NASA/TM—2019–220196

Air Traffic Management TestBed Simulation Architect: User's Guide

Chok Fung Lai
Ames Research Center, Moffett Field, California

April 2019

NASA/TM—2019–220196

Air Traffic Management TestBed Simulation Architect: User's Guide

Chok Fung Lai
Ames Research Center, Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, CA 94035-1000

April 2019

Acknowledgments

The author would like to thank the other ATM TestBed team members, John Robinson, James Murphy, Alan Lee, Dr. Gano Chatterji, Phu Huynh, Huu Huynh, Jimmy Nguyen, Yun Zheng, and Richard Jessop, for their contributions to the design and development of the Simulation Architect. The author would also like to thank Gregory Wong, Dr. Gano Chatterji, and Confesor Santiago, and Katharine Lee for reviewing this technical memorandum.

This report is available in electronic form at

https://ntrs.nasa.gov/

Abstract

The Air Traffic Management (ATM) TestBed is a Platform as a Service that is being developed by the National Aeronautics and Space Administration (NASA) to help design, configure, integrate, run, and monitor air traffic simulations. The platform provides cloud services including back-end big-data analytics tools, on-demand computing resource management, data storage, and communication middleware. The ATM TestBed reduces the time to test concepts and technologies, supports interactions among various concepts such as human-in-the-loop and automation-in-the-loop simulations, and enables collaborative simulations by sharing technologies and tools in the ATM community. The Simulation Architect application provides a graphical user interface tool for designing traffic scenarios and simulations using blocks representing components and links representing message channels linking them. This guide describes a high-level user interface design of Simulation Architect and provides information for a new user to compose traffic scenarios and simulations.

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

1. Introduction

Composing meaningful air traffic simulations is challenging. One reason is the task of manually generating realistic scenarios for human-in-the-loop simulations is difficult and time consuming.[1,2] Running distributed air traffic simulations usually requires simulation components—applications or data streams—via a network infrastructure interconnected in a simulation, geographically located at multiple facilities. Figure 1.1 shows a distributed air traffic simulation using simulation components located at three facilities. Each facility has its own network environment, the facilities are interconnected via a cloud network.

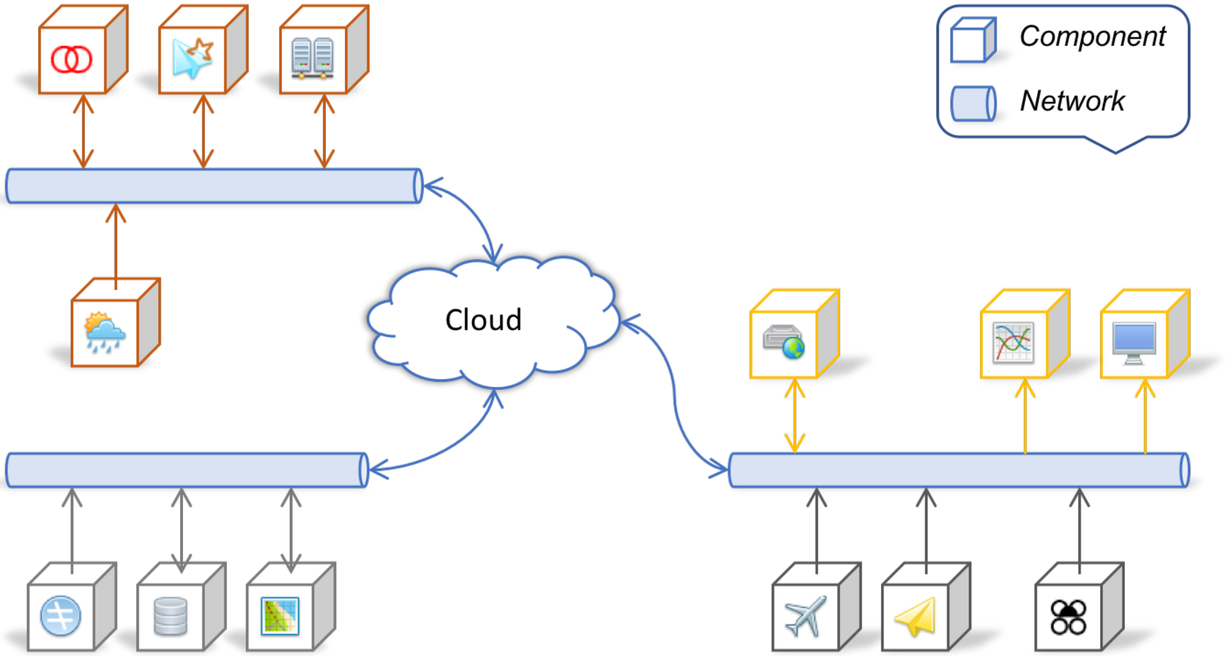


Figure 1.1. A distributed air traffic simulation

It is widely accepted that integrating various capabilities, such as assets, concepts and tools, into a unified simulation framework can provide potential benefits to many projects[3,4,5]. However, executing individual capabilities requires domain knowledge, and setting up correct configurations is error prone. In addition, in order to run live, virtual, and constructive simulations that can be distributed, scalable, asynchronous real-time or synchronized fast-time using different operating systems and machine capabilities, there is a need for the framework to support simulation design, execution, and configuration management.

To address these challenges, the Air Traffic Management (ATM) TestBed under the ATM-eXploration (ATM-X) Project[6] is being developed by the National Aeronautics and Space Administration (NASA) to accelerate the concept and technology development by standardizing the infrastructure and interface to integrate simulation assets, and by automating critical processes such as simulation setup, execution, and archiving.

Overall, the ATM TestBed consists of four layers—infrastructure, platform, framework, and application. The infrastructure layer provides cloud computing and resource management; the platform layer provides scalable communication middleware; the framework layer provides back-end big-data as well as real-time safety analytics tools, the ATM TestBed Software Development Kit (SDK) for simulation component developers; and the application layer allows researchers and partners to execute shareable simulation assets.

In addition, the ATM TestBed provides an automatic scenario generation capability and a graphical tool called Simulation Architect for composing a simulation. Simulation Architect is a

Java application that can be downloaded to a client machine using a web browser or can be run from a console using the ATM TestBed SDK. The tool is designed to run on major operating systems including Mac OS X, Windows, and Linux. The tool uses the Java Swing toolkit and the Prefuse[7] visualization toolkit to provide an interactive user interface for designing traffic scenarios and simulations. The idea is to use blocks to represent simulation components and links to represent communication channels. Simulation Architect has four main design features:

1. A graphical interface to display accessible simulation components
2. A drag-and-drop editor to specify and connect simulation components
3. An interface to configure individual simulation components
4. An interface to indicate misconfigured simulation components

An ATM TestBed user can launch Simulation Architect by: (1) accessing the Web Portal[8] from a browser[*] as a web-accessible tool, or (2) running an executable script file included in the ATM TestBed SDK[†] from a terminal (Mac) or a command prompt (Windows) as a standalone tool (see Table 1.1). The main difference between the two launch modes is that the web-accessible tool accesses layout files from the ATM TestBed Storage hosted on the Web Portal online, while the standalone tool accesses layout files from the user's hard drive offline.

Table 1.1. Running Simulation Architect from console

Operating System	**Commands**

Mac, Linux	`$ bash`
	`$ export SNTB_HOME=/path/to/smart-nas/`
	`$ cd $SNTB_HOME/TestBedCore/SimulationArchitect/`
	`$ ./bin/run_sim_architect.sh`

Windows	`> set SNTB_HOME=c:\path\to\smart-nas\`
	`> cd %SNTB_HOME%\TestBedCore\SimulationArchitect\`
	`> .\bin\run_sim_architect.bat`

Simulation Architect requires Java Standard Edition (SE)[9] Runtime Environment 8 to run. After launching the tool, the user will see an application frame. Figure 1.2 shows an initial view of Simulation Architect for creating a traffic scenario on the Mac operating system. The application consists of four important widgets: (a) a menu bar, (b) a tool bar, (c) a palette pane, and (d) an editor content.

[*] The site address of the Web Portal is https://atmtbportal.arc.nasa.gov:8000

[†] Please contact the ATM TestBed Development Team, email: chok.f.lai@nasa.gov

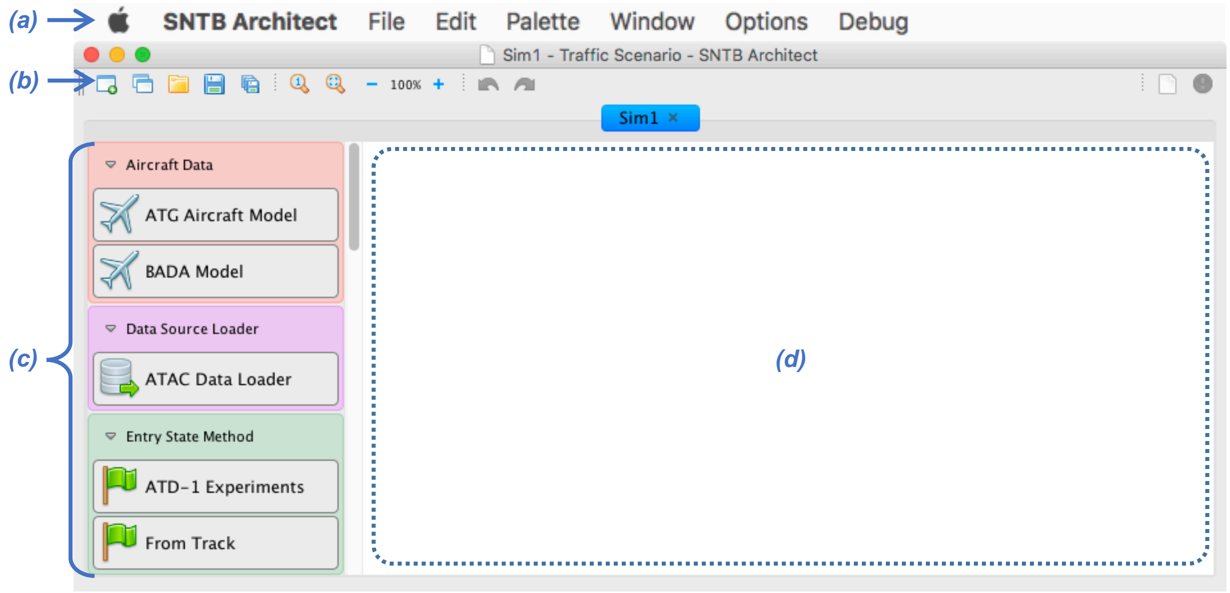


Figure 1.2. Initial view of Simulation Architect: (a) menu bar, (b) tool bar, (c) palette pane, and (d) editor content

Figure 1.3 shows a final view of Simulation Architect for creating a simulation. The application consists of four important widgets: (a) tabs, (b) blocks, (c) links, and (d) a simulation layout.

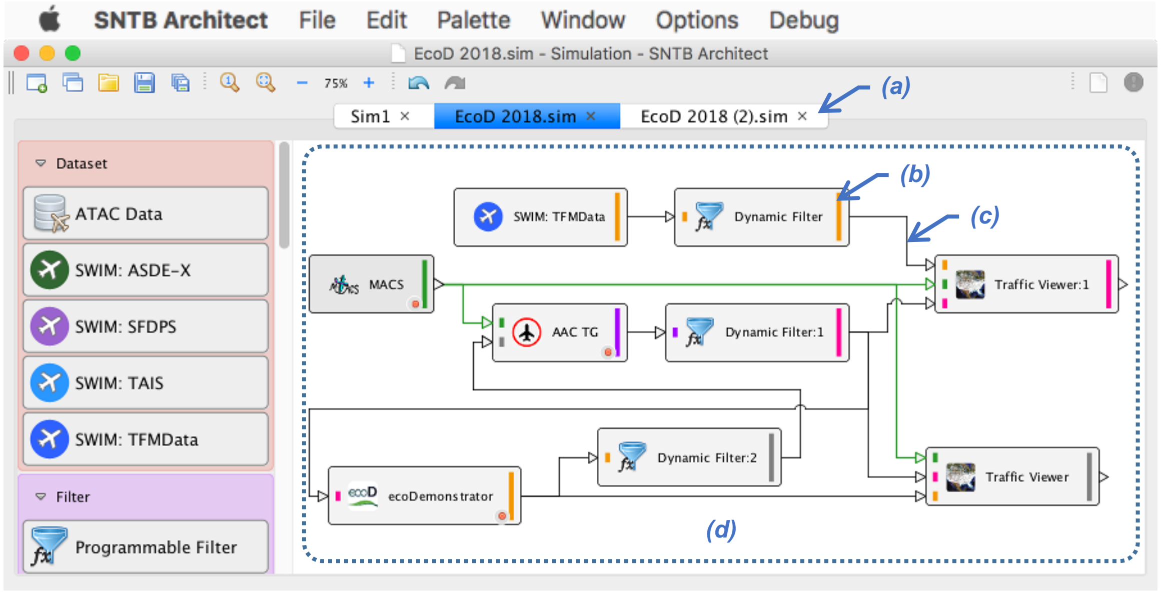


Figure 1.3. Final view of Simulation Architect: (a) tabs, (b) blocks, (c) links, and (d) simulation layout

The following sections describe the user interfaces and widget details.

2. Menu Bar

The menu bar contains six menus for creating, editing, and viewing Simulation Architect layouts. Each menu contains drop-down menu items of commands relating to specific widgets or features. Commands are context sensitive, and menu items with nonexecutable commands are disabled (grayed out). For example, the Cut edit command is disabled when no selection is made in the current editor content.

Shortcut keys are defined for commonly used commands to provide an efficient way to execute the commands without accessing the drop-down menus. Note that operating systems have specific shortcut key masks: Mac uses the command key (⌘), while both Windows and Linux use the control key (Ctrl).

Command and usage of each menu and the functions of its drop-down menu items are presented in the following subsections:

2.1. File—editor contents and layout files
2.2. Edit—edits in the current editor content
2.3. Palette—the palette pane associated with the current editor content
2.4. Window—views of the editor content
2.5. Options—settings for the current editor content
2.6. Debug—settings for software development

2.1. File Menu

Commands in the File menu allow the user to handle layout files associated with existing editor content tabs. Layout files use the JavaScript Object Notation (JSON) text format[10]. Figure 2.1 shows the drop-down menu and its 11 menu items divided into four categories—create, save, show, and exit.

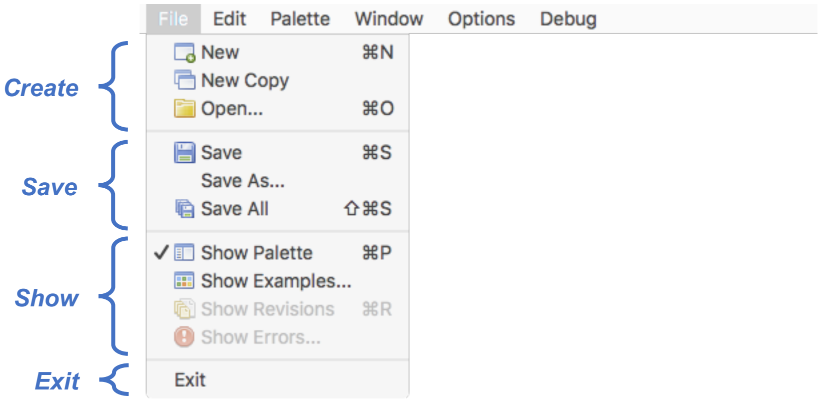


Figure 2.1. File menu items

2.1.1. New

The New command adds a new editor content to the tool. After running the command, a new tab is added and selected, and the user will see a new and empty editor content. The title of the new tab is defined based on the title of the last added tab. A new tab has the initial title *Sim1*; subsequently added tabs will have titles *Sim2*, *Sim3*, and so forth (see Figure 2.2).

