

Release 2023 R1 Highlights
Ansys SCADE



SCADE Innovations Overview (2022 R2 + 2023 R1)



Defense & Avionics

Safety & Interoperability Standards

Avionics Displays

- SCADE Display: 3D object importer

ARINC 661

- 'New' re-architected Server Creator & Widget Library → reduced certification effort, better performances
- 'New' UA Adaptor

Software Architecture & FACE

- Evolutive maintenance
- Improved/flexible Cameo import



Automotive

Safety & Interoperability Standards

Electrification

- Updates to the Electrification Package (new examples, connections to TB and medini)

AUTOSAR Classic




- Misc AUTOSAR SWC related improvements

Autonomy / ADAS

- Integration with Driving Simulators (IPG CarMaker, Carla) as extension in dedicated ADAS/AV Package with demos

App Package

App Package



Common Functionality & Other Industries (Rail, Energy, Industry)

Collaboration, DevOps, CI/CD

- Improved refactoring capability
- Archive management

Cloud

- SCADE on Ansys Gateway for AWS

Traceability & Requirements

- ALM Gateway improvements

Code Generation

- Security / CERT-C compliance report
- New SCADE Suite KCG 6.6.3 → Improved code generation so that local modification in model produces local modification in the code.

Software V&V workflow

- Test generation assistance improvements

SCADE Innovations Highlights in 2023 R1

Safe & Secure Embedded Control and HMI Software



Revamped SCADE Solutions for ARINC 661

CDS Server, Widget Library, UA Adaptor

- ✓ Up to 5x faster server generation to accelerate development
- ✓ Cleaner, better-structured, easier to read generated code
- ✓ Better documentation structure, Python test cases for all widgets, traceability between all assets
- ✓ Overall, the above improvements support an easier end-user customization and DO-178C certification

Expanded AUTOSAR Classic support

- ✓ Focusing on software components (SWC), including support of Application Value Specifications, and Memory Mapping
- ✓ Strengthened support of AUTOSAR software application development flow both at modeling and code generation levels, combined with unique ISO 26262 ASIL D tool qualification

SCADE on Cloud

now available on *Ansys Gateway powered by AWS!*

- ✓ Deploy, control and scale powerful machines in a few clicks
- ✓ Leverage any existing Ansys licenses and your existing agreement with AWS
- ✓ Integrate powerful SCADE runners into your CI/CD pipeline
- ✓ Enforce access control by running SCADE clients, directly in your AWS infrastructure

/ Agenda

- SCADE 2023 R1 / Innovations **Synthesis** (5 slides)
- SCADE 2023 R1 / Innovations **Highlights** (15+ slides)
- SCADE 2023 R1 / Innovations **Details** (50+ slides)



**2023 R1 / Innovations
Synthesis for Embedded
Software**



What's New – SCADE for Aerospace & Defense

What's New

- **Revamped ARINC 661 Server Generator & Widget Library** ①
New, modular architecture for CDS design
- **Revamped ARINC 661 UA Adaptor** ②
Improved code generation, messages priorities management
- **Enhanced FACE solution**
Enhanced model navigation and code generation

Customer Value and User Benefits

- Up to 5x faster server generation to accelerate development
- Cleaner, better-structured, easier to read generated code
- Better documentation structure, Python Test Cases for all delivered widgets, traceability between all assets
- Overall, the above improvements support an easier end-user customization and DO-178C certification

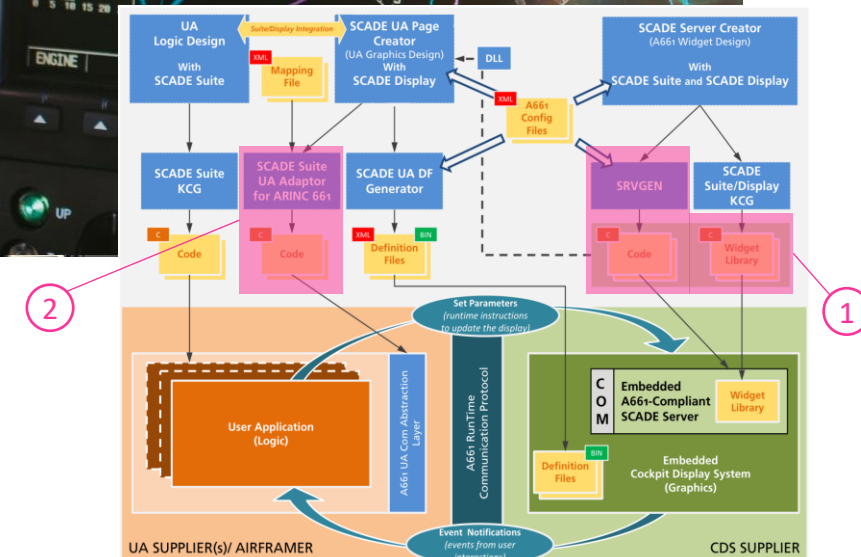
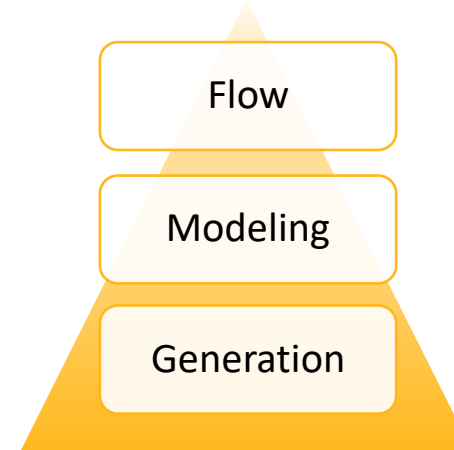


Photo credit: [Oskar Kadaksoo @ Unsplash](#)

What's New – SCADE for Automotive

AUTOSAR



What's New

- **Expanded AUTOSAR Classic support**, focusing on software components (SWC), including support of Application Value Specifications, and Memory Mapping
- Free “**Electrification Package**” on [Ansys Store](#) (reminder)

Customer Value and User Benefits

- Strengthened support of **AUTOSAR** software application development flow both at modeling and code generation levels, combined with unique **ISO 26262 ASIL D** tool qualification
- Step-by-step tutorial to support engineers in tool discovery and kickstarting design activities for **Battery Management Systems** and **Motor Controls** with Ansys SCADE and Twin Builder

Ansys / STORE

Careers Academic Customer Portal Resource Library

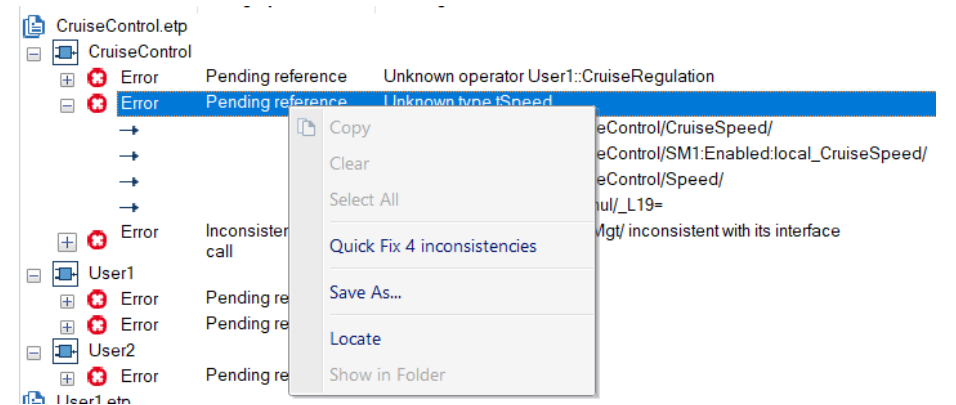
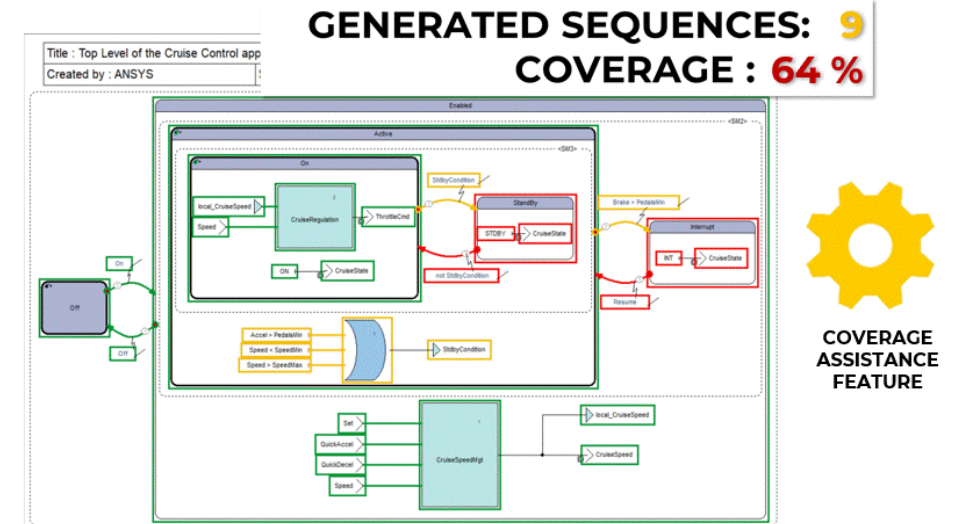
The screenshot shows the Ansys Store interface. On the left, there is a search bar with 'eliec' entered, and filters for Target Application, Product Version (2021 R1), and Price range (\$0 to \$15000). The main content area displays two product cards. The first card, 'Electrification Package', is highlighted with a red border and shows a 'FREE' price tag. The second card, 'Electronic Transformer', also shows a 'FREE' price tag. The interface includes a navigation bar at the top with 'Ansys / STORE' and 'Careers Academic Customer Portal Resource Library'. Below the search bar, there is a message about account creation and a search result summary showing 'Showing 1 - 4 of 4 items' and 'Results Per Page 24'.

What's New

- **Coverage Analysis Assistant improvements:** improved graphical user experience, improved analysis (abstraction, subnormal values)
- Enhanced **SCADE Suite model refactoring capabilities**, new capability to move entities from models to libraries

Customer Value and User Benefits

- Further speed-up complex and time-consuming model and code coverage analysis activities: better detect and report uncoverable parts of the model + generate justifications or test sequences for uncovered points
- Better support collaboration and teamwork when modeling with SCADE by adding refactoring capabilities



What's New – SCADE for Amazon Web Services (AWS) users

What's New

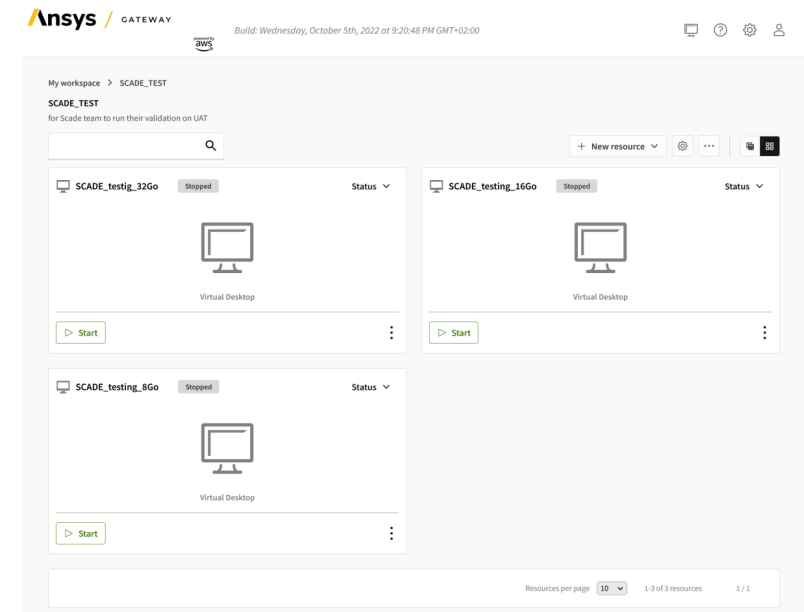
- SCADE is now available on Ansys Gateway powered by AWS!



REMOVE YOUR HARDWARE BARRIER

Customer Value and User Benefits

- Deploy, control and scale powerful machines in a few clicks (up to hundreds of cores and terabytes of RAM)
- Leverage any existing Ansys licenses and your existing agreement with AWS
- Integrate powerful SCADE runners into your CI/CD pipeline.
- Enforce access control by running SCADE clients, directly in your AWS infrastructure.





