

MATLAB EXPO

UNITED KINGDOM

Developing Architectures with System Composer

Mark Walker, MathWorks



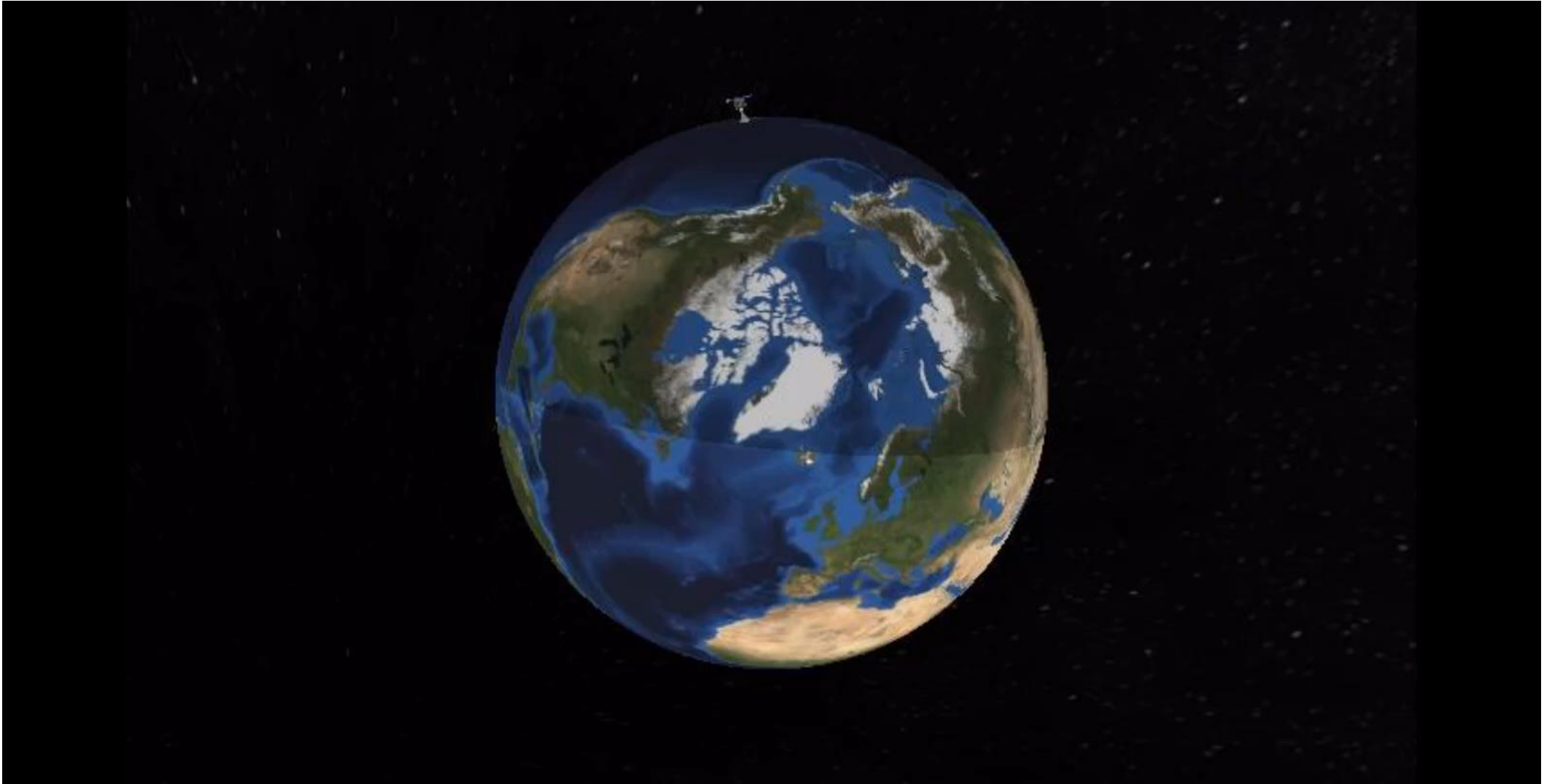
Mission

The system shall provide and store visual imagery of MathWorks headquarters [42.2775 N, 71.2468 W] 1 times daily at 10 metres resolution.

Mission



Mission



System Composer

The screenshot displays the MATLAB System Composer interface for a Simulink model named 'asbCubeSatArchModel'. The interface is divided into several sections:

- Toolbar:** Contains tabs for SIMULATION, DEBUG, MODELING, FORMAT, and APPS. The MODELING tab is active, showing options like Model Advisor, Interface Editor, Profile Editor, and Apply Stereotypes. It also includes a Stop Time field set to 86400/6, a Normal dropdown menu, Run and Stop buttons, and a Fast Restart option.
- Referenced Files:** A sidebar on the left showing the current model file 'asbCubeSatArchModel'.
- Model Diagram:** The main workspace shows a block diagram for 'asbCubeSatArchModel'. It consists of three main blocks: 'Space Environment' (containing 'CubeSatStates'), 'CubeSat Mission Enterprise' (containing 'Env' and 'CubeSatStates'), and 'Visualization' (containing 'CubeSatStates'). Data flow is indicated by arrows: 'CubeSatStates' from 'Space Environment' flows to 'Env' in 'CubeSat Mission Enterprise', which then flows to 'CubeSatStates' in 'Visualization'. An 'Env' input is also shown at the top of the 'Visualization' block. A 'CubeSatStates' output block is shown at the bottom right, connected to the 'CubeSatStates' block in 'Visualization' via an 'In' and 'Out' port.
- Copyright:** Copyright 2021-2022 The MathWorks, Inc.
- Status Bar:** Shows 'Ready', '100%', and 'VariableStepAuto'.

System Composer

- ...a tool for describing architectures
- ...but what is an architecture?
 - It describes how something works and why.
 - It will contain levels of details.
 - It will be traceable.
 - It may exist alongside other levels – each is a candidate design.

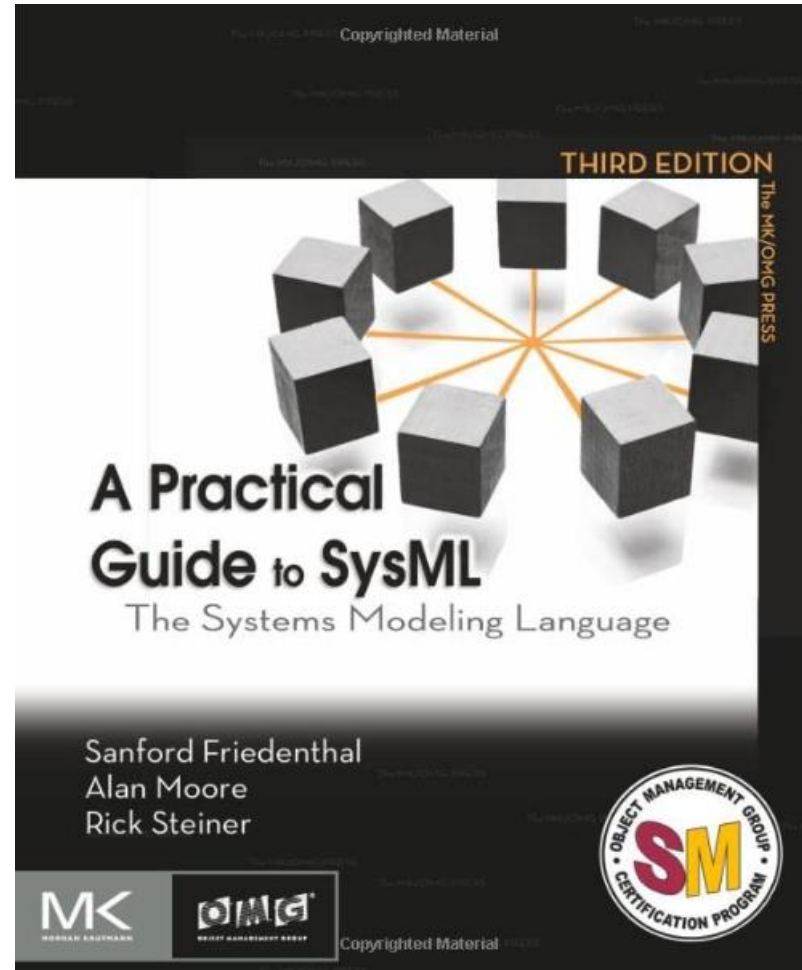
Systems Engineering

- From incose.org:

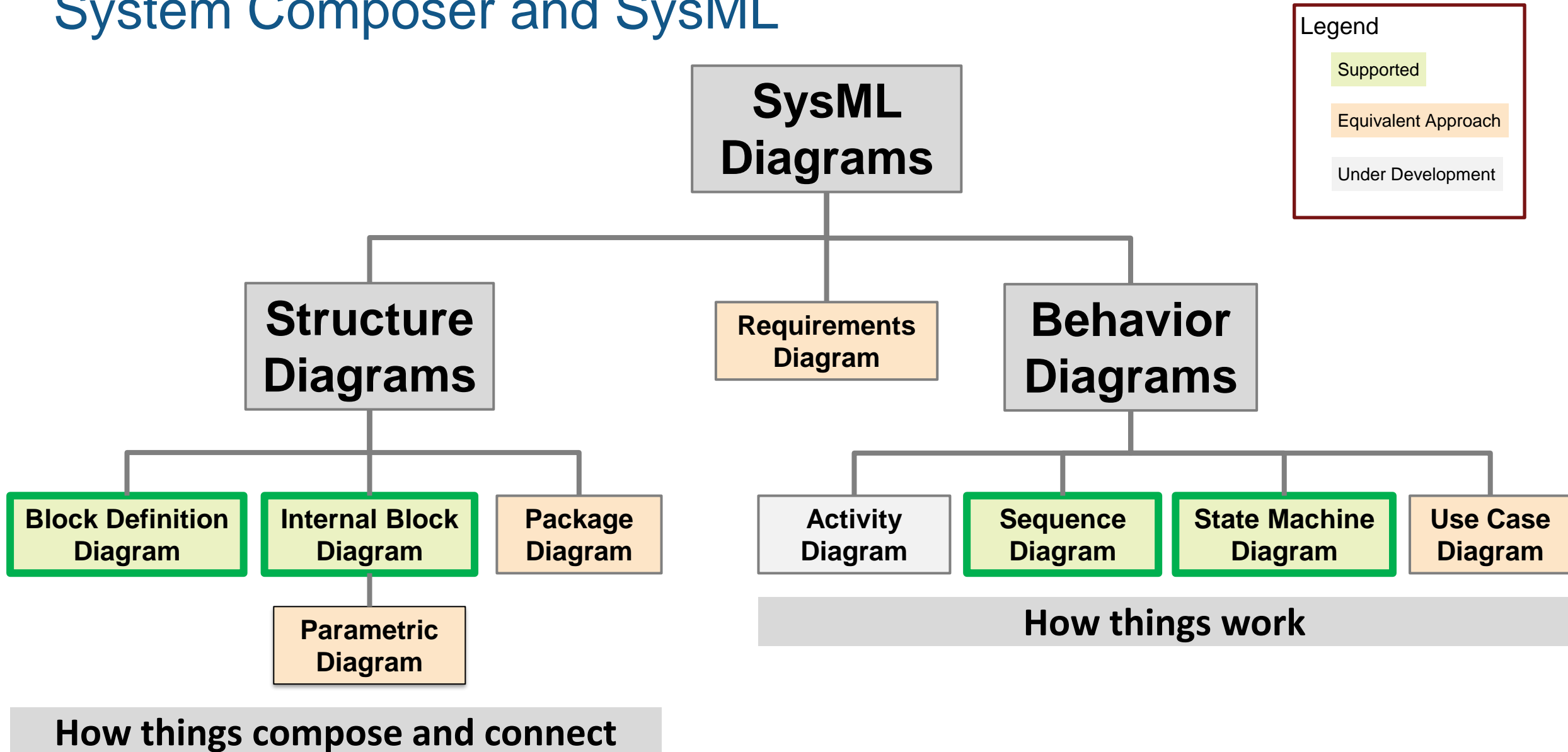
Systems Engineering is a **transdisciplinary** and **integrative** approach to enable the successful realization, use, and retirement of **engineered systems**, using **systems principles and concepts**, and scientific, technological, and management methods.

