edited to remove the Tool Use Input.

The text in your table cell will update to reflect the change.

Alternatively, you can remove a Tool Use Input from the **Tools Dashboard** by using the **Edit Tool Use Inputs** button.

Tool Use Input Type

Tool Use Input Types reference the **Tool Input Types** that are being used from the **Tools Table**. A Tool Use Input Type will auto-populate based off the Evidence Artifact Type of the Tool Use Input.

NOTE: The Evidence Artifact must have **Type** listed in the **Evidence Log** to auto-populate the Tools Use Table.

Tool Use Output

Tool Use Outputs are the **Evidence Artifacts** generated from using a Tool.

Create a New Tool Use Output

To create a new Tool Use Output in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Output cell** you wish to populate.
3. Enter a new value in the **Create a New Output…** textbox field.
4. Click **outside the cell** you populated to create your Tool Use Output.

The text in your table cell will update to reflect the change.

Add an Existing Tool Use Output

To add an existing Tool Use Output in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Output cell** you wish to populate.
3. In the drop-down menu, select the **checkbox** for the corresponding Output **Evidence Artifact** you wish to add.
4. Click **outside the cell** you edited to add the Tool Use Output.

The text in your table cell will update to reflect the change.

Alternatively, you can add an existing Tool Use Output from the **Tools Dashboard** by using the **Edit Tool Use Outputs** button.

Edit a Tool Use Output

To edit a Tool Use Output in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Output cell** you wish to edit.
3. Enter your edits in the **Output cell** textbox field.
4. Click **outside the cell** you edited to update the Tool Use Output.

The text in your table cell will update to reflect the change.

Remove a Tool Use Output

To remove a Tool Use Output in the **Tools Use Table**

1. Open your **Tools Use Table**.
2. In the Tools Log Editor, double-click the **Output cell** you wish to edit.
3. In the drop-down menu, deselect the **checkbox** for the corresponding Output **Evidence Artifact** you wish to remove.
4. Click **outside the cell** you edited to remove the Tool Use Output.

The text in your table cell will update to reflect the change.

Alternatively, you can remove a Tool Use Output from the **Tools Dashboard** by using the **Edit Tool Use Outputs** button.

Tool Use Output Type

Tool Use Output Types reference the **Tool Output Type** that is being used from the **Tools Table**. A Tool Use Output Type will auto-populate based off the Evidence Artifact Type of the Tool Use Output.

NOTE: The Evidence Artifact must have **Type** listed in the **Evidence Log** to auto-populate the Tools Use Table.

Toolset Table

The **Toolset Table** shows a list of all the Toolsets for an AdvoCATE Project.

A **Toolset** is a collection of related Tools. Some Toolset examples are when Tools are typically used together or even different modes of a single Tool (when it is natural to model each mode as a distinct Tool).

The **Toolset ID** interconnects the **Tools Table**, the **Tools Use Table**, and the **Toolset Table**. The Toolset ID is automatically generated once a new Toolset is created in the Tools Table.

The **Toolset Description** is the name of the Toolset. The Toolset Description is auto-populated based on the Toolset Description entry created in the Tools Table.

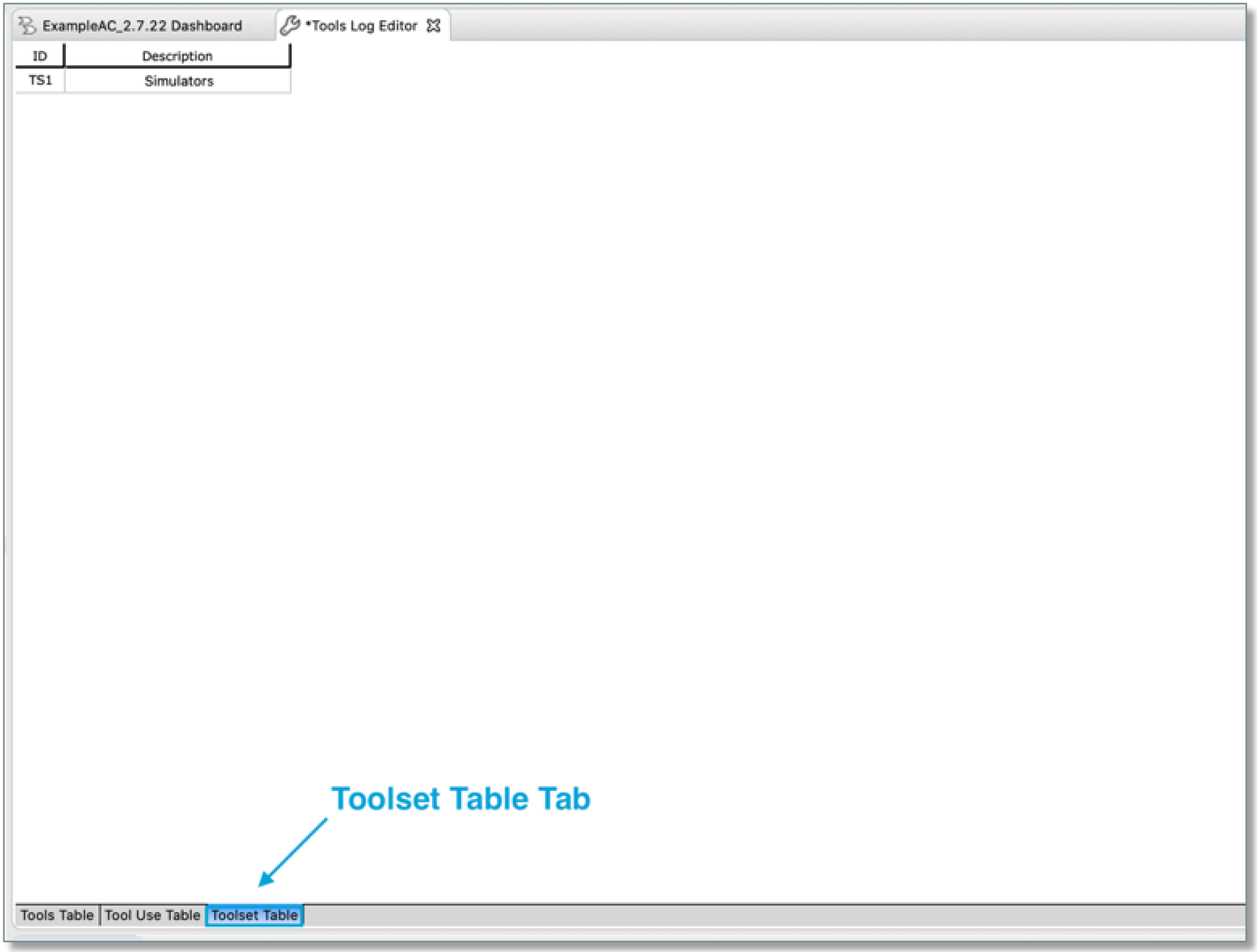
Open the Toolset Table

To open the Toolset Table

1. Open the **Tools Table**.
2. In the Tools Log Editor, select the **Toolset Table** tab (*Figure 56*).

The Toolset Table will open.

Alternatively, you can open the Toolset Table from the **Tools Dashboard**.



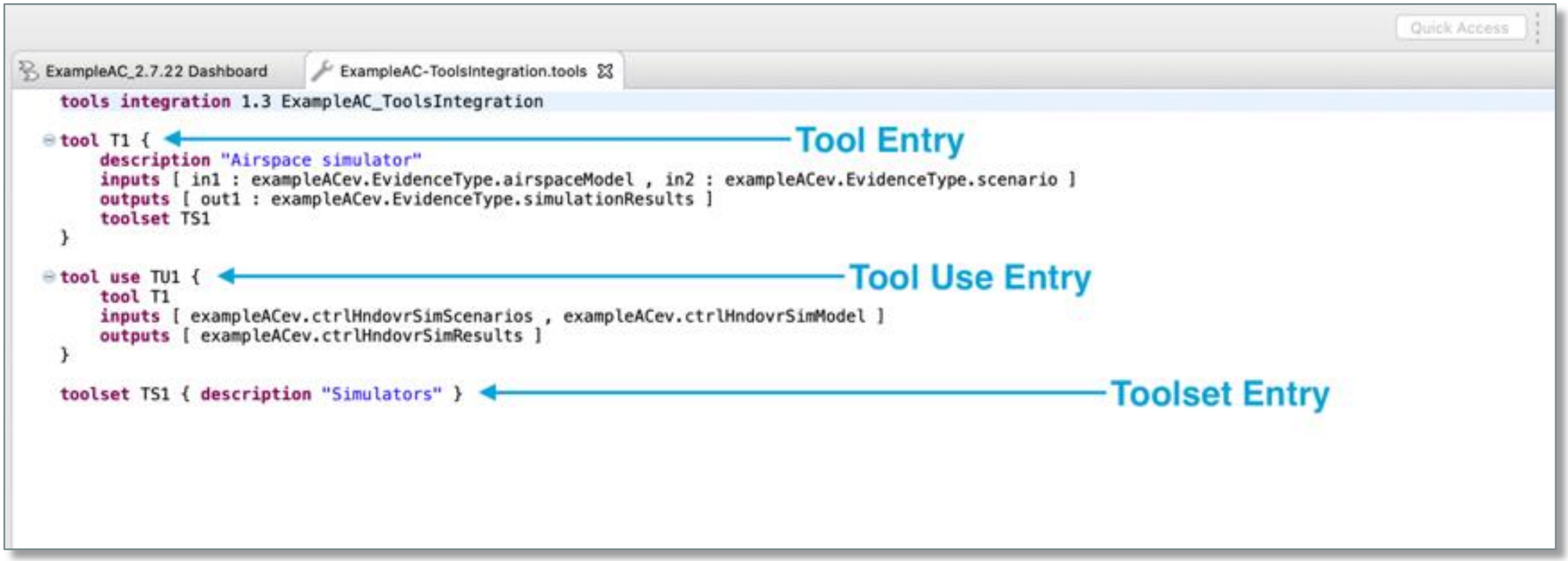
ID	Description
TS1	Simulators

Figure 56: In the Tools Log Editor, select the Toolset Table tab to open the Toolset Table.

Tools Log DSL

The **Tools Log domain-specific language (DSL)** can also be used to define and declare **Tools**, **Tool Uses**, and **Toolsets** (*Figure 57*).

The Tools Log DSL starts with the tools integration keyword, followed by the syntax version number and name of the Tools Log.



```
tools integration 1.3 ExampleAC_ToolsIntegration

tool T1 {
    description "Airspace simulator"
    inputs [ in1 : exampleACev.EvidenceType.airspaceModel , in2 : exampleACev.EvidenceType.scenario ]
    outputs [ out1 : exampleACev.EvidenceType.simulationResults ]
    toolset TS1
}

tool use TU1 {
    tool T1
    inputs [ exampleACev.ctrlHndovrSimScenarios , exampleACev.ctrlHndovrSimModel ]
    outputs [ exampleACev.ctrlHndovrSimResults ]
}

toolset TS1 { description "Simulators" }
```

Figure 57: In the Tools Log Editor, you can view and edit Tools, Tool Uses, and Toolsets.

Tools

Tool entries are how you define Tools and their Toolsets.

Tool entries start with the tool keyword, followed by an identifier. Each Tool entry is then enclosed in {} brackets.

Tool entries have multiple fields and can list the following details:

- **Description**: uses the description keyword to describe the Tool and is enclosed with “” quotation marks.
- **Inputs**: uses the inputs keyword to list the Evidence Type/s the Tool uses to generate Outputs.
    - The Tool Inputs entry is enclosed in [ ] square brackets.
    - Each Evidence Type uses an in# input number, followed by a : colon.
    - The Tool Input entry should then reference the Evidence file name, the Evidence Type list, and Evidence Type (each followed by a . period).
    - If your Tool uses multiple Evidence Types, enclosed each Evidence Type with a , comma.
- **Outputs**: uses the outputs keyword to list the Evidence Type/s the Tool generates.
    - The Tool Outputs entry is enclosed in [ ] square brackets.
    - Each Evidence Type uses an out# output number, followed by a : colon.
    - The Tool Output entry should then reference the Evidence file name, the Evidence Type list, and Evidence Type classification (each followed by a . period).
    - If your Tool uses multiple Evidence Types, offset each Evidence Type with a , comma.
- **Toolset**: uses the toolset keyword and then lists the Tool’s corresponding Toolset identifier.

Tool Uses

Tool Use entries are how you declare the application of a Tool to one or more **Evidence Artifacts** in your AdvoCATE Project called a **Tool Use**.

Tool Use entries start with the **tool use** keyword, followed by an identifier. Each Tool entry is then enclosed in {} brackets.

Tool Use entries have multiple fields and can list the following details:

- **Tool**: uses the **tool** keyword to list the Tool identifier associated to the Tool Use.
- **Inputs**: uses the **inputs** keyword to list the Evidence Artifact Definition the Tool Use uses to generate Outputs.
  - The Tool Use Inputs entry is enclosed in [ ] square brackets.
  - The Tool Use Input entry should then reference the Evidence file name, followed by a . period, and then the Evidence Artifact Definition identifier the Tool Use uses.
  - If your Tool uses multiple Evidence Types, offset each Evidence Type with a , comma.
- **Outputs**: uses the **outputs** keyword to list the Evidence Artifact Definition the Tool Use generates.
  - The Tool Outputs entry is enclosed in [ ] square brackets.
  - The Tool Use Output entry should then reference the Evidence file name, followed by a . period, and then the Evidence Artifact Definition identifier the Tool Use uses.
  - If your Tool uses multiple Evidence Types, offset each Evidence Type with a , comma.

Toolsets

Toolset entries declare the identifier and description of each Toolset. The Toolset identifier interconnects the **Tools Table**, the **Tools Use Table**, and the **Toolset Table**.

Toolset entries start with the toolset keyword, followed by an identifier. Within the toolset entry's {} brackets, each Toolset description is then enclosed within “” quotation marks.