This command is useful for creating a new editor content, for opening an example or a revision, and for keeping blocks on another editor content using copy (or cut) and paste commands before modifying the copied blocks. The shortcut key of this command is '⌘ N' (Mac) or 'Ctrl+N' (Windows).

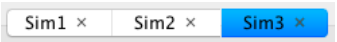


Figure 2.2. New editor content tabs

2.1.2. New Copy

The New Copy command duplicates the current editor content and its settings to a new editor content. After running the command, a new tab is added and selected, and the user will see a copy of the original editor content in the new window. Modifying the content in one window will not affect the content in other windows. The title of the new tab is defined based on the title of the original tab and a counter. The counter is incremented by one whenever the current window is duplicated. For example, if the title of the current tab is *Sim1*, running the New Copy command will add a new tab titled *Sim1 (2)*, and running the command again will add another new tab titled *Sim1 (3)* (see Figure 2.3).

This command is useful for creating a copy of the current editor content before modifications, so that the copy can be restored by using copy (or cut) and paste commands.

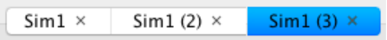


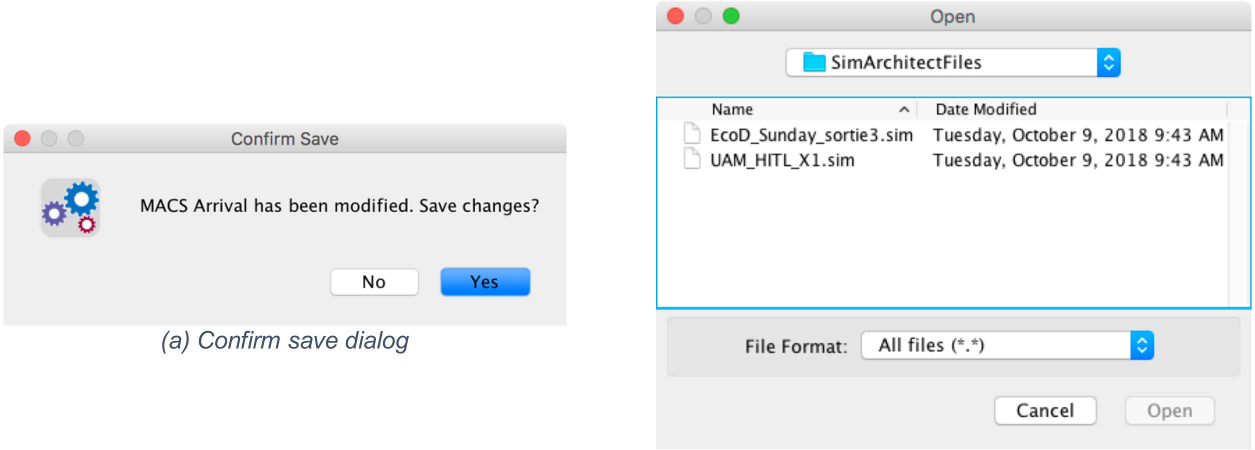
Figure 2.3. New copies of the first editor content

2.1.3. Open…

The Open command lets the user open a layout file. Execution of the command depends on how the user launches Simulation Architect:

1. If the tool is launched from the Web Portal, the command reopens the current layout file from its last revision. Note that if the current editor content is modified, a Confirm Save dialog is shown before the command execution (see Figure 2.4(a)).
2. If the tool is launched from the terminal, the command shows an Open dialog and asks the user to select a layout file to open (see Figure 2.4(b)).

This command is useful for opening a previously saved layout. The shortcut key of this command is '⌘ O' (Mac) or 'Ctrl+O' (Windows).



MACS Arrival has been modified. Save changes?

(a) Confirm save dialog

Name	Date Modified
EcoD_Sunday_sortie3.sim	Tuesday, October 9, 2018 9:43 AM
UAM_HITL_X1.sim	Tuesday, October 9, 2018 9:43 AM

File Format: All files (*.*)

(b) Open layout file dialog

Figure 2.4. Dialogs shown when opening layouts

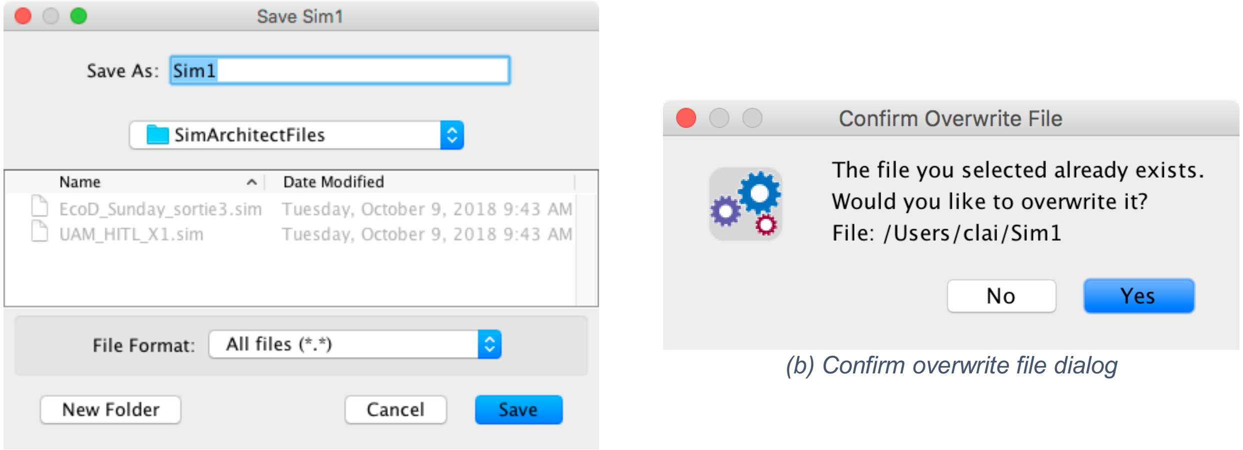
2.1.4. Save

The Save command saves the layout in the current editor content so that the layout can be reopened by running the Open command. Execution of the command depends on how the user launches Simulation Architect:

1. If the tool is launched from the Web Portal, the content is saved as a revision to the ATM TestBed Storage. Thus, all the previously saved layouts are not overwritten and the revisions can be retrieved by running the Show Revisions command.

2. If the tool is launched from the terminal, the command shows a Save dialog and asks the user to enter a filename (see Figure 2.5(a)). If the user enters a filename of an existing file, a Confirm Overwrite File dialog shows (see Figure 2.5(b)):
   a. Click on the Yes button will overwrite the existing file.
   b. Click on the No button will ask the user to enter another filename.

This command is useful for preserving the layout in the current editor content. Note that the command is disabled if Simulation Architect is launched from the Web Portal and the current editor content is not loaded from the ATM TestBed Storage. The shortcut key of this command is '⌘ S' (Mac) or 'Ctrl+S' (Windows).



Save Sim1

Save As: Sim1

SimArchitectFiles

Name	Date Modified
EcoD_Sunday_sortie3.sim	Tuesday, October 9, 2018 9:43 AM
UAM_HITL_X1.sim	Tuesday, October 9, 2018 9:43 AM

File Format: All files (*.*)

Confirm Overwrite File

The file you selected already exists.
Would you like to overwrite it?
File: /Users/clai/Sim1

(a) Save layout file dialog

(b) Confirm overwrite file dialog

Figure 2.5. Dialogs shown when saving layouts

2.1.5. Save As…

The Save As command saves the layout in the current editor content to a file and the layout can be reopened by running the Open command. Execution of the command depends on how the user launches Simulation Architect:

1. If the tool is launched from the Web Portal, the command saves the current editor content to the ATM TestBed Storage. The execution is the same as the Save command described above.
2. If the tool is launched from the terminal, the command shows a Save dialog and asks the user to enter a new filename (see Figure 2.5(a)). If the user enters a non-existent filename, the current editor content will not be saved to the currently opened file but a new file with the user specified filename; otherwise, a Confirm Overwrite File dialog will be shown (see Figure 2.5(b)):
   a. Click on the Yes button will overwrite the existing file.
   b. Click on the No button will ask the user to enter another filename.

This command is useful for saving a copy of the layout in the current editor content to another file. Note that the command is disabled if Simulation Architect is launched from the Web Portal and the current editor content is not loaded from the ATM TestBed Storage (e.g., a new copy of an existing editor content).

2.1.6. Save All

The Save All command saves all the layouts in the editor content tabs so that the layouts can be reopened by running the Open command. The command runs the Save command on each editor content tab. A save dialog is shown when an editor content has been modified and can be saved.

This command is useful for saving all the layouts to files. Note that the command is disabled if Simulation Architect is launched from the Web Portal and none of the editor contents is loaded from the ATM TestBed Storage. The shortcut key of this command is '⇧⌘ S' (Mac) or 'Shift+Ctrl+S' (Windows).

2.1.7. Show Palette

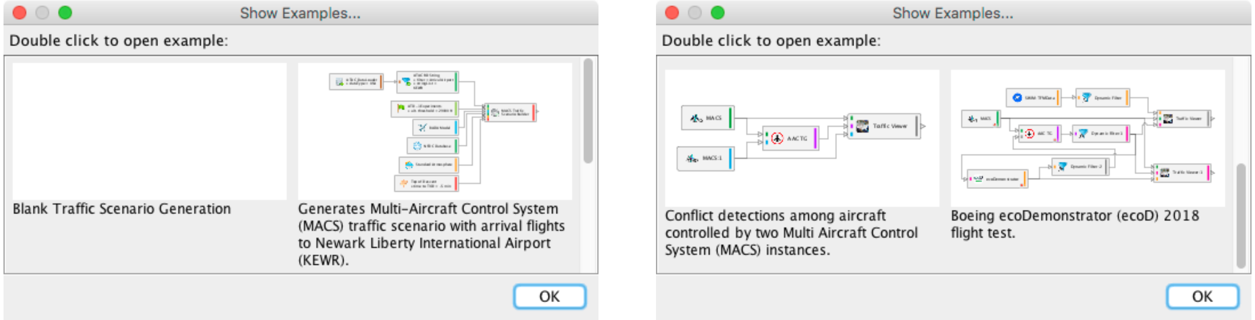
The Show Palette command shows or hides the palette pane associated with the current editor content. Showing the palette pane allows the user to add blocks to the current editor content, while hiding the palette pane provides extra space to show the layout. The palette pane is shown by default.

This command is useful for toggling the visibility of the palette pane. The shortcut key of this command is '⌘ P' (Mac) or 'Ctrl+P' (Windows).

2.1.8. Show Examples…

The Show Examples command shows a dialog listing predefined example layouts (see Figure 2.6). The examples demonstrate how to connect and configure blocks. The user can open an example layout in the current editor content by: (1) double clicking on an example content, or (2) right clicking on an example content and then selecting the Open menu item in the popup menu. Pressing the OK button closes the dialog.

This command is useful for viewing predefined examples. Since opening an example layout will modify the current editor content, the user is advised to create a new editor content before running the command.



Show Examples...

Double click to open example:

Blank Traffic Scenario Generation

Generates Multi-Aircraft Control System (MACS) traffic scenario with arrival flights to Newark Liberty International Airport (KEWR).

Show Examples...

Double click to open example:

Conflict detections among aircraft controlled by two Multi Aircraft Control System (MACS) instances.

Boeing ecoDemonstrator (ecoD) 2018 flight test.

(a) Traffic scenario generation　　*(b) Simulation*

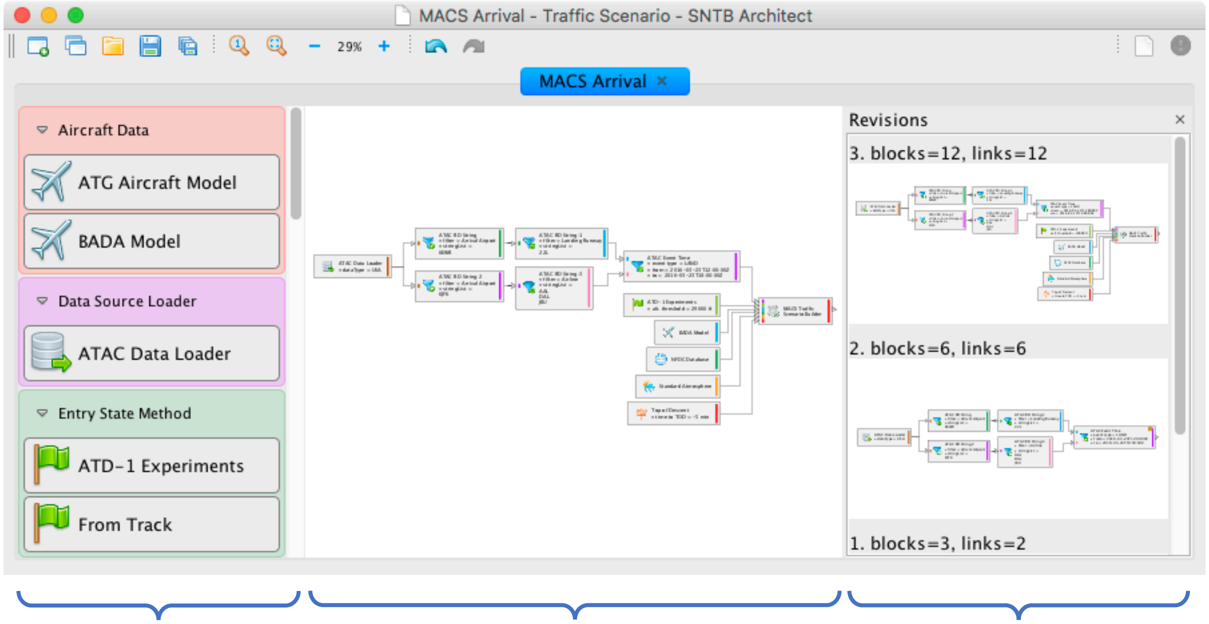
Figure 2.6. Show Examples dialogs

2.1.9. Show Revisions

The Show Revisions command shows or hides the revision pane listing all the previously saved layouts associated with the current file. In the revision pane, double clicking a revision opens a copy of the revision to a new tab titled *<filename> [Revision #]* where the *<filename>* is the name of the layout file and the # is the revision number. The user can copy and paste any blocks from the revision content to current editor content. Figure 2.7 shows the revision pane to the right of the editor content. If the second revision is opened, a new tab titled *MACS*[‡] *Arrival [Revision 2]* is added. Since the revisions are stored on the Web Portal Storage, the command is enabled only if Simulation Architect is launched from the Web Portal.

[‡] MACS stands for Multi-Aircraft Control System.

This command is useful for reviewing or retrieving previous versions of the layout. The shortcut key of this command is '⌘ R' (Mac) or 'Ctrl+R' (Windows).



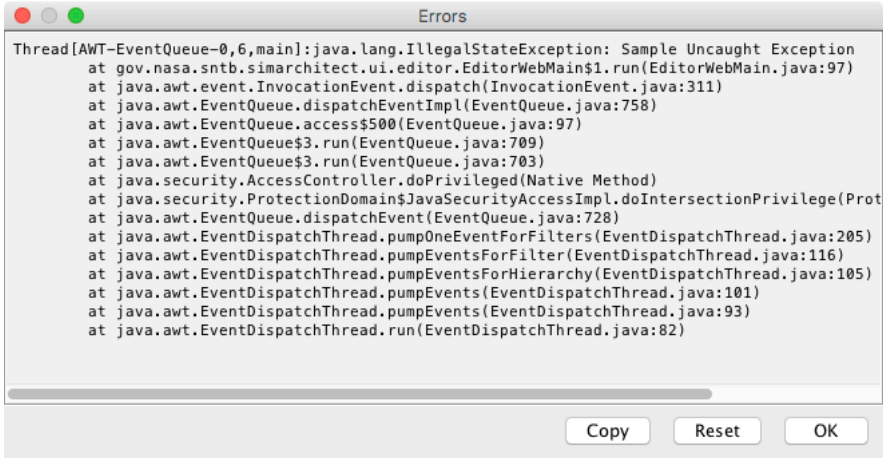
(a) Palette pane (b) Editor content (c) Revision pane

Figure 2.7. Revision pane with previously saved layouts

2.1.10. Show Errors…

The ❗ Show Errors command shows a dialog listing stack traces of uncaught exceptions. Exceptions indicate undesired conditions and they are generally caught in Simulation Architect and warning dialogs are shown. However, an uncaught exception may arise especially during application development. Figure 2.8 shows an error dialog with a sample uncaught exception and its stack trace.