- Multi-disciplinary
- Combines business and technical
- Uses formalisms such as SysML

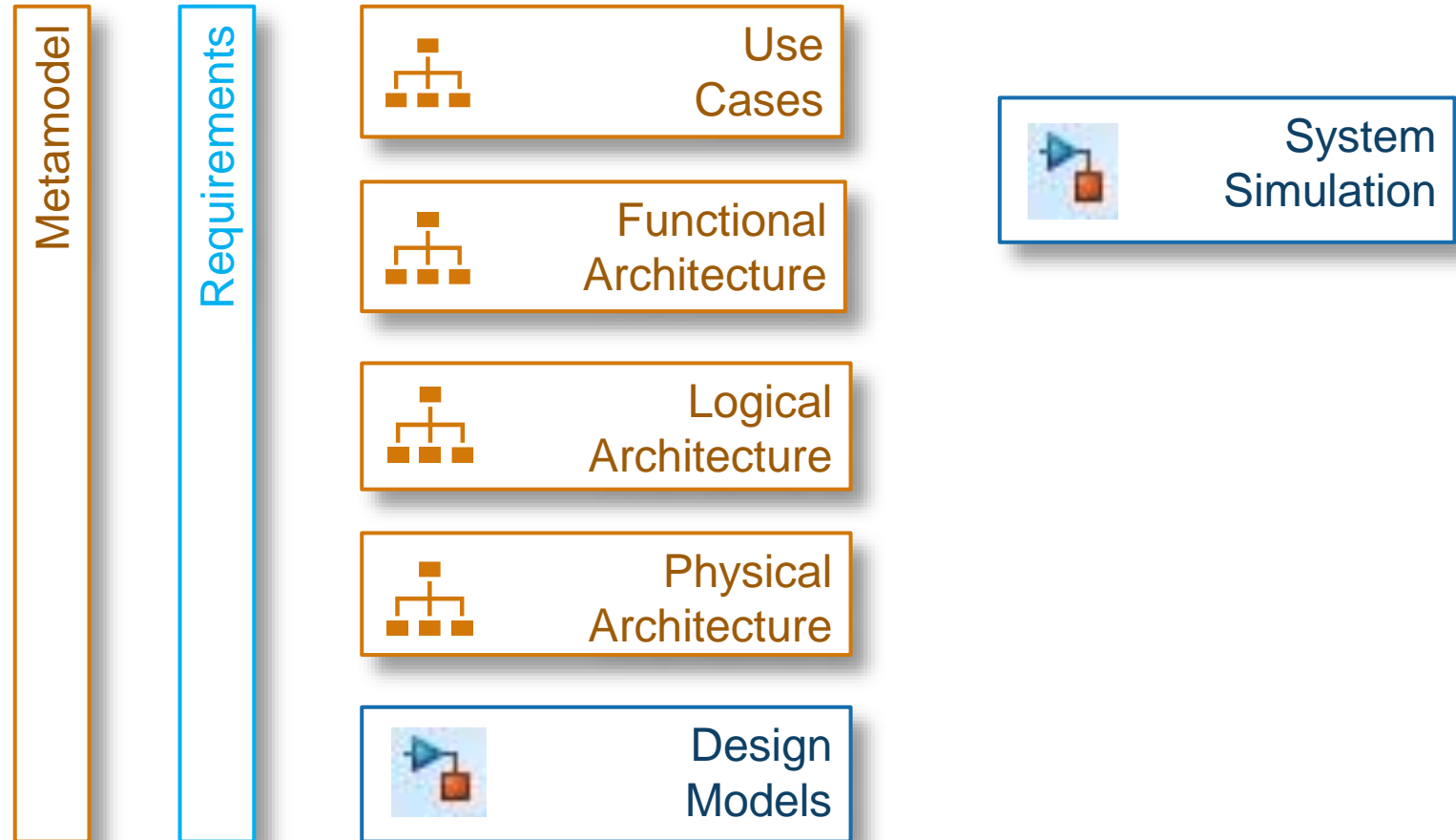
SysML



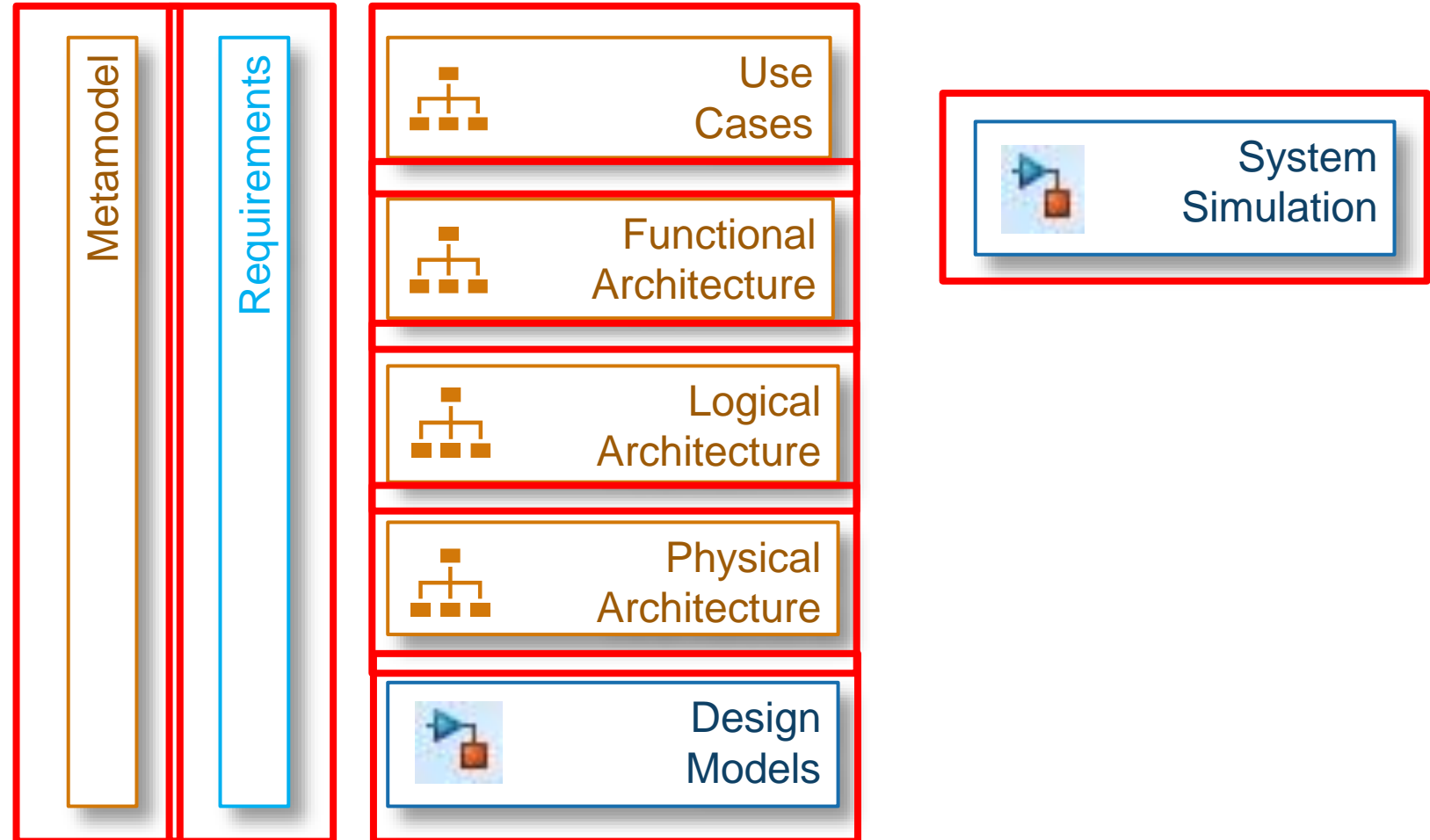
System Composer and SysML



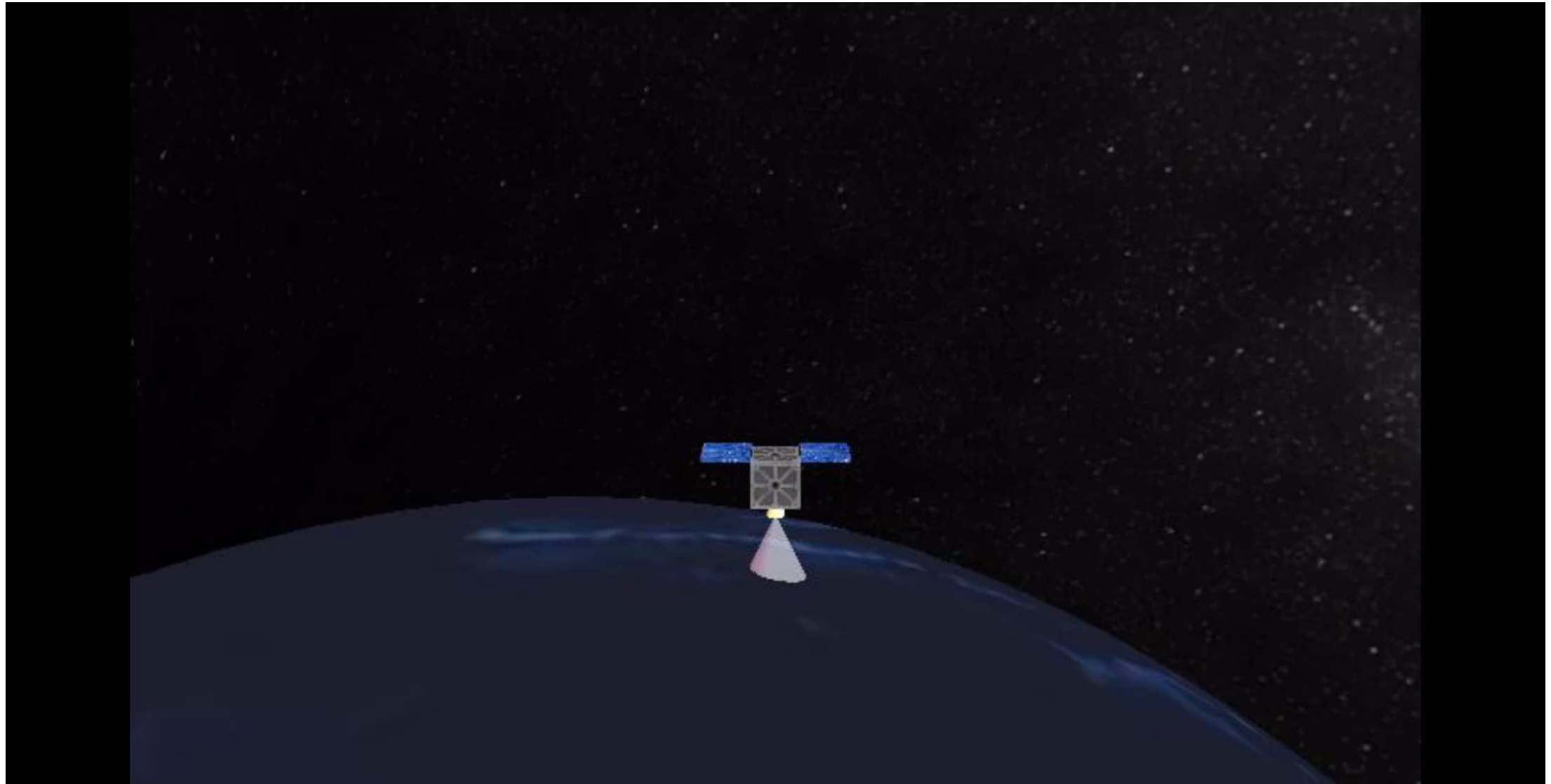
Bringing it all Together



Bringing it all Together



Mission



Core Concept: Projects

- ~~MBSE~~ PBSE: Project-Based Systems Engineering
- “Project” = a container for the design
 - Models
 - Data
 - Definitions
 - Requirements
 - Links
 - Tests
 - Reports
 - Code
- All items backed by a version control system (Git)

The Project

The screenshot shows the MATLAB Project Explorer window for a project named "EXPO23EXAMPLE". The interface is divided into several sections:

- PROJECT SHORTCUTS:** Includes icons for "New", "Open", and "Share" under the "FILE" tab, and "Dependency Analyzer", "Class Diagram", and "Model Testing Dashboard" under the "TOOLS" tab.
- ENVIRONMENT:** Includes "References", "Details", "Project Path", and "Startup Shutdown".
- SOURCE CONTROL:** Includes "Git Details", "Refresh", "Commit", "Fetch", "Push", "Pull", "Remote", "Branches", "Submodules", and "Stashes".
- Views:** Shows "All", "Project (78)", and "Modified (0)". The "Files" view is selected, showing a tree structure of folders and files.
- Labels:** Includes "Classification".
- Git:** Shows "Current branch: master", "Branch status: Normal", and "Coincident with /origin/master".
- Details:** A section at the bottom for file details.

The "Files" view displays a table with the following columns: Name, Status, Classification, and Git. The table contains the following entries:

Name	Status	Classification	Git
asbCubeSatSimulation	✓		.
CubeSatMBSEModel	✓		.
CubeSatRequirements	✓		.
MissionAnalysis	✓		.
presaved	✓		.
ProjectTasks	✓		.
work	✓		.
.gitattributes	✓		.