Innovations **Highlights for
Embedded Avionics &
Display Software**



ARINC 661 / New Server Generator

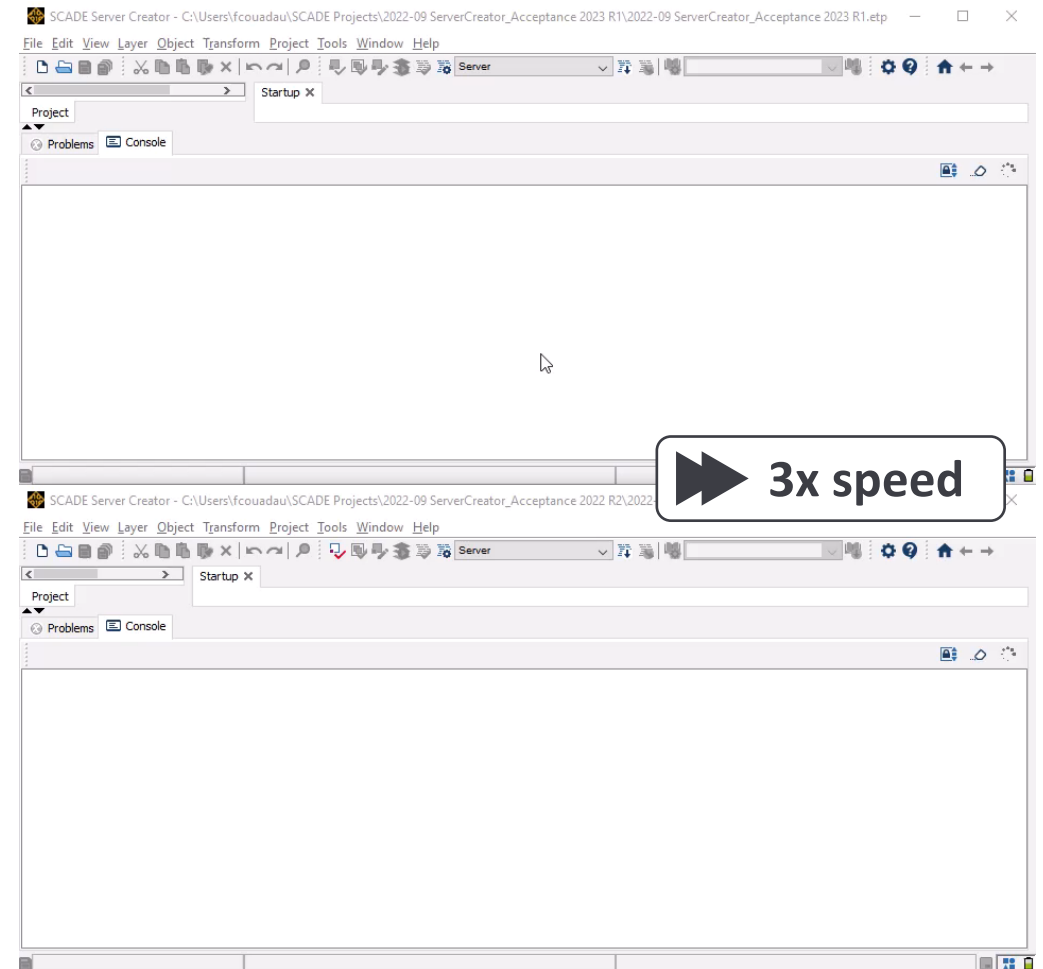
Up to 5x faster server generation to speed up development 👉

- Can be up to 7x faster with full cleanup
- Allows for faster iteration times for CDS customers

A **single Server Code Generator** replaces former SRVGEN and WWGEN code generators

~33% **less generated files** overall:

	2022/R2	2023/R1
server/a661_description	1490	1147
Server_output	2265	1344
<root>	2	4
	3757	2495



ARINC 661 / New Widget Library

Stepping stones for customers starting from our standard Widget Lib:

1. **One SRS per widget** reduces DO-178C certification workload
 - See <ANSYS Inc>\v231\SCADE\SCADE A661\Project Documentation\Widgets
2. **Improved widget architecture**: robustness, modularity, scalability, consistency
3. **Widget Library Test Cases** ease DO-178C certification / non-regression testing
4. **Traceability documentation** ensures proper linkage between Specification, Design, HLR, LLR and tests

Progress:

- **2023/R1**: 60 widgets redesigned under new architecture
- **2023/R2**: 34 remaining widgets will be released

Excerpt from PushButton\test_hlr_046.py:

```
import logging
from typing import Tuple

from tests.common.test_harness import EMPTY_LIST_WIDGETS, Status, TestCase, check_image

DEFAULT_SGFX = "graphic_pushbutton.sgfx"
logger = logging.getLogger("tests")

def tp_pushbutton_046(
    tc: TestCase, screenshot: str, position: Tuple[int, int], size: Tuple[int, int]
) -> Status:
    """Check BoundingBox values for PushButton
    """
    --- AUTHORS: WJC
    --- IMPLEMENTS: A661WL-TSTRAT-GRAPH-BOUNDING
    --- INPUT_MESSAGES: None
    --- INPUT_REQUESTS: None
    --- OTHER_INPUTS: See Test Case
    --- EXPECTED:
    --- * Screenshots: Display is matching BoundingBox computation
    """
    --- # P1
    --- logger.debug(" P1: Container Visibility.Value output is equal to True")
    --- # Step #01:
    --- logger.debug(" Step #01: Compare captured picture with expected picture")
    --- obt_img = tc.outdir / screenshot
    --- tc.te.screenshot_to_img(obt_img, position[0], position[1], size[0], size[1])
    --- exp_img = tc.expdir / screenshot
    --- check = check_image(obt_img, exp_img, obt_img.with_suffix(".diff.png"))
    --- assert (
    ---     check
    ---     ), f"Image is NOK, {obt_img} is different with expected picture: {exp_img}."
    --- return Status.PASSED

class TC_PushButton_GRAPH_046_01(TestCase): # pylint: disable=invalid-name
    """Test BoundingBox values of PushButton
    """
    --- AUTHORS: WJC
    --- OBJECTIVE:
    --- REQUIREMENTS:
    --- [A661WL-PUSHBUTTON-HLR-046]
    --- PRE-CONDITIONS: None
    --- INPUTS WIDGETS:
    --- All PushButton under input model
    --- INPUT ACTIONS: None
    --- INPUT_SERVER_CONFIGURATION: tests/server_config/default_conf.xml
    --- OTHER_INPUTS: See implementation of Test Procedure
    --- EXPECTED: See implementation of Test Procedure
    --- TEST ACCEPTANCE CRITERIA:
    --- Comparisons of expected and obtained screenshots return no difference
    --- No A661 error has been sent by the Server
    --- No Python Exception has been raised
    """
    --- TEST_PROCEDURE = tp_pushbutton_046
    --- INPUT_DF = DEFAULT_SGFX
    --- TP_ARGUMENTS = {
    ---     "screenshot": "boundingbox.png",
    ---     "position": (0, 0),
    ---     "size": (8_500, 1_5000),
    --- }
    --- INPUT_WIDGETS = EMPTY_LIST_WIDGETS
```


SCADE Architect

- Upgrade SysML importer to Sparx EnterpriseArchitect 16.0
- Flexible SysML importer for Cameo and Enterprise Architect
 - Customizable importer rules based on provided 2 sets of rules: « SysML subset », and « UML objects »
 - Detailed reporting log



SCADE Avionics Package solution for the FACE Standard

- Usability enhancements
 - New FACE elements « Realized by » and « Typed by » reverse links for enhanced navigation in the FACE data model
- Code generation for the FACE Standard
 - Upgrade of the FACE Wrapper for SCADE Suite KCG facilitating code integration with FACE OSS and TSS providers



Innovations **Highlights** for Embedded Automotive Software

AUTOSAR Support Enhancements



AUTOSAR

Flow

Modeling

Generation

- Syntactic editor for Value Specifications
- Arithmetic expressions with references to constants for Constants and Init attributes
- Support of Application Value Specifications
- Support of Memory Sections specification
- Constants for Init attributes with expressions
- Memory Sections in generated code
- ACG 2.4 & KCG 6.6.3 integration¹

¹: ACG 2.4 implements new features, KCG 6.6.3 new licensing. Not qualified



Common Innovations **Highlights** for Embedded Control Software

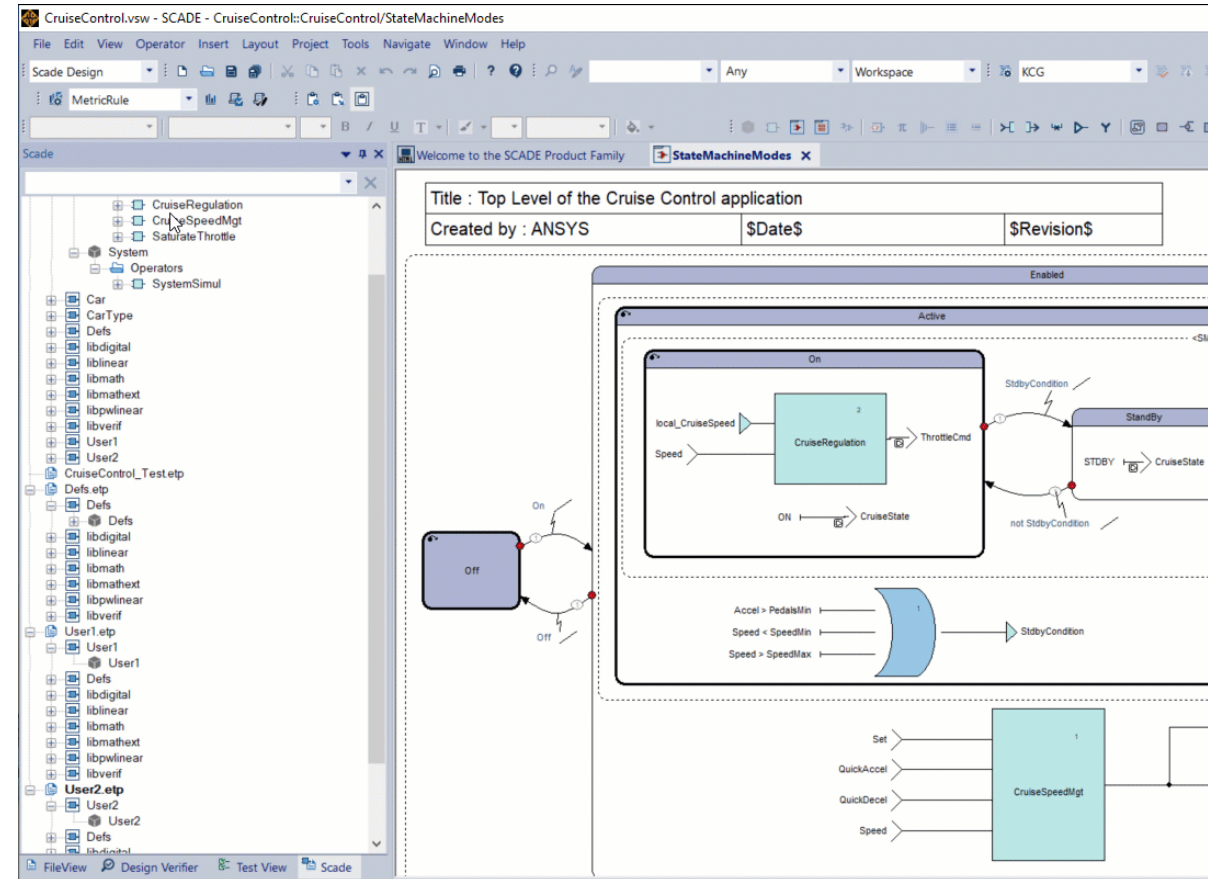


Refactoring Capability Improvements

Improved Teamwork and Collaboration capabilities

- Enable reducing development time
- Ease teamwork

New **refactoring features** enabling to move entities between models and libraries



Coverage Assistance Improvements

Improved User experience

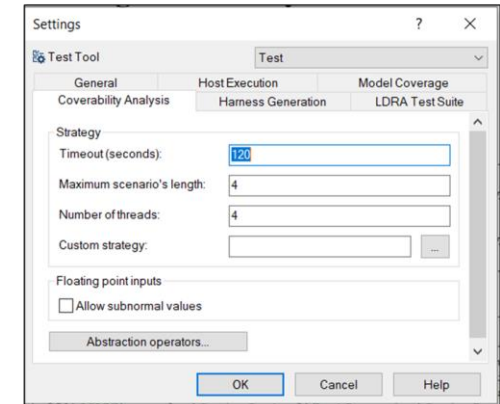
- New setting Panel
- Analysis no more modal

Improved analysis using abstraction

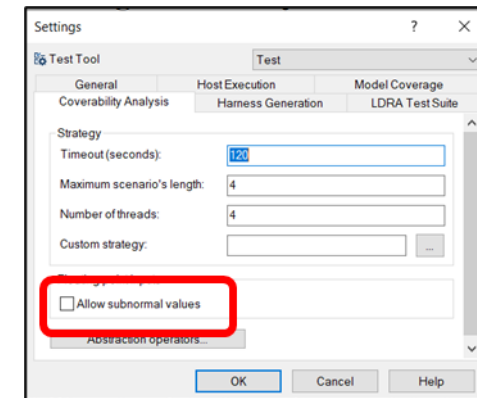
Better customization of the analysis allowing to remove or allow **subnormal values**

```
## Empty lines and lines starting with # are ignored
## @criterion@ is expanded to:
## - MC mode: option from active Test configuration
## - SCADE mode: '-criterion OMCDC'
@criterion@
## instrumentation options
#-instr P:N,P:N2
#-noinstr P:N,P:N2
## @expansion@ is expanded to:
## - MC mode: options from Suite configuration referenced by active Test configuration
## - SCADE mode: options from active Suite configuration
@expansion@
## @observation@ is expanded to:
## - MC mode: options from Suite configuration referenced by active Test configuration
## - SCADE mode: options from active Suite configuration
@observation@
## @root@ is expanded to:
## - MC mode: options from Suite configuration referenced by active Test configuration
## - SCADE mode: '-root <selected operator>'
@root@
## timeout in seconds
-timeout 20
## maximum depth for BMC (bounded model checking)
-bmc_depth 7
## number of threads used by PSL
## * strictly negative value: PSL is run using a single core.
## * 0: PSL is run with the maximum of cores.
## * strictly positive value: PSL is run with the specified value, this value
## being bounded by the maximum number of cores on host
-threads 4
## custom strategy
#-strategy "-solver current (inc=true) -bmc conj start 0 stop 5 -solver 5 -eqLemmas ind fu
```