Open Tool Integration View

To open the Tool Integration View

1. Open the **Tools Dashboard**.
2. In the Tools window, select the **Tools Log**.
3. Click the **Open Tool Integration View** button.

The Tool Integration View will open a single diagram that represents the integration of all tools in all Toolsets in an AdvoCATE Project.

Open Toolset Integration View

To open the Toolset Integration View

1. Open the **Tools Dashboard**.
2. In the Toolset window, select the **Toolset** you wish to open in the Toolset Integration View.
3. Click the **Open Toolset Integration View** button.

The Toolset Integration View will open a single diagram that represents the integration of all tools in the selected Toolset.

Allocations

Certain associations between assurance artifacts in AdvoCATE are called **Allocations**. These associations can be thought of as one element carrying some responsibility for another, in various senses. For example, an Allocation might represent the item which is to "blame" for a Hazard, or a Requirement which shows how to implement something. Depending on the particular Allocation, the allocated element may or may not be unique.

There are three main AdvoCATE assurance artifacts where these Allocations occur:

1. *Hazard Log Allocations*
2. *Requirements Log Allocations*
3. *Safety Architecture Allocations*

Hazard Log Allocations

A **Hazard Allocation** is the Component or Function where the Hazard is located. For more information about Hazard Allocations in the Hazard Log, see the *Hazard Allocation* section.

When a Hazard Allocation is associated to a Component or Function, the item is associated to an Event and all of Event's instances. These **Event Allocations** can be edited in the Safety Architecture Dashboard.

A **Hazard Source Allocation** is where a Hazard can originate or occur from. For more information about Hazard Source Allocations in the Hazard Log, see the *Hazard Sources* section.

A **BHA Allocation** is when an Event Instance is associated to a Barrier as a result from a BHA. BHA Allocations are not explicitly shown in the Hazard Log. They only appear implicitly in the description of the failure hazard that is generated (ex: "barrier is breached due to failure"). For more information about BHA, see the *Barrier Hazard Analysis* section.

Requirements Log Allocations

A **Requirement Allocation** associates a Requirement to a Function or Component that implements said Requirement. The Requirement Allocation is set in Allocation column in the Requirements Log. For more information about Requirement Allocations, see the *Requirement Allocation* section.

A **Verification Allocation** associates a Requirement to an Evidence Artifact that provides Evidence that the Requirement is met. The Verification Allocation is set in the Verification Allocation column in the Requirements Log. For more information about Verification Allocations, see the *Requirement Verification Allocation* section.

A **Requirement Source Allocation** is a person, organization, or a document from which a Requirement originates from or can be obtained. For more information about Requirement Source Allocations in the Requirements Log, see the *Requirement Source* section.

Safety Architecture Allocations

A **Barrier Requirement Allocation** associates a Barrier Node to a Requirement that describes the intended functionality of said Barrier. The Barrier Requirement Allocation is set in the Barrier Node’s Property View in a CES or BTD.

A **Barrier Functional Allocation** associates a Barrier Node to a Function that implements said Barrier. The Barrier Functional Allocation can be set in the Safety Architecture Dashboard or in the Barrier Node’s Property View in a CES or BTD.

A **Barrier Physical Allocation** associates a Barrier Node to a Component that implements said Barrier. The Barrier Physical Allocation can be set in the Safety Architecture Dashboard or in the Barrier Node’s Property View in a CES or BTD.

A **Control Requirement Allocation** associates a Control Node to a Requirement that describes the intended functionality of said Control. The Control Requirement Allocation is set in the Control Node’s Property View in a CES or BTD.

A **Control Functional Allocation** associates a Control Node to a Function that implements said Control. The Control Functional Allocation can be set in the Safety Architecture Dashboard or in the Control Node’s Property View in a CES or BTD.

A **Control Physical Allocation** associates a Control Node to a Component that implements said Control. The Control Physical Allocation can be set in the Safety Architecture Dashboard or in the Control Node’s Property View in a CES or BTD.

Traceability & Navigation

In systems and software engineering, **Traceability** refers to relationships that are established, either explicitly or implicitly, between work products at various phases of the project lifecycle.

For assurance cases in AdvoCATE, you can establish traceability links between various assurance artifacts, which reflect various semantic properties of the assurance case, and then navigate between those artifacts. More broadly, **Navigation** allows users to move between related but distinct artifacts, as well as different representations of the same underlying artifact.

There are six main AdvoCATE assurance artifacts where this traceability and navigation occurs:

1. *Hazard Log Tracing*
2. *Requirements Log Tracing*
3. *CESs/BTDs Tracing*
4. *Argument Structures Tracing*
5. *Evidence Log Tracing*
6. *Tools Tracing*

Hazard Log Tracing

Hazard Log to Requirements Log

Mitigation Requirement & Requirement Associations

Mitigation Requirements in the Hazard Log are linked implicitly as **Requirements** in the Requirements Log by the same identifier. When you create a new Mitigation Requirement in the Hazard Log, the corresponding Requirement will auto-populate in the Requirements Log.

Requirement identifiers can be seen in the Hazard Log's Mitigation Requirements column and the Requirements Log's ID column.

Mitigation Verification & Verification Method Associations

Mitigation Verifications in the Hazard Log are linked implicitly to the **Verification Methods** in the Requirements. When you create a new Mitigation Verification in the Hazard Log, the corresponding Verification Method will auto-populate in the Requirements Log.

Navigate from Mitigation Requirement to Requirement

To navigate from a Mitigation Requirement cell to its associated Requirement in the Requirements Log

1. Open the **Hazard Log**.
2. Right-click the **Mitigation Requirement cell**.
3. Select **Open Selected Requirement** from the right-click menu.

The Requirements Log will open with the associated Requirement’s row highlighted in dark gray.

Navigate from Hazard to Requirement Associations

To navigate from a Hazard to its associated Requirements in the Requirements Log

1. Open the **Hazard Log**.
2. Right-click the **Hazard cell**.
3. Select **Open All Hazard Requirements** from the right-click menu.

The Requirements Log will open with the associated Requirements’ rows highlighted in dark gray.

Hazard Log to CESs/BTDs

Hazard Log & Safety Architecture Element Associations

Hazardous Activities, **Hazards**, **Causes**, **Mitigations**, and **Effects** listed in the Hazard Log are implicitly linked with their CES/BTD node counterparts. When you create and edits these cells in the Hazard Log, the corresponding CESs and any created BTDs will update accordingly.

Mitigation Requirements & Barrier/Control Allocations

Mitigation Requirements (or Requirements) can be allocated to Barriers and Controls.

A **Barrier Requirement Allocation** associates a Barrier Node to a Requirement that describes the intended functionality of said Barrier.

A **Control Requirement Allocation** associates a Control Node to a Requirement that describes the intended functionality of said Control.

Navigate from Hazard to BTD Top Event

To navigate from a Hazard row to a BTD where that Event is depicted as a Top Event

1. Open the **Hazard Log**.
2. Right-click a **table cell** in the **Hazard row**.
3. Select **Open Corresponding Bowtie Diagram…** > **Where this Hazard is the Top Event** from the right-click menu.

The BTD/s where the Event is depicted as a Top Event will open in the BTD Diagram Editor with the Top Event Node selected.

Navigate from Hazard to BTD Threat Event

To navigate from a Hazard row to a BTD where that Event is depicted as a Threat Event

1. Open the **Hazard Log**.
2. Right-click a **table cell** in the **Hazard row**.
3. Select **Open Corresponding Bowtie Diagram…** > **Where this Hazard is a Threat** from the right-click menu.

The BTD/s where the Event is depicted as a Threat will open in the BTD Diagram Editor with the Threat Event Node selected.

NOTE: The **Cause column** in the Hazard Log is where Threats are listed for Hazards.

Navigate from Hazard to BTD Consequence Event

To navigate from a Hazard row to a BTD where that Event is depicted as a Consequence Event

1. Open the **Hazard Log**.
2. Right-click a **table cell** in the **Hazard row**.
3. Select **Open Corresponding Bowtie Diagram…** > **Where this Hazard is a Consequence** from the right-click menu.

The BTD/s where the Event is depicted as a Consequence will open in the BTD Diagram Editor with the Consequence Event Node selected.

NOTE: The **Effects Description column** in the Hazard Log is where the most severe Consequences are listed for Hazards.

Navigate from Mitigation Requirement to BTD Barrier

To navigate from a Mitigation Requirement cell to its associated BTD Barrier/s

1. Open the **Hazard Log**.
2. Right-click the **Mitigation Requirement cell**.
3. Select **Open Related Barrier in BowTie** from the right-click menu. *If there are multiple BTDs where the Barrier is used, a selection window will display the associated BTDs for you to choose to open.* If there are multiple options to choose from, continue with steps 4-5.
4. Select the **BTD/s** you wish to open in the popup window.
5. Click **Ok**.

The BTD/s will open in the BTD Diagram Editor.

Navigate from Mitigation Requirement to BTD Control

To navigate from a Mitigation Requirement cell to its associated BTD Control/s

1. Open the **Hazard Log**.
2. Right-click the **Mitigation Requirement cell**.
3. Select **Open Related Control in BowTie** from the right-click menu. *If there are multiple BTDs where the Control is used, a selection window will display the associated BTDs for you to choose to open.* If there are multiple options to choose from, continue with steps 4-5.
4. Select the **BTD/s** you wish to open in the popup window.
5. Click **Ok**.

The BTD/s will open in the BTD Diagram Editor.

Hazard Log to Argument Structures

Hazard & Argument Associations

Hazardous Activities and **Hazards** can be associated to Argument Structures or GSN Nodes that describe or reference said Hazardous Activity or Hazard.

Associate Hazard to an Argument Structure or GSN Nodes

To associate a Hazard to a GSN Node or an Argument Structure

1. Open the **Hazard Log**.
2. Right-click the **table cell** in the Hazard Log you wish to edit Argument Associations for.
3. Select **Edit Argument Node associations…** from the right-click menu.
4. In the pop-up window, select or deselect the checkboxes next to an Argument Structure or specific GSN Nodes.
5. Click **Ok**.

The Argument Structure and/or GSN Node/s will be associated to the selected Hazard.

Navigate from Hazard to GSN Nodes

To navigate from a Hazard to associated GSN Nodes

1. Open the **Hazard Log**.
2. Right-click the **table cell** in the Hazard Log you wish to view as a GSN Node in an Argument Structure.
3. Select **Open argument…** > **Open argument node associated with hazard…** from the right-click menu.

The corresponding Argument Structure will open with the associated GSN Node/s selected.

Navigate from Hazardous Activity to an Argument Structure

To navigate from a Hazardous Activity to an associated Argument Structure

1. Open the **Hazard Log**.
2. Right-click a **table cell** in the Hazard Log to view the Hazard Activity's associated Argument Structure.
3. Select **Open argument…** > **Open argument diagram associated with hazardous activity** from the right-click menu.

The corresponding Argument Structure will open in the Argument Diagram Editor.

Hazard Log to Evidence Log

In future AdvoCATE updates, traceability options from Risk Rationale and Likelihood entries in the Hazard Log to associated Evidence Artifacts will be created.

Hazard Log to Tools Log

In future AdvoCATE updates, traceability options from Risk Rationale and Likelihood entries in the Hazard Log to associated Evidence Artifacts that are generated by a Tool Use will be created.

Hazard Log to PhysArch & FuncSpec

Hazard Allocations & Components/Functions

Allocations in the Hazard Log are implicitly linked to **Components** and **Functions** listed in PhysArch and FuncSpec. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

Hazard Conditions & Failure Modes/Deviations

Conditions in the Hazard Log are implicitly linked to **Failure Modes** and **Deviations** listed in PhysArch and FuncSpec. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

Navigate from Hazard Log to the PhysArch/FuncSpec

To navigate from a Hazard Log association to the PhysArch or FuncSpec

1. Open the **Hazard Log**.
2. Right-click the **Allocation** or **Condition cell** in the Hazard Log you wish to view in the PhysArch or FuncSpec.
3. Select **Open Function Decomposition DSL Editor** or **Open Physical Decomposition DSL Editor** from the right-click menu. *If the selected table cell lists a Component or Failure Mode entry, the* ***Open Physical Decomposition DSL Editor*** *option will be present in the right-click menu. If the selected table cell lists a Function or Deviation entry, the* ***Open Function Decomposition DSL Editor*** *option will be present in the right-click menu.*

The corresponding PhysArch or FuncSpec will open with the associated entry highlighted.

Requirements Log Tracing

Requirements Log to Hazard Log

Requirement & Mitigation Requirement Associations

Requirements in the Requirements Log are linked implicitly to **Mitigation Requirements** in the Hazard Log by the same identifier. When you create or add an existing Mitigation Requirement in the Hazard Log, the corresponding Requirement will auto-populate in the Requirements Log.

Requirement identifiers can be seen in the Hazard Log's Mitigation Requirements column and the Requirements Log's ID column.

Requirement & Mitigation Associations

Requirements can be associated to Mitigations in the Hazard Log when said Requirement describes how the **Mitigation** will reduce the risk of **Cause** events.

Associate Requirement to a Mitigation

To associate a Requirement to a Mitigation in the Hazard Log

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to associate Hazard Mitigations to.
3. Select **Associate Requirement to Barrier/Mitigation** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to **Barrier/Mitigation Instances** to edit the selected Requirement's associations.
5. Click **Ok**.

The Hazard Mitigation associations for the selected Requirement will update accordingly.

Navigate from Requirement to Hazard Associations

To navigate from a Requirement to its associated Hazards in the Hazard Log

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to view associated Hazards for.
3. Select **Open associated artifact…** > **Show Associated Hazards in Hazard Log** from the right-click menu.

The Hazard Log will open with the associated Hazards highlighted in dark gray.

Requirements Log to CESs/BTDs

Requirements & Barrier/Control Allocations

Requirements can be allocated to Barriers and Controls.

A **Barrier Requirement Allocation** associates a Barrier Node to a Requirement that describes the intended functionality of said Barrier.