.gitignore	✓		.

Project Dependencies

The screenshot displays the MATLAB Project Explorer interface for a project named "Project - EXPO23EXAMPLE". The interface is divided into several sections:

- PROJECT SHORTCUTS:** Includes icons for New, Open, and Share.
- TOOLS:** Contains icons for Dependency Analyzer, Class Diagram, and Model Testing Dashboard.
- ENVIRONMENT:** Includes icons for References, Project Path, Startup Shutdown, Git, Refresh, Commit, and Details.
- SOURCE CONTROL:** Includes icons for Fetch, Remote, Submodules, Push, Branches, Stashes, and Pull.

The main area shows a list of project items with columns for Name, Status, Classification, and Git. The "Dependency Analyzer" view is active, showing a tree of folders and files. The "CubeSatMBSEModel" folder is selected and highlighted.

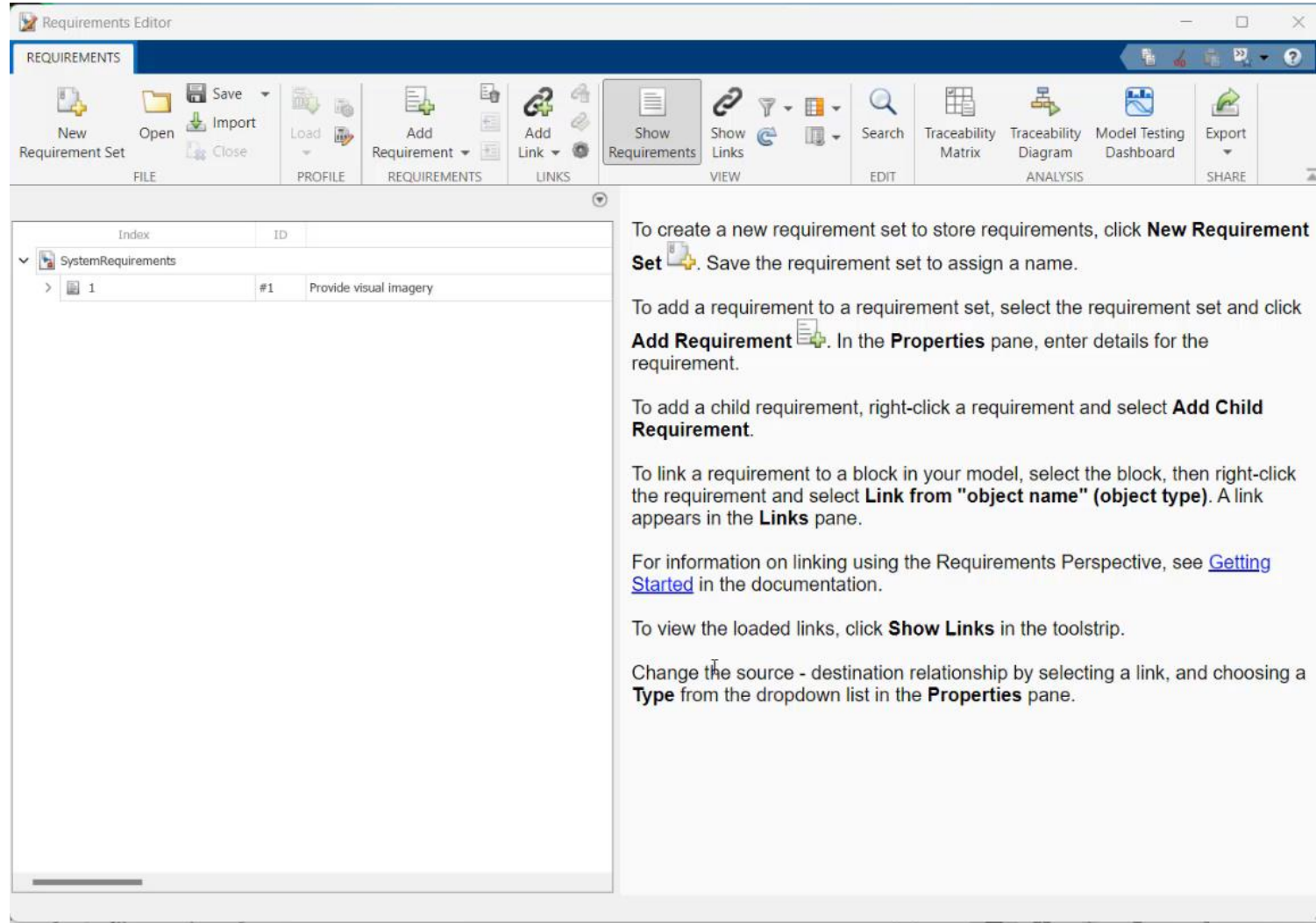
Name	Status	Classification	Git
asbCubeSatSimulation	✓		.
CubeSatMBSEModel	✓		.
CubeSatRequirements	✓		.
MissionAnalysis	✓		.
presaved	✓		.
ProjectTasks	✓		.
work	✓		.
.gitattributes	✓		.
.gitignore	✓		.

Labels: Classification

Git: Current branch: master, Branch status: Normal, Coincident with /origin/master

CubeSatMBSEModel (Folder) 0 labels

System Requirements



To create a new requirement set to store requirements, click **New Requirement Set**. Save the requirement set to assign a name.

To add a requirement to a requirement set, select the requirement set and click **Add Requirement**. In the **Properties** pane, enter details for the requirement.

To add a child requirement, right-click a requirement and select **Add Child Requirement**.

To link a requirement to a block in your model, select the block, then right-click the requirement and select **Link from "object name" (object type)**. A link appears in the **Links** pane.