Settings in 2022 R2



Settings in 2023 R1



- New ALM Gateway API

- ALM Gateway provides now a new Python API Allowing to get access to traceability information
- Examples are provided here : v231\SCADE\SCADE\APIs\Python\examples\traceability
- There is one script example to remove pending links:

```
79 def print_links(trace_project: traceability.Project):
80     print("List of links")
81     print("(oid, req_id, status)")
82     for link in trace_project.traceability_links:
83         print(f"({link.source.identifier}, {link.target.identifier}, {link.status.name})")
84
85
86 def print_pending_links(trace_project: traceability.Project):
87     print("List of pending links")
88     print("(oid, req_id, status)")
89     for link in traceability.get_pending_links(trace_project):
90         print(f"({link.source.identifier}, {link.target.identifier}, {link.status.name})")
91
92
93 def print_traceable_elements(trace_project: traceability.Project):
94     print("List of TraceableElements")
95     for element in trace_project.traceable_elements:
96         print_element(element)
97
98
99 def remove_all_pending_links(project: traceability.Project):
100     for link in traceability.get_pending_links(project):
101         project.remove_link(link.source.identifier, link.target.identifier)
102
```



Innovations **Highlights for
Amazon Web Services
users**



SCADE on Ansys Gateway powered by AWS / Benefits

Deployment

- Accelerate innovation by removing on-premise hardware barrier
- Manually install 3rd party and in-house tools for end-to-end MBSE workflows

Cost

- Leverage any existing Ansys licenses
- VM charges are based on your negotiated agreement with AWS + nominal fee
- Save on expensive hardware

Control

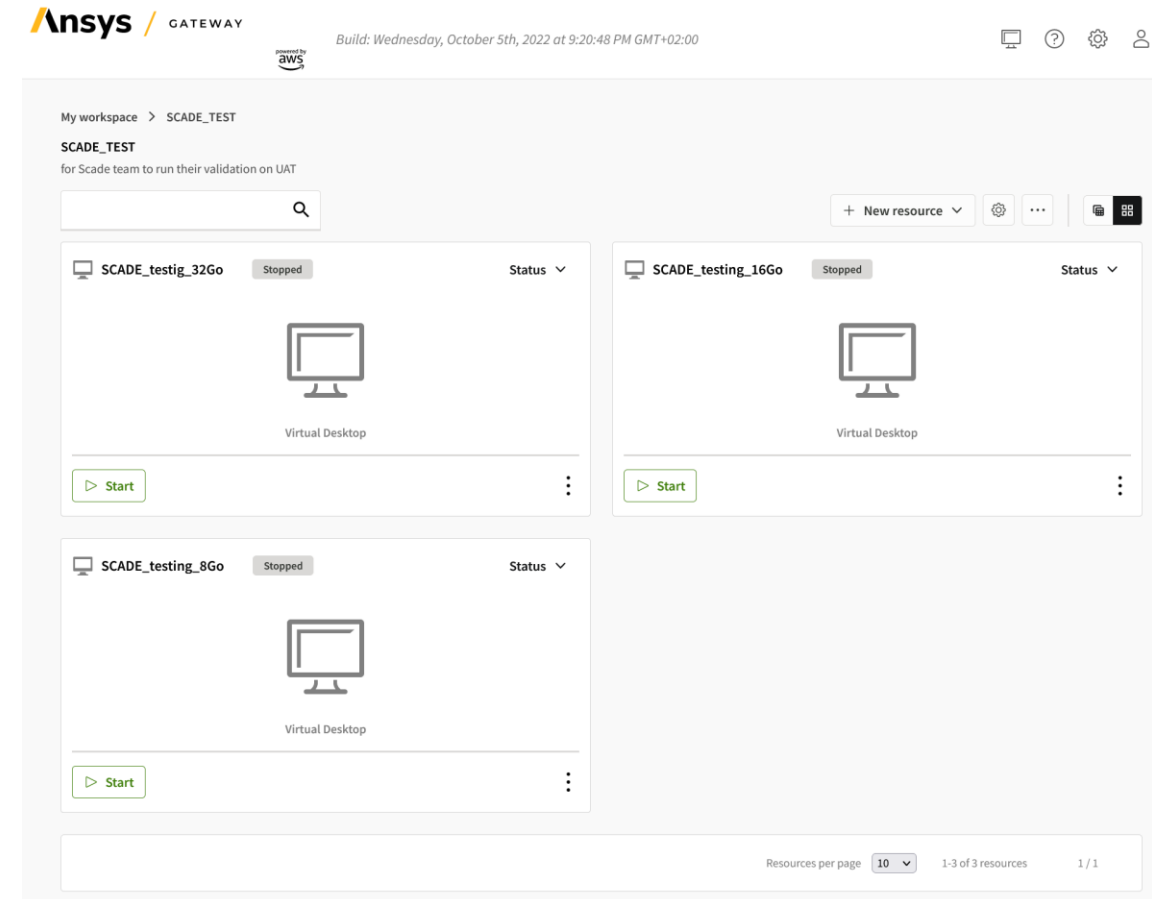
- Store VM configuration for easy, on-demand deployment by end users
- Manage access to workspaces, VMs and budgets through a single platform

SCADE on Ansys Gateway powered by AWS / In practice

Ansys-themed web application:

1. Log in with your **Ansys customer account**
2. Plug in your **Ansys license** and your **AWS account**
3. **Spawn virtual machines** (on your AWS account) with pre-installed Ansys products
4. **Remotely log** into your new machine
5. Manually install any 3rd party software
6. **Launch jobs** and collect results

No advanced technical skills required: spawn machines instantly, retrieve your files and start working





Innovations **Details for
Embedded Avionics &
Display Software**





**ARINC 661 / Revamped
Server Generator & Widget
Library**

Ansys

ARINC 661 / New Server Generator

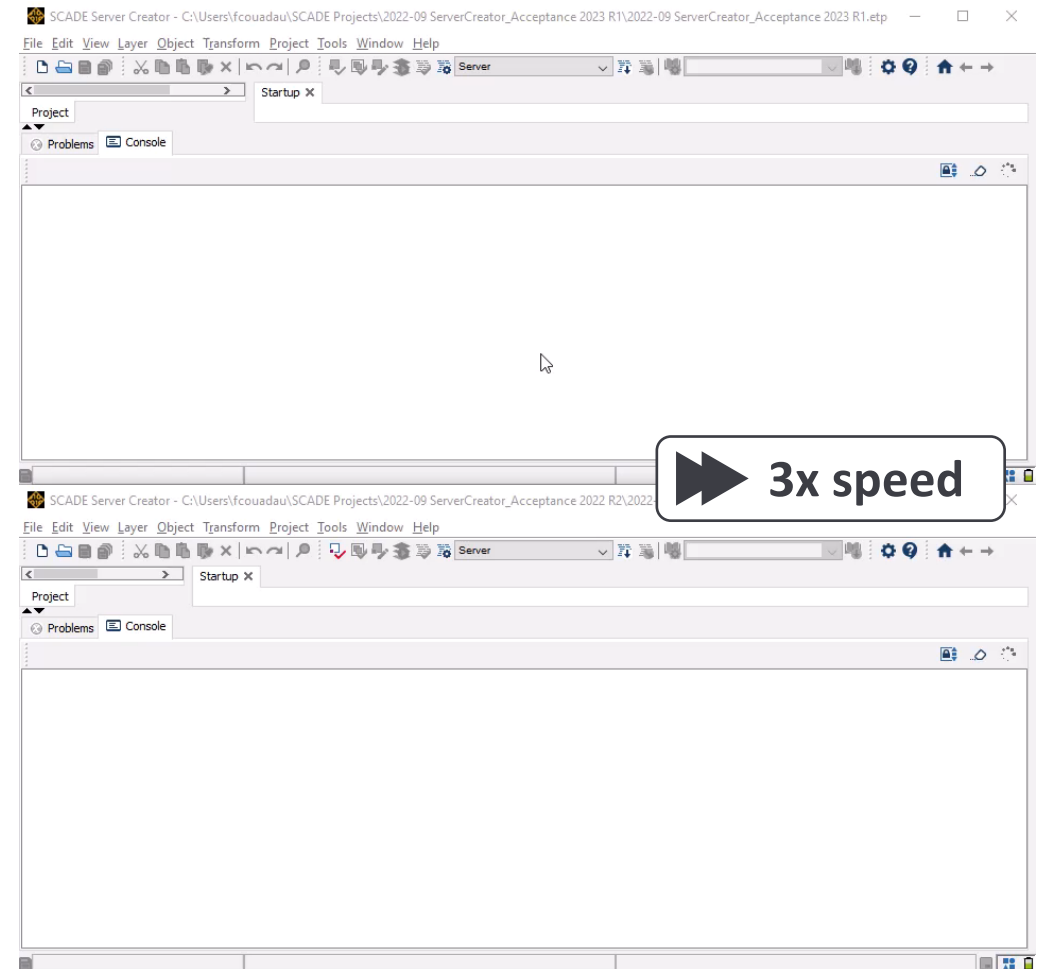
Up to 5x faster server generation to speed up development 👉

- Can be up to 7x faster with full cleanup
- Allows for faster iteration times for CDS customers

A **single Server Code Generator** replaces former SRVGEN and WWGEN code generators

~33% **less generated files** overall:

	2022/R2	2023/R1
server/a661_description	1490	1147
Server_output	2265	1344
<root>	2	4
	3757	2495



Stepping stones for customers starting from our standard Widget Lib:

1. One SRS per widget reduces DO-178C certification workload

- See <ANSYS Inc>\v231\SCADE\SCADE A661\Project Documentation\Widgets

2. Improved widget architecture

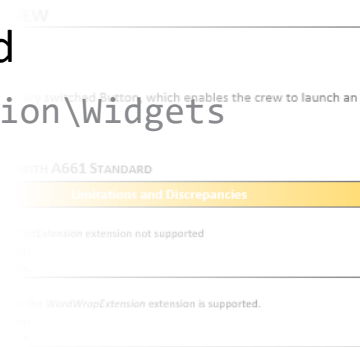
3. Widget Library Test Cases

4. Traceability documentation

Progress:

2023/R1: 60 widgets redesigned under new architecture

2023/R2: 34 remaining widgets will be released



3. INTERFACE REQUIREMENTS

The interfaces defined in this section follows rules and patterns described in chapter 3 "Rules on Widgets Requirements" of [SRS-WIDGET-COMMON].

3.1. SERVER INTERFACES

The PushButton widget implements the following widget/server interfaces:

#	Interface	Reference ¹
1	Cursor	A661WL-SRS-DEFINITION-HLR-012
2	Keyboard	A661WL-SRS-DEFINITION-HLR-013
3	Gesture	A661WL-SRS-DEFINITION-HLR-014
4	Visibility	A661WL-SRS-DEFINITION-HLR-015
5	Interactivity	A661WL-SRS-DEFINITION-HLR-016
6	BoundingBox	A661WL-SRS-DEFINITION-HLR-017
7	SensitiveArea	A661WL-SRS-DEFINITION-HLR-018
8	Focus	A661WL-SRS-DEFINITION-HLR-020
9	Selection	A661WL-SRS-DEFINITION-HLR-021
10	Error	A661WL-SRS-DEFINITION-HLR-022
11	Resizable	A661WL-SRS-DEFINITION-HLR-026
12	Animate	A661WL-SRS-DEFINITION-HLR-030

¹ The interfaces are defined in [SRS-WIDGET-COMMON].