A **Control Requirement Allocation** associates a Control Node to a Requirement that describes the intended functionality of said Control.

Associate Requirement to a Barrier

To associate a Requirement to a Barrier

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to associate Barriers to.
3. Select **Associate Requirement to Barrier/Mitigation** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to **Barrier/Mitigation Instances** to edit the selected Requirement's associations.
5. Click **Ok**.

The Barrier associations for the selected Requirement will update accordingly.

Associate Requirement to Control

To associate a Requirement to a Control

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to associate Controls to.
3. Select **Associate Requirement to Control** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to **Control Instances** to edit the selected Requirement's associations.
5. Click **Ok**.

The Control associations for the selected Requirement will update accordingly.

Navigate from Requirement to Hazard in a BTD

To navigate from a Requirement to its associated Hazards in a BTD

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to view associated Hazards for.
3. Select **Open associated artifact…** > **Show Associated Hazard in Bowtie** from the right-click menu.

Any BTD with the associated Hazard as an Event will open in the BTD Diagram Editor.

Navigate from Requirement to Control/Barrier Instances in a CES/BTD

To navigate from a Requirement to its associated Control/Barrier Instances in a CES or BTD

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to view associated Controls/Barriers Instances for.
3. Select **Open associated artifact…** > **Show Associated Control/Barrier Instances** from the right-click menu.
4. In the pop-up window, select the **CES** or **BTD** you wish to view the associated Controls/Barrier Instances in.
5. Click **Ok**.

The selected diagram will open with the associated Control/Barrier Nodes selected.

Requirements Log to Argument Structures

Requirement & Argument Associations

Requirements can be associated to GSN Nodes that describe or reference said Requirement.

Associate Requirement to GSN Nodes

To associate a Requirement to GSN Nodes

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to associate Argument Nodes to.
3. Select **Associate Requirement to Argument Nodes** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to **Argument Node Instances** to edit the selected Requirement's associations.
5. Click **Ok**.

The Requirement will update with the selected GSN Node associations.

Navigate from Requirement to GSN Nodes

To navigate from a Requirement to its associated GSN Nodes in an Argument Structure

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to view Argument Nodes for in an Argument Structure.
3. Select **Open associated artifact…** > **Show Associated Argument Nodes** from the right-click menu.

The associated Argument Structure will open.

Navigate from Requirement to Solution Nodes

To navigate from a Requirement to its associated Solution Nodes in an Argument Structure

1. Open the **Requirements Log**.
2. Right-click the **Requirement cell** you wish to view Solution Nodes for in an Argument Structure.
3. Select **Open associated artifact…** > **Show Associated Solutions** from the right-click menu.

The associated Argument Structure will open, and the Solution Nodes with the Requirement's associated Evidence Artifacts will be selected.

Requirements Log to Evidence Log

Verification Allocations & Evidence Artifacts

Verification Allocations in the Requirements Log are linked implicitly as **Evidence Artifacts** in the Evidence Log. When you create or add an existing Verification Allocation in the Requirements Log, the drop-down menu lists Evidence Artifacts from the Evidence Log.

Navigate from Verification Allocation to Evidence Artifacts

To navigate from a Verification Allocation to its associated Evidence Artifacts in the Evidence Log

1. Open the **Requirements Log**.
2. Right-click the **Verification Allocation cell** you wish to view in the Evidence Log.
3. Select **Open Selected Verification Allocation** from the right-click menu.

The Evidence Log will open with the associated Evidence Artifact entry highlighted.

Requirements Log to Tools Log

Verification Allocations & Tool Uses

Verification Allocations in the Requirements Log are linked implicitly as **Evidence Artifacts**. When the associated Evidence Artifact is generated by a Tool Use, the Verification Allocation becomes implicitly linked to the Tool in the Tools log.

Requirements Log to PhysArch & FuncSpec

Requirement Allocations & Components/Functions

Allocations in the Requirements Log are implicitly linked to **Components** and **Functions** listed in PhysArch and FuncSpec. For more information about PhysArch and FuncSpec files, see the *Physical Architecture* and *Functional Specifications* sections.

Navigate from Requirement Allocations to the PhysArch/FuncSpec

To navigate from a Requirement Allocation to the PhysArch or FuncSpec

1. Open the **Requirements Log**.
2. Right-click the **Allocation cell** in the Requirements Log you wish to view in the PhysArch or FuncSpec.
3. Select **Open Function Decomposition DSL Editor** or **Open Physical Decomposition DSL Editor** from the right-click menu. *If the selected table cell lists a Component entry, the **Open Physical Decomposition DSL Editor** option will be present in the right-click menu. If the selected table cell lists a Function entry, the **Open Function Decomposition DSL Editor** option will be present in the right-click menu.*

The corresponding PhysArch or FuncSpec will open with the associated entry highlighted.

CES/BTD Tracing

CES/BTD to Hazard Log

Safety Architecture Elements & Hazard Log Associations

Hazardous Activities, **Top Events**, **Threats**, **Barriers**, and **Consequences** listed in CESs/BTDs are implicitly linked with their Hazard Log counterparts. When you create and edits these nodes in a CES/BTD, the corresponding table cells in the Hazard Log will update accordingly.

Navigate from CES/BTD Node to Hazard Log

To navigate from a CES/BTD Node to an association in the Hazard Log

1. Open the **CES or BTD Diagram Editor**.
2. Right-click a **CES or BTD Node**.
3. Select **Open in** > **Hazard Log** from the right-click menu.

The Hazard Log will open with the associated CES/BTD elements highlighted in dark gray.

NOTES:

- CES/BTD elements will be highlighted in the Hazard Log only in the specific position that corresponds to their context in the specific CES/BTD, even if they are visible there in another capacity. Hence, when the Top Event of the BTD is not set to be visible in the Hazard Log, navigating to the Hazard Log will not highlight any table cells.
- You can also use the CES and BTD Barrier/Control Node's Properties View to navigate to the Hazard Log.

Navigate from CES/BTD Requirement to Hazard Log

To navigate from a CES/BTD Node associated Requirement to its Mitigation Requirement association in the Hazard Log

1. Open the **CES or BTD Diagram Editor**.
2. Select a **CES or BTD Mitigation Node**.
3. Select the **Control or Barrier tab** in the Properties View.
4. In the Control or Barrier's Properties View, locate the **Associated Requirements**.
5. Select an associated **Requirement**.
6. Click the **Open in Hazard Log button**.

The Hazard Log will open with the associated **Mitigation Requirement** highlighted in dark gray.

CES/BTD to Requirements Log

Barrier & Control Requirement Allocations

A **Barrier Requirement Allocation** associates a Barrier Node to a Requirement that describes the intended functionality of said Barrier.

A **Control Requirement Allocation** associates a Control Node to a Requirement that describes the intended functionality of said Control.

Associate CES/BTD Barrier Nodes to Requirements

To associate a CES/BTD Barrier Node to a Requirement in the Requirements Log

1. Open the **CES or BTD Diagram Editor**.
2. From the Association Links Palette, select the **Barrier Requirement Associations** tool.
3. Click a **Barrier Node** you wish to edit the associated Requirements for.
4. Use the **pop-up menu** to add or remove selected **Requirements** as Barrier Associations.
5. Click **Finish**.

The CES/BTD Barrier Node's Requirement associations will update accordingly.

NOTE: You can also edit CES/BTD Barrier Node associations to the Requirements Log using the **Associate Requirements button** in the CES/BTD Barrier Node's Properties View.

Associate CES/BTD Control Nodes to Requirements

To associate a CES/BTD Control Node to a Requirement in the Requirements Log

1. Open the **CES or BTD Diagram Editor**.
2. From the Association Links Palette, select the **Control Requirement Associations** tool.
3. Click a **Control Node** you wish to edit the associated Requirements for.
4. Use the **pop-up menu** to add or remove selected **Requirements** as Control Associations.
5. Click **Finish**.

The CES/BTD Control Node’s Requirement associations will update accordingly.

NOTE: You can also edit CES/BTD Control Node associations to the Requirements Log using the CES/BTD Control Node’s Properties View.

Create a New Requirement & Associate to a CES/BTD Barrier/Control Node

To create a new Requirement and associate it a CES/BTD Barrier/Control Node

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Barrier** or **Control Node** you wish to edit.
3. In the **Properties View**, select the **Barrier** or **Control tab**.
4. In the **Properties View**, select the **Create New Requirement** button.
5. In the pop-up window, enter the new **Requirement’s Description**.
6. Click **Ok**.

A new Requirement entry will be created and associated to your selected CES/BTD Barrier/Control node.

Navigate from CES/BTD Barrier/Control Node to Requirement

To navigate from a CES/BTD Barrier/Control Node to an associated Requirement in the Requirements Log

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Barrier** or **Control Node** you wish to edit.
3. In the **Properties View**, select the **Barrier** or **Control tab**.
4. Select the **Associated Requirement** you wish to navigate to.
5. Select the **Open Requirement** button.

The Requirements Log will open, and the selected Requirement will be highlighted.

CES/BTD to Argument Structures

CES/BTD Nodes & Argument Associations

CES/BTD Nodes can be associated to Argument Structures that describe or reference said CES/BTD Node.

Associate CES/BTD Node to Argument Structure

To associate a CES/BTD Node to an Argument Structure

1. Open the **CES or BTD Diagram Editor**.
2. Select the **CES/BTD Node** you wish to edit.
3. In the **Properties View**, select the **drop-down arrows** next to the **Associated Argument field**.
4. Select the **Argument Structure** you wish to associate to the CES/BTD Node.

The CES/BTD Node’s associations will update accordingly.

Navigate from CES/BTD Node to Argument Structure

To navigate from a CES/BTD Node to an associated Argument Structure

1. Open the **CES or BTD Diagram Editor**.
2. From the Association Links Palette, select the **Argument Link** tool.
3. Click a **CES/BTD Node** you wish to open the associated Argument Structure for. *If the node has associated Requirements, the cursor will turn into plus-sign cursor when you hover of the node.*
4. In the pop-up menu, select the **Argument Structure** you wish to open in the Argument Diagram Editor.
5. Click **Finish**.

The selected Argument Structure will open in the Argument Diagram Editor.

CES/BTD to Evidence Log

CES/BTD Barrier/Control Nodes & Integrity Evidence

Integrity Evidence in CESs/BTDs Barrier/Control Nodes is an Evidence Artifact that describes or references Integrity measures and standards, as well as system or safety features that said Barrier/Control Node implements.

Associate CES/BTD Barrier/Control Node to Integrity Evidence

To associate a CES/BTD Barrier/Control Node to an Integrity Evidence

1. Open the **CES or BTD Diagram Editor**.
2. Select the **CES/BTD Barrier/Control Node** you wish to edit.
3. In the **Properties View**, locate the **Integrity Evidence** section.
4. Select the **Manage Associated Evidence button**.
5. In the pop-up menu, select the checkboxes for the **Evidence Artifact/s** you wish to associate with the Barrier/Control Instance's Integrity.
6. Click **Ok**.

The CES/BTD Barrier/Control Node's Integrity Evidence associations will update accordingly.

Create a New Evidence Artifact & Associate to CES/BTD Barrier/Control Node's Integrity

To create a new Evidence Artifact and associate it a CES/BTD Barrier/Control Node's Integrity

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Barrier** or **Control Node** you wish to edit.
3. In the **Properties View**, select the **Barrier** or **Control tab**.
4. In the **Properties View**, select the **Create New Evidence Artifact button** in the Integrity Evidence section.
5. In the pop-up window, enter the new **Evidence Artifact's Description**.
6. Click **Ok**.

A new Evidence Artifact entry will be created and associated to your selected CES/BTD Barrier/Control node's Integrity.

CES/BTD Event Nodes & Initial Likelihood Evidence

Initial Likelihood Evidence in CESs/BTDs Event Nodes is an Evidence Artifact that describes or references the Initial Likelihood said Event will take place.

Associate CES/BTD Event Node to Initial Likelihood Evidence

To associate a CES/BTD Event Node to an Initial Likelihood Evidence

1. Open the **CES or BTD Diagram Editor**.
2. Select the **CES/BTD Event Node** you wish to edit.
3. In the **Properties View**, locate the **Initial Likelihood** section.
4. Select the **Manage Associated Evidence button**.
5. In the pop-up menu, select the checkboxes for the **Evidence Artifact/s** you wish to associate with the Event Instance's Initial Likelihood.
6. Click **Ok**.

The CES/BTD Event Node's Initial Likelihood associations will update accordingly.

Create a New Evidence Artifact & Associate to CES/BTD Event Node's Initial Likelihood

To create a new Evidence Artifact and associate it a CES/BTD Event Node's Initial Likelihood

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Event Node** you wish to edit.
3. In the **Properties View**, select the **Create New Evidence Artifact button** in the Initial Likelihood section.
4. In the pop-up window, enter the new **Evidence Artifact's Description**.
5. Click **Ok**.

A new Evidence Artifact entry will be created and associated to your selected CES/BTD Event node's Initial Likelihood.

CES/BTD Event Nodes & Initial Severity Evidence

Initial Severity Evidence in CESs/BTDs Event Nodes is an Evidence Artifact that describes or references the Initial Severity said Event will take place.

Associate CES/BTD Event Node to Initial Severity Evidence

To associate a CES/BTD Event Node to an Initial Severity Evidence

1. Open the **CES or BTD Diagram Editor**.
2. Select the **CES/BTD Event Node** you wish to edit.
3. In the **Properties View**, locate the **Initial Severity** section.
4. Select the **Manage Associated Evidence button**.
5. In the pop-up menu, select the checkboxes for the **Evidence Artifact/s** you wish to associate with the Event Instance's Initial Severity.
6. Click **Ok**.

The CES/BTD Event Node's Initial Severity associations will update accordingly.