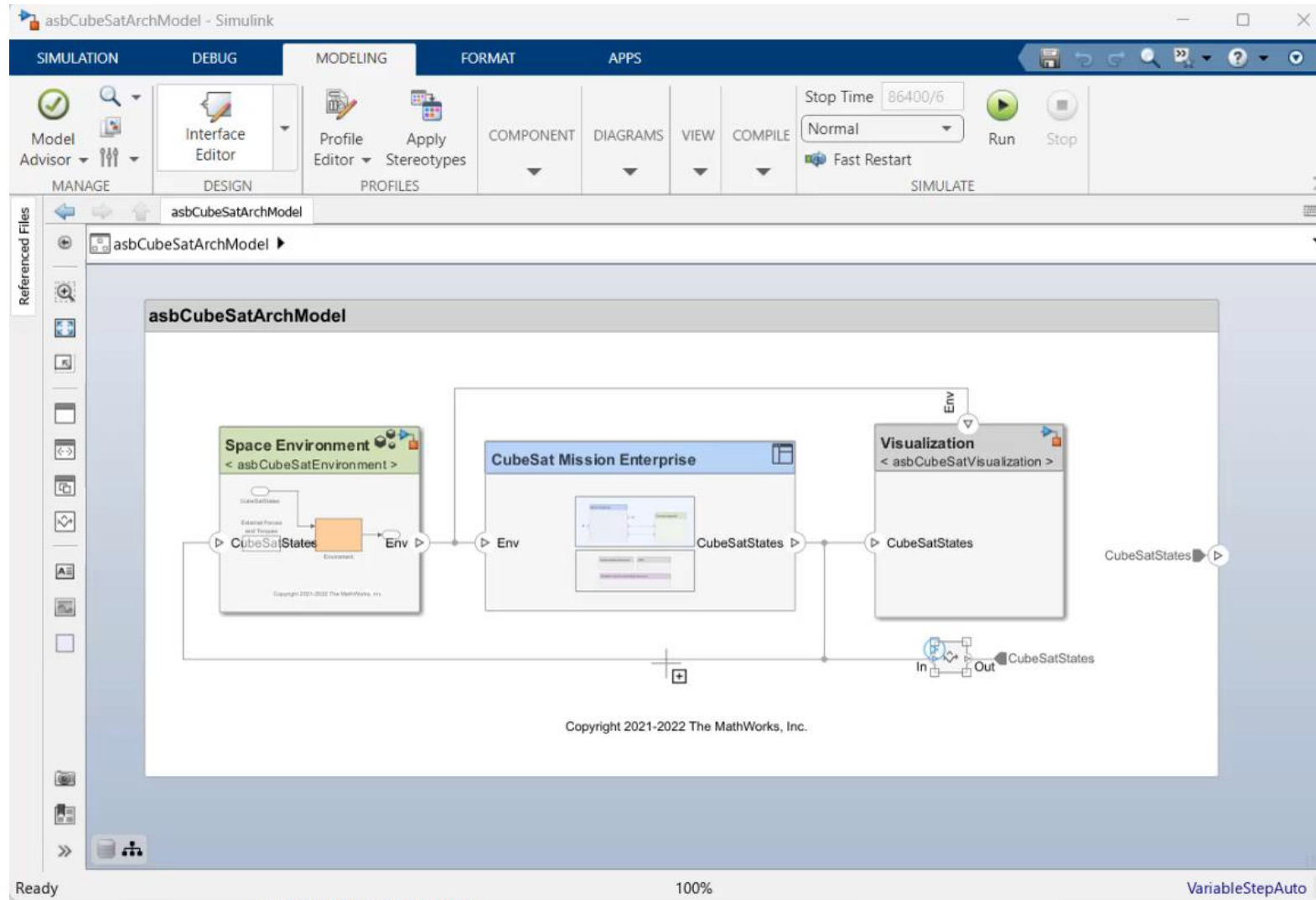
For information on linking using the Requirements Perspective, see [Getting Started](#) in the documentation.

To view the loaded links, click **Show Links** in the toolbar.

Change the source - destination relationship by selecting a link, and choosing a **Type** from the dropdown list in the **Properties** pane.

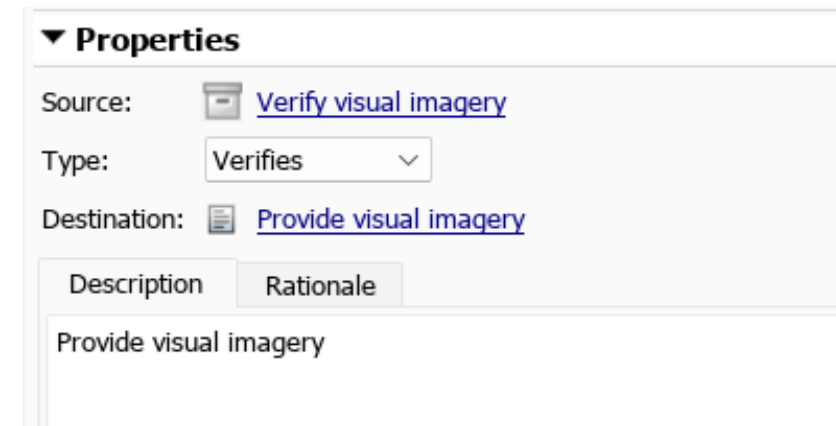
Index	ID
1	#1

System Model



Traceability

- The “Digital Thread” is built on a link primitive
 - Relates arbitrary things
 - Requirement, model, model element, test case, MATLAB code, C/C++ code
 - Custom targets, e.g. external SysML tool
 - Special types
 - Simulink or Stateflow implementation
 - Allocation
 - Categorised
 - Relate, Implement, Verify, Derive, Refine, Confirm
 - Custom
 - Directional
 - Traceable / reportable
 - Mergeable



Links

asbCubeSatArchModel - Simulink

SIMULATION DEBUG MODELING FORMAT APPS

Model Advisor Interface Editor Profile Editor Apply Stereotypes COMPONENT DIAGRAMS VIEW COMPILE

Stop Time: 86400/6 Normal Run Stop Fast Restart

MANAGE DESIGN PROFILES SIMULATE

Referenced Files asbCubeSatArchModel

Property Inspector

Component

Architecture Info

NAME	VALUE
▼ Main	
Name	CubeSat Mission Enterprise
Stereotype	Add...
▼ CubeSatEnterprise	Select
SystemOwner	'user'
MissionStartDate	juliandate(2019, 7, 1, 11, 0, 0)
> Parameters	Select

Ready 46% VariableStepAuto

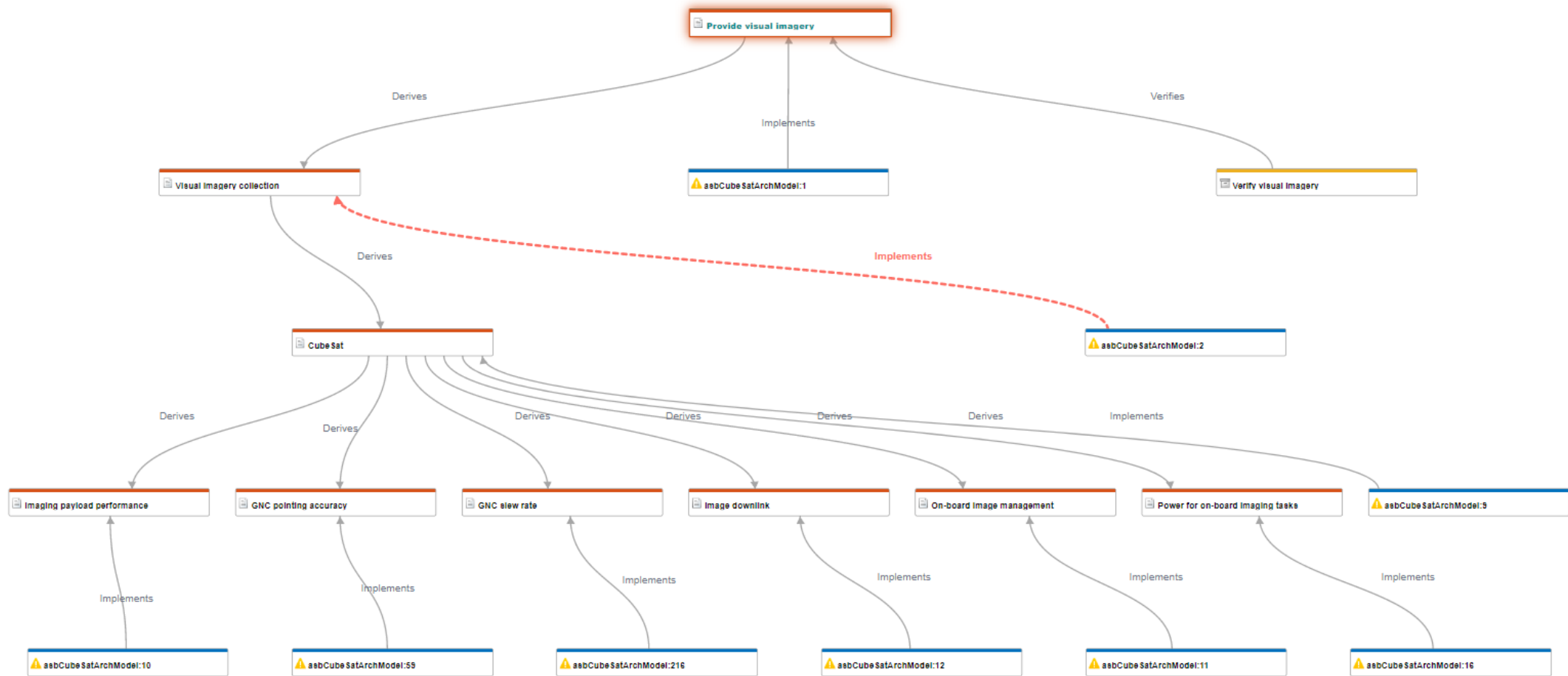
Link Traceability

The screenshot displays the Requirements Editor application window. The interface is divided into several sections:

- Toolbar:** Located at the top, it includes icons for 'New Requirement Set', 'Open', 'Save', 'Import', 'Close', 'Load', 'Add Requirement', 'Add Link', 'Show Requirements', 'Show Links', 'Search', 'Traceability Matrix', 'Traceability Diagram', 'Model Testing Dashboard', and 'Export'.
- Index Table:** A table on the left side of the main workspace. It has columns for 'Index', 'ID', and a description. The table contains one entry:

Index	ID	Description
1	#1	Provide visual imagery
- Properties Panel:** Located on the right side, it displays metadata for the selected requirement set:
 - Filepath:** C:\Projects\expo\EXPO23EXAMPLE\asbCubeSatMBSE\CubeSatRequirements\SystemRequirements.slreqx
 - Revision:** 13
 - Created by:** osaarela
 - Created on:** 25-Aug-2020 22:41:42
 - Modified by:** mwalker
 - Modified on:** 27-Sept-2023 15:54:07
 - Description:** (Empty text area)
- Custom Attribute Registries:** A section below the properties panel with an 'Attribute entries' table and 'Add', 'Remove', and 'Edit' buttons.
- Callbacks:** A section at the bottom of the right panel, currently empty.

Link Traceability



Stereotypes

The screenshot displays the Simulink environment for a model named 'asbCubeSatArchModel'. The interface includes a ribbon with tabs for SIMULATION, DEBUG, MODELING, FORMAT, and APPS. The MODELING tab is active, showing options like Profile Editor, Apply Stereotypes, and COMPONENT. The main workspace shows a block diagram with three main components: 'Space Environment' (stereotyped as '< asb.CubeSat:Environment >'), 'CubeSat Mission Enterprise', and 'Visualization' (stereotyped as '< asb.CubeSat:Visualization >'). Each component contains a 'CubeSatStates' block. Data flow is indicated by arrows labeled 'Env' and 'CubeSatStates'. The Property Inspector on the right shows the 'Architecture' section with a table of properties.