The error dialog is useful for troubleshooting Simulation Architect. The menu item is disabled by default, and it is enabled whenever an uncaught exception happens. The user may: (1) click on the Copy button to copy the errors to the clipboard, and then report the clipboard content to the ATM TestBed development team, (2) click on the Reset button to clear the errors, and the menu item becomes disabled again, or (3) click on the OK button to close the dialog.

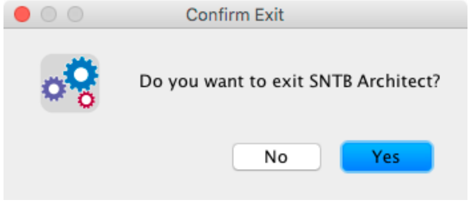


```
Thread[AWT-EventQueue-0,6,main]:java.lang.IllegalStateException: Sample Uncaught Exception
        at gov.nasa.sntb.simarchitect.ui.editor.EditorWebMain$1.run(EditorWebMain.java:97)
        at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:311)
        at java.awt.EventQueue.dispatchEventImpl(EventQueue.java:758)
        at java.awt.EventQueue.access$500(EventQueue.java:97)
        at java.awt.EventQueue$3.run(EventQueue.java:709)
        at java.awt.EventQueue$3.run(EventQueue.java:703)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(Prot
        at java.awt.EventQueue.dispatchEvent(EventQueue.java:728)
        at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:205)
        at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:116)
        at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:105)
        at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
        at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
        at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)
```

Figure 2.8. Error dialog with a sample uncaught exception

2.1.11. Exit

The Exit command shows a dialog and confirms the user to exit Simulation Architect (see Figure 2.9). Clicking on the Yes button confirms the command and the application exits; clicking on the No button cancels the command and the application continues. When editor contents have been modified, the user will be prompted to save them before confirming the exit command.



Do you want to exit SNTB Architect?

Figure 2.9. Confirm Exit dialog

2.2. Edit Menu

Commands in the Edit menu allow the user to edit the blocks and links associated with the current editor content. Figure 2.10 shows the drop-down menu and its 11 menu items divided into four categories—undo, edit, select, and properties.

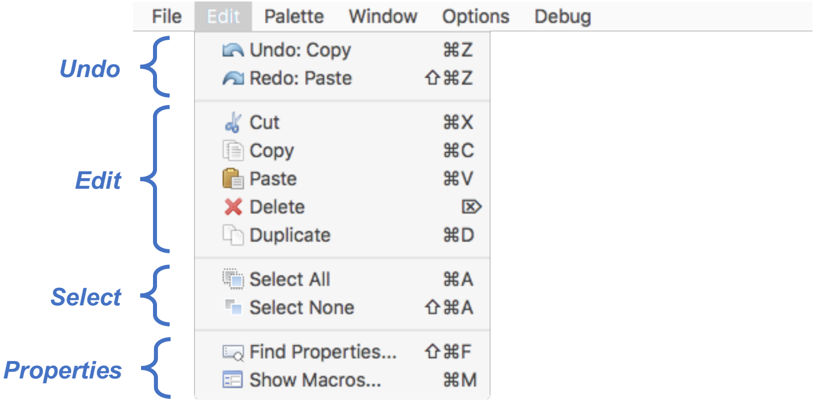


Figure 2.10. Edit menu items

2.2.1. Undo

The Undo command cancels the previously executed command by rolling back its execution. A canceled command may be re-executed by running the Redo command. Simulation Architect stores the last 50 cancelable commands. Most commands are cancelable but some are not such as the Save command and the Exit command.

The menu item is disabled by default, and it is enabled whenever a cancelable command is executed. If the menu item is enabled, the name of the previously executed command will be included in the label, e.g., *Undo: Add Block*.

This command is useful for canceling the execution of one or more unintended commands. The shortcut key of this command is '⌘ Z' (Mac) or 'Ctrl+Z' (Windows).

2.2.2. Redo

The Redo command executes the previously canceled command. The menu item is disabled by default, and it is enabled whenever a command is canceled. If the menu is enabled, the name of the previously canceled command will be included in the label, e.g., *Redo: Add Block*.

This command is useful for re-executing one or multiple previously canceled commands. The shortcut key of this command is '⇧ ⌘ Z' (Mac) or 'Shift+Ctrl+Z' (Windows).

2.2.3. Cut

The Cut command cuts (copies and deletes) the selected items in the current editor content to the clipboard. The copied items can be pasted into any editor content later. The menu item is enabled whenever a selection is made in the current editor content.

This command is useful for moving the selected blocks and links from the current editor content to another editor content. The shortcut key of this command is '⌘ X' (Mac) or 'Ctrl+X' (Windows).

2.2.4. Copy

The Copy command copies the selected items in the current editor content to the clipboard. The copied items can be pasted into any editor content later. The menu item is enabled whenever a selection is made in the current editor content.

This command is useful for duplicating the selected blocks and links to the current or other editor contents. The shortcut key of this command is '⌘ C' (Mac) or 'Ctrl+C' (Windows).

2.2.5. Paste

The Paste command adds the copied items from the clipboard to the current editor content. The menu item is enabled whenever the clipboard contains layout items that are copied via the Cut command or the Copy command.

This command is useful for moving or duplicating the previously copied blocks and links to the current or other editor contents. The shortcut key of this command is '⌘ V' (Mac) or 'Ctrl+V' (Windows).

2.2.6. Delete

The Delete command deletes the selected items in the current editor content. The menu item is enabled whenever a selection is made in the current editor content.

This command is useful for removing the selected blocks and links from the current editor content. The shortcut key of this command is '⌦' (Mac) or 'Delete' (Windows).

2.2.7. Duplicate

The Duplicate command duplicates the selected items in the current editor content. The command performs a copy-and-paste command without using the clipboard; thus, the clipboard content is not modified.

This command is useful for duplicating the selected blocks and links to the current editor content without modifying the clipboard. The shortcut key of this command is '⌘ D' (Mac) or 'Ctrl+D' (Windows).

2.2.8. Select All

The Select All command selects all the items in the current editor content. The user can also select all the items by right clicking on the current editor content and then dragging the cursor to select the items.

This command is useful for selecting all the blocks and links in the current editor content. The shortcut key of this command is '⌘ A' (Mac) or 'Ctrl+A' (Windows).

2.2.9. Select None

The Select None command clears selections of all the items in the current editor content. The user can also clear the selections by right clicking on an empty area in the current editor content.

This command is useful for disabling the Cut, Copy, Delete, and Duplicate commands for the current editor content. The shortcut key of this command is '⇧ ⌘ A' (Mac) or 'Ctrl+Shift+A' (Windows).

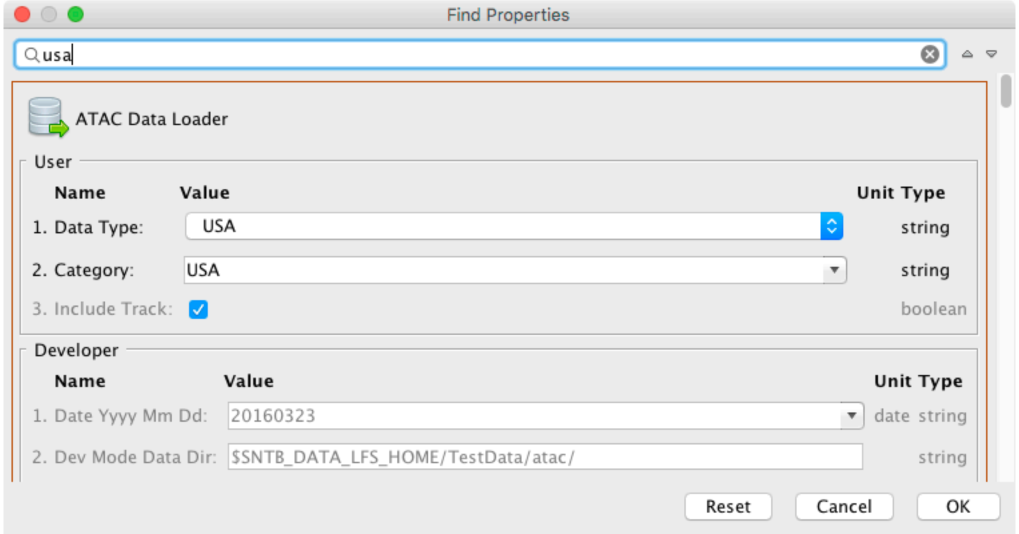
2.2.10. Find Properties…

The Find Properties command shows a dialog listing all user-accessible block properties in the current editor content. The dialog allows the user to view, find, and modify the values of the properties. This menu item is enabled whenever the current editor content is not empty, i.e., contains one or multiple blocks.

In the dialog, the user can search for properties by their names, values, or types. Whenever a search text is entered in the search bar, nonmatching properties are grayed out (see Figure 2.11). Three buttons are located at the bottom of the dialog:

1. Clicking on the Reset button cancels all the current changes without closing the dialog.
2. Clicking on the Cancel button cancels all the current changes and closes the dialog.
3. Clicking on the OK button accepts all the current changes and closes the dialog.

This command is useful for handling all the blocks properties in a single dialog. The shortcut key of this command is '⇧⌘F' (Mac) or 'Ctrl+Shift+F' (Windows).



Find Properties

usa

ATAC Data Loader

User

Name	Value	Unit Type
1. Data Type:	USA	string
2. Category:	USA	string
3. Include Track:	☑	boolean

Developer

Name	Value	Unit Type
1. Date Yyyy Mm Dd:	20160323	date string
2. Dev Mode Data Dir:	$SNTB_DATA_LFS_HOME/TestData/atac/	string

Figure 2.11. Find Properties dialog with a user specified search text

2.2.11. Show Macros…

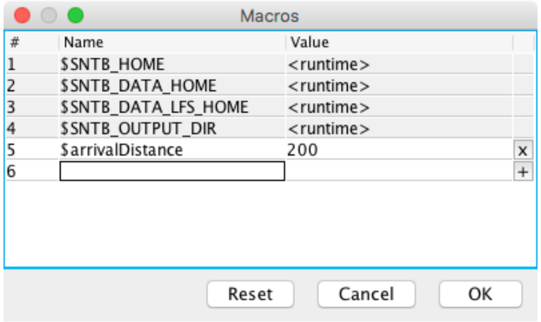
The Show Macros command shows a dialog listing all the defined macros in the current editor content (see Figure 2.12). Macros provide a mechanism for defining shareable values among multiple properties. Macro names are strings starting with a dollar sign ($), while macro values can be any string.

The dialog has a table containing four columns: macro number, macro name, macro value, and action button. Each row in the table represents a macro. The add (+) button in the action button column indicates a new macro to be added. The delete (×) buttons are available to user-defined macros. Clicking on the delete button removes the macro from the table.

In addition, three buttons are located at the bottom of the dialog:

1. Clicking on the Reset button cancels all the current changes without closing the dialog.
2. Clicking on the Cancel button cancels all the current changes and closes the dialog.
3. Clicking on the OK button accepts all the current changes and closes the dialog.

This command is useful for defining reference values in one or multiple properties. For example, in Figure 2.11, the value of the *Dev Mode Data Dir* property is a string containing a macro named *$SNTB_DATA_LFS_HOME*. The shortcut key of this command is '⌘ M' (Mac) or 'Ctrl+M' (Windows).



#	Name	Value
1	$SNTB_HOME	<runtime>
2	$SNTB_DATA_HOME	<runtime>
3	$SNTB_DATA_LFS_HOME	<runtime>
4	$SNTB_OUTPUT_DIR	<runtime>
5	$arrivalDistance	200
6		

Figure 2.12. Macros dialog with predefined and user-defined macros

2.3. Palette Menu

Commands in the Palette menu allow the user to handle the blocks in the palette pane associated with the current editor content. Details of the palette pane are described in the Section 4. Figure 2.13 shows the drop-down menu and its nine menu items divided into three categories—find, collapse, and group by.

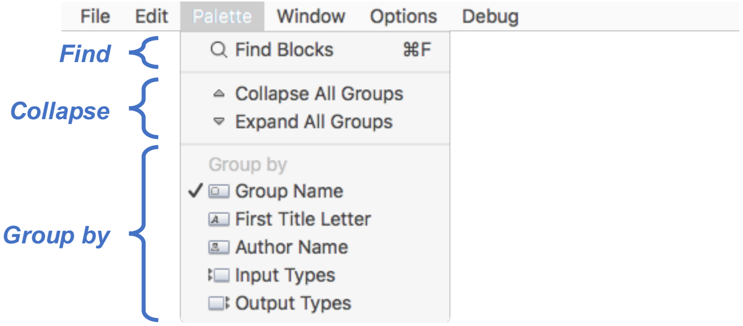


Figure 2.13. Palette menu items

2.3.1. Find Blocks

The 🔍 Find Blocks command shows the find bar located at the top of the palette pane. The user can enter search terms to search for the blocks. Matching blocks will be shown and nonmatching blocks will be hidden. Groups that do not have matching blocks will be hidden.

This command is useful for finding blocks by their names. The shortcut key of this command is '⌘ F' (Mac) or 'Ctrl+F' (Windows).

2.3.2. Collapse All Groups

The ▴ Collapse All *Groups* command collapses all the groups in the palette pane. This command is useful for viewing the groups without showing their blocks.

2.3.3. Expand All Groups

The ▾ Expand All Groups command expands all the groups in the palette pane. This command is useful for showing all the blocks after running the Collapse All Groups command.

2.3.4. Group By

The Group By menu item indicates a header for the following five mutually exclusive settings: Group Name, First Title Letter, Author Name, Input Types, and Output Types. This menu item is always disabled.

2.3.5. Group Name

The Group Name setting allows the blocks in the palette pane to be grouped by their group names. Each block will be shown in exactly one group. This setting is useful for grouping the blocks based on their classification. This menu item is selected by default.

2.3.6. First Title Letter

The First Title Letter setting allows the blocks in the palette pane to be grouped by the first letter in each block's title. Each block will be shown in exactly one group. This setting is useful for listing the blocks in alphabetical order.

2.3.7. Author Name

The Author Name setting allows the blocks in the palette pane to be grouped by their author names. Each block will be shown in exactly one group. This setting is useful for identifying the creators of the blocks.

2.3.8. Input Types

The Input Types setting allows the blocks in the palette pane to be grouped by their input types. A block can be shown in multiple groups if it supports more than one input type. This setting is useful for listing the input types and their supported blocks.

2.3.9. Output Types

The Output Types setting allows the blocks in the palette pane to be grouped by their output types. A block can be shown in multiple groups if it supports more than one output type. This setting is useful for listing the output types and their supported blocks.

2.4. Window Menu

Commands in the Window menu allow the user to perform editor content related actions: (1) switching to another editor content window identified by its tab title, (2) controlling the zoom level of the current editor content, and (3) closing one or more editor content windows.

Simulation Architect supports multiple editor contents by using a tab pane. The menu items are disabled whenever there is no editor content. Figure 2.14 shows the drop-down menu and its 10 menu items divided into three categories—select, zoom, and close.