Create a New Evidence Artifact & Associate to CES/BTD Event Node's Initial Severity

To create a new Evidence Artifact and associate it a CES/BTD Event Node's Initial Severity

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Event Node** you wish to edit.
3. In the **Properties View**, select the **Create New Evidence Artifact button** in the Initial Severity section.
4. In the pop-up window, enter the new **Evidence Artifact's Description**.
5. Click **Ok**.

A new Evidence Artifact entry will be created and associated to your selected CES/BTD Event node's Initial Severity.

Navigate from CES/BTD Barrier/Control Node Integrity Evidence to Evidence Log DSL

To navigate from a CES/BTD Barrier/Control Node to an associated Integrity Evidence Artifact in the Evidence Log DSL

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Barrier** or **Control Node** you wish to view Integrity Evidence for.
3. In the **Properties View**, select the **Barrier** or **Control tab**.
4. Select the **Integrity Evidence** you wish to navigate to.
5. Select the **Open Selected Evidence in DSL button**.

The Evidence Log DSL will open, and the selected Evidence Artifact will be highlighted.

Navigate from CES/BTD Barrier/Control Node Integrity Evidence to Evidence Log Diagram

To navigate from a CES/BTD Barrier/Control Node to an associated Integrity Evidence Artifact in the Evidence Log Diagram

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Barrier** or **Control Node** you wish to view Integrity Evidence for.
3. In the **Properties View**, select the **Barrier** or **Control tab**.
4. Select the **Integrity Evidence** you wish to navigate to.
5. Select the **Open Selected Evidence in Diagram button**.

The Evidence Log Diagram will open, and the selected Evidence Artifact will be selected.

Navigate from CES/BTD Event Node Initial Likelihood Evidence to Evidence Log DSL

To navigate from a CES/BTD Event Node to an associated Initial Likelihood Evidence Artifact in the Evidence Log DSL

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Event Node** you wish to view Initial Likelihood Evidence for.
3. In the **Properties View**, select the **Initial Likelihood Evidence** you wish to navigate to.
4. Select the **Open Selected Evidence in DSL button**.

The Evidence Log DSL will open, and the selected Evidence Artifact will be highlighted.

Navigate from CES/BTD Event Node Initial Likelihood Evidence to Evidence Log Diagram

To navigate from a CES/BTD Event Node to an associated Initial Likelihood Evidence Artifact in the Evidence Log Diagram

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Event Node** you wish to view Initial Likelihood Evidence for.
3. In the **Properties View**, select the **Initial Likelihood Evidence** you wish to navigate to.
4. Select the **Open Selected Evidence in Diagram button**.

The Evidence Log Diagram will open, and the selected Evidence Artifact will be selected.

Navigate from CES/BTD Event Node Initial Severity Evidence to Evidence Log DSL

To navigate from a CES/BTD Event Node to an associated Initial Severity Evidence Artifact in the Evidence Log DSL

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Event Node** you wish to view Initial Severity Evidence for.
3. In the **Properties View**, select the **Initial Severity Evidence** you wish to navigate to.
4. Select the **Open Selected Evidence in DSL button**.

The Evidence Log DSL will open, and the selected Evidence Artifact will be highlighted.

Navigate from CES/BTD Event Node Initial Severity Evidence to Evidence Log Diagram

To navigate from a CES/BTD Event Node to an associated Initial Severity Evidence Artifact in the Evidence Log Diagram

1. Open the **CES or BTD Diagram Editor**.
2. Select the **Event Node** you wish to view Initial Severity Evidence for.
3. In the **Properties View**, select the **Initial Severity Evidence** you wish to navigate to.
4. Select the **Open Selected Evidence in Diagram button**.

The Evidence Log Diagram will open, and the selected Evidence Artifact will be selected.

CES/BTD to Tools Log

Integrity Evidence, Initial Likelihood Evidence, Initial Severity Evidence & Tool Uses

Integrity Evidence, Initial Likelihood Evidence, and Initial Severity Evidence in CESs and BTDs are linked implicitly as **Evidence Artifacts**. When the associated Evidence Artifact is generated by a Tool Use, those Integrity Evidence, Initial Likelihood Evidence, and Initial Severity Evidence become implicitly linked to the Tool/s in the Tools log.

CES/BTD to PhysArch & FuncSpec

CES/BTD Barrier/Control Nodes & Physical Allocations

A **Barrier Physical Allocation** associates a Barrier Node to a Component in the PhysArch that implements said Barrier. A **Control Physical Allocation** associates a Control Node to a Component in the PhysArch that implements said Control.

Allocate CES/BTD Barrier/Control Nodes to Components

To associate a CES/BTD Barrier/Control Node to a Component

1. Open the **CES or BTD Diagram Editor**.
2. Select the **CES/BTD Barrier/Control Node** you wish to edit.
3. In the **Properties View**, locate the **Barrier/Control Physical Allocations** section.
4. Select the **Select Physical Allocations button**.
5. In the pop-up menu, select the checkboxes for the **Components** you wish to associate with the Barrier/Control Instance’s Integrity.
6. Click **Ok**.

The CES/BTD Barrier/Control Node’s Physical Allocations will update accordingly.

CES/BTD Barrier/Control Nodes & Functional Allocations

A **Barrier Functional Allocation** associates a Barrier Node to a Function that implements said Barrier. A **Control Functional Allocation** associates a Control Node to a Function that implements said Control.

Allocate CES/BTD Barrier/Control Nodes to Functions

To associate a CES/BTD Barrier/Control Node to a Function

1. Open the **CES or BTD Diagram Editor**.
2. Select the **CES/BTD Barrier/Control Node** you wish to edit.
3. In the **Properties View**, locate the **Barrier/Control Functional Allocations** section.
4. Select the **Select Function Allocations button**.
5. In the pop-up menu, select the checkboxes for the **Functions** you wish to associate with the Barrier/Control Instance’s Integrity.
6. Click **Ok**.

The CES/BTD Barrier/Control Node’s Function Allocations will update accordingly.

Argument Structure Tracing

Argument Structure to Hazard Log

Arguments & Safety Architecture Element Associations in the Hazard Log

Safety Architecture Elements that describe or reference an Argument Structure or its GSN Node/s can be associated to said Argument Structure or GSN Node/s.

Associate Argument Structures to Safety Architecture Elements in the Hazard Log

To associate an Argument Structure to Safety Architecture Elements

1. Open the **Argument Diagram Editor**.
2. Right-click a **blank space** in the **Argument Structure's canvas**.
3. Select **Edit Associations** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to the **Safety Architecture Instances** to edit the Argument Structure's associations.
5. Click **Ok**.

The Argument Structure's associations will update accordingly.

NOTE: While the available associations are organized in a Data Tree, the actual associations are at the Event Instance, Control Instance, and Barrier level. Thus, adding a Safety Architecture or Hazardous Activity will result in the Argument Structure being associated with all eligible elements contained within in the selected Safety Architecture or Hazardous Activity.

Associate GSN Nodes to Safety Architecture Elements in the Hazard Log

To associate an Argument Structure’s GSN Node to Safety Architecture Elements

1. Open the **Argument Diagram Editor**.
2. Right-click a **GSN Node** in the **Argument Structure’s canvas**.
3. Select **Edit Associations** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to the **Safety Architecture Instances** to edit the GSN Node’s associations.
5. Click **Ok**.

The GSN Node’s associations will update accordingly.

NOTE: While the available associations are organized in a Data Tree, the actual associations are at the Event Instance, Control Instance, and Barrier level. Thus, adding a Safety Architecture or Hazardous Activity will result in the GSN Node being associated with all eligible elements contained within in the selected Safety Architecture or Hazardous Activity.

Navigate from Argument Structure to Safety Architecture Elements in the Hazard Log

To navigate from an Argument Structure to a Safety Architecture Element association in the Hazard Log

1. Open the **Argument Diagram Editor**.
2. Right-click a **blank space** in the **Argument Structure’s canvas**.
3. Select **Open in** > **Associated Artifacts** from the right-click menu.
4. In the pop-up menu, select the **association** you wish to open.
5. Click **Ok**.
6. In the next pop-up menu, select the **Hazard Log Editor button**.

The Hazard Log will open with the associated Safety Architecture Element highlighted in dark gray.

Navigate from GSN Node to Safety Architecture Elements in the Hazard Log

To navigate from a GSN Node to a Safety Architecture Element association in the Hazard Log

7. Open the **Argument Diagram Editor**.
8. Right-click a **GSN Node**.
9. Select **Open in** > **Associated Artifacts** from the right-click menu.
10. In the pop-up menu, select the **association** you wish to open.
11. Click **Ok**.
12. In the next pop-up menu, select the **Hazard Log Editor button**.

The Hazard Log will open with the associated Safety Architecture Element highlighted in dark gray.

Argument Structure to Requirements Log

GSN Nodes & Requirements

Safety Architecture Elements that describe or reference an Argument Structure or its GSN Node/s can be associated to said Argument Structure or GSN Node/s.

Associate GSN Nodes to Requirements

To associate an Argument Structure's GSN Nodes to Requirements

1. Open the **Argument Diagram Editor**.
2. Right-click a **GSN Node** in the **Argument Structure's canvas**.
3. Select **Edit Associations** from the right-click menu.
4. In the pop-up window, select the **drop-down arrow** next to the **Requirement data tree**.
5. Select or deselect the **checkboxes** next to the listed **Requirements** to edit the GSN Node's associations.
6. Click **Ok**.

The GSN Node's associations will update accordingly.

Navigate from GSN Node to Requirement

To navigate from a GSN Node to a Requirement

1. Open the **Argument Diagram Editor**.
2. Right-click a **GSN Node**.
3. Select **Open in** > **Associated Artifacts** from the right-click menu.
4. In the pop-up menu, select the **association** you wish to open.
5. Click **Ok**.

The Requirements Log will open with the associated Requirement highlighted in dark gray.

Argument Structure to CESs/BTDs

Arguments & Safety Architecture Element Associations in CESs/BTDs

Safety Architecture Elements that describe or reference an Argument Structure or its GSN Node/s can be associated to said Argument Structure or GSN Node/s.

Associate Argument Structures to Safety Architecture Elements in CESs/BTDs

To associate an Argument Structure to Safety Architecture Elements

1. Open the **Argument Diagram Editor**.
2. Right-click a **blank space** in the **Argument Structure's canvas**.
3. Select **Edit Associations** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to the **Safety Architecture Instances** to edit the Argument Structure's associations.
5. Click **Ok**.

The Argument Structure's associations will update accordingly.

NOTE: While the available associations are organized in a Data Tree, the actual associations are at the Event Instance, Control Instance, and Barrier level. Thus, adding a Safety Architecture or Hazardous Activity will result in the Argument Structure being associated with all eligible elements contained within in the selected Safety Architecture or Hazardous Activity.

Associate GSN Nodes to Safety Architecture Elements in CESs/BTDs

To associate an Argument Structure’s GSN Node to Safety Architecture Elements

1. Open the **Argument Diagram Editor**.
2. Right-click a **GSN Node** in the **Argument Structure’s canvas**.
3. Select **Edit Associations** from the right-click menu.
4. In the pop-up window, select or deselect the **checkboxes** next to the **Safety Architecture Instances** to edit the GSN Node’s associations.
5. Click **Ok**.

The GSN Node’s associations will update accordingly.

NOTE: While the available associations are organized in a Data Tree, the actual associations are at the Event Instance, Control Instance, and Barrier level. Thus, adding a Safety Architecture or Hazardous Activity will result in the GSN Node being associated with all eligible elements contained within in the selected Safety Architecture or Hazardous Activity.

Navigate from Argument Structure to Safety Architecture Elements in CESs/BTDs

To navigate from an Argument Structure to a Safety Architecture Element association in CESs/BTDs

1. Open the **Argument Diagram Editor**.
2. Right-click a **blank space** in the **Argument Structure’s canvas**.
3. Select **Open in** > **Associated Artifacts** from the right-click menu.
4. In the pop-up menu, select the **association** you wish to open.
5. Click **Ok**.
6. In the next pop-up menu, select the **Diagram button**.
7. In the next pop-up menu, select the **CES/BTD** you wish to open.
8. Click **Ok**.

The diagram will open.

Navigate from GSN Node to Safety Architecture Elements in CESs/BTDs

To navigate from a GSN Node to a Safety Architecture Element association in CESs/BTDs

1. Open the **Argument Diagram Editor**.
2. Right-click a **GSN Node**.
3. Select **Open in** > **Associated Artifacts** from the right-click menu.
4. In the pop-up menu, select the **association** you wish to open.
5. Click **Ok**.
6. In the next pop-up menu, select the **Diagram button**.
7. In the next pop-up menu, select the **CES/BTD** you wish to open.
8. Click **Ok**.

The diagram will open.

Argument Structure to Evidence Log

GSN Nodes & Evidence Artifacts

There are four **GSN Node Types** that you can associate Evidence Artifacts to”

- **Assumption Node**: The associated Evidence Artifacts describe or refence the unsubstantiated statements said Assumption Node dictates.
- **Context Node**: The associated Evidence Artifacts describe or refence the contextual information said Context Node dictates.
- **Solution Node**: The associated Evidence Artifacts describe or refence said Solution Node.
- **Justification Node**: The associated Evidence Artifacts describe or refence the rationale statement said Justification Node dictates.

Associate GSN Nodes to Evidence Artifacts

To associate a GSN Node to Evidence Artifacts

1. Open the **Argument Diagram Editor**.
2. Select the **Assumption**, **Context**, **Solution**, or **Justification Node** you wish to associate to an Evidence Artifact.
3. In the **Properties View**, select the **Manage Associate Evidence** button.
4. In the Associated Evidence window, toggle the **checkbox** for the corresponding **Evidence Artifact** you wish to associate or disassociate from the selected GSN Node.
5. Click **Ok**.

The GSN Node’s Properties will update to reflect the change.

Create a New Evidence Artifact & Associate to a GSN Node

To create a new Evidence Artifact and associate it a GSN Node

1. Open the **Argument Diagram Editor**.
2. Select the **GSN Node** you wish to edit.
3. In the **Properties View**, select the **Create New Evidence Artifact** button.
4. In the pop-up window, enter the new **Evidence Artifact Description**.
5. Click **Ok**.