NAME	VALUE
Main	
Name	asbCubeSatArchModel
Stereotype	Add...
Parameters	
	Select

Copyright 2021-2022 The MathWorks, Inc.

Ready 70% VariableStepAuto

Shared Interfaces

The screenshot displays the Simulink software interface for a project named "asbCubeSatArchModel". The interface is divided into several sections:

- Top Menu Bar:** Includes tabs for SIMULATION, DEBUG, MODELING, FORMAT, and APPS. The MODELING tab is active.
- Toolbars:** Contains various icons for simulation control (Run, Stop), modeling (Interface Editor, Profile Editor, Apply Stereotypes), and other functions. The "Run" button is highlighted.
- Property Inspector:** Located on the right side, it shows the "Architecture" tab for the selected component. It displays the following information:

NAME	VALUE
Main	
Name	asbCubeSatArchModel
Stereotype	Add...
Parameters	
	Select
- Diagram Area:** The central workspace shows a block diagram of the "asbCubeSatArchModel" architecture. It consists of three main blocks: "Space Environment" (containing a "CubeSatStates" block), "CubeSat Mission Enterprise" (containing a "CubeSatStates" block), and "Visualization" (containing a "CubeSatStates" block). These blocks are interconnected via shared interfaces labeled "Env" and "CubeSatStates".
- Bottom Status Bar:** Shows "Ready", "70%", and "VariableStepAuto".

Shared Interfaces

The screenshot shows the MATLAB Model Explorer window. The left pane displays the Model Hierarchy with 'asbCubeSatModelData*' expanded to 'Design Data'. The main pane shows the contents of a dictionary object, with 'StatesOutBus' selected. The right pane shows the 'Simulink.Bus: StatesOutBus' configuration window.

Model Explorer Contents:

Name	Status	Value	DataType	Dimensions	Complexity
CubeSatTimeStep		1	double (auto)	[1 1]	real
visOff					
visSL3D					
variantVisualization		1	double (auto)	[1 1]	real
StatesOutBus	Mod				
ACSOOutBus	Mod				
FCSOutBus	Mod				
AttitudeErrorBus	Mod				
EnvBus	Mod				
initCond	Mod	<1x1 struct>	struct	[1 1]	N/A
gains		<1x1 struct>	struct	[1 1]	N/A
vehicle		<1x1 struct>	struct	[1 1]	N/A

Simulink.Bus: StatesOutBus Configuration:

Launch Type Editor

Design

Bus elements

Name	DataType	Complexity	Dimensions	Min	Max	Di
utc_JD	double	real	1	0	0	Fix
X_ecef	double	real	3	0	0	Fix
V_ecef	double	real	3	0	0	Fix
X_eci	double	real	3	0	0	Fix
V_eci	double	real	3	0	0	Fix

Description:

Revert Help Apply

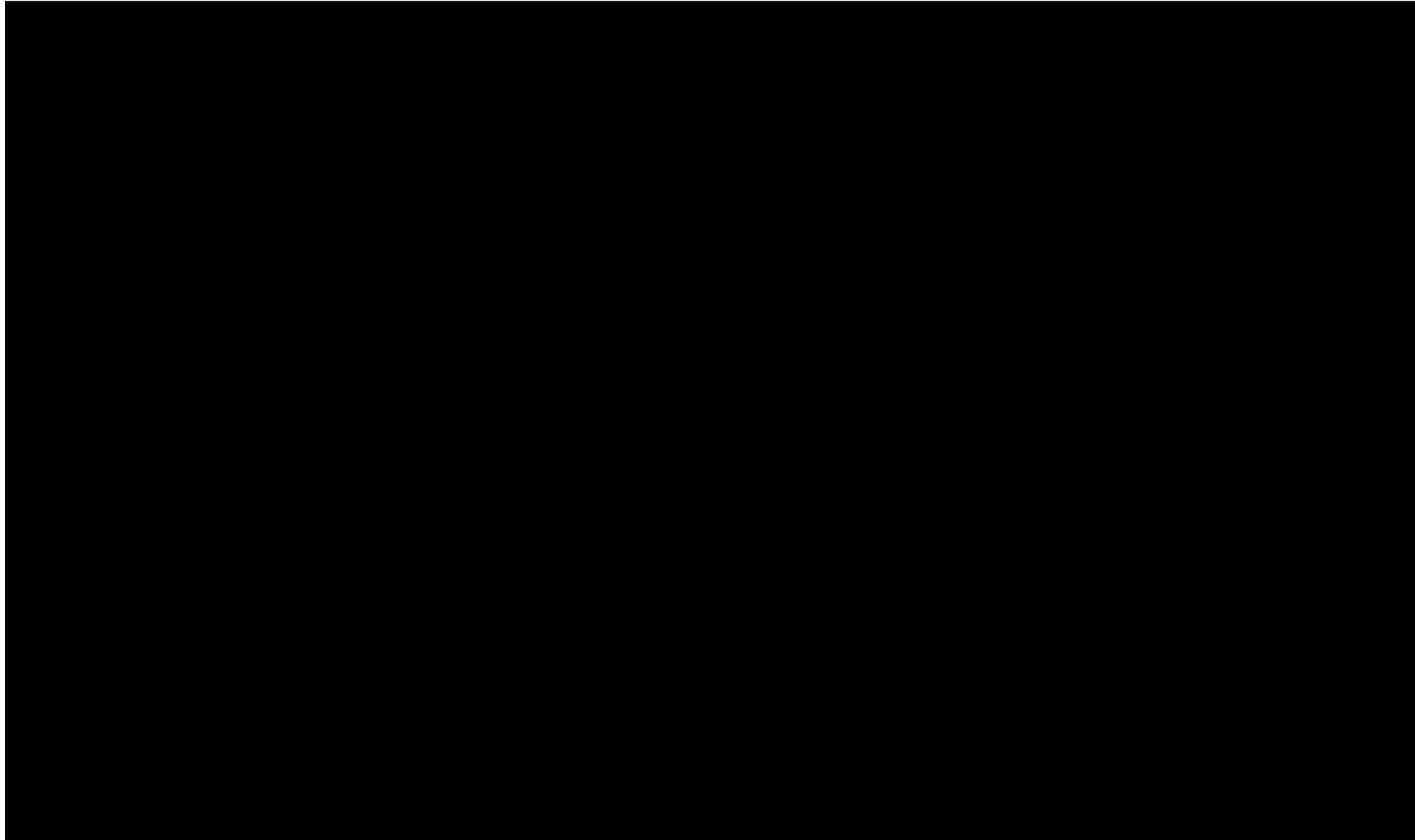
Shared Interfaces

The screenshot displays the MATLAB Dependency Analyzer interface for a project named EXPO23EXAMPLE. The window title is "Dependency Analyzer - EXPO23EXAMPLE". The interface is divided into several sections:

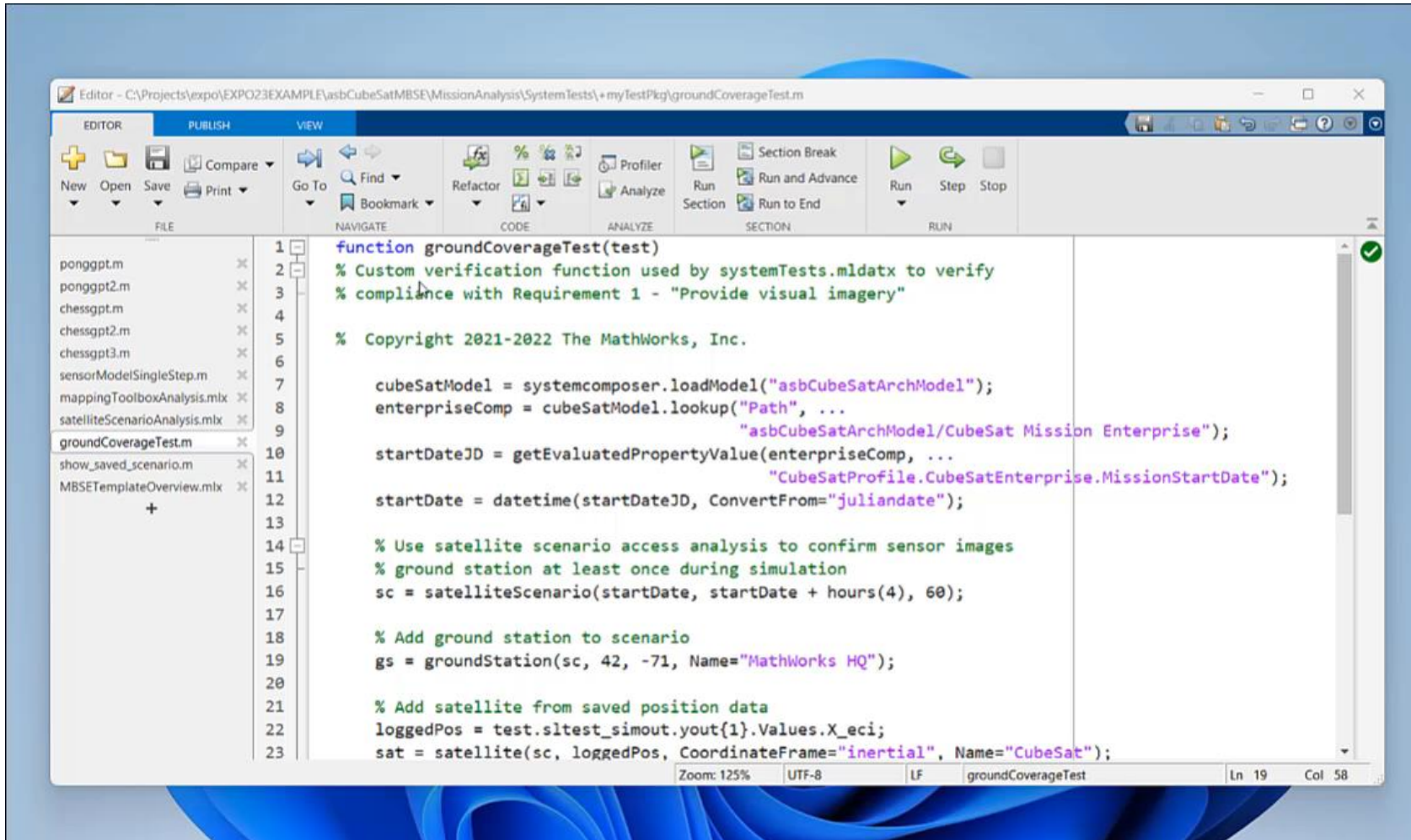
- ANALYZER** (top bar): Contains a search icon and a help icon.
- Toolbars:**
 - ANALYZE:** Includes "Analyze", "Restore to Default", "MATLAB Files", "Class Hierarchy", and "Model Hierarchy".
 - IMPACT ANALYSIS:** Includes "All Dependencies", "Impacted", and "Required".
 - LAYOUT:** Includes "Horizontal", "Vertical", and "Fit to View".
 - NAVIGATE:** Includes "Zoom In", "Zoom Out", and "Fit to View".
 - SHOW:** Includes "File List".
 - FIND:** Includes "Find".
 - EXPORT:** Includes "Project" and "Export".
- Legend:** Lists categories and counts:
 - MATLAB Code (28 of 28)
 - Simulink Models and Libraries (12 of 12)
 - Data (4 of 4)
 - Requirements (6 of 6)
 - Other Files (10 of 10)
- Overview:** A small thumbnail of the dependency graph.
- FILE LIST:** A list of files at the bottom.
- Properties Panel (right):**
 - Details:** Project: EXPO23EXAMPLE, Root: ...AMPLE\asbCubeSatMBSE.
 - Products:** Aerospace Blockset, Aerospace Toolbox, Mapping Toolbox, MATLAB, Requirements Toolbox, Simulink, Simulink 3D Animation, and [and 3 more](#).
 - Problems:** Missing file, [How to fix problems](#).