3.2. INTERNAL ATTRIBUTES

The PushButton widget defines the following internal attributes:

#	Internal attribute	Nature	Type	Range/Value
1	GraphicalState	Variable	#enum#	EnabledUnselected (1) DisabledUnselected (2) WaitForValidUnselected (3) EnabledSelected (4)
2	WaitForValidation	State	#bool#	False (initial) True
3	Selected	Event	None	N/A
4	SelectionRequest	Event	None	N/A
5	CursorPressedState	State	#enum#	NotPressed (0) (initial) PressedIn (1) PressedOut (2)
6	ButtonPushed	Event	None	N/A
7	Highlight	Event	None	N/A

ARINC 661 / New Widget Library

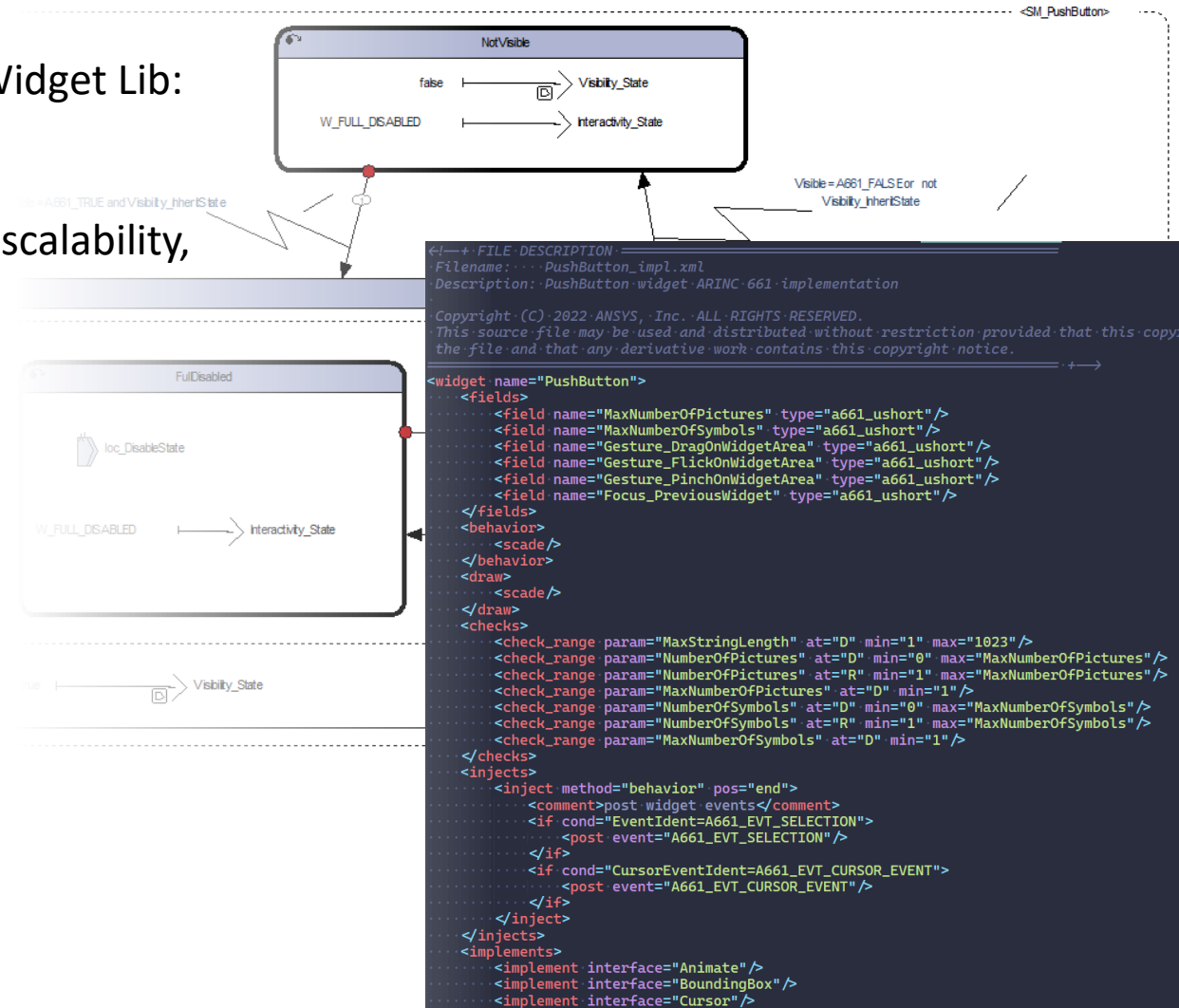
Stepping stones for customers starting from our standard Widget Lib:

1. One SRS per widget
2. **Improved widget architecture:** robustness, modularity, scalability, consistency
3. Widget Library Test Cases
4. Traceability documentation

Progress:

2023/R1: 60 widgets redesigned under new architecture

2023/R2: 34 remaining widgets will be released



ARINC 661 / New Widget Library

Stepping stones for customers starting from our standard Widget Lib:

1. One SRS per widget
2. Improved widget architecture
3. **Widget Library Test Cases** ease DO-178C certification / non-regression testing
 - See <ANSYS Inc>\v231\SCADE\SCADE A661\tests\widgets\tests
 - Uses the Test Automation Framework
 - Docstrings provide requirement traceability 👉
4. Traceability documentation

Progress:

2023/R1: 60 widgets redesigned under new architecture

2023/R2: 34 remaining widgets will be released

Excerpt from PushButton\test_hlr_046.py:

```
import logging
from typing import Tuple

from tests.common.test_harness import EMPTY_LIST_WIDGETS, Status, TestCase, check_image

DEFAULT_SGFX = "graphic_pushbutton_sgfx"
logger = logging.getLogger("tests")

def tp_pushbutton_046(
    tc: TestCase, screenshot: str, position: Tuple[int, int], size: Tuple[int, int]
) -> Status:
    """Check BoundingBox values for PushButton"""
    ...
    AUTHORS: WJC
    ...
    IMPLEMENTS: A661WL-TSTRAT-GRAPH-BOUNDING
    ...
    INPUT_MESSAGES: None
    ...
    INPUT_REQUESTS: None
    ...
    OTHER_INPUTS: See Test Case
    ...
    EXPECTED:
    ...
    * Screenshots: Display is matching BoundingBox computation
    ...
    # P1
    ...
    logger.debug(f" P1: Container Visibility.Value output is equal to True")
    ...
    # Step #01:
    ...
    logger.debug(f" Step #01: Compare captured picture with expected picture")
    ...
    obt_img = tc.outdir / screenshot
    ...
    tc.te.screenshot_to_img(obt_img, position[0], position[1], size[0], size[1])
    ...
    exp_img = tc.expdir / screenshot
    ...
    check = check_image(obt_img, exp_img, obt_img.with_suffix(".diff.png"))
    ...
    assert (
        ...
        check
        ...
    ), f"Image is NOK, {obt_img} is different with expected picture: {exp_img}."
    ...
    return Status.PASSED

class TC_PushButton_GRAPH_046_01(TestCase): # pylint: disable=invalid-name
    """Test BoundingBox values of PushButton"""
    ...
    AUTHORS: WJC
    ...
    OBJECTIVE:
    ...
    REQUIREMENTS:
    ...
    [A661WL-PUSHBUTTON-HLR-046]
    ...
    PRE-CONDITIONS: None
    ...
    INPUTS_WIDGETS:
    ...
    All PushButton under input model
    ...
    INPUT_ACTIONS: None
    ...
    INPUT_SERVER_CONFIGURATION: tests/server_config/default_conf.xml
    ...
    OTHER_INPUTS: See implementation of Test Procedure
    ...
    EXPECTED: See implementation of Test Procedure
    ...
    TEST_ACCEPTANCE_CRITERIA:
    ...
    Comparisons of expected and obtained screenshots return no difference
    ...
    No A661 error has been sent by the Server
    ...
    No Python Exception has been raised
    ...
    TEST_PROCEDURE = tp_pushbutton_046
    ...
    INPUT_DF = DEFAULT_SGFX
    ...
    TP_ARGUMENTS = {
        ...
        "screenshot": "boundingbox.png",
        ...
        "position": (0, 0),
        ...
        "size": (8_500, 1_5000),
        ...
    }
    ...
    INPUT_WIDGETS = EMPTY_LIST_WIDGETS
```

ARINC 661 / New Widget Library

Stepping stones for customers starting from our standard Widget Lib.

1. One SRS per widget
2. Improved widget architecture
3. Widget Library Test Cases
4. Traceability documentation ensures proper linkage between Specification, Design, HLR, LLR and tests

ID	Doc Reference	Release	Title
MTX-W-ACA	2023 R1	2023 R1	SCADE Server for ARINC 661 - Widgets HLR - Tests Traceability Matrix
MTX-W-ANG	2023 R1	2023 R1	SCADE Server for ARINC 661 - AnimationGroup Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-AOP	2023 R1	2023 R1	SCADE Server for ARINC 661 - AnimationOnParam Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-ANR	2023 R1	2023 R1	SCADE Server for ARINC 661 - AnimationRotation Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-ANS	2023 R1	2023 R1	SCADE Server for ARINC 661 - AnimationScale Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-ANT	2023 R1	2023 R1	SCADE Server for ARINC 661 - AnimationTranslation Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-BAC	2023 R1	2023 R1	SCADE Server for ARINC 661 - BasicContainer Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-BLC	2023 R1	2023 R1	SCADE Server for ARINC 661 - BlinkingContainer Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-BRR	2023 R1	2023 R1	SCADE Server for ARINC 661 - BroadcastReceiver Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-BUF	2023 R1	2023 R1	SCADE Server for ARINC 661 - BufferFormat Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-CHB	2023 R1	2023 R1	SCADE Server for ARINC 661 - CheckButton Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-COB	2023 R1	2023 R1	SCADE Server for ARINC 661 - ComboBox Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-CBE	2023 R1	2023 R1	SCADE Server for ARINC 661 - ComboBoxEdit Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-CON	2023 R1	2023 R1	SCADE Server for ARINC 661 - Connector Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-CUO	2023 R1	2023 R1	SCADE Server for ARINC 661 - CursorOver Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-CPO	2023 R1	2023 R1	SCADE Server for ARINC 661 - CursorPosOverlay Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-CUR	2023 R1	2023 R1	SCADE Server for ARINC 661 - CursorRef Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-DAC	2023 R1	2023 R1	SCADE Server for ARINC 661 - DataConnector Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-DSF	2023 R1	2023 R1	SCADE Server for ARINC 661 - DataScalingFR180 Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-DSL	2023 R1	2023 R1	SCADE Server for ARINC 661 - DataScalingLong Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-DAS	2023 R1	2023 R1	SCADE Server for ARINC 661 - DataScalingULong Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-EMA	2023 R1	2023 R1	SCADE Server for ARINC 661 - EditBoxMasked Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-EMU	2023 R1	2023 R1	SCADE Server for ARINC 661 - EditBoxMultiLine Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-EBV	2023 R1	2023 R1	SCADE Server for ARINC 661 - EditBoxNumeric Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-EBC	2023 R1	2023 R1	SCADE Server for ARINC 661 - EditBoxNumericBCD Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-EBT	2023 R1	2023 R1	SCADE Server for ARINC 661 - EditBoxText Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-EVH	2023 R1	2023 R1	SCADE Server for ARINC 661 - EventHandler Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-EXS	2023 R1	2023 R1	SCADE Server for ARINC 661 - ExternalSource Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-FOI	2023 R1	2023 R1	SCADE Server for ARINC 661 - FocusIn Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-FOL	2023 R1	2023 R1	SCADE Server for ARINC 661 - FocusLink Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-FOO	2023 R1	2023 R1	SCADE Server for ARINC 661 - FocusOut Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-GEA	2023 R1	2023 R1	SCADE Server for ARINC 661 - GestureArea Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-GPA	2023 R1	2023 R1	SCADE Server for ARINC 661 - GpArcCircle Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-GPE	2023 R1	2023 R1	SCADE Server for ARINC 661 - GpArcEllipse Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-GPC	2023 R1	2023 R1	SCADE Server for ARINC 661 - GpCrown Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-GPL	2023 R1	2023 R1	SCADE Server for ARINC 661 - GpLine Widget High Level Requirements to Low Level Requirements Traceability Matrix
MTX-W-GLP	2023 R1	2023 R1	SCADE Server for ARINC 661 - GpLinePolar Widget High Level Requirements to Low Level Requirements Traceability Matrix

Progress:

2023/R1: 60 widgets redesigned under new architecture

2023/R2: 34 remaining widgets will be released

/ ARINC 661 / Other Server Creator Improvements

Server port revamping eases end-user customization and DO-178C certification

- Here again, this is a stepping stone for CDS customers starting from our standard Server Creator / Widget Lib
- We provide a hand coded A661 server porting layer for Windows
- The code now has improved modularity & structure
- Performance monitoring code (on Windows host) is now included
- No FACE CDS integration yet

ARINC 661 / Revamped UA Adaptor

Ansys

ARINC 661 UA Adaptor / Revamped Code Generation

2022 R2 / (N + 2) .c files

- One file per layer (encode)
- Private utilities
- Glue code (encode entry point + decode)

Information	Generated Files	GENFIL	SCADE UA Adaptor Generated files
→			UA_FMS_UA_1.h
→			UA_FMS_UA_1.c
→			UA_FMS_UA_1_constants.h
→			UA_FMS_UA_1_private.h
→			UA_FMS_UA_1_private.c
→			UA_FMS_UA_1_LAY_1.c

2023 R1 / 2 .c files

- Decode
- Encode

Information	Generated Files	GENFIL	SCADE UA Adaptor Generated files
→			UA_FMS_UA_1.h
→			UA_FMS_UA_1_constants.h
→			UA_FMS_UA_1_decode.c
→			UA_FMS_UA_1_encode.c