A new Evidence Artifact entry will be created and associated to your selected GSN node.

Navigate from GSN Node to Evidence Artifacts

To navigate from an Argument Structure's GSN Node to an associated Evidence Artifact

1. Open the **Argument Diagram Editor**.
2. Select the **GSN Node** you wish to edit.
3. In the **Properties View** under Evidence, select the associated **Evidence Artifact** you wish to open.
4. Select the **Open Selected Evidence** button.

The **Evidence Log** will open, and the selected **Evidence Artifact entry** will be highlighted.

Argument Structure to Tools Log

GSN Node Evidence & Tool Uses

GSN Node Evidence in Argument Structures are linked implicitly as **Evidence Artifacts**. When the associated Evidence Artifact is generated by a Tool Use, that GSN Node Evidence becomes implicitly linked to the Tool/s in the Tools log.

Evidence Log Tracing

Evidence Log to Argument Structures

In future AdvoCATE updates, traceability and navigation options from an Evidence Artifact entry to an associated GSN Node in an Argument Structure will be created.

Evidence Log to Tools Log

Evidence Artifacts & Tool Uses

Evidence Artifacts in the Evidence Log are linked implicitly to **Tool Uses**. When the Evidence Artifact is generated by a Tool Use, that Evidence Artifact becomes implicitly linked to the Tool/s in the Tools log.

Tools Log Tracing

Tools Log to Requirements Log

Tool Uses & Requirements

Verification Allocations in the Requirements Log are linked implicitly to **Tool Uses**. When the associated Evidence Artifact is generated by a Tool Use, the Verification Allocation becomes implicitly linked to the Tool in the Tools log.

Navigate from Tool Use to Requirements

To navigate from a Tool Use to its associated Requirement in the Requirements Log

1. Open the **Tool Use Table**.
2. Right-click the **Tool cell** you wish to view in the Requirements Log.
3. Select **Navigate to Requirements of Tool Use** from the right-click menu.

The Requirements Log will open with the associated Requirement/s highlighted.

Tools Log to Argument Structures

Tool Uses & GSN Nodes

GSN Nodes in an Argument Structure are linked implicitly to **Tool Uses**. When the GSN Node's associated Evidence Artifact is generated by a Tool Use, the GSN Node becomes implicitly linked to the Tool in the Tools log.

Navigate from Tool Use to GSN Nodes

To navigate from a Tool Use to its associated GSN Nodes in an Argument Structure

1. Open the **Tool Use Table**.
2. Right-click the **Tool cell** you wish to view in an Argument Structure.
3. Select **Navigate to Argument Nodes of Tool Use** from the right-click menu.

The Argument Structure will open with the associated GSN Nodes selected.

Tools Log to Evidence Log

Tool Uses & Evidence Artifacts

Evidence Artifacts in the Evidence Log are linked implicitly to **Tool Uses**. When the Evidence Artifact is generated by a Tool Use, that Evidence Artifact becomes implicitly linked to the Tool/s in the Tools log.

Navigate from Tool Use to Evidence Artifacts

To navigate from a Tool Use to its associated Evidence Artifacts

1. Open the **Tool Use Table**.
2. Right-click the **Inputs or Outputs cell** you wish to view in an Evidence Dependency Graph.

The Evidence Dependency Graph will open with the Evidence Artifact selected.

File Export

AdvoCATE allows users to export files in the following formats:

- *Export an AdvoCATE Project or File*
- *Export a Table as an Excel XLS or CSV File*
- *Export a Table as a PDF*
- *Export a Diagram in an Image Format*

Export an AdvoCATE Project or File

To export an existing AdvoCATE Project or File

1. Select **File** > **Export…** from the main menu.
2. In the Export window, toggle the **drop-down arrow** for the General folder.
3. In the drop-down menu for the General folder, select **File System**.
4. Click **Next**.
5. Toggle the **checkbox** for the **AdvoCATE Project folder**/s you wish to export. *When exporting an AdvoCATE Project, be sure to include the backup files (.bak), which are needed for backwards compatibility if the AdvoCATE assurance case model changes between versions.*
6. In the sub-column, toggle the **checkbox** for the **AdvoCATE file**/s you wish to export.
7. Click the **Browse…** button to select the location where you want to export your desired AdvoCATE file/s to.
8. Click **Open**.
9. Click **Finish**.

Your corresponding AdvoCATE files will export.

Export a Table as an Excel XLS or CSV File

To export an existing table in AdvoCATE as an Excel file

1. Open your desired **table file**.
2. Select the **Export Table to Excel** icon in the toolbar. *Depending on the table file, additional pop-up window/s may appear to refine your selection.*
3. In the Save window, select the **formatting specifications** you wish to implement.
4. Select whether you want to save the new Excel in the **.xls** or **.csv** format.

   NOTE: In addition to selecting the file format, you must also change the file format manually, e.g., by replacing .xls with .csv.

5. Click **Save**.

Your corresponding AdvoCATE table will export as an Excel XLS or CSV file.

Export a Table as a PDF

To export an existing table in AdvoCATE as a PDF

1. Open your desired **table file**.
2. Select the **Export Table to PDF** icon in the toolbar. *Depending on the table file, additional pop-up window/s may appear to refine your selection.*
3. In the Save window, select the **formatting specifications** you wish to implement.
4. Click **Save**.

Your corresponding AdvoCATE table will export as a PDF file.

Export a Diagram in an Image Format

To export an existing diagram in AdvoCATE in an image format

1. Open your desired **diagram file**.
2. Select the 📷 **Export diagram as image** icon in the Diagram Editor toolbar.
3. In the Save window, select the **formatting specifications** you wish to implement. *Image format options include JPG, PNG, SVG, and PDF.*
4. Click **Ok**.

Your corresponding AdvoCATE diagram will export in the selected image format.

Troubleshooting

The Troubleshooting section describes 22 common problems users experience within the AdvoCATE application:

1. *I closed a Perspective View on accident.*
2. *I opened my Perspective Views in different windows, and now I have too many windows open.*
3. *I have an error pop-up message.*
4. *My AdvoCATE application froze.*
5. *The same error message keeps popping up.*
6. *A security warning pops up when launching AdvoCATE.*
7. *I cannot scroll to the right or down.*
8. *My BTD nodes do not connect where I expect.*
9. *My GSN nodes, added by shortcut keys, are not connected in my Argument Structure.*
10. *When I click a GSN or Pattern node, the edit textbox does not always appear.*
11. *I cannot collapse/expand a drop-down arrow menu in a dialog window.*
12. *I cannot access nodes at the edge of the diagram canvas.*
13. *My diagram does not exist even though its model object appears in the Model Explorer.*
14. *My AdvoCATE Project cannot be migrated.*
15. *Some of my links do not appear in my diagram.*
16. *A diagram looks good on my colleague's computer, but it does not look good on mine (e.g., not all node text is visible and/or links look terrible).*
17. *I created an AdvoCATE Artifact but it's not visible from another AdvoCATE Artifact.*
18. *I have a lot of warnings in the Problems View.*
19. *I can't find my AdvoCATE Project or Workspace.*
20. *AdvoCATE starts, but the dashboard doesn't load, or many errors appear on startup.*
21. *The Content Assist feature in the DSL Editor does not list all the available options.*
22. *My Hazard has the same Initial and Residual Risk Levels.*

I closed a Perspective View on accident.

If you closed a Perspective View on accident, you can

1. Re-open the **Perspective View** using the *Open a Perspective View* section.

NOTES:

- Alternatively, if you wish to restore the **AdvoCATE Perspective** completely, see the *Reset AdvoCATE Perspective* section.
- For more information about the **AdvoCATE Perspective** and **Perspective Views**, see the *AdvoCATE Perspective* section.

I opened my Perspective Views in different windows, and now I have too many windows open.

If you have too many Perspective Views open in different windows, you can

1. Drag a **Perspective View's window tab** over to another window. *The two windows should combine into one window.*
2. If a blank window remains, it can be minimized by clicking the **Minimize icon** in the top-right corner of the window.

NOTES:

- Alternatively, if you wish to restore the **AdvoCATE Perspective** completely, see the *Reset AdvoCATE Perspective* section.
- For more information about the **AdvoCATE Perspective** and **Perspective Views**, see the *AdvoCATE Perspective* section.

I have an error pop-up message.

If you have an error pop-up message, you can

1. Close the specific **Editor** you are using.
2. If the error message cannot be resolved, restart the **AdvoCATE application**.
3. Send your **AdvoCATE Configuration Log** to our team for additional troubleshooting. For more information about how to find your AdvoCATE Configuration Log, see the *Send AdvoCATE Configuration Log* section.

My AdvoCATE application froze.

If your AdvoCATE application freezes, you can

1. Click the **Finish** button in the dialog window.
2. If the dialog window does not have a **Finish** button, exit out of the dialog window by clicking the **Close icon** in the top-right corner of the dialog window.
3. If the AdvoCATE application will not close the dialog window or unfreeze, restart the **AdvoCATE application**.
4. Send your **AdvoCATE Configuration Log** to our team for additional troubleshooting. For more information about how to find your AdvoCATE Configuration Log, see the *Send AdvoCATE Configuration Log* section.

The same error message keeps popping up.

If an error message keeps popping up, you can

1. Deselect the **Save options** when exiting out of the AdvoCATE application. *Otherwise, you might get stuck in an endless loop of error pop-up messages.*
2. If the error pop-up message doesn’t have **Save options**, force quit the **AdvoCATE application**.
3. Send your **AdvoCATE Configuration Log** to our team for additional troubleshooting. For more information about how to find your AdvoCATE Configuration Log, see the *Send AdvoCATE Configuration Log* section.

A security warning pops up when launching AdvoCATE.

Depending on your security settings, your OS might not let you launch the AdvoCATE executable file (AdvoCATE.exe). If you are using a PC, you can often run the eclipsec.exe instead.

To run the eclipsec.exe file

1. Locate the **eclipse.exe** file in your AdvoCATE package folder.
2. Double click the **eclipse.exe** file.

The AdvoCATE application should start to run.

I cannot scroll to the right or down.

Scrolling is handled by drivers that are hardware specific. We've noticed that on some machines scrolling in specific directions does not work with touchpads or touch screens. If you cannot scroll to the right or down, you can

1. Use a **wired** or **wireless mouse**.
2. If you cannot scroll with a wired or wireless mouse, please send an e-mail to *advocate-feedback@lists.nasa.gov*.

My BTD nodes do not connect where I expect.

If your BTD nodes do not connect where you expect them to, you can

1. Drop the **BTD node** precisely on the BTD link that you wish it is connected to (not to the side of the BTD link).

NOTES:

- Alternatively, you can delete your existing BTD Links and recreate them using the **Generic Connection Creation Tool**.
- For more information about **BTD Nodes** and **BTD Links**, see the *BTD Event Node Types*, *BTD Mitigation Node Types*, and *BTD Link Types* sections.

My GSN nodes, added by shortcut keys, are not connected in my Argument Structure.

If your GSN nodes, added by shortcut keys, are not connected in an Argument Structure, this error can occur when rapidly adding a series of GSN nodes. You can

1. Pause after **adding a GSN node**. *This pause allows the AdvoCATE application to update the selection, so subsequent nodes connect appropriately.*

NOTES:

- Alternatively, you can add GSN Links using the **Generic Connection Creation Tool**.
- For more information about **GSN Nodes** and **GSN Links**, see the *GSN Node Types* and *GSN Link Types* sections.

When I click a GSN or Pattern node, the edit textbox does not always appear.

If the edit textbox does not always appear when you click a GSN or Pattern node, you might be clicking the node too quickly. You can

1. Select the **GSN** or **Pattern node**.
2. Click the same **GSN** or **Pattern node** again to bring up the edit textbox (not a double click).

NOTES:

- Alternatively, you can edit the node’s description in the Properties Panel.
- For more information about **GSN Nodes** and **Pattern Nodes**, see the *GSN Node Types* and *Pattern Node Types* sections.

I cannot collapse/expand a drop-down arrow menu in a dialog window.

If you cannot collapse/expand a drop-down arrow menu in a dialog window, you can

1. Click just slightly to the right of the **drop-down arrow**.

I cannot access nodes at the edge of the diagram canvas.

If you cannot access nodes at the edge of the diagram canvas, you can

1. Use the **Zoom Out** tool in the Editor's toolbar until there is some white space between the node and the edge of the canvas (typically 50 - 75%).
2. Select the **node** you could not access before and proceed as desired.

My diagram does not exist even though its model object appears in the Model Explorer.

If your diagram (i.e., representation) does not appear in its model object in the Model Explorer, you can

1. Right-click the model object in the Model Explorer.
2. From the right-click menu, select **New Representation** > **Other…**.
3. In the pop-up window, select the **Diagram** representation option.
4. Click **Finish**.
5. Enter the **name** for your new diagram.
6. Click **Ok**.

The diagram will be created and open in the Diagram Editor.

My AdvoCATE Project cannot be migrated.

When you import an old AdvoCATE Project, you can sometimes get a message that says the AdvoCATE Project (or some parts of the project) could not be migrated.

Generally, AdvoCATE is backwards compatible with older versions, but occasionally a change is introduced that requires a manual edit to AdvoCATE Projects to make them work with the latest application version. If your AdvoCATE Project cannot be migrated, you can

1. Send an e-mail to *advocate-feedback@lists.nasa.gov* and attach a copy of the AdvoCATE Project you wish to migrate.

Some of my links do not appear in my diagram.

There used to be a bug in our layout code that occasionally caused horizontal links to disappear near curves. As of AdvoCATE 0.33, this bug should be resolved.