The central area displays a dependency graph with nodes represented by colored bars (blue for MATLAB Code, orange for Simulink Models and Libraries, yellow for Data, green for Requirements, and grey for Other Files) connected by lines. A mouse cursor is visible over the graph.

Parametric Studies / Analysis

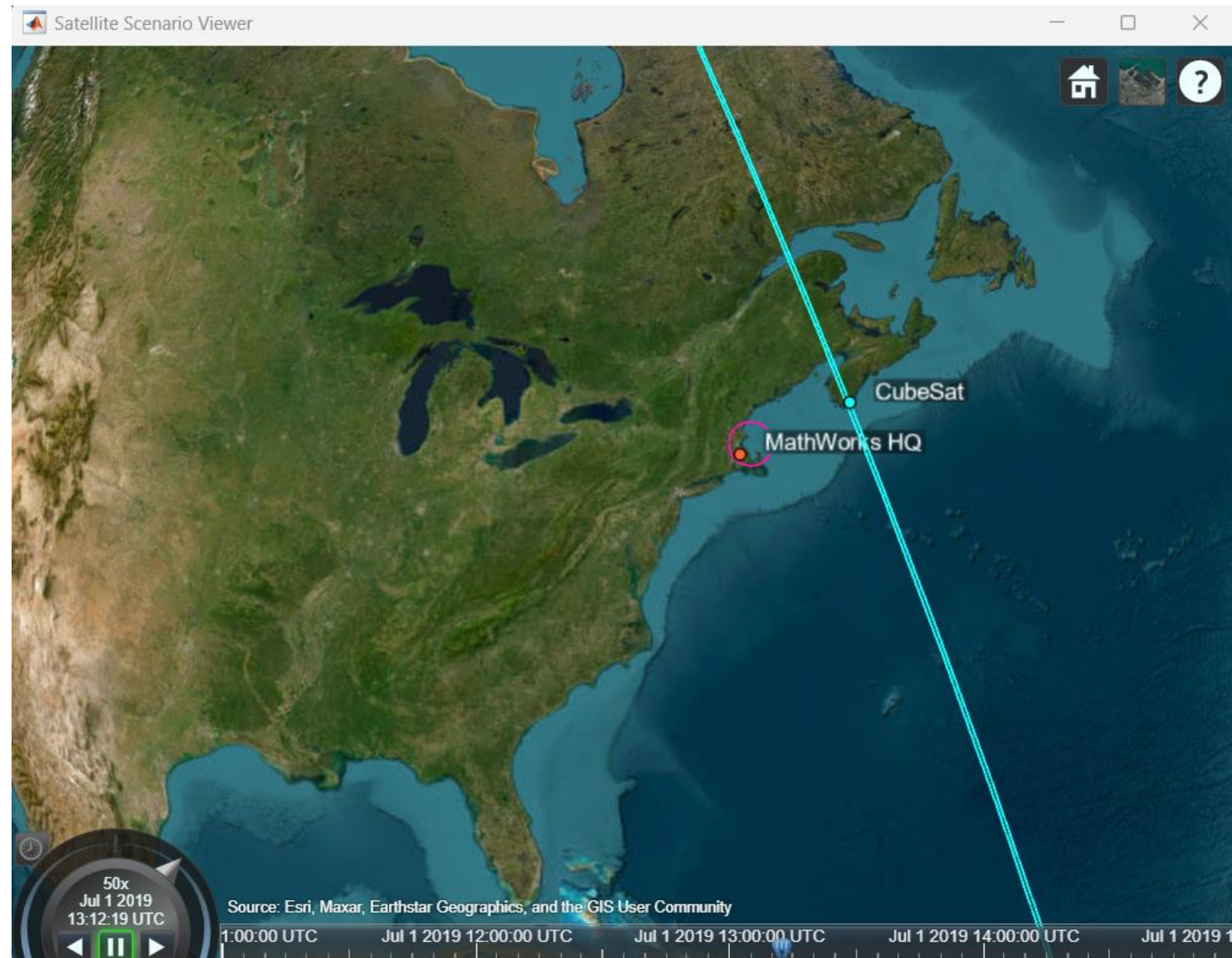


Parametric Studies / Analysis

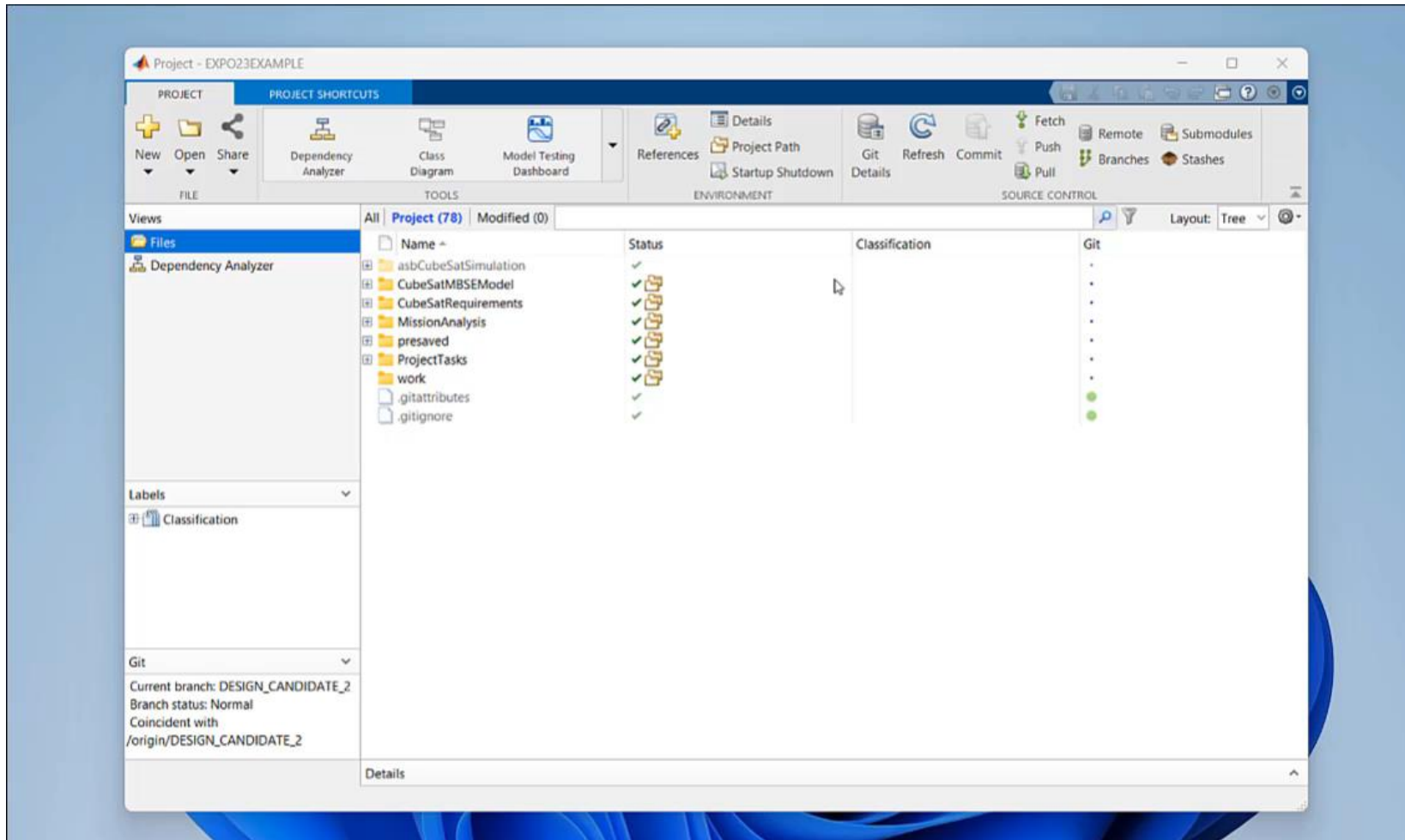


```
Editor - C:\Projects\expo\EXPO23EXAMPLE\asbCubeSatMBSE\MissionAnalysis\System Tests\+myTestPkg\groundCoverageTest.m
EDITOR PUBLISH VIEW
New Open Save Print Compare Go To Find Refactor Profiler Analyze Run Run and Advance Run Step Stop
FILE NAVIGATE CODE ANALYZE SECTION SECTION RUN
1 function groundCoverageTest(test)
2 % Custom verification function used by systemTests.mdatx to verify
3 % compliance with Requirement 1 - "Provide visual imagery"
4
5 % Copyright 2021-2022 The MathWorks, Inc.
6
7 cubeSatModel = systemcomposer.loadModel("asbCubeSatArchModel");
8 enterpriseComp = cubeSatModel.lookup("Path", ...
9 "asbCubeSatArchModel/CubeSat Mission Enterprise");
10 startDateJD = getEvaluatedPropertyValue(enterpriseComp, ...
11 "CubeSatProfile.CubeSatEnterprise.MissionStartDate");
12 startDate = datetime(startDateJD, ConvertFrom="juliandate");
13
14 % Use satellite scenario access analysis to confirm sensor images
15 % ground station at least once during simulation
16 sc = satelliteScenario(startDate, startDate + hours(4), 60);
17
18 % Add ground station to scenario
19 gs = groundStation(sc, 42, -71, Name="MathWorks HQ");
20
21 % Add satellite from saved position data
22 loggedPos = test.sltest_simout.yout{1}.Values.X_eci;
23 sat = satellite(sc, loggedPos, CoordinateFrame="inertial", Name="CubeSat");
Zoom: 125% UTF-8 LF groundCoverageTest Ln 19 Col 58
```