ARINC 661 UA Adaptor / Revamped Code Generation

2022 R2

Single function handling

A661 decoding + event triage + processing



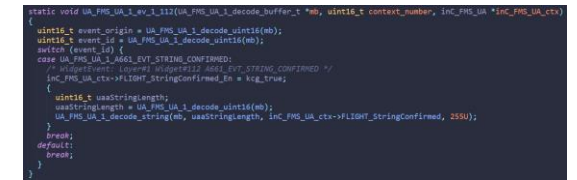
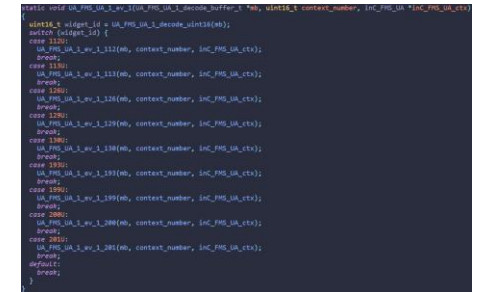
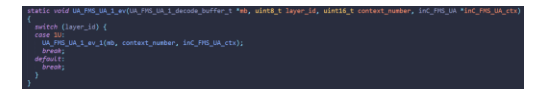
210 LOC
13 indentation levels

2023 R1

Modular code with successive function calls:

- Decode A661 notification
- Decode widget events
- Triage events
- Process event (one function per event)

Generally < 30 LOC per function
Max. 3 indentation levels



ARINC 661 UA Adaptor / Revamped Code Generation

Changelog

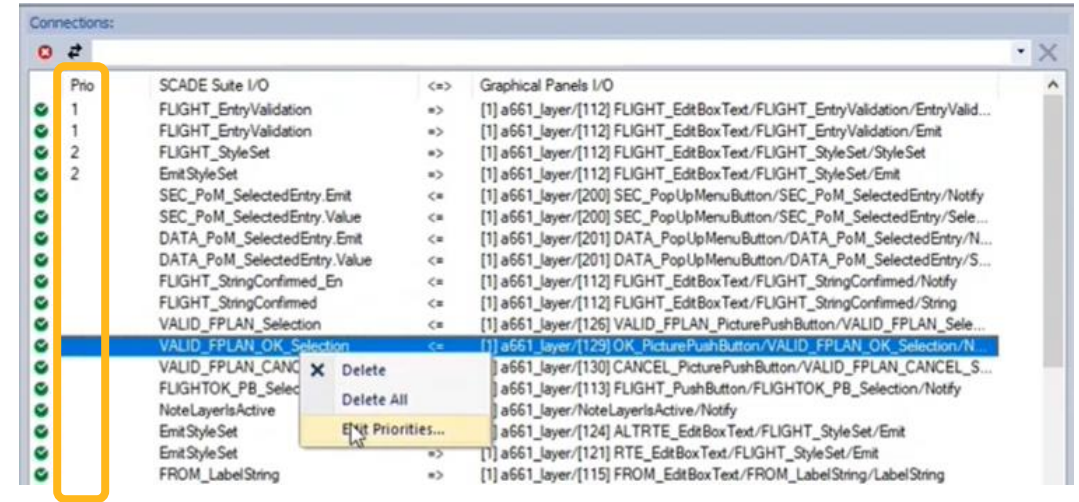
- Add emit priority support
 - Parse attribute `priority` on `A661Data/Emit` elements.
 - Build first emit commands then other valued field commands to ensure priority is well recorded in commands.
- Improve generated code
 - Improve stability of generated code following model changes: no more code factoring per widget.
 - Renaming for clarity: "receive" → "decode", "send" → "encode".
 - Remove useless `void* uaa_ctxt` last parameter.
 - Shorten function names to be distinct in first 30 characters.
 - Compute max output buffer size depending of existing connections.
 - Compute max input buffer size as the sum of max events sizes of all layers, widgets, extensions in the DF.
 - Add size to encode function, so user can pass a different size than the `MAX_SIZE_OUTPUT_BUFFER`
 - Decode consecutive layer blocks in the input buffer. The user must pass the correct `bufsize` to limit the number of read blocks.
 - Compute message size before switch block to avoid block overflow.
 - Add option `-datagroup` to encode/decode DataGroup header: `0x4753 <uint16 pad> <uint32 length>`
 - Avoid `SetParameter` command size overflow, forcing new command if necessary.
 - Use memory buffer structure instead of pointer arithmetic.

ARINC 661 UA Adaptor / Revamped Code Generation

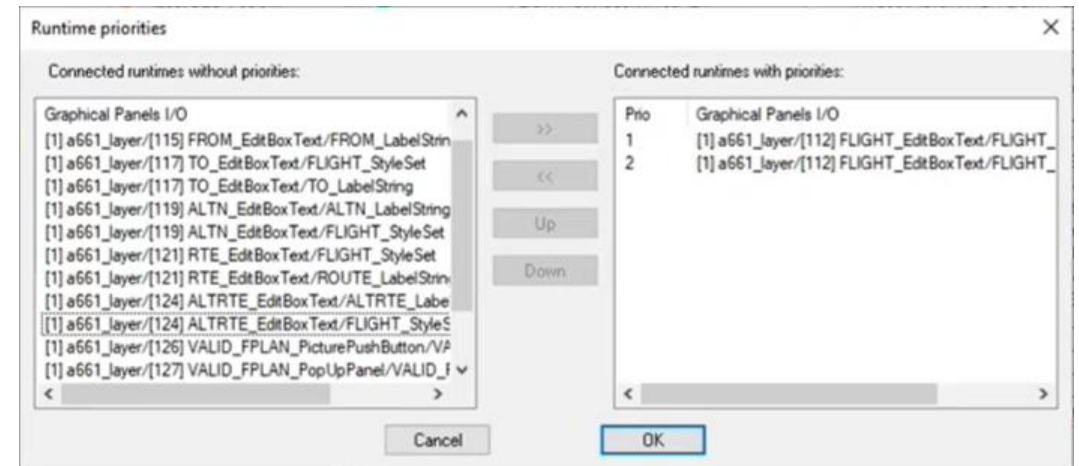
File enhancements

- Support new XML A661 description file construct (**DFGEN 6.8**)
- New **versioning property** in the .sdy file to control model releases w.r.t. tool releases
- Automated **file upgrade** from SCADE 2022 R2 (manual upgrade needed for map items)

- New column **Prio** in Connections view
 - Allows sorting messages according to their priority



- New pop-up dialog from Connections view
 - Only messages with connected 'emit' are listed
 - Select any message to set a priority
 - Order selected messages to set priority (higher = 1)
 - Unselected messages all have equal, lowest priority (same default behavior as previous SCADE versions)
 - Deleting a runtime message
 - Does not change the priority of other runtime messages
 - Automated re-indexing to avoid priority index gaps

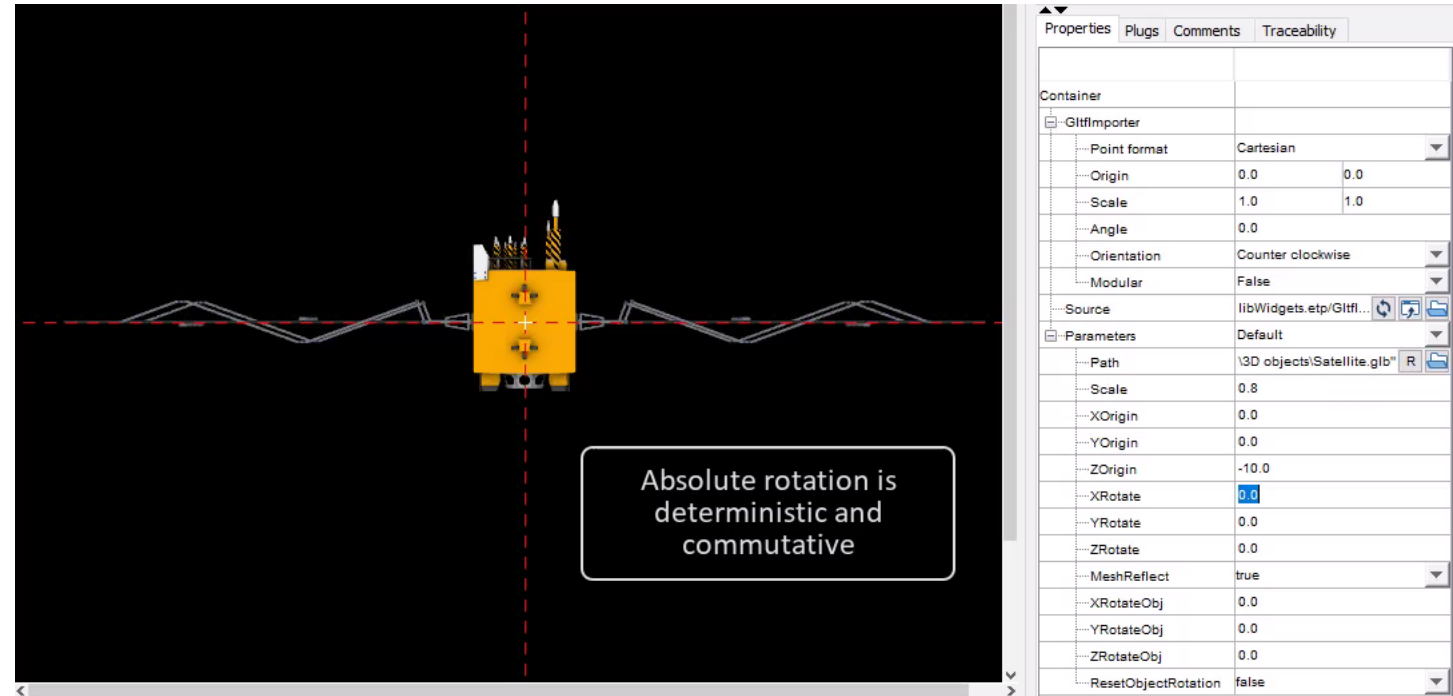


Misc. SCADE Display improvements

2023 R1 SCADE Display / 3D object local rotation reset (CR26633)

2 rotation modes are available:

- Fixed **world origin** reference frame
 - 👍 Rotation order doesn't matter
 - 🗨️ Requires sines/cosines to compute rotation
- Moving **object-local** reference frame
 - 👍 Conveniently matches pitch/roll/yaw
 - 🗨️ **Rotation order matters**
 - NEW Rotation reset feature for convenience



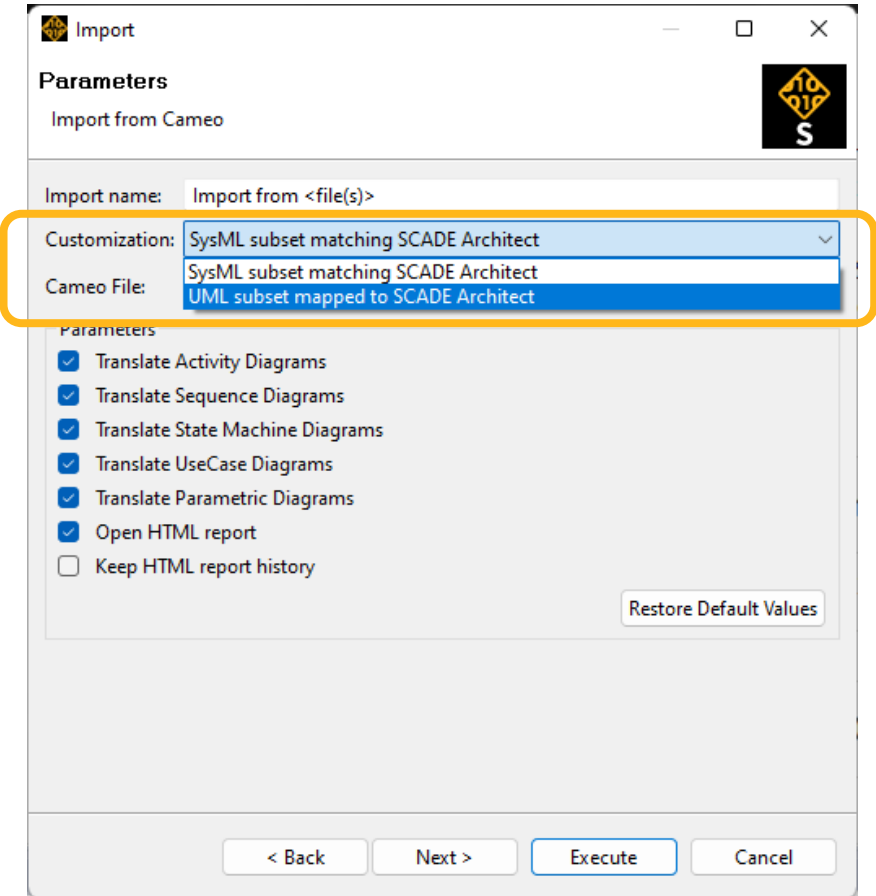
Innovations in SCADE Architect 2023 R1

Detailed Presentation



SCADE Architect 2023 R1

- Extended SysML importer to facilitate digital continuity from Cameo or EA to SCADE (1/2)
 - Customizable importer
 - Template-based user mapping customization (Java and QVT expertise)
 - Allows importing custom Stereotypes values into SCADE Architect configured properties
 - Product release including 2 sets of import rules
 - “SysML matching SCADE Architect subset”
 - Imports only elements matching SCADE Architect SysML subset (Same behavior as previous SCADE release)
 - “UML mapped to SCADE Architect subset”
 - Imports all SysML/UML element matching “approximately” to SCADE Architect objects
 - Example
 - UML standard port mapped to SCADE Architect Flowport, direction set to inout