If some of your links do not appear in your diagram, you can

1. Click the **Arrange All** icon in the Diagram Editor’s toolbar. *Sometimes, you must click the Arrange All icon twice before you can see the link/s reappear.*
2. If the Arrange All icon does not help, try to move the link by dragging one of its segments, which will refresh the link.
3. If your link still does not appear, please send an e-mail to *advocate-feedback@lists.nasa.gov*.

A diagram looks good on my colleague's computer, but it does not look good on mine (e.g., not all node text is visible and/or links look terrible).

When two computers have different resolutions and/or display scaling, diagram nodes and links can appear skewed. Unfortunately, the diagram's layout (stored in the AdvoCATE Project) will need to be different on each computer. If your diagram layout appears skewed, you can

1. Click the **Autosize all nodes to fit text** icon in the Diagram Editor's toolbar.
2. Click the **Arrange All** icon in the Diagram Editor's toolbar. *Sometimes, you must click the Arrange All icon twice before you can see the link/s update.*

I created an AdvoCATE Artifact but it's not visible from another AdvoCATE Artifact.

If you cannot see an AdvoCATE Artifact you created, you can

1. Close the **AdvoCATE Artifact**. *This error is due to known synchronization issues.*
2. Re-open the **AdvoCATE Artifact**.

I have a lot of warnings in the Problems View.

By default, AdvoCATE warns about just about everything. In future versions, we plan to add preferences to allow some of these warnings to be turned off. However, in general, warnings provide a reminder that something non-essential is missing from your AdvoCATE Project, whereas errors indicate something that needs to be addressed.

I can't find my AdvoCATE Project or Workspace.

There are various options for where to save an AdvoCATE Project or Workspace. If you selected the default option upon creation, the AdvoCATE Project will be inside the Workspace. Also, when importing an AdvoCATE Project, there is also an option to copy it into the Workspace, so you may or may not have been editing the original version.

AdvoCATE starts, but the Dashboard doesn't load, or many errors appear on startup.

If the AdvoCATE application starts, but the Dashboard does not load or there are many error upon starting the application, you can

1. Check to see if you are using the correct **Java version**. For more information about the necessary system requirements to run the AdvoCATE application, see the *System Requirements* section.
2. Check to see if the **AdvoCATE application points to the correct version of Java**.
3. Check to see if you are using the **compatible OS version**. For more information about the necessary system requirements to run the AdvoCATE application, see the *System Requirements* section.

The Content Assist feature in the DSL Editor does not list all the available options.

The Contest Assist feature in the DSL Editor lists attributes based on the specified order of your selected entry. If you wish to view all of the DSL attributes available, you can

1. Open the **Content Assist** feature directly after the { bracket for your DSL entry by using the **Control** and **Space** keyboard shortcut.
2. If you have additional DSL entry lines to edit, you can move to the DSL entry line by using the **Alt** and **Up/Down arrow keys** (PC) or **Option** and **Up/Down arrow keys** (Mac) keyboard shortcuts.

My Hazard has the same Initial and Residual Risk Levels.

If a Hazard has the same Initial and Residual Risk Level, you can

1. Check that the relevant **Hazard Mitigations** are a mixture of **Existing** and **New** classifications. For more information about the Existing and New classifications, see the *New?* section.
2. Check that all **Controls** have associated **Barriers** and their Integrities have been changed from the default of 1. For more information about Control and Barrier Integrities, see the *CES Barrier Properties*, *CES Control Properties*, and the *Risk Analysis Calculations* sections.

Index

TBD

Glossary

TBD

Appendices

There are four appendices within this AdvoCATE User Guide:

1. *Risk Analysis Calculations*
2. *Scoping Table*
3. *Validations*
4. *Traceability Table*

Risk Analysis Calculations

Risk analysis calculations are performed by computing **Initial Likelihoods** and **Residual Likelihoods** and **Initial Severities** and **Residual Severities**, using **Control Integrities** and **Barrier Integrities**.

All computations of Likelihood and Severity are specific to a CES. The same Event in a different CES within the same Safety Architecture can have different risk values since it is a different **Event Instance**.

Initial Likelihood is only specified for **Global Threats** (or Global Causes), that is, those events that do not have any preceding events. AdvoCATE derives the **Residual Likelihood** for all other Events (working rightwards).

Initial Severity is only specified for **Global Consequences** (or Global Effects), that is, those events that do not have any following events. AdvoCATE derives severity for all other events (working leftwards). **Residual Severity** is currently set to always be the same as Initial Severity since AdvoCATE does not model severity-reducing Barriers.

The Initial Likelihood and Initial Severity of **Global Events** can be specified by the user in the Hazard Log, CES Event Properties, or BTD Event Properties.

If **Barriers** are present on a path between a Global Threat and a Global Consequence, **Barrier Integrity** is used to compute the likelihood of the remaining **Non-Global Events** on the path, given the mitigation measure/s, using the **Inclusion-Exclusion Principle**.

This computation of Non-Global Event Likelihood, in turn, relies on the computation of path probabilities, which is determined for a path as the joint probability that all the events on that path (including barrier breach events) occur.

NOTE: Here, a key assumption is that barrier breaches occur independently. In future versions of AdvoCATE, we plan to model dependent and mutually exclusive probabilities as well. However, currently AdvoCATE only models independent events, in the sense that the risk calculation assumes this for Threats.

The safety target against which AdvoCATE compares the Residual Likelihood is based upon a qualitative risk acceptance matrix from the FAA Air Traffic Organization: Safety Management System Manual. These Risk Matrices are viewable in the **Safety Architecture Analytics Dashboard**.

Control Integrities only influence Event Likelihoods if Barrier Integrity is computed from Control Integrity.

There are several options for how **Barrier Integrity** is determined:

- **Specified**: In this case, the user provides an Integrity value for each Barrier.
- **Computed from Control Integrities**: The user selects the **Derive from Control Integrities** checkbox in the CES or BTD Barrier Properties. The Barrier's Integrity then is computed from the Control Integrities of the Barrier's constituent Controls. The user then species the Control Integrities.
- **Computed from Control Integrities and Escalation Branches**: The user selects the **Derive from Control Integrities** checkbox in the CES or BTD Barrier Properties and the **Derived from Escalation Paths** checkbox in the CES or BTD Control Properties. The Barrier's Integrity then is computed from the Control Integrities, with the Control Integrities, in turn, computed from their Escalation Branches. The user then species the Integrities of the Escalation Factor Barriers and Escalation Factor Controls along the Escalation Branch.
- **Computed from Escalation Branches**: The preference to compute Control Integrities from their Escalation Branches is independent of whether Barrier Integrities are computed from Control Integrities. Although, if Barrier Integrities are specified and Control Integrities are computed, the latter will have no effect on the overall risk levels.

The preference whether to compute or specify Barrier Integrity is specific to each Barrier, Control, and Escalation Instances.

Barrier Integrity (whether specified or computed) is unique to a path between events. All Barrier Instances between events have the same integrity. When Barrier Integrity is derived from Control Integrity, it is the Controls on the path between the relevant events that are used.

Repetition of Barriers on a path is allowed, but individual Controls represent separate instances, and since Control Instances can have distinct Escalations, they can have distinct Integrities.

Scoping Table

Scoping groups model elements within certain AdvoCATE areas.

Below is a **Scope Table** that lists model elements, where model elements are scoped within AdvoCATE, naming conventions, and additional notes.

Model Element	Scoped Within	Naming	Notes
Argument Node	Argument Structure		
Evidence Artifact	Evidence Log		
Evidence Type	Evidence Log		
(Evidence) Dependency Type	Evidence Log		
Deviation	Functional Specifications File		
Function	Functional Specifications File		
Component	Physical Architecture File		
Failure Mode	Component		

Data Tree Node	Data Tree		
Parameter Value	Data Tree Node		
Pattern Node	Argument Pattern		
Parameter	Argument Pattern		
Requirement	Requirements Log		
Source	Requirements Log		
Verification Method	Requirements Log		
Event	Safety Architecture		
Hazardous Activity	Safety Architecture		
Control	Safety Architecture		
Barrier	Safety Architecture		

System State	Safety Architecture		
Environmental Condition	Safety Architecture		
Event Instance	Hazardous Activity	Events live at the Safety Architecture level, and each Event has at most one Event Instance per CES. However, Event Instance names are unique within a Hazardous Activity (not a CES), so, e.g. if event E has instances in CES1 = (HA1, SS1, EC1), CES2 = (HA1, SS2, EC2), and CES3 = (HA2, SS1, EC1), then the names of the instances in CES1 and CES2 must be distinct, but the name of the instance in CES3 can be the same as either of the other two.	Naming is linked to the parent Event when created outside the DSL. E1_1 would be the first Event Instance for Event E1. You can only have one Event Instance called EI_1 per Hazardous Activity.
Control Instance	Hazardous Activity	You can create as many Control Instances of a given Control in the same CES as you like.	Naming is linked to the parent Control when created outside the DSL. You can only have one Control Instance called CI_1 per Hazardous Activity.
Barrier Instance	Hazardous Activity	You can create as many barrier instances of a given barrier in the same CES as you like.	Naming is linked to the parent Barrier when created outside the DSL. You can only have one

			Barrier Instance called BI_1 per Hazardous Activity.
Tool	Tools Log		
Tool Use	Tools Log		
Toolset	Tools Log		

Validations

Validations are indications when data entries conflict with validation rules and constraints to reduce inaccuracies within AdvoCATE Projects.

Validations are viewable in the **Problems View** as Errors, Warnings, and Infos.

Errors are violations that prevent some other action from taking place, whereas **Warnings** are considered possibly temporary problems at the user's discretion to fix. **Infos** are clarifications about AdvoCATE Artifacts settings.

Below are the existing Validations for AdvoCATE Projects within the DSL, Diagram, and Table Editors:

DSL Editor Validations

Argument DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Argument version tag is not “supported” or non-existent. Currently, the supported version tags are 1.0 and 1.1.	Error	“The Argument «argument.name» should declare a syntax version” OR “The syntax version of the Argument «argument.name» is unsupported. Supported versions are: «ArgumentConstants.SUPPORTED_VERSIONS.join(', ')»” OR “The syntax version of the Argument «argument.name» is not the latest. This file will be migrated upon close/reopen”	
Check Multiple Parents	Checks if an argument node has two or more links coming into it from multiple different nodes.	Warning	“Node «node.name» has multiple parents («String.join(", ", parentNames)»)”	
Check Multiple Roots	Checks if there are multiple root nodes in an Argument.	Warning	“There are «roots.size» root nodes («root, root2, root3, ...)»)”	

Check Non-Goal Root	Checks if a root node is not a Goal node.	Warning	“Node «node.name» should have parents, as only Goals should be root nodes”	
Check Goal to Goal	Checks if a Goal node is supported by another Goal node.	Warning	“Goal «goal.name» is directly supported by Goal «it.name» without an intervening Strategy”	
Check Strategy Is Supported By	Checks if a Strategy node is supported by another Strategy or a Solution node.	Warning	“Strategy «strategy.name» is supported by Strategy «it.name»” OR “Strategy «strategy.name» is supported by Solution «it.name»”	
Check Solution In Context Of	Checks if a Solution node is in context of a Justification, Context, or Assumption node.	Warning	“Solution «solution.name» is in context of Justification/Context/Assumption «it.name»”	
Check Cycle	Checks if there is a cycle (circular reasoning) in the Argument.	Error	“Node «node.name» is part of a cycle”	

Check Solution has Evidence	Checks if Solution nodes have associated evidence.	Warning	“Solution «solution.name» should have evidence”	
Check Goal Strategy References Evidence	Checks if a Goal or Strategy node references evidence.	Warning	“Node «node.name» references («node.evidenceArtifacts»). Goal and Strategy nodes can not reference evidence artifacts”	
Check Duplicate Evidence Use	Checks if an argument node references the same evidence more than once.	Error	“Evidence Artifact «evidenceArtifact.name» is referenced more than once by Node «node.name»”	
Check Strategy TBD	Checks if a childless Strategy node is not marked TBD.	Warning	“Strategy «strategy.name» is undeveloped, and should therefore be marked TBD”	
Check Goal TBD	Checks if a childless Goal node is not marked TBD.	Warning	“Goal «goal.name» is undeveloped, and should therefore be marked TBD”	
Check Undeveloped	Checks if a node is marked as undeveloped.	Info	“Node «node.name» yet to be developed”	Mark developed

Safety Architecture DSL Validations

Safety Architecture DSL Validations				
Name	**Description**	**Validation Severity**	**Validation Text**	**Quick Fix**
Check Valid Version	Checks if the Safety Architecture version tag is not "supported" or non-existent. Currently, the supported version tags are 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, and 1.7.	Error	“The Safety Architecture «safetyArchitecture.name» should declare a syntax version” OR “The syntax version of the Safety Architecture «safetyArchitecture.name» is unsupported. Supported versions are: «BowtieModelConstants.SUPPORTED_VERSIONS.join(', ')” OR “The syntax version of the Safety Architecture «safetyArchitecture.name» is not the latest. This file will be migrated upon close/reopen”	
Check Duplicate Requirements	Checks if a Barrier Instance (BI) is allocated multiple times to the same Requirement.	Error	“The Requirement «it.name» occurs multiple times for «barrierInstance.name»”	Remove all redundant associations to the requirement
Check Repeated Controls	Checks if a Control is repeated on a non-escalation path.	Error	“The Control «control.name» is repeated on a non-escalation path”	

Check Repeated Events	Checks if an Event has multiple Event Instances (EIs) in a CES.	Error	“The Event «it.name» occurs multiple times in the CES”	
Check Event Directly Connected to Event	Checks if two connected events have a Control between them.	Warning	“The Event «eventInstance.name» is connected to the Event «it.to.name» with no intervening control”	
Check Event Causality Hierarchy	Checks if an Event and its Consequence event have allocations with a common ancestor, and the Consequence allocation is neither a parent or a sibling to the Event allocation.	Warning	“Event «event.name» has consequence «consequence.event.name» which is allocated to an ancestor of the super-«AllocationType» («event.allocation.description», «consequence.event.allocation.description», respectively)” OR “Event «event.name» has consequence «consequence.event.name» which is allocated to a sub-«AllocationType» («event.allocation.description», «consequence.event.allocation.description», respectively)”	