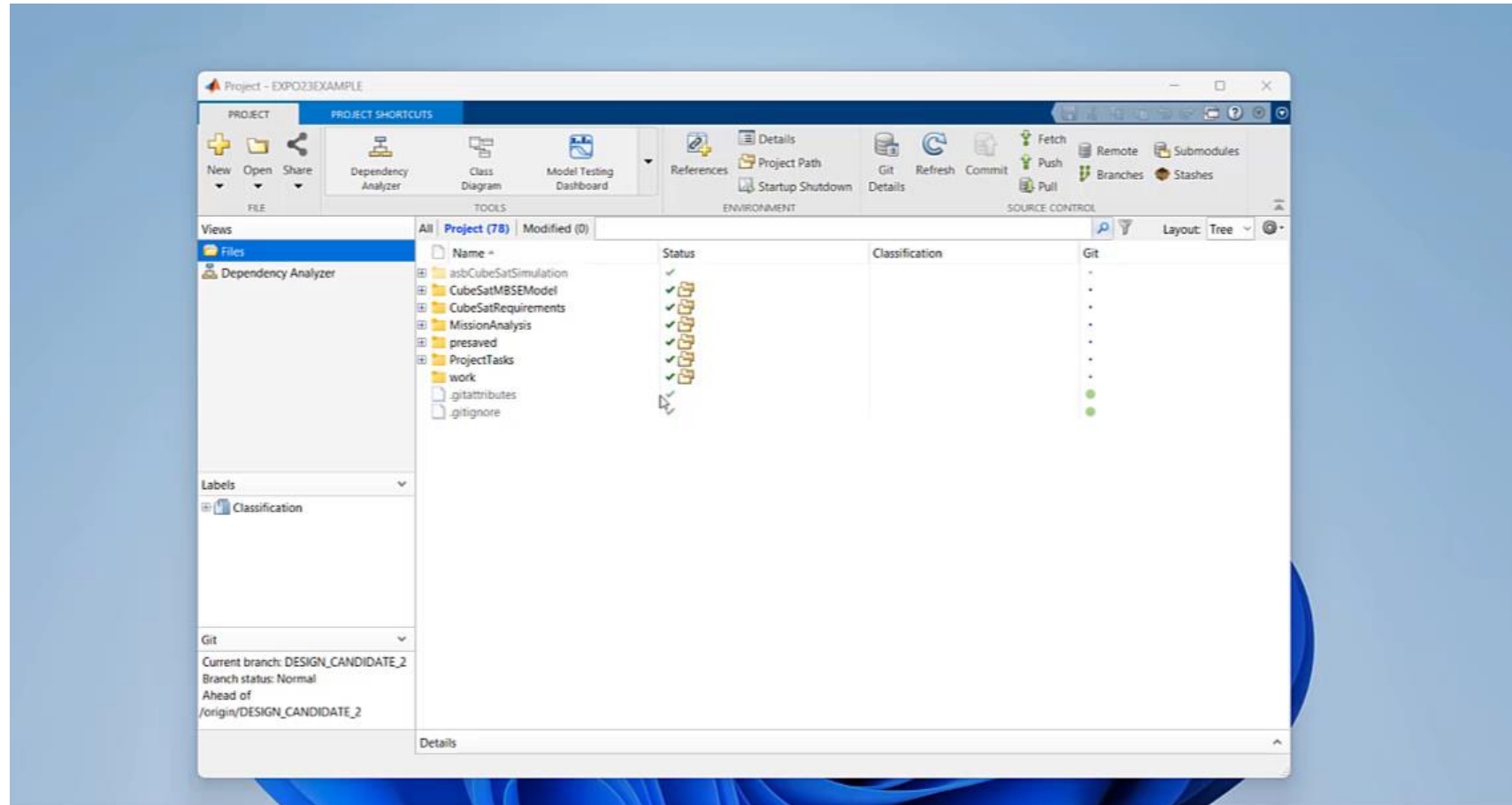
Parametric Studies / Analysis



Concurrent Working



Comparisons



Concurrent Working

The screenshot displays the MATLAB Branches dialog box for a project named "EXPO23EXAMP". The current branch is "DESIGN_CANDIDATE_2" with a HEAD hash of "ffd68c61cb8fb1e50f96865e03842d7433cf2bf3". The Branch Browser shows a list of branches: "DESIGN_CANDIDATE_2" (origin/DESIGN_CANDIDATE_2, HEAD), "master" (origin/master), and two other branches. The selected commit details are as follows:

ID	Author
ffd68c61cb8fb1e50f96865e03842d7433cf2bf3	mwalker

Commit Message: Added fallback spacecraft command capability to the ground station.

Committer: mwalker (mwalker@MATHWORKS-CYMPj.dhcp.mathworks.com)
Date: 2023-09-29 11:24:50

Differences from parent 5be55f764d25293fe14d58ec9c1e82848:
CubeSatM8SEModel

Git status: Current branch: DESIGN_CAN, Branch status: Conflicts, Coincident with /origin/DESIGN_CANDIDATE_2.

Concurrent Working

The screenshot shows the MATLAB Branches window with the following details:

Current Branch
 Name: master
 HEAD: 8d0a452b49637abe26d70943f6e93953f281b985
 [Revert to HEAD]

Branch Browser
 Branches: master [Switch] [Merge]

Commit	Author
DESIGN_CAND... master origin/mast... HEAD	mwalker ...
origin/DESI... Added fallback spacecraft comm and	mwalker ...
DESIGN_CAND... origin/DESI... Added mission per	mwalker ...
Updated test to show HQ better and share results for	mwalker ...
Added derives traceability for requirements	mwalker ...
Moved simulation models into same project (to enable	mwalker ...
Project snapshot taken from CubeSat example. 23b	mwalker ...

Commit Details:
 ID: 8d0a452b49637abe26d70943f6e93953f281b985
 Author: mwalker (mwalker@MATHWORKS-CYMpj.dhcp.mathworks.com)
 Committer: mwalker (mwalker@MATHWORKS-CYMpj.dhcp.mathworks.com)
 Date: 2023-09-29 12:38:13
 Message: Accepted changes from design candidate 1

Differences from parent 5562aa3a6cff5ba2516930f9d657b0eb7:
 - CubeSatMBSEModel

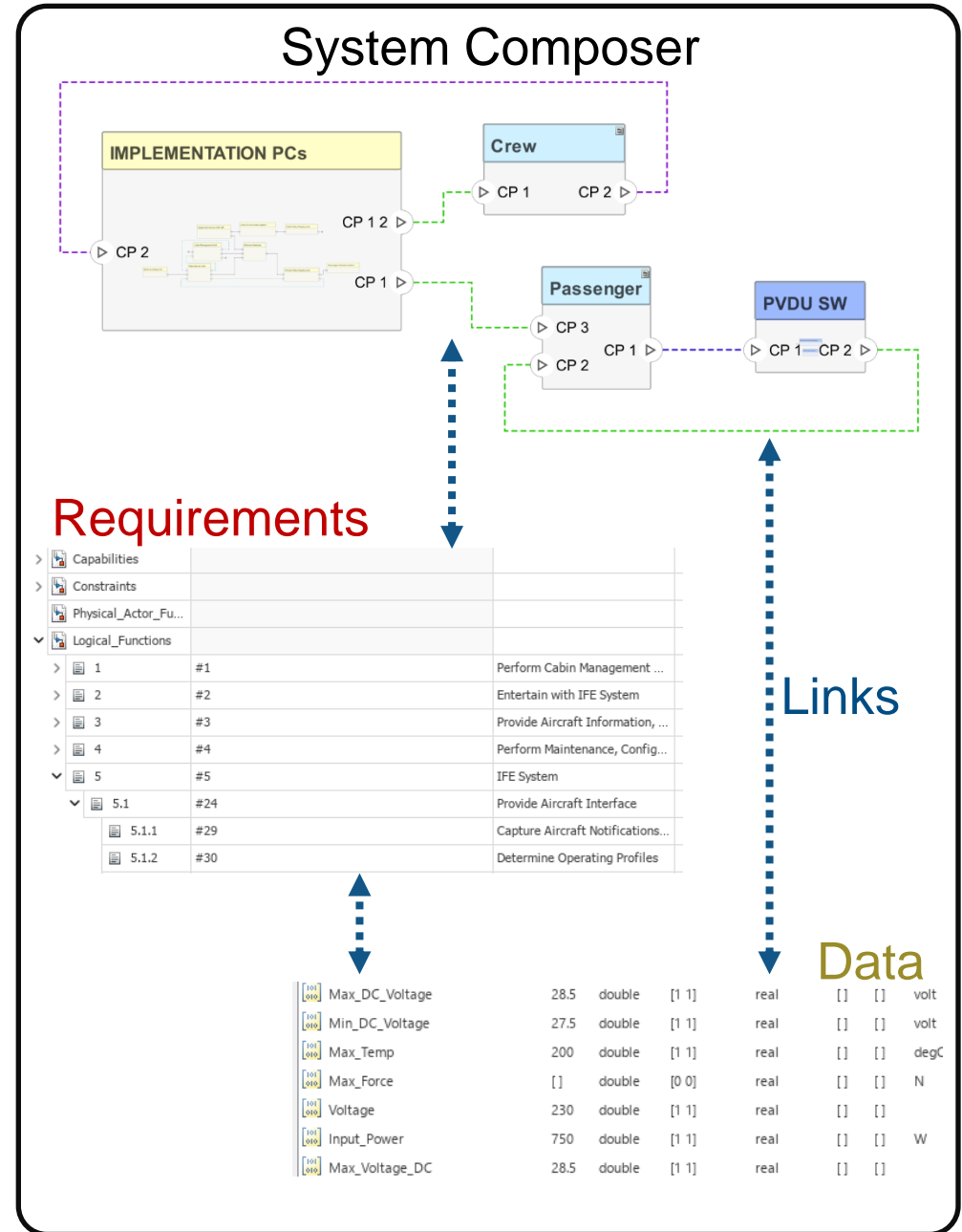
Differences from parent ffd68c61cb8fb1e50f96865e03842d7433:
 - CubeSatMBSEModel