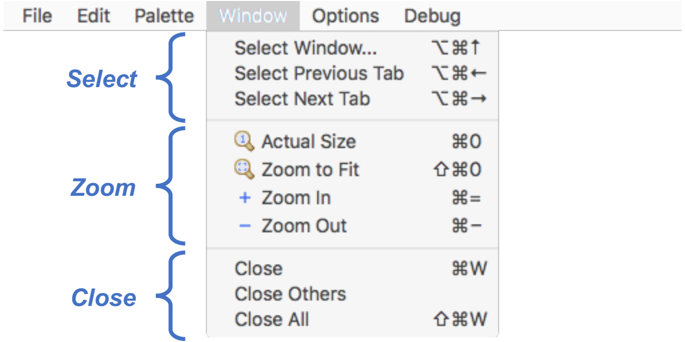


Figure 2.14. Window menu items

2.4.1. Select Window…

The Select Window command shows a dialog listing all the tab titles. The dialog lets the user to search for tab titles and switch to an existing editor content window (see Figure 2.15). To switch to another window, in the dialog either select the tab title and then click on the OK button, or simply double click on the tab title. Clicking on the Cancel button closes the dialog without switching to another editor content.

This command is useful for viewing all the tab titles and switching to another editor content by its tab title. The shortcut key of this command is '⌥ ⌘ ↑' (Mac) or 'Ctrl+Alt+Up Arrow' (Windows).

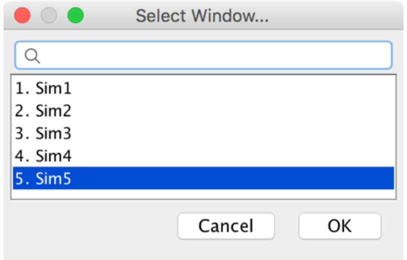


Figure 2.15. Select window dialog

2.4.2. Select Previous Tab

The Select Previous Tab command switches to the previous editor content window. Note that if the editor content associated with the first tab is currently showing, running the command switches to the last editor content window. The menu item is enabled whenever the tab pane has more than one tab.

This command is useful for switching to the previous or the last tab. The shortcut key of this command is '⌥ ⌘ ←' (Mac) or 'Ctrl+Alt+Left Arrow' (Windows).

2.4.3. Select Next Tab

The Select Next Tab command switches to the next editor content window. Note that if the editor content associated with the last tab is currently showing, running the command switches to the first editor content window. The menu item is enabled whenever the tab pane has more than one tab.

This command is useful for switching to the next or the first tab. The shortcut key of this command is '⌥ ⌘ →' (Mac) or 'Ctrl+Alt+Right Arrow' (Windows).

2.4.4. Actual Size

The 🔍 Actual Size command sets the zoom level of the current editor content window to 100% and centers the layout horizontally and vertically. The default zoom level of a new editor content window is 100%.

This command is useful for resetting the zoom level after applying panning or zooming commands in the current editor content. The shortcut key of this command is '⌘ 0' (Mac) or 'Ctrl+0' (Windows).

2.4.5. Zoom to Fit

The 🔍 Zoom to Fit command scales the zoom level of the current editor content window to cover all the blocks and links and centers the layout horizontally and vertically. Depending on the placement of the blocks and links, this command may zoom in or zoom out the editor content window.

This command is useful for having an overview of the layout within the current editor content window. The shortcut key of this command is '⇧ ⌘ 0' (Mac) or 'Ctrl+Shift+0' (Windows).

2.4.6. Zoom In

The + Zoom In command increases the zoom level of the current editor window (blocks look larger). The menu item is disabled whenever the zoom level is set to 500% or higher.

This command is useful for increasing the sizes of the blocks. The shortcut key of this command is ‘⌘ =’ (Mac) or ‘Ctrl+Equal’ (Windows).

2.4.7. Zoom Out

The - Zoom Out command decreases the zoom level of the current editor window (blocks look smaller). The menu item is disabled whenever the zoom level is set to 10% or lower.

This command is useful for decreasing the sizes of the blocks. The shortcut key of this command is ‘⌘ -’ (Mac) or ‘Ctrl+Minus’ (Windows).

2.4.8. Close

The Close command closes the current editor content window. If the current editor content has been modified, a save dialog will be shown to let the user save the layout. Canceling the save dialog will cancel the close command. The menu item is disabled if there is no editor content window, i.e., all the editor content windows are closed.

This command is useful for removing the current editor content from the tab pane. The shortcut key of this command is ‘⌘ W’ (Mac) or ‘Ctrl+W’ (Windows).

2.4.9. Close Others

The Close Others command closes other editor content windows except the current one. If an editor content to be closed has been modified, a save dialog will be shown to let the user save the layout. Canceling the save dialog will cancel the Close Others command. The menu item is enabled only if there are two or more editor content windows.

This command is useful for keeping only the current editor content in the tab pane.

2.4.10. Close All

The Close All command closes all the editor content windows. When an editor content to be closed has been modified, a save dialog will be shown to let the user save the layout. Canceling the save dialog will cancel the Close All command. The menu item is disabled if there is no editor content window, i.e., all the editor content windows are closed.

This command is useful for removing all the editor contents from the tab pane. The shortcut key of this command is ‘⇧ ⌘ W’ (Mac) or ‘Ctrl+Shift+W’ (Windows).

2.5. Options Menu

Commands in the Options menu allow the user to control visual features of the items in the current editor content. Each menu item is a toggle command to enable or disable a visual feature. Figure 2.16 shows the drop-down menu and its 10 menu items divided into four categories—point, block, tool tip, and feature.

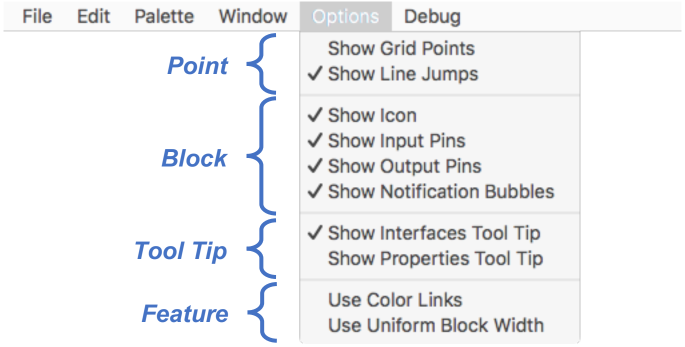
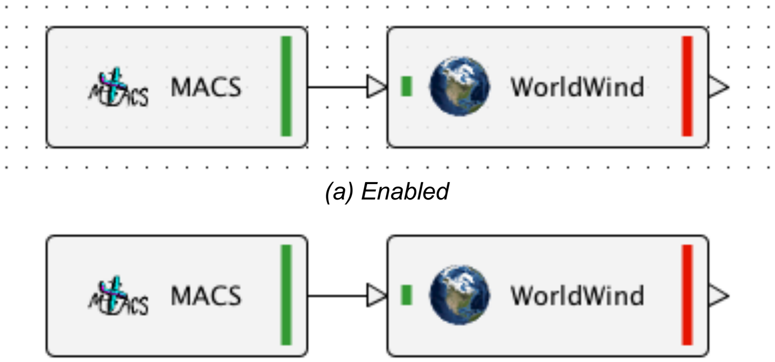


Figure 2.16. Options menu items

2.5.1. Show Grid Points

The Show Grid Points option controls whether to display grid points in the current editor content. Grid points are uniformly spaced. Since blocks and links are rendered based on the grid points, the option is useful for horizontally or vertically aligning the blocks. Figure 2.17 shows the effects when the option is (a) enabled or (b) disabled. The menu item is not selected by default, and the grid points are hidden.



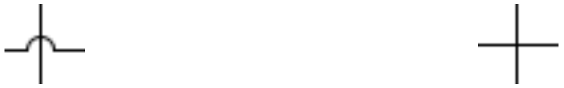
(a) Enabled

(b) Disabled

Figure 2.17. Show Grid Points option

2.5.2. Show Line Jumps

The Show Line Jumps option controls whether to display line jumps for non-intersecting links in the current editor content. Line jumps are useful for identifying whether two crossing links are connected or not. Figure 2.18 shows the effects when the option is (a) enabled or (b) disabled. The menu item is selected by default, and the line jumps are displayed.



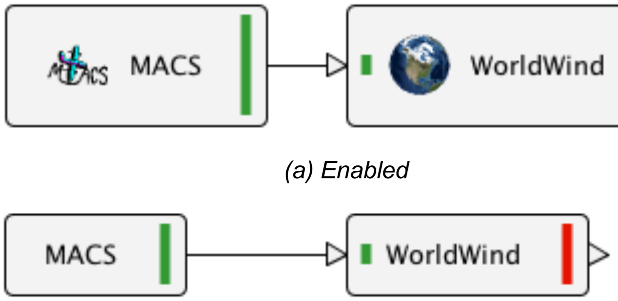
(a) Enabled

(b) Disabled

Figure 2.18. Show Line Jumps option

2.5.3. Show Icon

The Show Icon option controls whether to display block icons in the current editor content. Block icons are located on the left side of block titles, regardless the directions of the blocks. They are useful for visually identifying blocks. Figure 2.19 shows the effects when the option is (a) enabled or (b) disabled. The menu item is selected by default, and the block icons are shown.



(a) Enabled

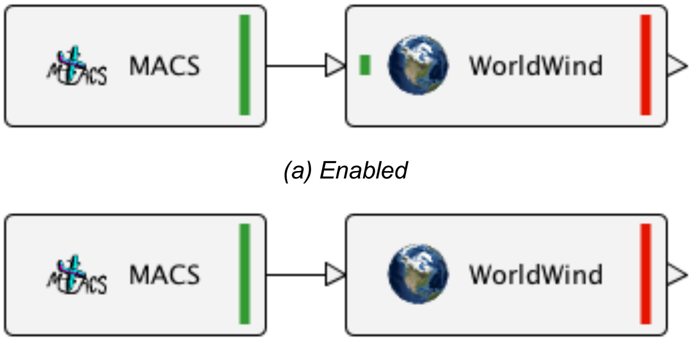
(b) Disabled

Figure 2.19. Show Icon option

2.5.4. Show Input Pins

The Show Input Pins option controls whether to display block input pins in the current editor content. The input pins provide a visual aid for the user to find connected blocks. Input pins are located on the input side of a block: (1) on the left side for left-to-right blocks, and (2) on the right side for right-to-left blocks.

Whenever this option is enabled, an input pin is shown if a block is connected from another source block. The color of the input pin matches the color of the output pin of the connected source block. Figure 2.20 shows the effects when the option is (a) enabled or (b) disabled. The menu item is selected by default, and the input pins are shown.



(a) Enabled

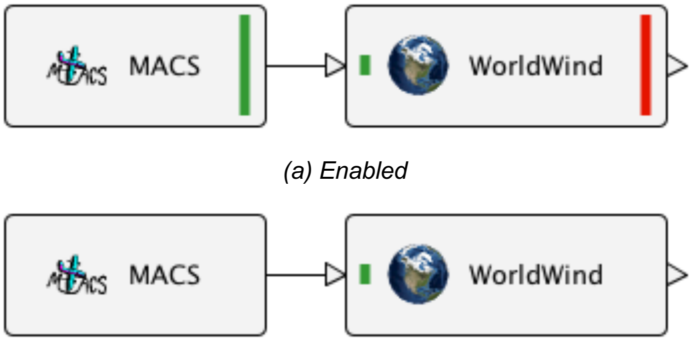
(b) Disabled

Figure 2.20. Show Input Pins option

2.5.5. Show Output Pins

The Show Output Pins option controls whether to display block output pins in the current editor content. The output pins provide a visual aid for the user to find connected blocks. Output pins are located on the output side of a block: (1) on the right side for left-to-right blocks, and (2) on the left side for right-to-left blocks.

Whenever this option is enabled, an output pin is shown if a block is connected to another target block. The color of the output pin matches the color of the input pin of the connected target block. Figure 2.21 shows the effects when the option is (a) enabled or (b) disabled. The menu item is selected by default, and the output pins are shown.



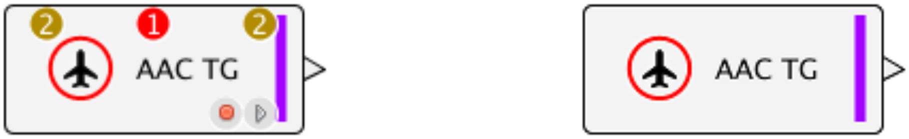
(a) Enabled

(b) Disabled

Figure 2.21. Show Output Pins option

2.5.6. Show Notification Bubbles

The Show Notification Bubbles option controls whether to display notification bubbles for each block in the current editor content. The notification bubbles provide a visual aid for the user to identify blocks that may have misconfigured settings or have enabled component features. Figure 2.22 shows the effects when the option is (a) enabled or (b) disabled. The menu item is selected by default, and the notification bubbles are shown. For details, please refer to Section 5.2.1.



(a) Enabled

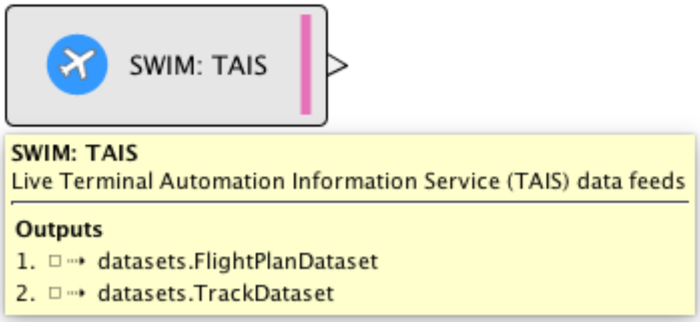
(b) Disabled

Figure 2.22. Show Notification Bubbles option

2.5.7. Show Interfaces Tool Tip

The Show Interfaces Tool Tip option controls whether to display a tool tip window listing a block's supported input and output interfaces whenever the user hovers over the block icon region. Note that even though the Show Icon option is disabled, the region is still active.

The tool tip window is useful for indicating input and output interfaces whether (1) they are required or optional, and (2) they are connected or missing. The menu item is selected by default, and the interfaces tool tip is used.



SWIM: TAIS
Live Terminal Automation Information Service (TAIS) data feeds

Outputs
1. □⇢ datasets.FlightPlanDataset
2. □⇢ datasets.TrackDataset

Figure 2.23. Block and its interfaces tool tip window

Figure 2.23 shows a block titled *SWIM: TAIS*[§] and a tool tip window listing two optional output interfaces, Flight Dataset and Track Dataset, supported by the block. Table 2.1 lists the symbols to be used in the tool tip window as well as the meanings of the symbols. Black squares (■) represent required interfaces while white squares (□) represent optional interfaces, and solid arrows (→ or ←) represent the connected interfaces while dashed arrows (⇢ or ⇠) represent unconnected interfaces. The directions of the arrows indicate the direction of the blocks.

[§] SWIM: TAIS stands for System Wide Information Management: Terminal Automation Information Service.

Table 2.1. Symbols and meanings in block interfaces tool tip window

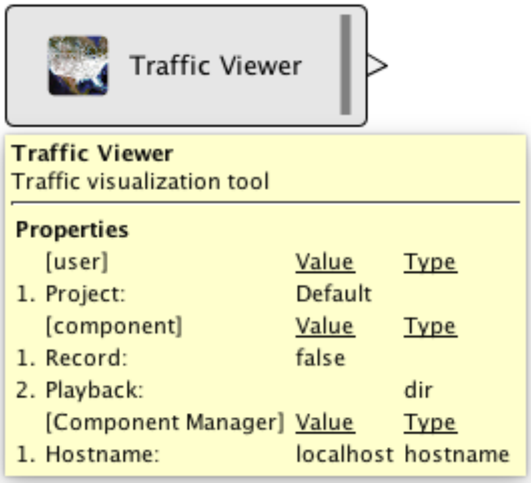
Block Direction		Meaning
Left-to-right	Right-to-left	
→ ■	■ ←	Required input is connected
⇢ ■	■ ⇠	Required input is missing
→ □	□ ←	Optional input is connected
⇢ □	□ ⇠	Optional input is not used
■ →	← ■	Required output is connected
■ ⇢	⇠ ■	Required output is missing
□ →	← □	Optional output is connected
□ ⇢	⇠ □	Optional output is missing

2.5.8. Show Properties Tool Tip

The Show Properties Tool Tip option controls whether to display a tool tip window listing the block's properties when the user hovers over the block icon region. Note that even though the Show Icon option is disabled, the region is still active.