SCADE Architect 2023 R1

- Extended SysML importer to facilitate digital continuity from Cameo or EA to SCADE (2/2)
 - Comprehensive HTML import reporting pointing
 - All imported / not imported model elements
 - Per package / per object kind

Import settings : Import from CruiseControlSystem.mdzip
 Customization : SysML subset matching SCADE Architect
 Wednesday November 09 2022 11:45:28

Synthesis by Package

Name	From settings		Model elements		Total
	Selectable	Selected	Imported	Not Imported	
CruiseControlSystem	42 (4)	42 (4)	397 (5)	18 (6)	415 (11)
Functional	16 (16)	16 (16)	144 (144)	0 (0)	144 (144)
Architecture	22 (22)	22 (22)	225 (225)	10 (10)	235 (235)
FunctionsAllocation	0 (0)	0 (0)	20 (20)	1 (1)	21 (21)
InterfacesAllocation	0 (0)	0 (0)	3 (3)	1 (1)	4 (4)

Object Kind for package CruiseControlSystem

Name	From settings		Model elements		Total
	Selectable	Selected	Imported	Not Imported	
All	4	4	5	6	11
Abstraction	0 (0)	0 (0)	23 (0)	0 (0)	23 (0)
Association	0 (0)	0 (0)	26 (0)	0 (0)	26 (0)
Class	27 (0)	27 (0)	27 (0)	0 (0)	27 (0)
Connector	0 (0)	0 (0)	50 (0)	0 (0)	50 (0)
DataType	0 (0)	0 (0)	2 (0)	0 (0)	2 (0)
Diagram	11 (0)	11 (0)	11 (0)	2 (0)	13 (0)
InstanceSpecification	0 (0)	0 (0)	6 (0)	0 (0)	6 (0)
Interface	0 (0)	0 (0)	3 (0)	0 (0)	3 (0)
Package	4 (4)	4 (4)	4 (4)	0 (0)	4 (4)
Port	0 (0)	0 (0)	65 (0)	0 (0)	65 (0)
Property	0 (0)	0 (0)	66 (0)	0 (0)	66 (0)

/ SysML models import

- Upgrade to Sparx Systems Enterprise Architect new release

Dassault Systèmes Cameo 2021x R2

CAMEO
SYSTEMS MODELER™



Sparx Systems Enterprise Architect 16.0

ENTERPRISE
ARCHITECT



IBM Rhapsody 8.2

Rhapsody®



Ansys
SCADE ARCHITECT

- Imports SysML elements and graphical diagrams:

- BDD
- IBD
- Parametric

Cameo Systems Modeler
Enterprise Architect
Rhapsody

- State machine
- Activity
- Sequence
- Use case

Cameo Systems Modeler

Ansys

Innovations in SCADE Avionics Package 2023 R1

Detailed Presentation



SCADE Solution for FACE

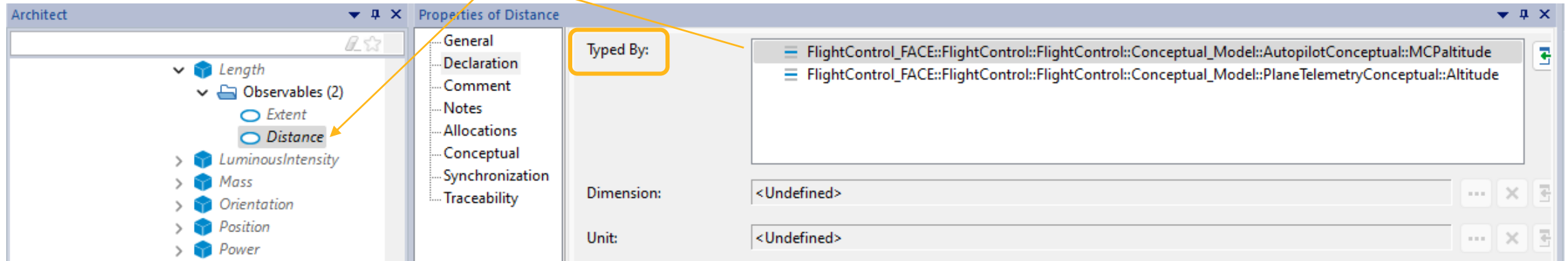
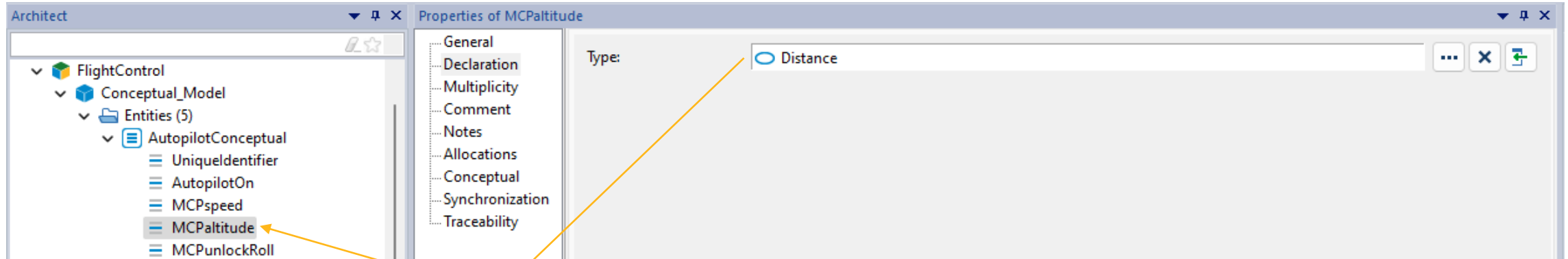
- FACE model navigation (1/2)
 - New « Realized by » and « Typed by » properties

The screenshot shows the 'Architect' software interface. On the left, a tree view under 'Logical_Model' shows 'Entities (5)' including 'AutopilotLogical', 'AutopilotOn', 'MCPspeed', 'MCPaltitude', and 'MCPunlockRoll'. The 'Properties of MCPaltitude' dialog box is open, showing a 'Realized by' property with the value 'FlightControl_FACE::FlightControl::FlightControl::Platform_Model::AutopilotPlatform::MCPaltitude'. The 'Lower bound' and 'Upper bound' are both set to 1.

The screenshot shows the 'Architect' software interface. On the left, a tree view under 'FlightControl' shows 'Conceptual_Model' with 'Entities (5)' including 'AutopilotConceptual', 'Uniquelidentifier', 'AutopilotOn', 'MCPspeed', 'MCPaltitude', and 'MCPunlockRoll'. The 'Properties of MCPaltitude' dialog box is open, showing a 'Realized by' property with the value 'FlightControl_FACE::FlightControl::FlightControl::Logical_Model::AutopilotLogical::MCPaltitude'. The 'Lower bound' and 'Upper bound' are both set to 1. The 'Realized by' property name is highlighted with a yellow box.

SCADE Solution for FACE

- FACE model navigation (2/2)
 - New « Realized by » and « Typed by » properties



SCADE Solution for FACE

- FACE wrapper enhancement

- Both C and C++ generations
- UoP code now independent from OSS
 - OSS integration part now generated in dedicated directory to ease customization
- Better usage of injectable interface API
 - Ease TSS code integration with third parties
- Upgrade CTS versions support
 - Versions 3.1.2, 3.0.3, and 2.1.3

```
// FACE_Factory_Functions.cpp - FACE factory functions
// Injectable Factory function
FACE::TSS::Base_Injectable::Injectable* Get_FACE_TSS_Base_Injectable_Injectable() {
    FACE::TSS::Base_Injectable::Injectable* FACE_TSS_Base_Injectable_obj =
    new FACE::TSS::Base_Injectable::Injectable_impl();
    FACE::TSS::Base* FACE_TSS_Base_obj = Get_FACE_TSS_Base();
    FACE_TSS_Base_Injectable_obj->Set_Reference("FACE::TSS::Base", *FACE_TSS_Base_obj, id++, status);
}
```

```
// FACE_TSS_Base_Injectable.hpp - FACE Base Injectable

#ifndef _FACE_TSS_BASE_INJECTABLE_IMPL_HPP_
#define _FACE_TSS_BASE_INJECTABLE_IMPL_HPP_
#include <FACE/TSS/Base_Injectable.hpp>

namespace FACE {
    namespace TSS {
        namespace Base_Injectable {
            class Injectable_impl : public FACE::TSS::Base_Injectable::Injectable {
            public:
                Injectable_impl() {}
                ~Injectable_impl() {}
                Injectable_impl& operator=(const Injectable_impl&) { return *this; }
                Injectable_impl(const Injectable_impl&) {}
                void Set_Reference(
                    /* in */ const FACE::STRING_TYPE& interface_name,
                    /* in */ FACE::TSS::Base& interface_reference,
                    /* in */ FACE::GUID_TYPE id,
                    /* out */ FACE::RETURN_CODE_TYPE::Value& return_code) {
                    m_FACE_TSS_Base = &interface_reference;
                    return_code = FACE::RETURN_CODE_TYPE::NO_ERROR;
                }
                FACE::TSS::Base* Get_Reference() { return m_FACE_TSS_Base; }
            private:
                FACE::TSS::Base* m_FACE_TSS_Base;
            }; // class Injectable_impl
        };
    };
};
#endif // _FACE_TSS_BASE_INJECTABLE_IMPL_HPP_
```

```
// FACE_UoP_Model_FlightControl_PSS.cpp - FACE transport

// Get TSS objects from injectable API
FACE_TSS_Base_Injectable_obj =
    Get_FACE_TSS_Base_Injectable_Injectable();
FACE::TSS::Base* FACE_TSS_Base_obj =
    static_cast<FACE::TSS::Base_Injectable::Injectable_impl*>{
        FACE_TSS_Base_Injectable_obj->Get_Reference()};

// Call TSS Create_Connection API
FACE_TSS_Base_obj->Create_Connection(
    /* in */ FlightControlTelemetrySubscriber_connection_name,
    /* in */ 1000000,
    /* out */ FlightControlTelemetrySubscriber_connection_id,
    /* out */ max_message_size,
    /* out */ return_code);

// Call TSS Destroy_Connection API
FACE_TSS_Base_obj->Destroy_Connection(
    /* in */ FlightControlTargetSubscriber_connection_id,
    /* out */ return_code);
```

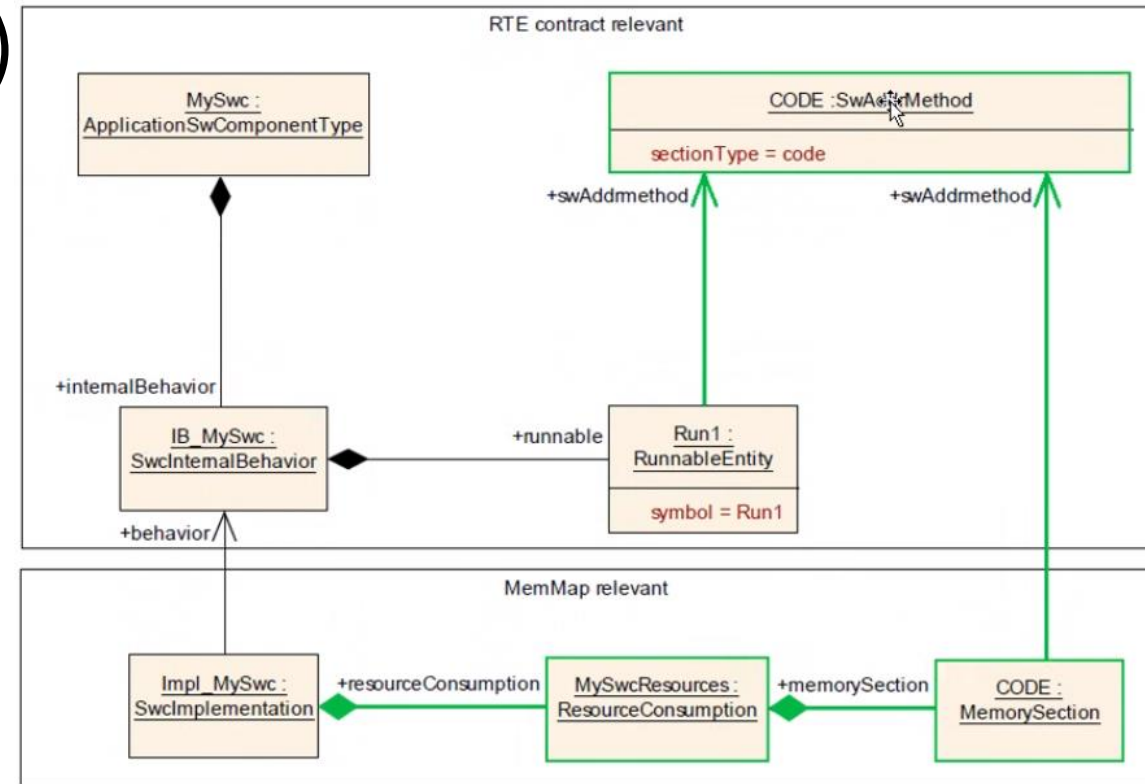
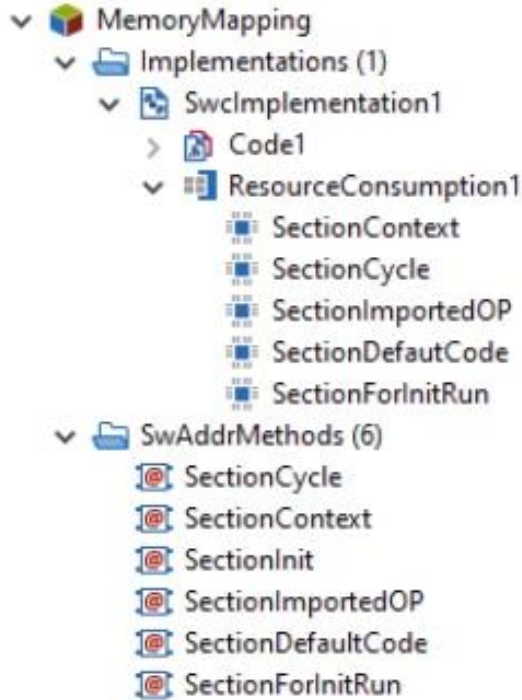