Multiple Systems Tools

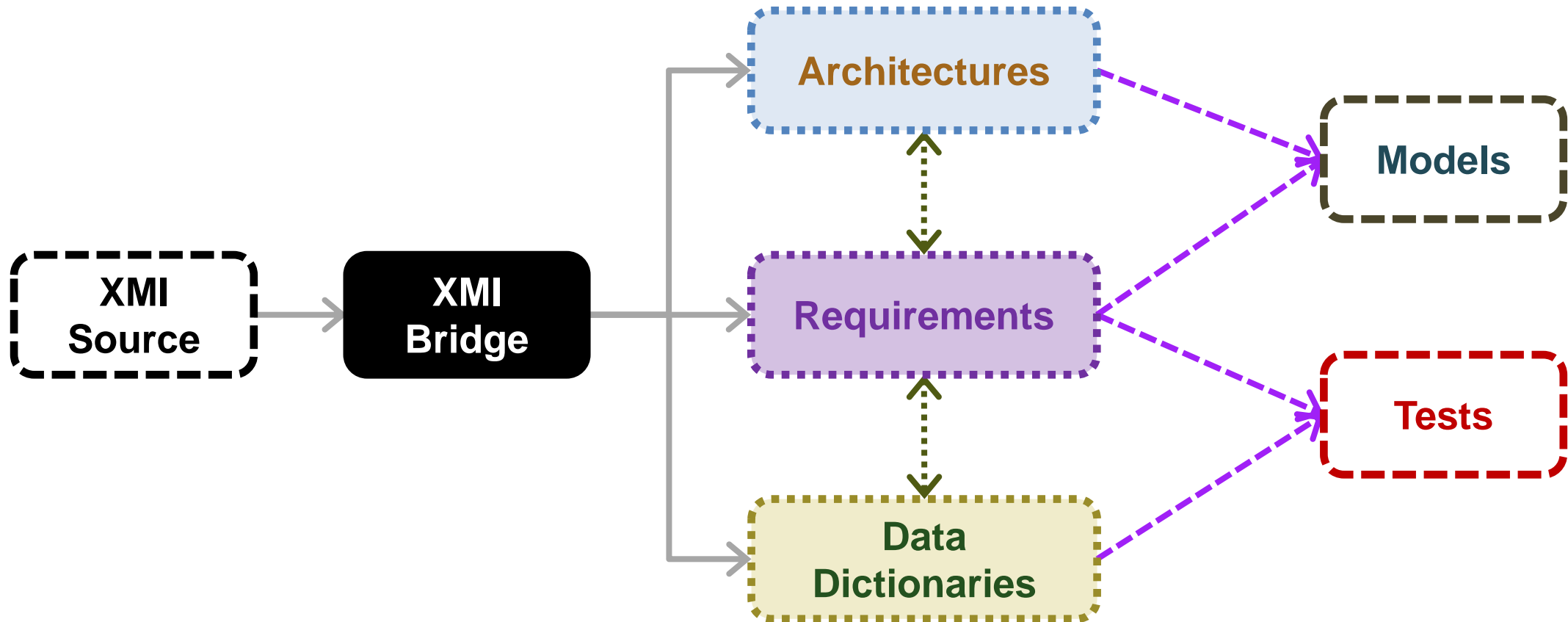


ReqIF
Requirements Interchange Format

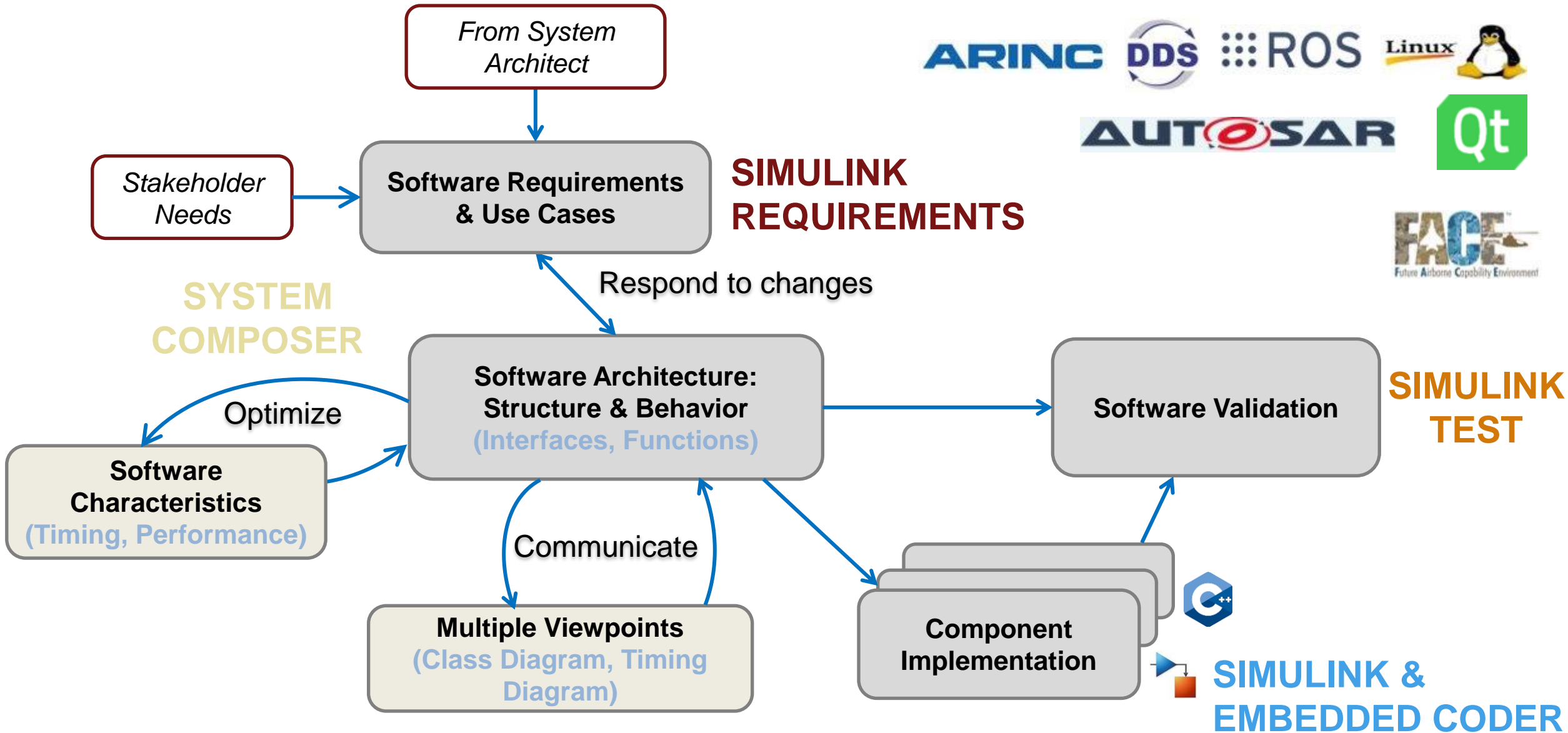
XMI



Linking Apples with Apples

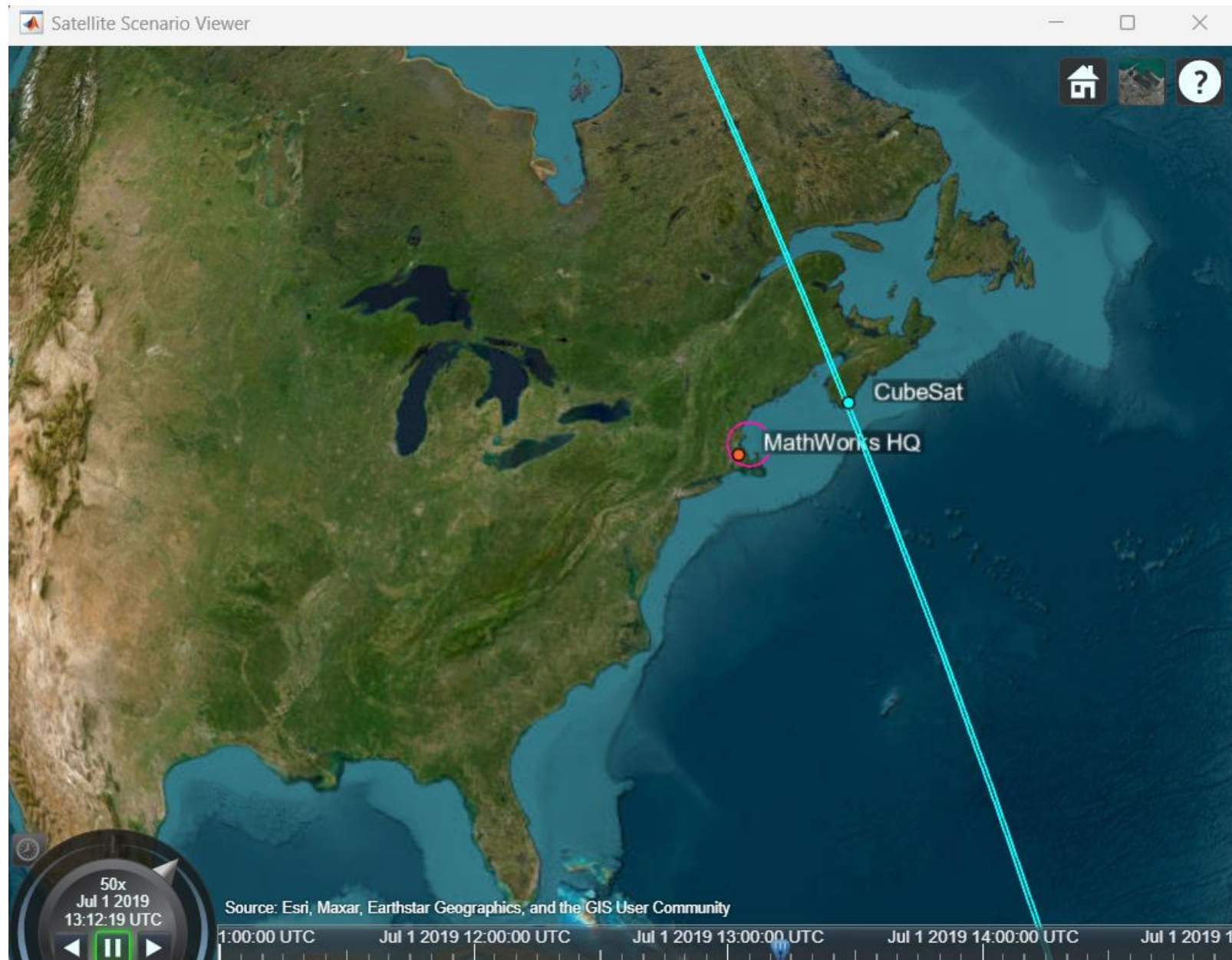


System Composer for Software Architectures



We Have Seen...

- Systems engineering and notation
- A top-down walk through the levels
- Traceability
- Versioning
- System simulation
- Concurrent working
- Connections to external tools



MATLAB EXPO

UNITED KINGDOM

Thank you



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.