Check Intermediate Event has Allocation	Checks if an unallocated Event has Event Instances (EIs) with both a Consequence event and a Cause event that are allocated.	Warning	“Event «event.name» should be allocated because it has an event instance («threat.name», «threat.hazardousActivity.name») with both an allocated cause and an allocated consequence” OR “Event «event.name» should be allocated because it has an event instance («threat.name», «threat.hazardousActivity.name») with an allocated cause and an event instance («consequence.name», «consequence.hazardousActivity.name») with an allocated consequence”	
Check Escalation Control Duplicates	Checks if the Control is used as both an Escalation Factor Control and a regular Control on the same path.	Warning	“Control «control.name» is used as both an escalation factor control and a regular control on the same path”	

Check Circular Consequences	Checks if the Safety Architecture contains any cyclical chains (circular reasoning) of Event Instances (EIs), considered over all CESs. Such a cycle would indicate a failure of the causal consistency between CESs.	Warning	“Events form a circular chain of consequences, e1→ (e2 ~ e3) → (e4 ~ e1)” OR “Event «instance.name» is directly connected to itself”	
Check Short Circuits	Checks if there is a direct link between two nodes, as well as a longer path containing a Control Instance (CI).	Warning	“Short circuit exists from «element.name» to «element.name»”	
Check Event has Assurance Type	Checks if an Event has been given an assurance type.	Error	“Event '«event.name»: «event.description»' has no assurance type(s) set.”	
Check CES has Environmental Condition	Checks if a CES has an Environmental Condition.	Error	“CES has no environmental condition”	
Check CES has System State	Checks if a CES has a System State.	Error	“CES has no system state”	

Check Multiple Barrier Instances Between Events	Checks if a Barrier has multiple Barrier Instances (BIs) between two Event Instances (EIs).	Error	“Barrier «currentBarrier.name» is used by multiple barrier instances between two events («ci1.barrier.name», «ci2.barrier.name»)”	Merges the two barrier instances
Check Barrier Requirement Allocation	Checks if a Barrier is missing an allocated Function or Component that is allocated to the Requirements of its Barrier Instances (BIs).	Warning	“Barrier «barrierInstance.barrier.name» is missing component allocation «component.name» from its requirement «requirement.name»” OR “Barrier «barrierInstance.barrier.name» is missing function allocation «function.name» from its requirement «requirement.name»”	Allocates the function to the barrier
Check Barrier Function Allocation	Checks if a Barrier is missing an allocated Component that is allocated to the Functions of its Barrier Instances (BIs).	Warning	“Barrier «barrier.name» is missing component allocation «component.name» from its function allocation «function.name»”	
Check Control Requirement Allocation	Checks if a Control is missing an allocation of its Control Instances' (CIs) Requirements.	Warning	“Control «control.name» is missing function allocation «function.name» from its requirement «requirement.name»” OR	Allocates the function to the control OR

			“Control «control.name» is missing component allocation «component.name» from its requirement «requirement.name»”	Allocates the component to the control
Check Control Function Allocation	Checks if a Control is missing a component of its Control Instances' (CIs) Functions	Warning	“Control «control.name» is missing component allocation «component.name» from its function allocation «function.name»”	
Check Control Instance has Links	Checks if a Control Instance (CI) has input/output links.	Error	“Control Instance «controlInstance.name» is missing an input link' OR “Control Instance «controlInstance.name» is missing an output link”	
Check Control Instance has Barrier Instance	Checks if a Control Instance (CI) has a Barrier Instance (BI) set.	Error	“Control «controlInstance.name» has no associated barrier instance”	
Check Control has Barrier	Checks if a Control has a Barrier set.	Error	“Control «control.name» has no associated barrier”	

Check Control Barrier Matches Barrier Instance	Checks if a Control Instance's (CI) Control's Barrier is the same as the Control Instance's (CI) Barrier Instance's (BI) Barrier.	Error	“Control instance «controlInstance.name» is associated with barrier «controlInstance.barrier.barrier.name» but the parent control «controlInstance.control.name» does not have an associated barrier” OR “Control instance «controlInstance.name» is associated with barrier «controlInstance.barrier.barrier.name» that is different from the parent control's barrier «controlInstance.control.controlBarrier.name»”	Sets barrier to the control
Check Hazard Function Allocation Deviation	Checks if a Hazard has an allocated Function and a Deviation, and the Function does not have the Deviation as an allocation.	Warning	“Hazard «event.name» has allocation Function «event.allocation.name» and Deviation «event.deviation.name», but «event.allocation.name» does not have «event.deviation.name» as a Deviation”	
Check Hazard Allocation Hierarchy	Checks if a Subhazard has an allocation that is lower in the allocation hierarchy than its super-hazard.	Error	“Hazard «event.name» is a subhazard of «parent.name» but «event.name»'s allocation is higher in the allocation hierarchy than «parent.name»'s allocation”	
Check Hazard Allocation Deviation Hierarchy	Checks if two events are allocated to the same Function, but with different	Warning	“Event «event.name» and «conflictingEvent.name» are allocated to the same function, but have related deviations («event.deviation.name», and «conflictingEvent.deviation.name» respectively)”	

	Deviations belonging to the same family.			
Check Mitigation Requirement is Assigned in Requirement	Checks if a Barrier Instance's (BI) Mitigation Requirement does not have the BI as its source.	Warning	“Mitigation «barrier.name» should be a source for its mitigation requirement «requirement.name»”	Add mitigation as new source to requirement AND/OR Replace existing source with mitigation for the requirement
Check Event has Null Condition	Checks if an Event condition is a string rather than a Failure Mode.	Warning	“Hazard «event.name» has an empty condition string” OR “Hazard «event.name» has condition string «condition» rather than a deviation”	Remove empty condition string from the Event OR Replace condition of the Event with a Failure Mode
Check Escalates and Mitigated By	Checks if an Event Instance (EI) has paths that both escalate and are mitigated by a Control.	Warning	“«event.name» contains paths that both lead to and escalate «controlInstance.name»”	

Check Subhazards are Valid	Checks if an Event has an Event Instance (EI) which does not lead to any of its parent's EIs.	Warning	“Event «event.name» is a subhazard of Event «parent.name» and EventInstance «eventInstance.name» in «hazardousActivity.name», «environmentConditionName», «systemStateName» does not contain an instance of «parent.name» as a consequence” OR “Event «parent.name» is a parent of Event «event.name» and EventInstance «parentEventInstance.name» in «hazardousActivity.name», «environmentConditionName», «systemStateName» does not contain an instance of «event.name» as a threat”	Add the parent instance as an effect of the Event Instance OR Create new instance of the event as (an effect/ a cause) of the existing Event Instance

Data Map DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Data Map version tag is not "supported" or non-existent. Currently the only supported version tag is 1.0.	Error	“The Data Mapping «dataMap.name» should declare a syntax version” OR “The syntax version of the Data Mapping «dataMap.name» is unsupported. Supported versions are: «DataMapConstants.SUPPORTED_VERSIONS.join(', ')».” OR “The syntax version of the Data Mapping «dataMap.name» is not the latest. This file will be migrated upon close/reopen.”	

Check Repeated Node Targets	Checks if a node N is set to repeat but is not within an enclosing node N. This is either because the node is not defined at all, or the use is outwith the loop. If the node is not defined, then there will also be a linking error.	Error	“There is no containing mapping for «node.externalData.segments.map[targetName].join(".")» to repeat”	

Evidence DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Evidence Definition version tag is not "supported" or non-existent. Currently, the supported version tags are 1.0, 1.1, 1.2, and 1.3.	Warning	“The Evidence Definition «evidenceDef.name» does not declare a syntax version” OR “The syntax version of the Evidence Definition «evidenceDef.name» is unsupported. Supported versions are: «String.join(", ", EvidenceDefinitionModelConstants.SUPPORTED_VERSIONS)»” OR “The syntax version of the Evidence Definition «evidenceDef.name» is not the latest. This file will be migrated upon close/reopen”	
Check Evidence Artifact Used	Checks if an Evidence Artifact is referenced by either an argument node or a Requirement as verification allocation.	Warning	“Evidence artifact «evidenceArtifact.name» is not referenced by any requirements or any argument nodes”	

Check Location URI Valid	Checks if the location of an Evidence Artifact specified cannot be found in the workspace.	Warning	“The referenced file '«location.uri»' does not exist”	
Check Type Is Set	Checks that an Evidence Artifact has a type.	Warning	“Evidence artifact: «evidenceArtifact.name» has no type set”	
Check Dependency Is Valid	Checks if an Evidence Artifact has a dependency with a reference of the wrong type. Or, if the dependency has too many references listed.	Error	“«evidenceArtifact.name» has a generatedBy dependency that does not reference an Evidence Artifact Tool” OR “evidenceArtifact.name» has a «dependency.type.name» dependency that does not reference any Evidence Artifact(s)” OR “«evidenceArtifact.name» uses «dependency.type.name» in a one-to-many dependency, but it is not declared as one-to-many”	

Functional Specifications DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Functional Specifications version tag is not "supported" or non-existent. Currently, the supported version tags are 1.0, 1.1, and 1.2.	Error	“The Functional Decomposition «funcSpec.name»should declare a syntax version” OR “The syntax version of the Functional Decomposition «funcSpec.name» is unsupported. Supported versions are: «FuncSpecConstants.SUPPORTED_VERSIONS.join(', ')»” OR “The syntax version of the Functional Decomposition «funcSpec.name» is not the latest. This file will be migrated upon close/reopen”	Add missing version number OR Correct version number
Check Func Dev Has Event	Check if some Function has a Deviation, but there is no Event to which this Function and Deviation are allocated.	Warning	“No hazard is allocated to this combination of allocation and deviation («function.name», «deviation.name»)”	
Check Duplicate Function Name	Checks if a Function shares an unqualified name with a Deviation or another Function.	Error	'There is another Deviation with the same name as Deviation «deviation.name»'	

Check Duplicate Deviation Name	Checks if a Deviation shares a name with a Function or another Deviation.	Error	“There is a Function with the same name as Deviation «deviation.name»”	
Check Duplicate Allocations	Checks if a Function is allocated to the same Component multiple times.	Warning	“Duplicate allocation: Function «function.name» has allocation «it.name» listed multiple times”	
Check Function Missing Deviations	Checks if a Function has no Deviations.	Warning	“Function «function.name» has no deviations”	
Check Duplicate Deviations	Checks if a Function has the same Deviation associated twice.	Warning	“Duplicate deviations: Function «function.name» has deviation «it.name» listed multiple times”	
Check Allocation Same Group	Checks if a Function has multiple allocations belonging to the same group (e.g., component + sub-component as allocation).	Warning	“Redundant allocation: Function «function.name» has two allocations of the same group («component.name», «ancestorComponent.name»)”	
Check Deviation Same Group	Checks if a Function has multiple Deviations	Warning	“Redundant deviations: Function «function.name» has two deviations of the same group («deviation.name», «ancestorDeviation.name»)”	

	belonging to the same group.			
Check Deviation Hierarchy	Checks if a Function's children have Deviations of a higher order than itself (if they are within the same Deviation tree).	Warning	“Inconsistent deviation hierarchy: Function «ancestorFunction.name» has sub-function «function.name» which is allocated to a super-deviation («ancestorDeviation.name», «searchDeviation.name», respectively)”	
Check Allocation Hierarchy	Checks if a Function's children have allocations of a higher order than itself (if they are within the same Component tree).	Warning	“Inconsistent allocation hierarchy: Function «ancestorFunction.name» has sub-function «function.name» which is allocated to a super-component («ancestorComponent.name», «searchComponent.name», respectively)”	

Data Tree DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Data Tree version tag is not "supported" or non-existent. Currently, the supported version tags are 1.0 and 1.1.	Error	“The Evidence Definition «evidenceDef.name» does not declare a syntax version” OR “The sytax version of the Evidence Definition «evidenceDef.name» is unsupported. Supported versions are: «String.join(", ", EvidenceDefinitionModelConstants.SUPPORTED_VERSIONS)»” OR “The syntax version of the Evidence Definition «evidenceDef.name» is not the latest. This file will be migrated upon close/reopen”	

Pattern DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Pattern version tag is not "supported" or non-existent. Currently, the supported version tag are 1.0, 1.1, and 1.2.	Error	“The Pattern «pattern.name» should declare a syntax version” OR “The syntax version of the Pattern «pattern.name» is unsupported. Supported versions are: «PatternConstants.SUPPORTED_VERSIONS.join(', ')»” OR “The syntax version of the Pattern «pattern.name» is not the latest. This file will be migrated upon close/reopen”	
Check Strategy Is Supported By	Checks if a Strategy node is supported by another Strategy or a Solution node.	Warning	“Strategy «from.name» is supported by «to.name». A Strategy is generally not supported by a «to.typeString»”	
Check Solution In Context Of	Checks if a Solution node is supported by a Justification, Context, or Assumption node.	Warning	“Solution «from.name» is in context of «to.name». A Solution is generally not in context of a(n) «to.typeString»”	
Check Goal Supported By Goal	Checks if a Goal node is supported by another Goal node.	Warning	“Goal «from.name» is supported by «to.name». A Goal is generally not supported by another Goal”	

Check Choice Links	Checks if not all options on a Choice node are of the same pattern node type.	Warning	“Choice «choice.name» has options of different types. All options for a Choice should be of the same node type”	
Check Templated Parameter Redefinition	Checks if a Local Parameter has a name that is shared with a Local Parameter defined in terms of a Top-Level-Named Object.	Error	“Parameter '«parameter.name»' is defined in terms of a top-level object and may not be shadowed by definitions further down the pattern chain”	
Check Duplicate Global Parameter	Checks if more than one Global Parameter has the same name.	Error	“Duplicate parameter name: «it.name»”	Remove Duplicate