Innovations **Details** for Embedded Automotive Software



AUTOSAR – Memory Mapping (1/2)

- Memory sections for code, constants, and variables
 - Modeling supported through AUTOSAR objects
ResourcesConsumption, MemorySection and SwAddrMethod



Properties of AdaptiveCruiseControl

General Declaration Implementation Comment Appearance Notes Allocations AUTOSAR Synchronization Traceability

Applied notes:

- ScInitRunnableMemoryMapping (from autosar_architect)
 - codeSwAddrMethod: SwAddrMethod [1] = SectionInit

Properties of HighLevelController

General Declaration Implementation Comment Appearance Notes Allocations AUTOSAR Synchronization Traceability

Applied notes:

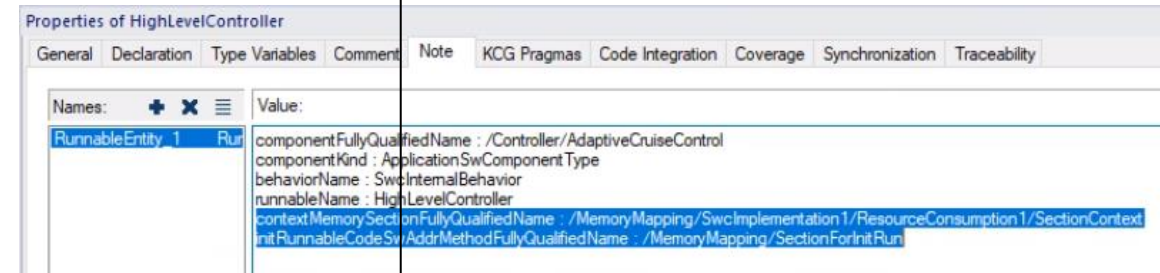
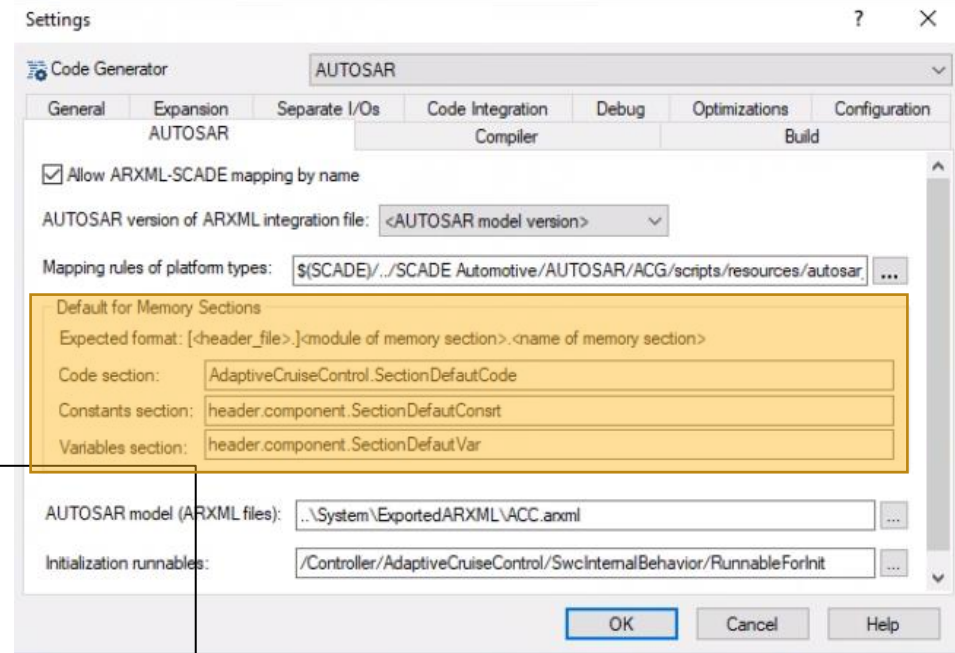
- AdditionalMemorySections (from autosar_architect)
 - contextMemorySection: MemorySection [1] = SectionContext



AUTOSAR – Memory Mapping (2/2)

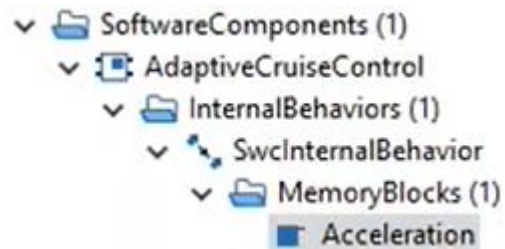
- Memory sections for code, constants, and variables
 - Synchronization with SCADE Suite supported through dedicated annotations
 - Code generator settings extended with default sections definition
 - Code generation introduces C macro to start/end the memory sections

```
/* *****  
 * RunnableEntity: /Controller/AdaptiveCruiseControl/SwcInternalBehavior/HighLevelController  
 * Scade Operator: Controller::HighLevelController/  
 * *****  
  
#define AdaptiveCruiseControl_START_SEC_SectionContext  
#include "AdaptiveCruiseControl_MemMap.h"  
outC_HighLevelController_Controller AdaptiveCruiseControl_HighLevelController_ScContext;  
#define AdaptiveCruiseControl_STOP_SEC_SectionContext  
#include "AdaptiveCruiseControl_MemMap.h"  
  
#define AdaptiveCruiseControl_START_SEC_SectionCycle  
#include "AdaptiveCruiseControl_MemMap.h"  
void HighLevelController(void){  
    /* Call cycle function */  
    HighLevelController_Controller(  
        /* DataReadAccess: VehSpeed --> Scade Input: VehSpeed */ Rte_IRead_HighLevelController_VehSpeed_speed(),  
<...>  
        /* Scade Output: CruiseSpeed --> DataWriteAccess: CruiseSpeed */ Rte_IWriteRef_HighLevelController_CruiseSpeed_speed(),  
<...>  
        &AdaptiveCruiseControl_HighLevelController_ScContext);  
}  
#define AdaptiveCruiseControl_STOP_SEC_SectionCycle  
#include "AdaptiveCruiseControl_MemMap.h"
```

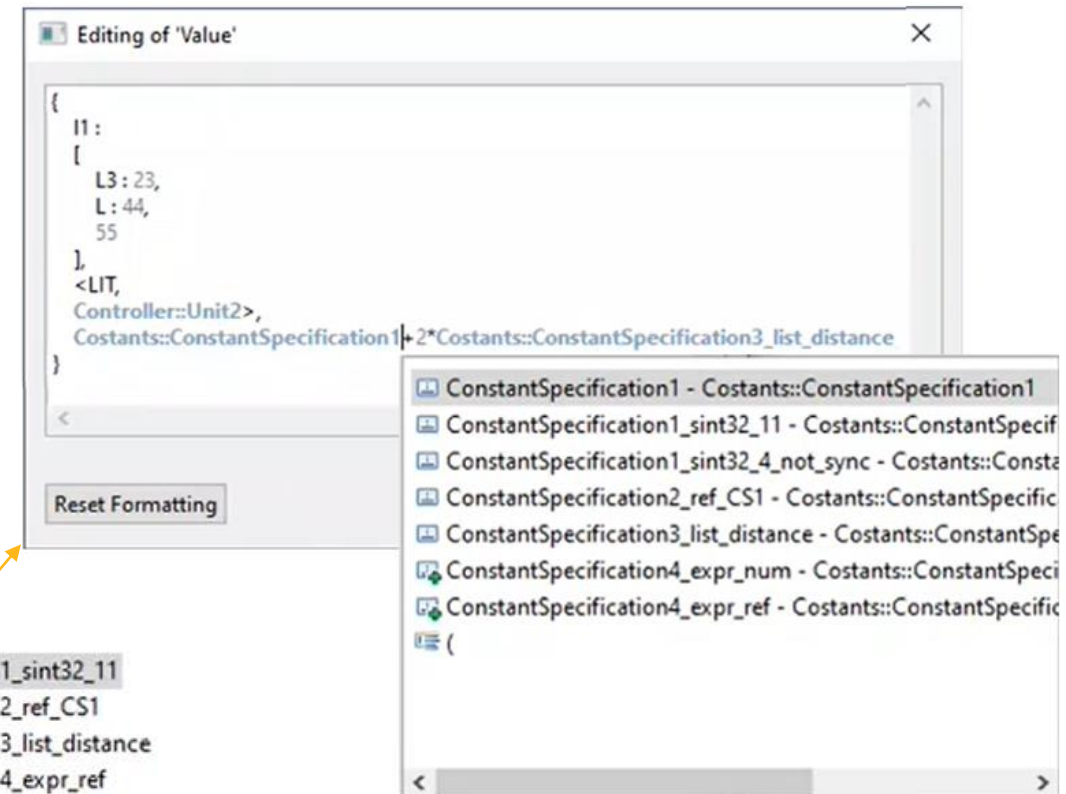
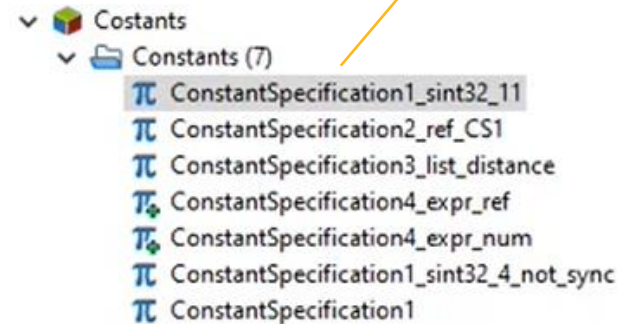


- Extended support of Value Specification

- Syntactic editor with auto-completion
- Support of expressions
 - Arithmetic expressions, record, arrays
 - Including references to constants
- Use cases
 - Constants definitions
 - All AUTOSAR objects “init” attribute
 - Support of Application Value Specification
 - Value specified between ‘<’, ‘>’ with Unit



Init value <23.0 + 2.4, Controller::Unit2>



Application Value Specifications (2/2)

- Extended support of Value Specification
 - ARXML export
 - Expressions evaluation
 - Synchronization with SCAD Suite
 - Option to keep constants with their expressions
 - Option to create constants for Init attributes value with their expressions

Synchronization

Parameters

Synchronization between SCAD Architect and SCAD Suite projects

Synchronization name: Synchronization with ACC_Suite.etp

Customization: AUTOSAR

Project file: ./Design/ACC_Suite.etp

Parameters

Platform Types Mapping Rules: \$(SCADE)/lib/autosar_platform_types_map_rules

Create Getters and Setters for Pointer Types

Synchronize ConstantSpecifications into Scade Constants

Synchronize InitValues into Scade Constants

Synchronize Implicit VariableAccesses into Scade imported operators

Restore Default Values

SoftwareComponents (1)

AdaptiveCruiseControl

InternalBehaviors (1)

SwcInternalBehavior

MemoryBlocks (1)

Acceleration

Init value <23.0 + 2.4, Controller::Unit2>



Constant	Type	Value	Comments
SwcInternalBehavi...	tAcceleration	23.0 + 2.4	DataPrototype InitValue translated to AUTOSAR element



Innovations **Details** for Embedded Control Software





Refactoring Capability Improvements

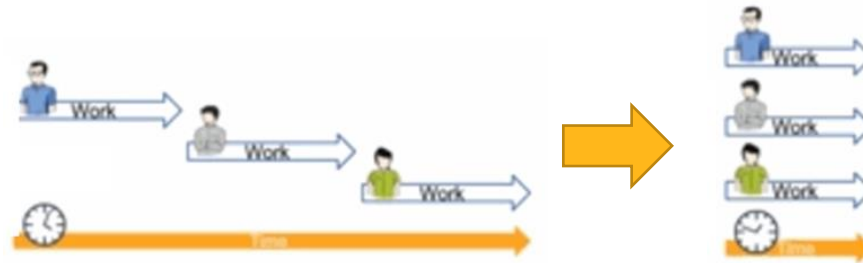
Ansys

Better support collaboration and teamwork with SCADE Suite

- Efficient collaborative work is a key driver for project success

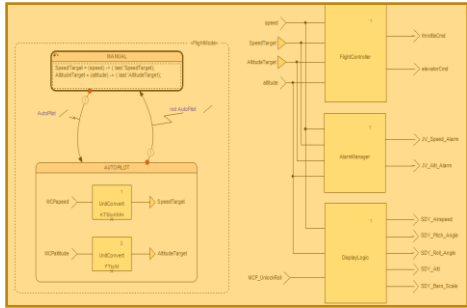


- It is one of the best way to save time and improve Go to market challenges



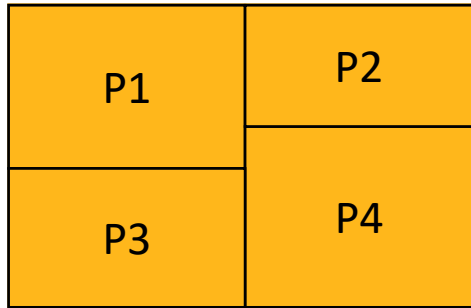
These slides shows a SCADE 2023 R1 collaborative workflow