The tool tip window is useful for viewing all the properties of a block without showing the Edit Properties dialog (see Section 5.2.2.5. Edit Properties…) The menu item is not selected by default, and the properties tool tip is not used.

Figure 2.24 shows a block titled *Traffic Viewer* and a tool tip window listing the properties supported by the block. In the tool tip window, the group names of the properties are enclosed by square brackets. Thus, the tool tip window in the figure indicates that the block has three groups of properties: user, component, and Component Manager. For example, the "user" group has the property named "Project" and its value is "Default."



Traffic Viewer
Traffic visualization tool

Properties

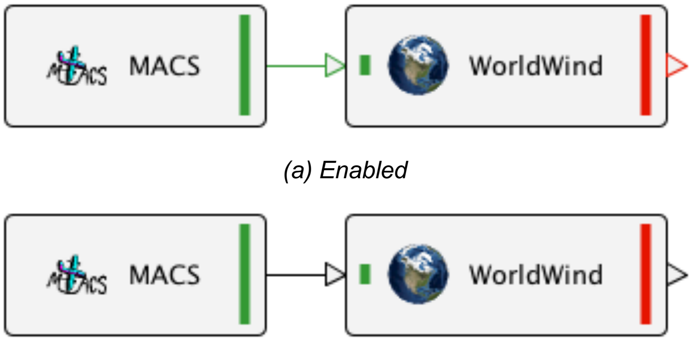
[user]	Value	Type
1. Project:	Default	
[component]	Value	Type
1. Record:	false	
2. Playback:		dir
[Component Manager]	Value	Type
1. Hostname:	localhost	hostname

Figure 2.24. Block and its properties tool tip window

2.5.9. Use Color Links

The Use Color Links option controls whether to display color links even though the user has no interaction with the blocks or links in the current editor content. The color links provide a visual aid for the user to trace connections among blocks. Figure 2.25 shows the effects when the option is (a) enabled or (b) disabled. The menu item is not selected by default, and the black links are used.

Note that when the user interacts with the items, the color links will always be used regardless the option value: (1) hovering over a link changes its color, and (2) hovering over a block changes the colors of its associated links.



(a) Enabled

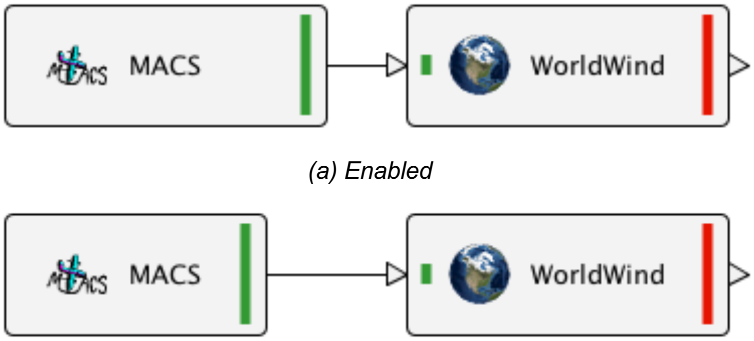
(b) Disabled

Figure 2.25. Use Color Links option

2.5.10. Use Uniform Block Width

The Use Uniform Block Width option controls whether to use the same width for all the blocks in the current editor content. The width is determined based on the widest block in the current layout.

The option is useful for showing all the blocks with the same width. Figure 2.26 shows the effects when the option is (a) enabled or (b) disabled. The menu item is not selected by default, and the uniform block width is not used.



(a) Enabled

(b) Disabled

Figure 2.26. Use Uniform Block Width option

2.6. Debug Menu

Commands in the Debug menu allow the user to control settings for troubleshooting the current editor content. The settings are primarily for the ATM TestBed developers. Figure 2.27 shows the drop-down menu and its six menu items divided into four categories—mode, blocks, grid, and status.

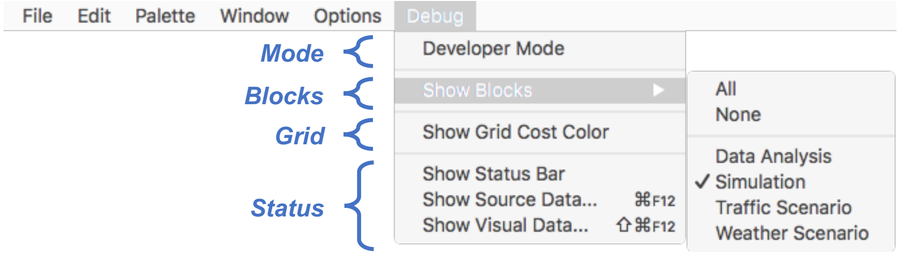


Figure 2.27. Debug menu items

2.6.1. Developer Mode

The Developer Mode option enables features to ease software development. If the developer mode is enabled, the layout file of the current editor content will be pretty printed when executing the save command. Table 2.2 lists the effects when the option is (a) enabled or (b) disabled.

This option is useful for viewing the saved layout file in order to troubleshoot the current editor content. The menu item is not selected by default, and the developer mode is not used.

Table 2.2. Layout file content and developer mode

```
{
  "architect": {
    "at": {
      "x": 0.0,
      "y": 0.0
    },
    "options": {
      "developerMode": true,
      ...
```

(a) Enabled

```
{"architect":{"at":{"x":-1.0,
"y":-1.0},"options":{"develop
erMode":false,"showBlocksForD
ataAnalysis":true,"showBlocks
ForSimulation":true,"showBloc
ksForTrafficGeneration":true,
"showBlocksForWeatherGenerati
on":true,"showGridCostColor":
false,"showGridPoints":fal...
```

(b) Disabled

2.6.2. Show Blocks

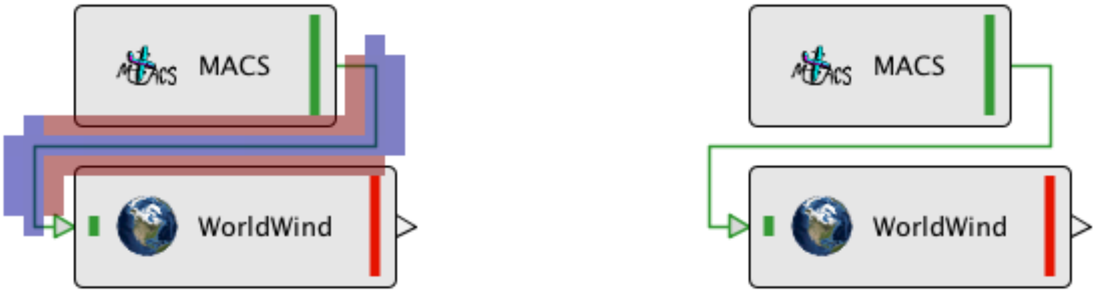
The Show Blocks menu provides sub-menu items for the user to control the types of the blocks to be shown in the palette pane. The supported types are: Data Analysis, Simulation, Traffic Scenario, and Weather Scenario. This option is useful for developing the component library and allowing the user to selectively show or hide blocks in the palette pane. The default selection of the sub-menu item depends on how the user launches Simulation Architect:

1. If the tool is launched from the Web Portal, only one item is selected. For example, the Simulation item is selected when opening a simulation layout file.
2. If the tool is launched from the terminal, all the items are selected.

2.6.3. Show Grid Cost Color

The Show Grid Cost Color option controls whether to display a color grid map for the cost function in the pathfinder algorithm whenever the user interacts with the links in the current editor content (see Section 5.3. Links).

The option is useful for developing and troubleshooting the path evaluator and the heuristic cost function. Figure 2.28 shows the effects when the option is (a) enabled or (b) disabled. The menu item is not selected by default, and the grid color map is hidden.



(a) Enabled

(b) Disabled

Figure 2.28. Show Grid Cost Color option

2.6.4. Show Status Bar

The Show Status Bar option controls whether to show the status bar on the bottom of Simulation Architect. The status bar provides the following information of the current editor content window, as shown in Figure 2.29:

(a) Zoom level in percentage
(b) Translated position, in pixels, of the top-left corner of the editor content
(c) Grid position, in pixels, mapped from the cursor position
(d) Cursor position, in pixels
(e) Memory monitor indicating current memory usage of Simulation Architect

This option is useful for developing and troubleshooting the panning and zooming commands as well as the memory usage of the application.

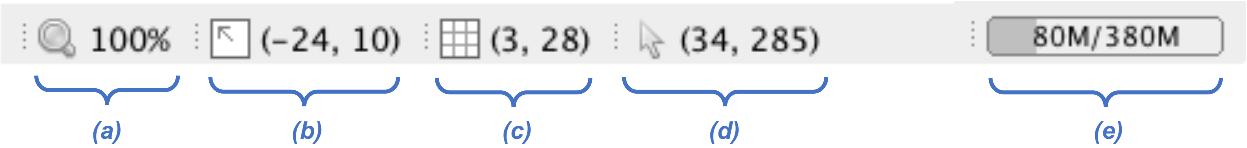
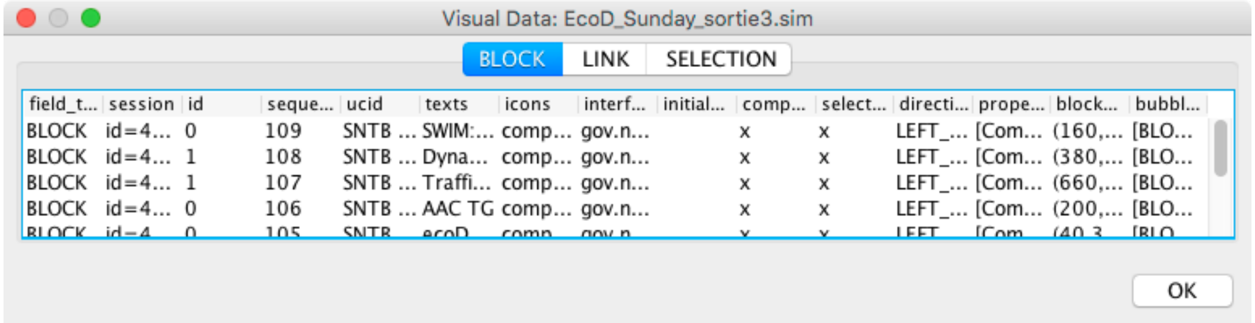


Figure 2.29. Status bar with five groups of buttons:
(a) zoom level, (b) translated position, (c) grid position, (d) cursor position, and (e) memory monitor

2.6.5. Show Source Data…

The Show Source Data command shows a new table dialog with the ATM TestBed specific fields of visual items in the current editor content. The dialog (see Figure 2.30) is useful for developing and troubleshooting the current editor content with the Prefuse library. The shortcut key of this command is '⌘ F12' (Mac) or 'Ctrl+F12' (Windows).

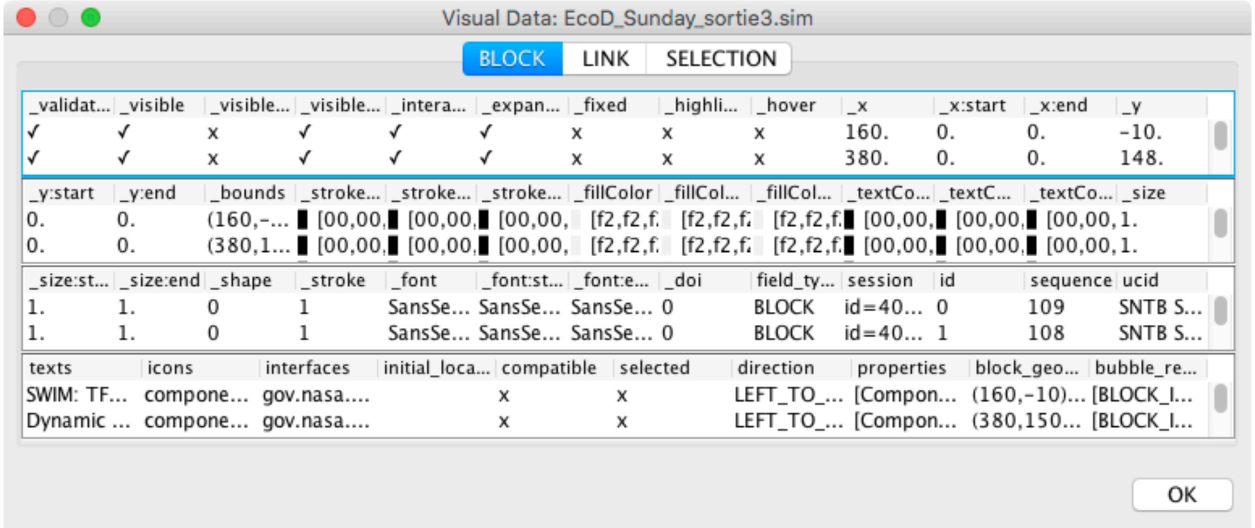


field_t...	session	id	seque...	ucid	texts	icons	interf...	initial...	comp...	select...	directi...	prope...	block...	bubbl...
BLOCK	id=4...	0	109	SNTB ...	SWIM:...	comp...	gov.n...		x	x	LEFT_...	[Com...	(160,...	[BLO...
BLOCK	id=4...	1	108	SNTB ...	Dyna...	comp...	gov.n...		x	x	LEFT_...	[Com...	(380,...	[BLO...
BLOCK	id=4...	1	107	SNTB ...	Traffi...	comp...	gov.n...		x	x	LEFT_...	[Com...	(660,...	[BLO...
BLOCK	id=4...	0	106	SNTB ...	AAC TG	comp...	gov.n...		x	x	LEFT_...	[Com...	(200,...	[BLO...
BLOCK	id=4	0	105	SNTB	ecoD	comp	gov.n		x	x	LEFT	[Com	(40,3	[BLO

Figure 2.30. Visual data dialog with ATM TestBed specific fields

2.6.6. Show Visual Data…

The Show Visual Data command shows a new table dialog with all the fields of visual items in the current editor content window. The dialog (see Figure 2.31) is useful for developing and troubleshooting the current editor content with the Prefuse library. The shortcut key of this command is ‘⇧ ⌘ F12’ (Mac) or ‘Ctrl+Shift+F12’ (Windows).



Visual Data: EcoD_Sunday_sortie3.sim

_validat…	_visible	_visible…	_visible…	_intera…	_expan…	_fixed	_highli…	_hover	_x	_x:start	_x:end	_y
✓	✓	x	✓	✓	✓	x	x	x	160.	0.	0.	-10.
✓	✓	x	✓	✓	✓	x	x	x	380.	0.	0.	148.

_y:start	_y:end	_bounds	_stroke…	_stroke…	_stroke…	_fillColor	_fillCol…	_fillCol…	_textCo…	_textC…	_textCo…	_size
0.	0.	(160,-…	■ [00,00,	■ [00,00,	■ [00,00,	□ [f2,f2,f.	□ [f2,f2,f.	□ [f2,f2,f.	■ [00,00,	■ [00,00,	■ [00,00,	1.
0.	0.	(380,1…	■ [00,00,	■ [00,00,	■ [00,00,	□ [f2,f2,f.	□ [f2,f2,f.	□ [f2,f2,f.	■ [00,00,	■ [00,00,	■ [00,00,	1.