Physical Architecture DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Physical Architecture version tag is not "supported" or non-existent. Currently, the supported version tag are 1.0, 1.1, and 1.2.	Error	“The Physical Decomposition «physArch.name» should declare a syntax version” OR “The syntax version of the Physical Decomposition «physArch.name» is unsupported. Supported versions are: «PhysArchConstants.SUPPORTED_VERSIONS.join(', ')»” OR “The syntax version of the Physical Decomposition «physArch.name» is not the latest. This file will be migrated upon close/reopen”	
Check Comp Fail Has Event	Checks if some Component has a Failure Mode, but there is no Event to which this Component and Failure Mode are allocated.	Warning	"No hazard is allocated to this combination of component and failure mode («component.name», «failureMode.name»)"	
Check Missing Failure Modes	Checks if a Component is missing Failure Modes.	Warning	"Component «component.name» has no failure modes"	

Check Duplicate Component Name	Checks if there are multiple Components with the same name.	Error	“There are other Components with the same name as Component «component.name»”	

Requirements DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Requirements Log version tag is not "supported" or non-existent. Currently, the supported version tags are 1.0 and 1.1.	Error	“The Requirements Table «requirementsTable.name» should declare a syntax version” OR “The syntax version of the Requirements Table «requirementsTable.name» is unsupported. Supported versions are: «RequirementsConstants.SUPPORTED_VERSIONS.join(', ')»” OR “The syntax version of the Requirements Table «requirementsTable.name» is not the latest. This file will be migrated upon close/reopen”	
Check Allocation Consistency	Checks if a Requirement is missing any physical allocations of its functional allocations (the Requirement/ Function/ Component triangle).	Warning	“Requirement «requirement.name» is missing component(s) of its allocation «function.name». ("'«alloc1»,«alloc2», ...)”	Add missing allocations from the sub-allocation

Check Duplicate Allocations	Checks if a Requirement is allocated to the same item (Function or Component) multiple times.	Warning	“Duplicate allocations: Requirement "«requirement.name»" has allocation "«alloc.name»" listed multiple times”	
Check Allocation Same Group	Checks if a Requirement has multiple allocations belonging to the same group (e.g., component + sub-component as allocation).	Warning	“Redundant allocations: Requirement «requirement.name» has two allocations of the same group («item1.name», «item2.name»)”	
Check Cyclic Parents	Checks if a Requirement has itself as an ancestor.	Error	“Requirements form a cyclic parent relationship, R1 -> R2 -> R1”	
Check Allocation Hierarchy	Checks if a Requirement's ancestors have allocations of a lower order than itself (if they are within the same Component tree).	Warning	“Inconsistent allocation hierarchy: Requirement "«ancestorRequirement.name»" has sub-requirement "«requirement.name»" which is allocated to a super-«allocationType». («ancestorItem.name», «searchItem.name», respectively)”	
Check Safety Requirement	Checks if a “safety” Component is allocated to a Requirement with a type other than "safety".	Warning	“Safety component "+ (it as Component).name + " is allocated to non-safety requirement "«Requirement.name»”	

Table Specification DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the Table Definition version tag is not "supported" or non-existent. Currently, the only supported version tag is 1.0.	Error	“The Table Definition «tableDef.name» should declare a syntax version” OR “The syntax version of the Table Definition «tableDef.name» is unsupported. Supported versions are: «TableDefConstants.SUPPORTED_VERSIONS.join(', ')»” OR “The syntax version of the Table Definition «tableDef.name» is not the latest. This file will be migrated upon close/reopen”	

Check Parent Cycle	Checks if there is a cycle (circular reasoning of any length) in the column parent chain (i.e., either directly via parent declarations or through accessing other columns' values). Also checks if the parent column is never used.	Error	“This column is referenced in the parent hierarchy already; cycles are not allowed” OR “This column cannot be a child of the parent column” OR “The parent column is specified but never used; this may result in different spanning than you expected”	
Check Rows Exists	Checks if a Table Specification has no rows defined.	Warning	“Columns specified with no rows”	
Check Duplicate Header Columns	Checks if more than one header column is defined.	Error	“Cannot have more than one header column”	
Check Rows Are Valid	Checks if rows are not defined to be a list type.	Error	“Rows must represent a model element of list type”	
Check Column Value is Valid	Checks if a value of a column is not a valid model element.	Error	“Value must represent a model element”	

View Specification DSL Validations

Name	Description	Validation Severity	Validation Text	Quick Fix
Check Valid Version	Checks if the View Specification version tag is not "supported" or non-existent. Currently, the supported version tags are 1.0 and 1.1.	Error	“The View Specification «viewSpec.name» should declare a syntax version”	
Check Unused Figures	Checks that all defined Figures are used in the View Specification. "Figure" is the Sirius term for the graphical elements that form a diagram. Only applies to graph views.	Warning	“Figure «name»” is not used in this view specification”	
Check Unused Links	Checks if a defined link is not used in the View Specification.	Warning	“Link «name»is not used in this view specification.”	

Check Hidden Figures	Checks if a Figure is hidden that is not an implicit argument figure (i.e., one of the argument node types, such as Goal, Strategy, etc.). "Figure" is the Sirius term for the graphical elements that form a diagram. Only applies to graph views.	Error	“Only implicit argument figures in argument views can be hidden”	

Diagram Validations

Common Diagram Validations

Name	Description	Validation Severity	Validation Text
Validate Name	A name must start with a letter or underscore, and following that can only contain letters, numbers, underscores, or dashes. Or, it can consist of "^" followed by such a well-formed name.	Impossible ("Impossible" means that the validation prevents this happening.)	“That name is either empty, invalid, or duplicates the name of another «Type» in the diagram.”

Pattern Diagram Validations

Name	Description	Validation Severity	Validation Text
Unambiguous Multiplicity Validation	Checks if a node has any ambiguous multiplicities per the Pattern Theory.	Error	“Multiplicity is ambiguous for nodes " + resolversNames.toString”
Strong Unambiguous Multiplicity Validation	Checks if a node has any strong ambiguous multiplicities per the Pattern Theory.	Warning	“It is good practice to resolve a multiplicity directly after the multiplicity symbol. This multiplicity is not always resolved directly”
Unambiguous Choice Validation	Checks if a link has any ambiguous choices per the Pattern Theory.	Error	“Choice is ambiguous for nodes " + resolversNames.toString”
Strong Unambiguous Choice Validation	Checks if a node has any strong ambiguous choices per the Pattern Theory.	Warning	“It is good practice to resolve a choice directly after the choices symbol. This choice is not always resolved directly”

Table Validations

Requirements Table Validations

Name	Description	Validation Severity	Validation Text
Requirements Name Validation	Checks if a Requirement has a name.	Error	“Missing Requirement ID”
Requirements Unique Name Validation	Checks if two or more Requirements have the same name.	Error	“The Requirements %s share the same ID”
Requirements Description Validation	Checks if a Requirement has a Description.	Warning	“Requirement %s does not have a description”
Requirements Type Validation	Checks if a Requirement has at least one Type.	Error	“Requirement %s does not have a Type”
Requirements Source Validation	Checks if a Requirement has at least one Source.	Warning	“Requirement %s does not have any Sources”
Requirements Allocation Validation	Checks if a Requirement has at least one Allocation.	Warning	“Requirement %s does not have any Allocations”
Requirements Verification Method Validation	Checks if a Requirement has at least one Verification Method.	Warning	“Requirement %s does not have any Verification Methods”
Requirements Verification Allocation Validation	Checks if a Requirement has at least one Verification Allocation.	Warning	“Requirement %s does not have any Verification Allocations”

Verification Method Verification Allocation Validation	Checks if a Verification Method has at least one Verification Allocation.	Warning	“Verification Method %s does not have any Verification Allocations”
Unused Source Validation	Checks if a Source is not used.	Warning	“Source %s is unused”
Unused Verification Method Validation	Checks if a Verification Method is not used.	Warning	“Verification Method %s is unused”

Traceability Table

The **Traceability Table** lists the current traceability and navigation possibilities between various AdvoCATE Artifacts.

From \ To	Hazard Log	Requirements Log	CESs / BTDs	Argument Structures	Evidence Log	Tools Log	PhysArch / FuncSpec
Hazard Log		*Mitigation Requirement & Requirement Associations* *Mitigation Verification & Verification Method Associations* *Navigate from Mitigation Requirement to Requirement* *Navigate from Hazard to Requirement Associations*	*Hazard Log & Safety Architecture Element Associations* *Mitigation Requirements & Barrier/Control Allocations* *Navigate from Hazard to BTD Top Event* *Navigate from Hazard to BTD Threat Event* *Navigate from Hazard to BTD Consequence Event* *Navigate from Mitigation Requirement to BTD Barrier* *Navigate from Mitigation Requirement*	*Hazard & Argument Associations* *Associate Hazard to an Argument Structure or GSN Node* *Navigate from Hazard to GSN Nodes* *Navigate from Hazardous Activity to an Argument Structure*	TBD	TBD	*Hazard Allocations & Components/ Functions* *Hazard Conditions & Failure Modes/ Deviations* *Navigate from Hazard Log to the PhysArch/ FuncSpec*

			to BTD Control				
Requirements Log	*Requirement & Mitigation Requirement Associations* *Requirement & Mitigation Associations* *Associate Requirement to Hazard Mitigation* *Navigate from Requirement to Hazard Associations*		*Requirements & Barrier/Control Allocations* *Associate Requirement to Barrier* *Associate Requirement to Control* *Navigate from Requirement to Hazard in a BTD* *Navigate from Requirement to Control/Barrier Instances in a CES/BTD*	*Requirement & Argument Associations* *Associate Requirement to GSN Nodes* *Navigate from Requirement to GSN Nodes* *Navigate from Requirement to Solution Nodes*	*Verification Allocations & Evidence Artifacts* *Navigate from Verification Allocation to Evidence Artifacts*	*Verification Allocations & Tool Uses*	*Requirement Allocations & Components/ Functions* *Navigate from Requirement Allocations to the PhysArch/ FuncSpec*
CESs / BTDs	*Safety Architecture Elements & Hazard Log Associations* *Navigate from CES/BTD Node to Hazard Log* *Navigate from CES/BTD Requirement to Hazard Log*	*Barrier & Control Requirement Allocations* *Associate CES/BTD Barrier Nodes to Requirements* *Associate CES/BTD Control Nodes to Requirements*		*CES/BTD Nodes & Argument Associations* *Associate CES/BTD Node to Argument Structure* *Navigate from CES/BTD Node to Argument Structure*	*CES/BTD Barrier/Control Nodes & Integrity Evidence* *Associate CES/BTD Barrier/Control Node to Integrity Evidence* *Create a New Evidence Artifact &*	*Integrity Evidence, Initial Likelihood Evidence, Initial Severity Evidence & Tool Uses*	*CES/BTD Barrier/Control Nodes & Physical Allocations* *Allocate CES/BTD Barrier/Control Nodes to Components* *CES/BTD Barrier/Control Nodes &*

		Create a New Requirement & Associate to a CES/BTD Barrier/Control Node *Navigate from CES/BTD Barrier/Control Node to Requirement*			*Associate to CES/BTD Barrier/Control Node's Integrity* *CES/BTD Event Nodes & Initial Likelihood Evidence* *Associate CES/BTD Event Node to Initial Likelihood Evidence* *Create a New Evidence Artifact & Associate to CES/BTD Event Node's Initial Likelihood* *CES/BTD Event Nodes & Initial Severity Evidence* *Associate CES/BTD Event Node to Initial Severity Evidence* *Create a New Evidence Artifact & Associate to CES/BTD*		*Functional Allocations* *Allocate CES/BTD Barrier/Control Nodes to Functions*

					Event Node’s Initial Severity *Navigate from CES/BTD Barrier/Control Node Integrity Evidence to Evidence Log DSL* *Navigate from CES/BTD Barrier/Control Node Integrity Evidence to Evidence Log Diagram* *Navigate from CES/BTD Event Node Initial Likelihood Evidence to Evidence Log DSL* *Navigate from CES/BTD Event Node Initial Likelihood Evidence to Evidence Log Diagram* *Navigate from CES/BTD Event Node Initial Severity Evidence to Evidence Log DSL*		

					Navigate from CES/BTD Event Node Initial Severity Evidence to Evidence Log Diagram		
Argument Structures	*Arguments & Safety Architecture Element Associations in the Hazard Log* *Associate Argument Structures to Safety Architecture Elements* *Associate GSN Nodes to Safety Architecture Elements* *Navigate from Argument Structure to Safety Architecture Elements in the Hazard Log* *Navigate from GSN Node to*	*GSN Nodes & Requirements* *Associate GSN Nodes to Requirements* *Navigate from GSN Node to Requirement*	*Arguments & Safety Architecture Element Associations in CESs/BTDs* *Associate Argument Structures to Safety Architecture Elements in CESs/BTDs* *Associate GSN Nodes to Safety Architecture Elements in CESs/BTDs* *Navigate from Argument Structure to Safety Architecture Elements in CESs/BTDs* *Navigate from GSN Node to Safety*		*GSN Nodes & Evidence Artifacts* *Associate GSN Nodes to Evidence Artifacts* *Create a New Evidence Artifact & Associate to a GSN Node* *Navigate from GSN Node to Evidence Artifacts*	*GSN Node Evidence & Tool Uses*	N/A

	Safety Architecture Elements in the Hazard Log		*Architecture Elements in CESs/BTDs*				
Evidence Log	N/A	N/A	N/A	TBD		*Evidence Artifacts & Tool Uses*	N/A
Tools Log	N/A	*Tool Uses & Requirements* *Navigate from Tool Use to Requirements*	N/A	*Tool Uses & GSN Nodes* *Navigate from Tool Use to GSN Nodes*	*Tool Uses & Evidence Artifacts* *Navigate from Tool Use to Evidence Artifacts*		N/A
PhysArch / FuncSpec	N/A	N/A	N/A	N/A	N/A	N/A	