_size:st…	_size:end	_shape	_stroke	_font	_font:st…	_font:e…	_doi	field_ty…	session	id	sequence	ucid
1.	1.	0	1	SansSe…	SansSe…	SansSe…	0	BLOCK	id=40…	0	109	SNTB S…
1.	1.	0	1	SansSe…	SansSe…	SansSe…	0	BLOCK	id=40…	1	108	SNTB S…

texts	icons	interfaces	initial_loca…	compatible	selected	direction	properties	block_geo…	bubble_re…
SWIM: TF…	compone…	gov.nasa.…		x	x	LEFT_TO_…	[Compon…	(160,-10)…	[BLOCK_I…
Dynamic …	compone…	gov.nasa.…		x	x	LEFT_TO_…	[Compon…	(380,150…	[BLOCK_I…

Figure 2.31. Visual data dialog with all fields

3. Tool Bar

The tool bar provides commonly used action buttons. Figure 3.1 shows the action buttons divided into four groups:

1. Content Group contains five buttons related to layout files associated with the existing editor content tabs.
2. Zoom Group contains four buttons related to zoom level control of the current editor content.
3. Edit Group contains two buttons related to edits associated with the current editor content.
4. Debug Group contains two buttons related to software development and troubleshooting.

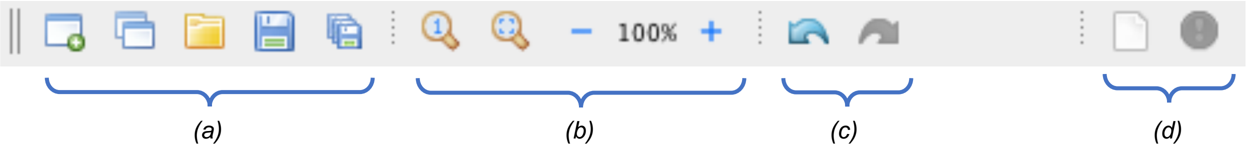


Figure 3.1. Tool bar with four groups of buttons: (a) content, (b) zoom, (c) edit, and (d) debug

The command of each action button is listed in Table 3.1.

Table 3.1. Tool bar buttons in Simulation Architect

Button	Command	Description
🗋	New	See Section 2.1.1. on page 4
🗐	New Copy	See Section 2.1.2. on page 5
📁	Open	See Section 2.1.3. on page 5

💾	Save	See Section 2.1.4. on page 5
💾	Save All	See Section 2.1.6. on page 6
🔍	Actual Size	See Section 2.4.4. on page 14
🔍	Zoom to Fit	See Section 2.4.5. on page 14
−	Zoom Out	See Section 2.4.7. on page 15
+	Zoom In	See Section 2.4.6. on page 14
↶	Undo	See Section 2.2.1. on page 9
↷	Redo	See Section 2.2.2. on page 9
🗋	Show Log…	Shows a dialog listing the console output text that has been logged by Simulation Architect. This command is useful for reporting the console to the ATM TestBed software development team.
❗	Show Errors…	See Section 2.1.10. on page 8

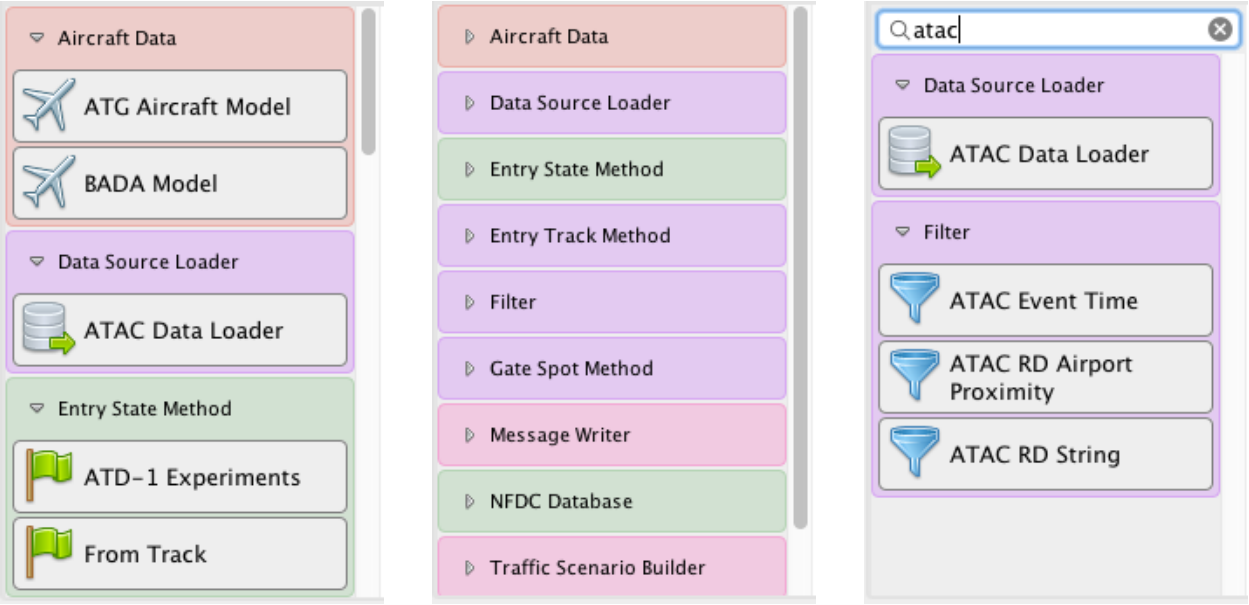
4. Palette Pane

A palette pane, located on the left side of Simulation Architect, contains blocks representing simulation components available to the current editor content. The palette pane provides an easy way for the user to know all the accessible simulation components. Figure 4.1(a) shows a palette pane with the traffic scenario blocks. In the palette pane, each block is visually identified by an icon and a title. Similar blocks are grouped based on the Group By setting, which can be changed from the Palette menu as described in Section 2.3. Each group is visually identified by a group name and a colored background. Each group has two states:

1. Collapsed—all containing blocks are hidden to the user.
2. Expanded—all containing blocks are visible to the user.

Initially, all groups are expanded. Clicking on the group name toggles the state of the group. Figure 4.1(b) shows a palette pane with all the groups collapsed.

Blocks in the palette pane can also be hidden when the user specifies a search text in the find bar (see Figure 4.1(c)). The find bar is hidden by default, and it can be made visible by running the Find Blocks command from the Palette menu.

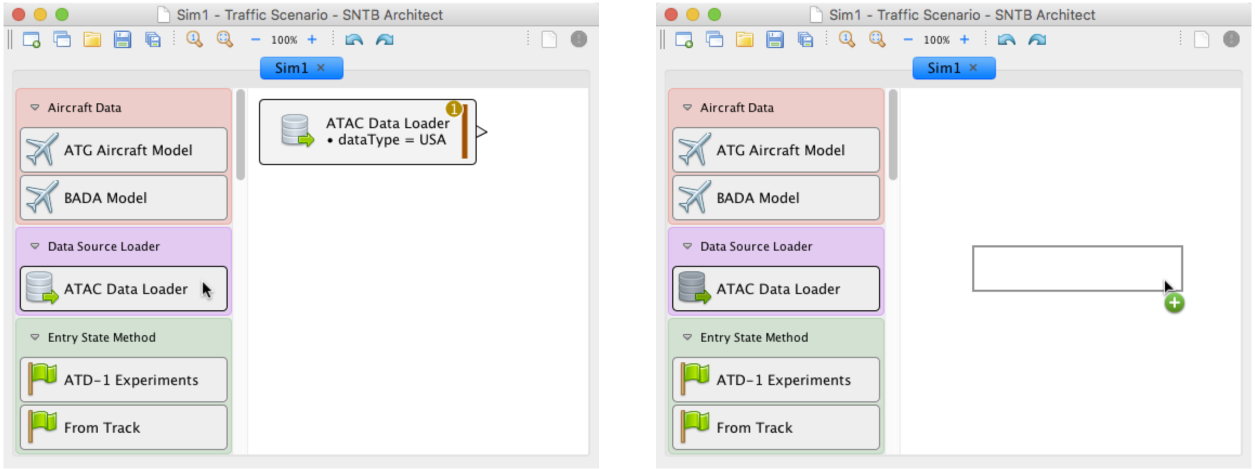


(a) Expanded groups *(b) Collapsed groups* *(c) Find blocks*

Figure 4.1. Three views of a palette pane

Hovering over a block icon will show a tool tip window describing the information of the block including title, description, input interfaces and output interfaces (see Section 2.5.7. Show Interfaces Tool Tip). The user can add a block from the palette pane to the current editor content using one of the following methods:

1. Click on the block and a new block will be added to the top-left corner of the current editor content (see Figure 4.2(a)). When the same block is added multiple times, the subsequently added blocks are shown in a cascading style.
2. Drag the block from the palette pane and drop it to the current editor content (see Figure 4.2(b)). Using the drag-and-drop method allows the user to place the block at any location in the current editor content.



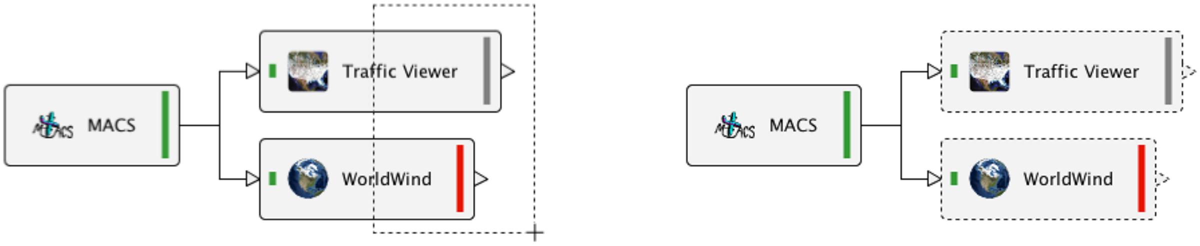
(a) Single click *(b) Drag and drop*

Figure 4.2. Two ways to add a block from the palette pane to the current editor content

5. Editor Content

An editor content provides a working space for creating, designing and configuring a layout. The user can use the mouse to interact with the current editor content in the following ways:

1. Left click on an empty area and then drag in any direction to pan the working space
2. Mouse scroll up or down to zoom out or in the working space
3. Right click on an empty area and then drag in any direction to select blocks (see Figure 5.1)



(a) Selecting blocks

(b) Selected blocks

Figure 5.1. Right click and drag to select blocks

Simulation Architect provides various features for the user to explore and examine interfaces among blocks. For example, two blocks can connect to each other only if they share one or more common message interfaces. In addition, the tool supports Undo and Redo commands so that the user may cancel or rerun an operation any time. This section gives a detailed description of the editor content how an ATM TestBed user interacts with it.

5.1. Tab Pane

Simulation Architect supports multiple editor contents by using a tab pane. Commands such as New, New Copy, and Open create a new editor content. Each editor content can be identified by a title in the tab. The new editor content tab is generally added to the right side. An exception to this rule is running an Undo command after a Close command. In this case, the tab is added to the previously defined position. The user has three ways to switch to another editor content:

1. By clicking on a tab
2. By using Window menu (see Section 2.4. Window Menu)
3. By pressing shortcut keys '⌘ 1' through '⌘ 8' (Mac) or 'Ctrl+1' through 'Ctrl+8' (Windows) for a specific tab, and '⌘ 9' (Mac) or 'Ctrl+9' (Windows) for the last tab

5.1.1. Tab Popup Menu

To close a tab, click on the close button (×) located on the right side of the tab title. Hovering over the close button changes its background color to red (see Figure 5.2).

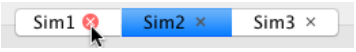


Figure 5.2. Close a tab

To close multiple tabs, right click on a tab. The user can then select a menu item from the popup menu to perform the close action (see Figure 5.3).

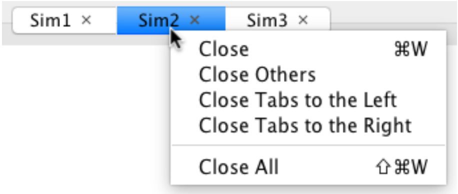


Figure 5.3. Tab popup menu items

Before closing a modified editor content, a save dialog lets the user decide whether to save the editor content. Canceling the save dialog will not close the corresponding modified editor content.

5.1.1.1. Close

The Close command closes the current editor content window. This command is a shortcut to the Close command in the Window menu (see Section 2.4.8. Close).

5.1.1.2. Close Others

The Close Others command closes other editor content windows except the current one. The command is available whenever the tab pane has more than one tab. This command is a shortcut to the Close Others command in the Window menu (see Section 2.4.9. Close Others).

5.1.1.3. Close Tabs to the Left

The Close Tabs to the Left command closes all editor content windows located on the left side of the current tab. This menu item is available whenever the first tab is not right clicked.

5.1.1.4. Close Tabs to the Right

The Close Tabs to the Right command closes all editor content windows located on the right side of the current tab. This menu item is available whenever the last tab is not right clicked.

5.1.1.5. Close All

The Close All command closes all the editor content windows. This command is available whenever the tab pane has more than one tab. The command is a shortcut to the Close All command in the Window menu (see Section 2.4.10. Close All).

5.2. Blocks

In Simulation Architect, a block represents a simulation component, either an application or a data stream. A simulation component developer can create pluggable adapters using the ATM TestBed SDK and then upload the adapters to the TestBed Component Library. In an editor content, each block has four main visual sections:

1. Icon—an image representation allowing the user to easily identify the block
2. Title—a text representation allowing the user to easily find the block
3. Output arrow—a left-pointing triangle (◁) for right-to-left blocks and a right-pointing triangle (▷) for left-to-right blocks indicating outputs of the block
4. Notification bubbles—circled symbols indicating potential issues and enabled features of the block

For example, Figure 5.4 shows a simulation block on an editor content. The icon assembled by a red circle and an aircraft indicates a simulation component for detecting conflicts among flight tracks. The title text, *AAC TG*[**], indicates the name of the simulation component. The output arrow, right-pointing and located on the right edge, indicates the block has a left-to-right direction and the block's output has not yet connected to another block. Four groups of notification bubbles are shown inside the block and they are described in the following subsection.

The user can move a block on the editor content by left clicking on the block and then dragging it to a desired location. Whenever a block is being hovered or dragged, its background

[**] AAC TG stands for Advanced Airspace Concept with built-in Trajectory Generator.

color is changed from light gray to dark gray. In addition, if the hovered block is selected (i.e., the outline is dashed), moving it will also move the other selected blocks.

The user can link two blocks on the same editor content by left clicking on the source block, dragging and dropping it onto the target block.

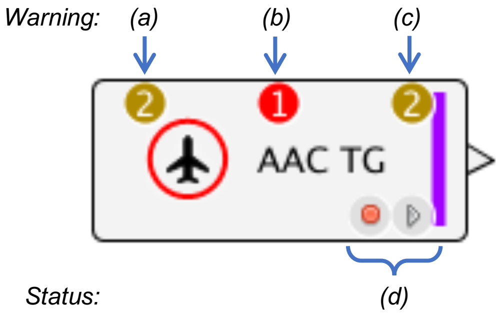


Figure 5.4. Notification bubbles: (a) inputs, (b) properties, (c) outputs, and (d) component features

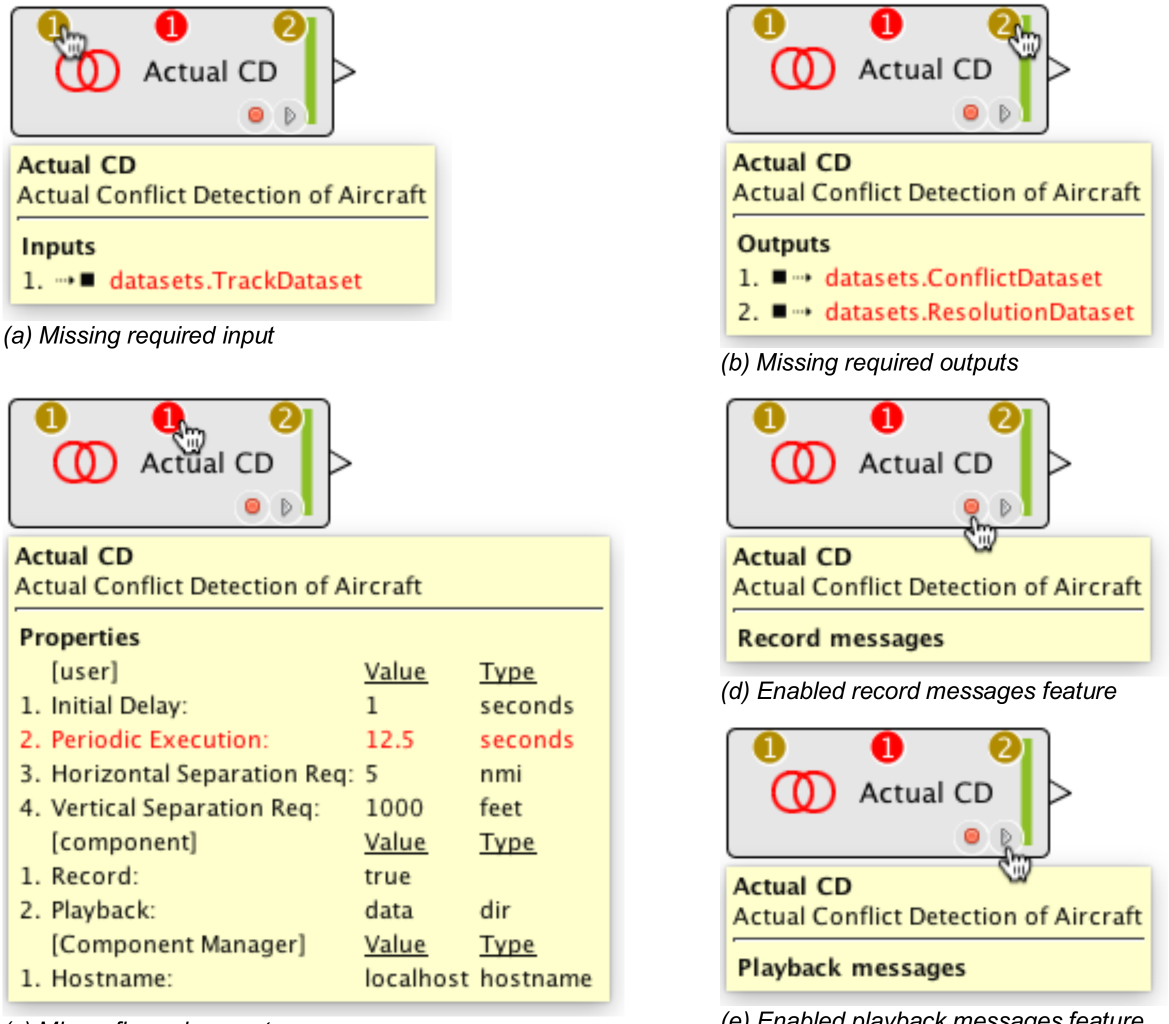
5.2.1. Notification Bubbles

Four groups of notification bubbles may be shown on a block: three groups are on the top of the block indicating warnings and one group is on the bottom of the block indicating enabled component features. The locations of the notification bubbles matter:

1. Missing inputs, located at the top corner of the input edge, indicates the number of required inputs that are missing.
2. Misconfigured properties, located at the top-center, indicates the number of properties that are misconfigured.
3. Missing outputs, located at the top corner of the output edge, indicates the number of required outputs that are missing.
4. Component features, located at the bottom corner of the output edge, indicates enabled features of the simulation component such as record (●) and playback messages (▷).

When hovering over a notification bubble, a tool tip window is displayed to inform the user the reason why the notification bubble is shown. For warnings, the cause is highlighted in red in the tool tip window. Figure 5.5 shows five examples of the tool tip windows:

(a) One required input is missing—The block expects a source block to provide track dataset messages.
(b) Two required outputs are missing—The block expects one or more target blocks to handle the conflict dataset and resolution dataset messages.
(c) One property is misconfigured—The value of the property Periodic Execution is invalid;
(d) The “record messages” feature is enabled.
(e) The “playback messages” feature is enabled.



Actual CD
Actual Conflict Detection of Aircraft

Inputs
1. ⋯▸■ datasets.TrackDataset

(a) Missing required input

Actual CD
Actual Conflict Detection of Aircraft

Outputs
1. ■⋯▸ datasets.ConflictDataset
2. ■⋯▸ datasets.ResolutionDataset

(b) Missing required outputs

Actual CD
Actual Conflict Detection of Aircraft

Properties

[user]	Value	Type
1. Initial Delay:	1	seconds
2. Periodic Execution:	12.5	seconds
3. Horizontal Separation Req:	5	nmi
4. Vertical Separation Req:	1000	feet
[component]	Value	Type
1. Record:	true	
2. Playback:	data	dir
[Component Manager]	Value	Type
1. Hostname:	localhost	hostname

(c) Misconfigured property

Actual CD
Actual Conflict Detection of Aircraft

Record messages

(d) Enabled record messages feature

Actual CD
Actual Conflict Detection of Aircraft

Playback messages

(e) Enabled playback messages feature

Figure 5.5. Tool tip windows

5.2.2. Block Popup Menu

Right clicking on a block shows a context sensitive popup menu. Figure 5.6 shows the popup menu and its seven menu items divided into four categories—add, connect, edit, and delete. Instead of using the popup menu, shortcut keys can also be used to execute the popup menu commands whenever a block is hovered.

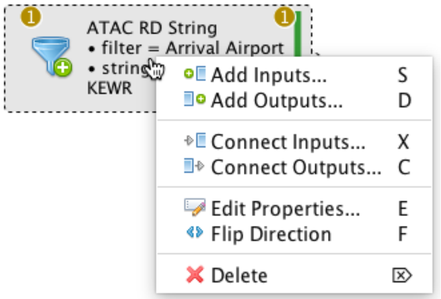


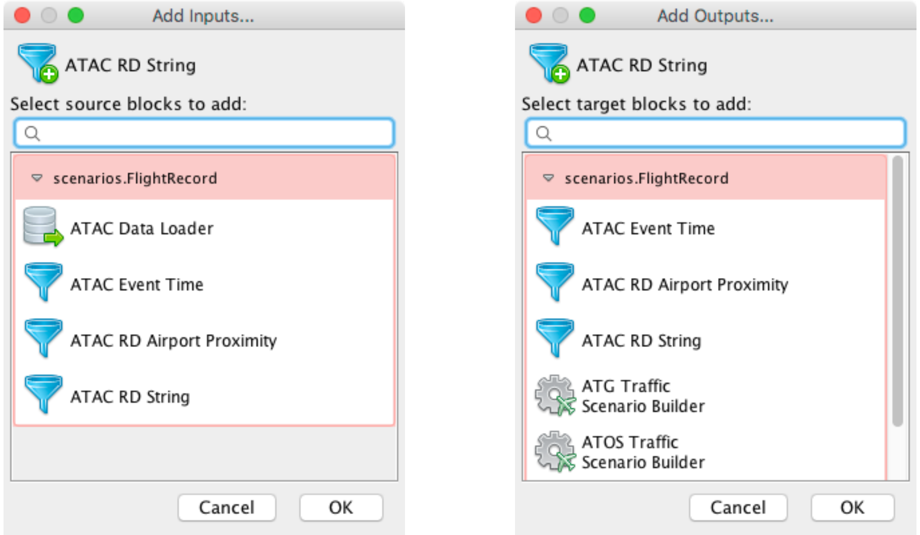
Figure 5.6. Block popup menu items

Executing Add or Connect commands may show a dialog listing compatible blocks for the user to select. Two blocks are compatible if they share at least one common interface between the outputs of the source block and the inputs of the target block. In the dialog, the compatible blocks are grouped by the interfaces. The user can select a single block by clicking on the block. To select multiple blocks, press the command key (Mac) or the control key (Windows) and then left click a block to select.

5.2.2.1. Add Inputs…

The Add Inputs command treats the hovered block as a target block, and allows the user to add compatible source blocks and then connect them *to* the target block. The command obtains the compatible source blocks from the Palette Pane. This command is useful for finding and adding blocks that can be connected to the target block.

The command is disabled when there is no compatible source block. When there is only one compatible source block, such block will be added to the current editor content and a link is added from the hovered block to the block added. When there are multiple compatible source blocks, the command shows a dialog listing them and asks the user to select one or multiple blocks (see Figure 5.7(a)). The shortcut key of this command is ‘S’.



(a) Add Inputs

(b) Add Outputs

Figure 5.7. Add Inputs and Outputs dialogs

5.2.2.2. Add Outputs…

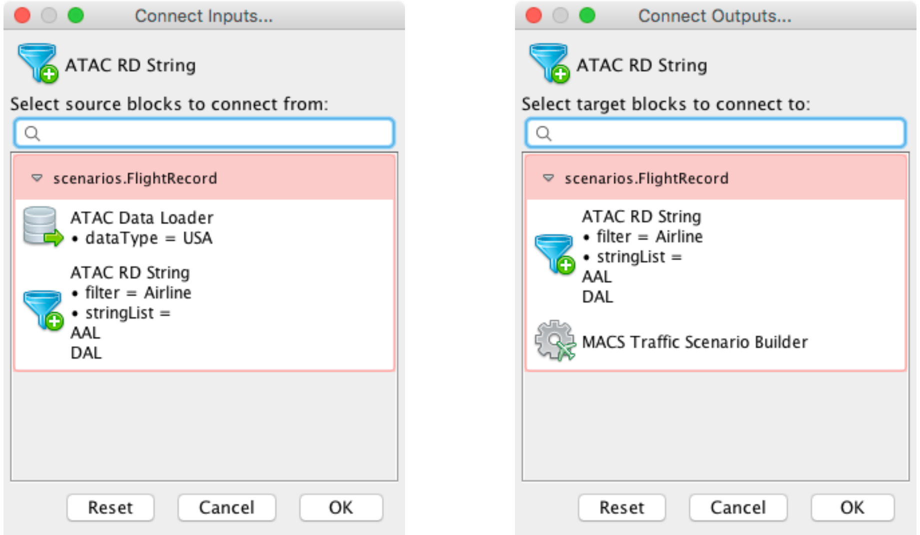
The Add Outputs command treats the hovered block as a source block, and allows the user to add compatible target blocks and then connect them *from* the source block. The command obtains the compatible target blocks from the Palette Pane. This command is useful for finding and adding blocks that can be connected from the source block.

The command is disabled when there is no compatible target block. When there is only one compatible target block, such block will be added to the current editor content and a link is added from the block added to the hovered block. When there are multiple compatible target blocks, the command shows a dialog listing them and asks the user to select one or multiple blocks (see Figure 5.7(b)). The shortcut key of this command is ‘D’.

5.2.2.3. Connect Inputs…

The Connect Inputs command treats the hovered block as a target block, and allows the user to connect compatible source blocks *to* the target block. The command obtains the compatible source blocks from the current editor content. This command is useful for finding blocks that can be connected to the target block.

The command is disabled when there is no compatible source block. When there is only one compatible source block, a link is added from such block to the hovered block. When there are multiple compatible source blocks, the command shows a dialog listing them and asks the user to select one or multiple blocks (see Figure 5.8(a)). The shortcut key of this command is ‘X’.



(a) Add Inputs *(b) Add Outputs*

Figure 5.8. Connect Inputs and Outputs dialogs

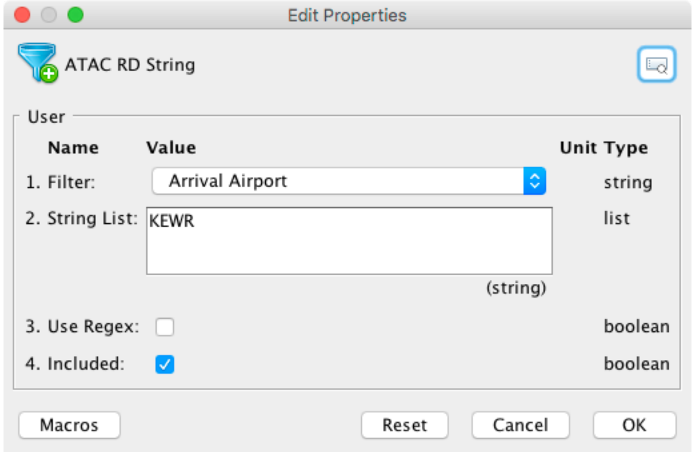
5.2.2.4. Connect Outputs…

The ⇨ Connect Outputs command treats the hovered block as a source block, and allows the user to connect compatible target blocks *from* the source block. The command obtains the compatible target blocks from the current editor content. This command is useful for finding blocks that can be connected from the source block.

The command is disabled when there is no compatible target block. When there is only one compatible target block, a link is added from the hovered block to such block. When there are multiple target blocks, the command shows a dialog listing them and asks the user to select one or multiple blocks (see Figure 5.8(b)). The shortcut key of this command is ‘C’.

5.2.2.5. Edit Properties…

The Edit Properties (✎) command shows a dialog listing all properties associated with the hovered block in the current editor content. The user can view the properties and edit their values (see Figure 5.9). The shortcut key of this command is ‘E’ or double click on a block.



Edit Properties

ATAC RD String

User

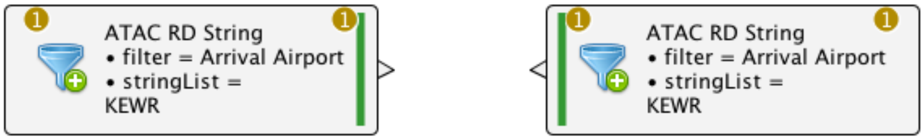
Name	Value	Unit Type
1. Filter:	Arrival Airport	string
2. String List:	KEWR (string)	list
3. Use Regex:	□	boolean
4. Included:	☑	boolean

Figure 5.9. Edit Properties dialog

5.2.2.6. Flip Direction

The ↔ Flip Direction command allows the user to flip the direction of the hovered block in the current editor content. Directions of a block can be either left-to-right or right-to-left, as shown in Figure 5.10. Whenever a block is added from the palette pane to the editor content, its default direction is left-to-right.

This command is useful for creating a pretty layout involving many blocks. The shortcut key of this command is ‘F’.



(a) Left to right

(b) Right to left

Figure 5.10. Directions of block

5.2.2.7. Delete

The ✕ Delete command allows the user to delete the block and its associated links in the current editor content. This command is useful for removing blocks. The shortcut key of this command is ‘*delete*’.

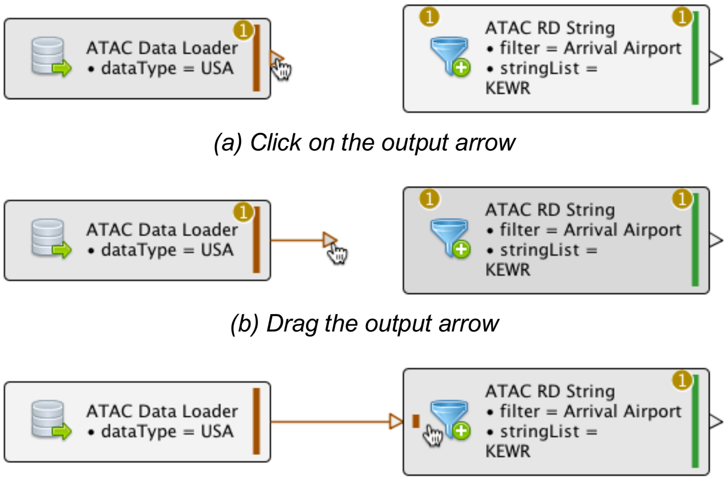
5.2.3. Block Connection

Two blocks can connect if and only if they are compatible, i.e., they share at least one common interface between the outputs of the source block and the inputs of the target block. A simple way to connect two compatible blocks is using the drag-and-drop feature as shown in Figure 5.11:

(a) The user clicks on the output (triangle) arrow of the source block and the output arrow is highlighted.

(b) The user drags the output arrow to a target block. In this step, the background color of the compatible target blocks changes from light gray to dark gray.

(c) The user then releases the mouse button. A link is created from the source block to the target block. Note that the output notification bubble of the source block and the input notification bubble of the target block disappear.



(a) Click on the output arrow

(b) Drag the output arrow

(c) Drop the output arrow

Figure 5.11. Connecting two blocks

To delete a link, the user can either right click on the link and select the delete command or hover a link and press the delete key. Deleting a block will also delete its associated links.

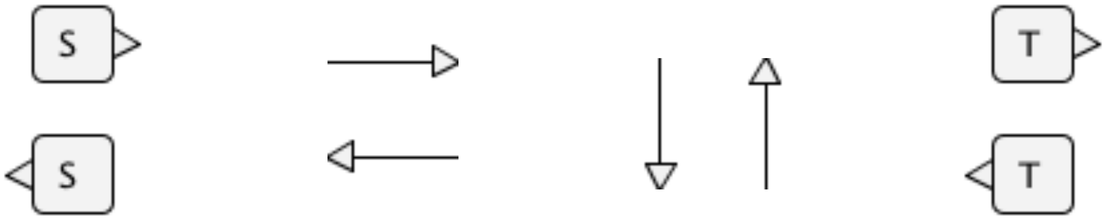
5.3. Links

In Simulation Architect, a link represents an ATM TestBed data exchange message channel and indicates a message flow between two simulation components. In an editor content, links are directed and consist of a series of horizontal or vertical straight lines connecting from the output of a source block to the input of a target block. The tool applies an A* search algorithm[11] with a path evaluator and a heuristic cost function to guide links connecting from source blocks to target blocks. By dividing an editor content into square grids, the algorithm finds an optimal path from a source block's output pin to a target block's input pin. The path evaluator avoids link overlapping, and the heuristic cost function avoids links crossing existing blocks or making undesired turns.

A link connecting from a source block to a target block is determined by four direction groups[††] as shown in Figure 5.12:

(a) Direction of the source block—either Left-to-Right (LR) or Right-to-Left (RL)
(b) Horizontal direction from the source block to the target block—either West-to-East (WE) or East-to-West (EW)
(c) Vertical direction from the source block to the target block—either North-to-South (NS) or South-to-North (SN)
(d) Direction of the target block—either Left-to-Right (LR) or Right-to-Left (RL)

[††] In order to differentiate the directions of blocks and links, block directions are represented using *Left* and *Right*, and link directions are represented using the four cardinal directions, *North*, *East*, *South*, and *West*.

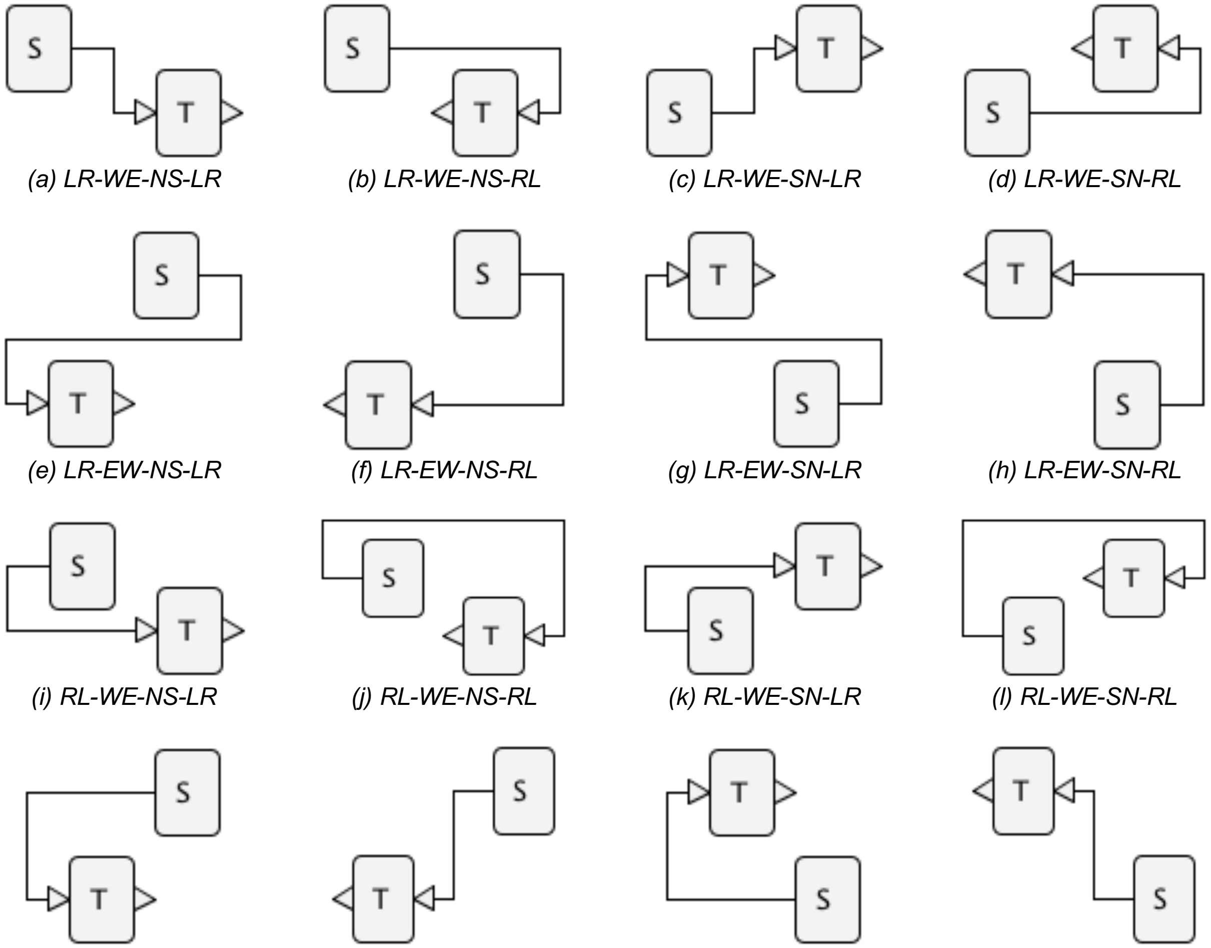


(a) Source block *(b) Horizontal* *(c) Vertical* *(d) Target block*

Figure 5.12. Four direction groups

Since each group contains two possible directions, a total of 2×2×2×2=16 cases are considered in the path evaluator (see Figure 5.13). For example, Figure 5.13(b) shows the link generated based on the following four conditions:

1. The source block has a Left-to-Right direction (LR).
2. The link is horizontally connecting from the West to the East (WE).
3. The link is vertically connecting from the North to the South (NS).
4. The target block has a Right-to-Left direction (RL).



(a) LR-WE-NS-LR *(b) LR-WE-NS-RL* *(c) LR-WE-SN-LR* *(d) LR-WE-SN-RL*

(e) LR-EW-NS-LR *(f) LR-EW-NS-RL* *(g) LR-EW-SN-LR* *(h) LR-EW-SN-RL*

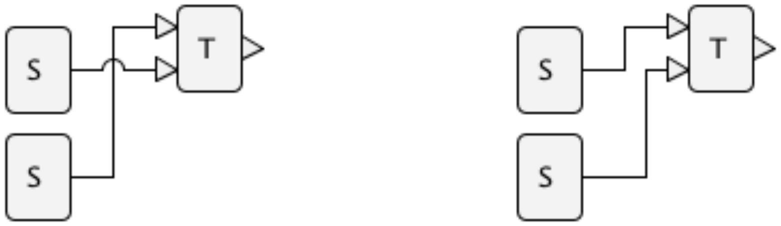
(i) RL-WE-NS-LR *(j) RL-WE-NS-RL* *(k) RL-WE-SN-LR* *(l) RL-WE-SN-RL*

(m) RL-EW-NS-LR *(n) RL-EW-NS-RL* *(o) RL-EW-SN-LR* *(p) RL-EW-SN-RL*

Legend: [Source]-[Horizontal]-[Vertical]-[Target]
Block: S=Source, T=Target, LR=Left-to-Right, RL=Right-to-Left;
Horizontal: EW=East-to-West, WE=West-to-East;
Vertical: NS=North-to-South, SN=South-to-North

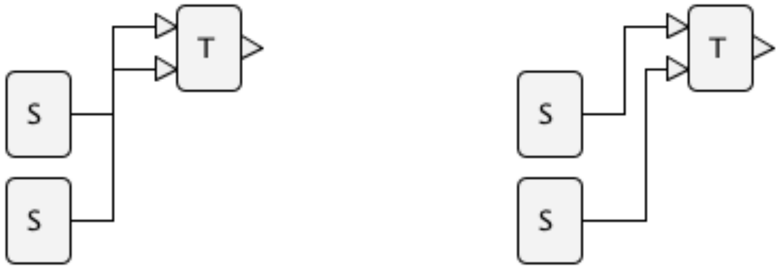
Figure 5.13. Layouts of a link connecting from a source block to a target block

When multiple blocks are defined in the editor content, the path evaluator used in the A* search algorithm avoids link overlapping by determining the order of the input pins (see Figure 5.14) and by determining the separation distances between adjacent horizontal or vertical lines (see Figure 5.15).



(a) Undesired (b) Desired

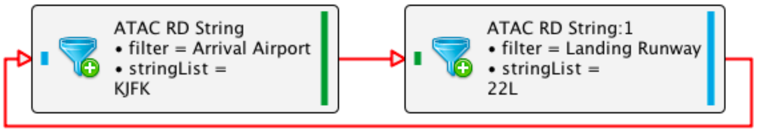
Figure 5.14. Link overlapping is avoided by reordering input pins



(a) Undesired (b) Desired

Figure 5.15. Link overlapping is avoided by separating vertical lines

For traffic scenario generation layouts, Simulation Architect applies Tarjan's algorithm[12] to detect cycles with the Flight Record interface. For example, the input and output interfaces of the ATAC RD String block are Flight Record; Figure 5.16 shows a simple case that two blocks form a cycle and their links are highlighted in red.



ATAC RD String
- filter = Arrival Airport
- stringList = KJFK

ATAC RD String:1
- filter = Landing Runway
- stringList = 22L

Figure 5.16. Cycle detected when using flight records

6. Shortcut Keys

Shortcut keys provide a faster way to perform action commands without accessing drop-down menus or pressing tool bar buttons. Table 6.1 lists the shortcut keys defined in Simulation Architect.

Table 6.1. Shortcut keys in Simulation Architect

	Shortcut Keys	
To Do This	**Mac**	**Windows or Linux**
Open a new layout file	⌘ O	Ctrl+O
Add a new editor content	⌘ N	Ctrl+N
Save the current editor content	⌘ S	Ctrl+S
Save all editor contents	⇧ ⌘ S	Shift+Ctrl+S

Cancel the current dialog	⇧ Esc	Shift+Esc
Add compatible source blocks	S	S
Add compatible target blocks	D	D
Connect from compatible source blocks	X	X
Connect to compatible target blocks	C	C
Flip direction of block	F	F
Copy the selected blocks to clipboard	⌘ C	Ctrl+C
Cut the selected blocks to clipboard	⌘ X	Ctrl+X
Paste the copied/cut blocks from clipboard	⌘ V	Ctrl+V
Delete the selected blocks	⌦	Delete
Duplicate the selected blocks	⌘ D	Ctrl+D
Select all the blocks	⌘ A	Ctrl+A
Clear the selection	⇧ ⌘ A	Shift+Ctrl+A
Find blocks in the palette pane	⌘ F	Ctrl+F
Find properties of blocks in the current layout	⇧ ⌘ F	Shift+Ctrl+F
Show an edit properties dialog	E	E
Select a window to switch to	⌘ ⌥ ↑	Ctrl+Alt+Up Arrow
Switch to a specific (the first to eighth) window	⌘ 1 through ⌘ 8	Ctrl+1 through Ctrl+8
Switch to the last window	⌘ 9	Ctrl+9
Switch to the next window	⌘ ⌥ →	Ctrl+Alt+Right Arrow
Switch to the previous window	⌘ ⌥ ←	Ctrl+Alt+Left Arrow
Close the current window	⌘ W	Ctrl+W
Close all the windows	⇧ ⌘ W	Shift+Ctrl+W
Reset zoom to 100%	⌘ 0	Ctrl+0
Zoom to fit the current view space	⇧ ⌘ 0	Shift+Ctrl+0
Zoom in	⌘ =	Ctrl+Equal
Zoom out	⌘ -	Ctrl+Minus
Undo the most recent command	⌘ Z	Ctrl+Z
Redo the previously undone command	⇧ ⌘ Z	Shift+Ctrl+Z
Show a macros dialog	⌘ M	Ctrl+M
Show or hide the palette pane	⌘ P	Ctrl+P
Show or hide the revision pane	⌘ R	Ctrl+R
Show data tables (ATM TestBed specific)	⌘ F12	Ctrl+F12
Show data tables (all visual items)	⇧ ⌘ F12	Shift+Ctrl+F12

7. References

1. Kupfer, M., Mercer, J., Cabrall, C., and Callantine T., “Designing Scenarios for Controller-in-the-Loop Air Traffic Simulations,” AIAA Modeling and Simulation Technologies (MST) Conference, Guidance, Navigation, and Control and Co-located Conferences, (AIAA 2013-5156).
2. Chatterji, G. B. and Zheng, Y., “Automated Scenario Generation for Meeting Human-in-the-Loop Simulation Requirements,” AIAA Scitech 2019 Forum, AIAA SciTech Forum, (AIAA 2019-1710).
3. Glaab, P. and Madden M., “Benefits of a unified LaSRS++ simulation for NAS-wide and high-fidelity modeling,” 2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC).
4. Mehlitz, P., Shafiei, N., Tkachuk, O., and Davies M, “RACE: Building airspace simulations faster and better with actors,” 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC).
5. Ruiz, S. and Piera, M., “TPAS: Test-bed Platform for ATM Studies,” 6th International Conference on Research in Air Transportation, Turkey, May 26-30, 2014.
6. Chan, W. N., Barmore, B. E., Kibler, J. L.; Lee, P. U.; O’Connor, C. J., Palopo, K., Thipphavong, D. P., Zelinski, S. J., “Overview of NASA’s Air Traffic Management - eXploration (ATM-X) Project,” 2018 Aviation Technology, Integration, and Operations Conference, Atlanta, Georgia.
7. Prefuse, website: https://github.com/prefuse/, February 2019.
8. Robinson, J. E., III, Lee, A., and Lai, C. F., “Development of a High-Fidelity Simulation Environment for Shadow-Mode Assessments of Air Traffic Concepts,” RAeS and AIAA Modelling and Simulation in Air Traffic Management Conference, London, 14-15 November, 2017.
9. Java Platform, Standard Edition, website: https://www.oracle.com/technetwork/java/javase/overview/index.html, February 2019.
10. JSON, website: https://www.json.org/, February 2019.
11. Hart, P. E.; Nilsson, N. J.; Raphael, B., “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” 1968, IEEE Transactions on Systems Science and Cybernetics SSC4. 4 (2): 100-107.
12. Tarjan, R. E., “Depth-First Search and Linear Graph Algorithms,” 1972, SIAM Journal on Computing, 1 (2): 146–160.