Collaboration principle – Step by step

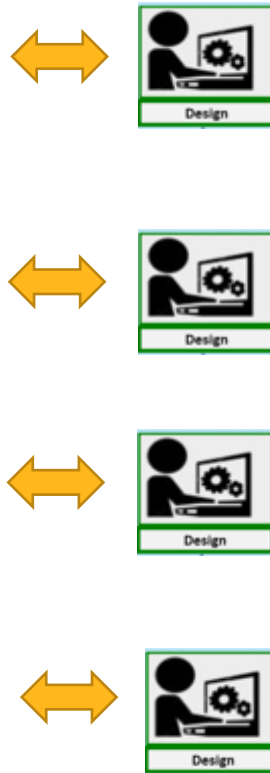


SCADE Model

Collaboration principle – Step 1 (split for collaboration)



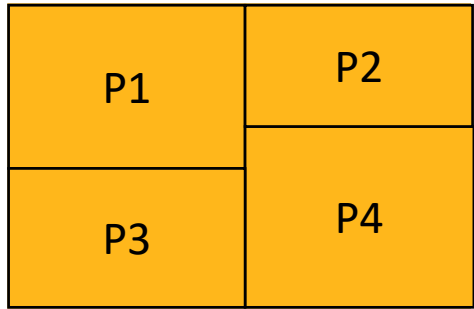
SCADA Model



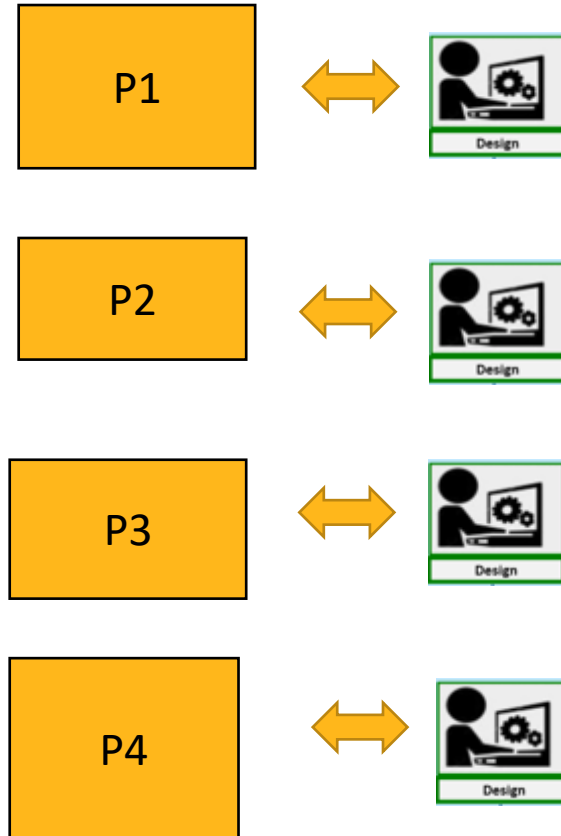
- Split model in sub-projects or Libraries. One sub-project (Library per stakeholder)
- New refactoring capabilities added in 2023 R1 allows building sub-projects or libraries



Collaboration principle – Step 2 (Concurrent work on *parts*)

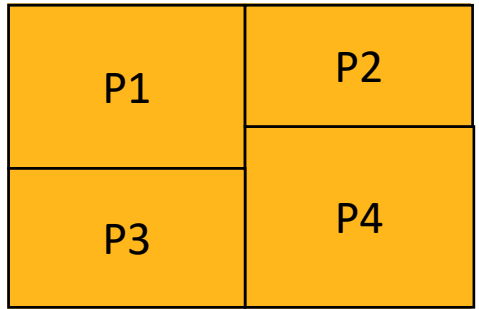


SCADE Model

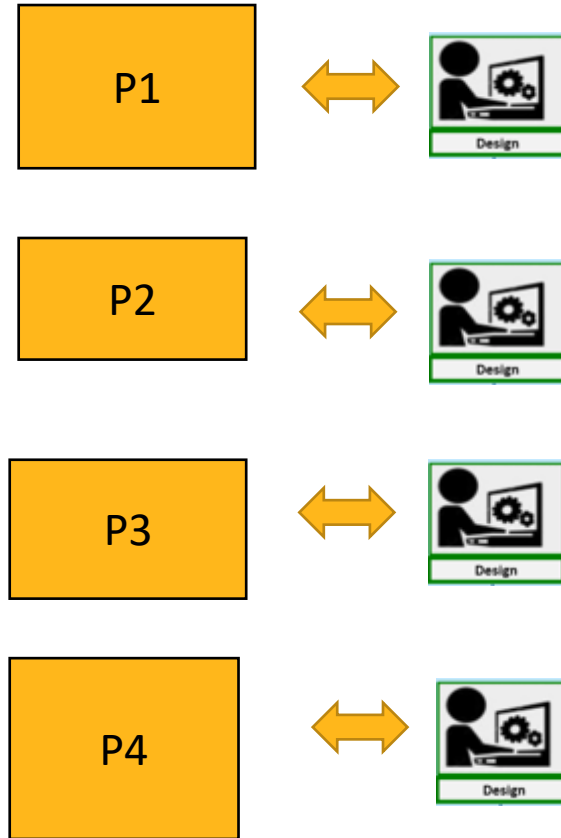


- Each project stakeholder will work on one or several *Sub-projects*

Collaboration principle – Step 3 (Integration)



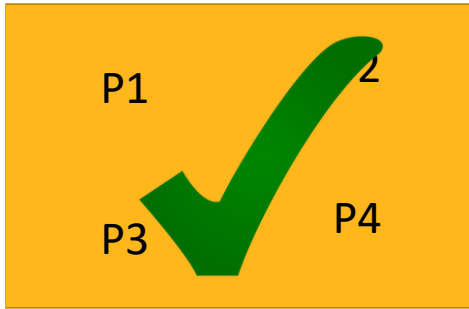
SCADA Model
(integration model)



- Each updated sub-project can be integrated in the whole SCADA Model (integration Model)



Collaboration principle – Step 4 (manage conflicts if any)



SCADE Model
(integration model)

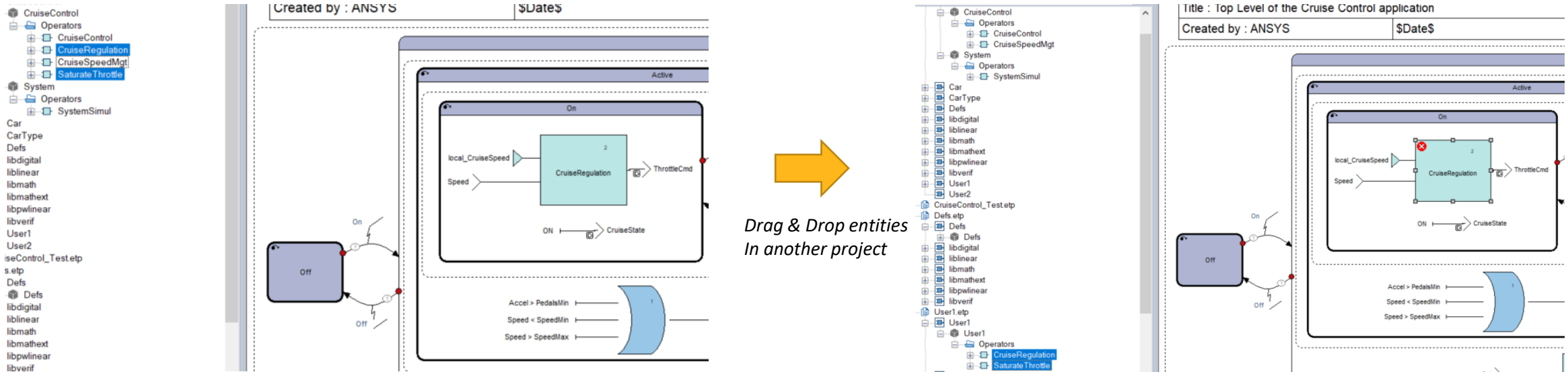


- Quick fix is available to resolve inconsistencies



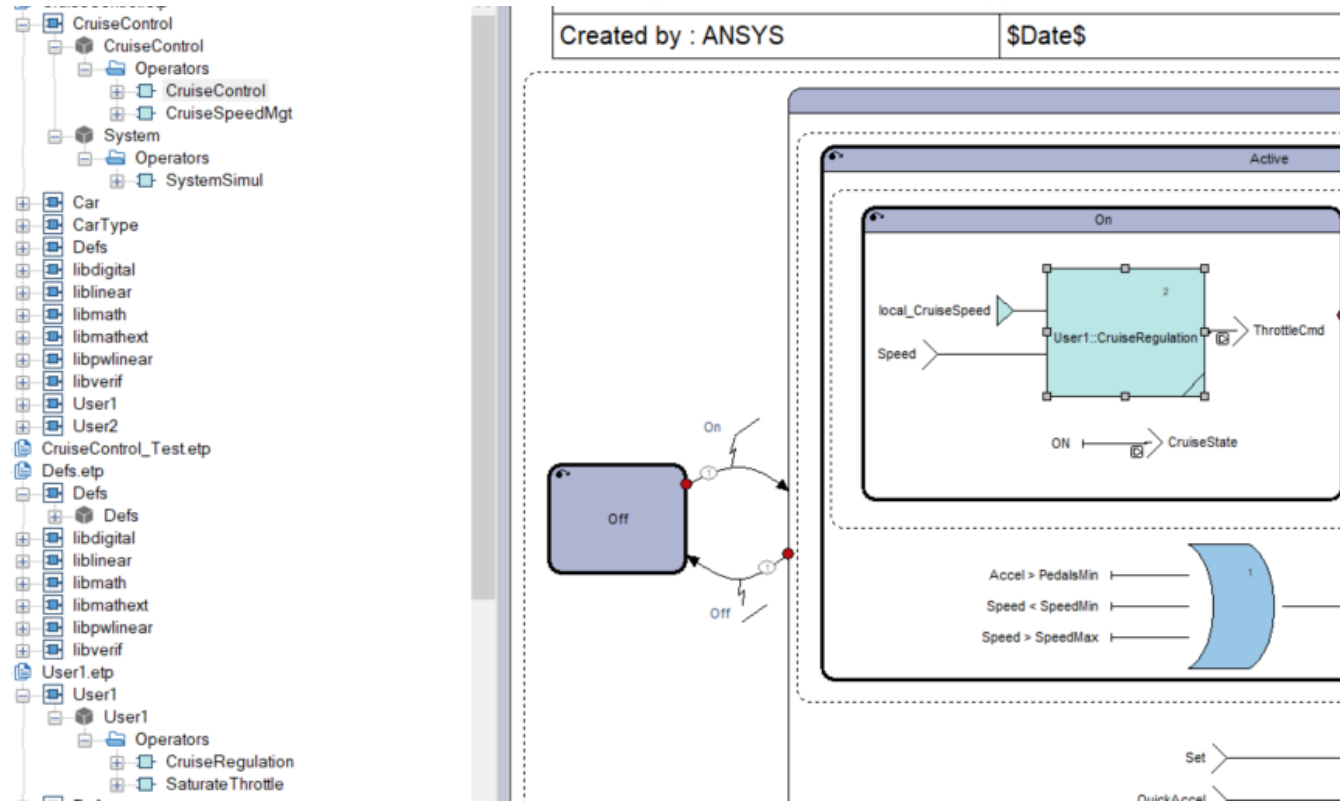
Move entities from a project to a library or another project

- Pre-requisite : All projects (or libraries) should be editable = in a same workspace



Move entities from a project to a library or another project

- Apply Quick fix → Done



Coverage Assistance Improvements

Ansys

Coverage Assistance – Improved GUI integration

```
## Empty lines and lines starting with # are ignored
## @criterion@ is expanded to:
## - MC mode: option from active Test configuration
## - SCADE mode: '-criterion OMCDC'
@criterion@

## instrumentation options
#-instr P::N,P::N2
#-noinstr P::N,P::N2

## @expansion@ is expanded to:
## - MC mode: options from Suite configuration referenced by active Test configuration
## - SCADE mode: options from active Suite configuration
@expansion@

## @observation@ is expanded to:
## - MC mode: options from Suite configuration referenced by active Test configuration
## - SCADE mode: options from active Suite configuration
@observation@

## @root@ is expanded to:
## - MC mode: options from Suite configuration referenced by active Test configuration
## - SCADE mode: '-root <selected operator>'
@root@

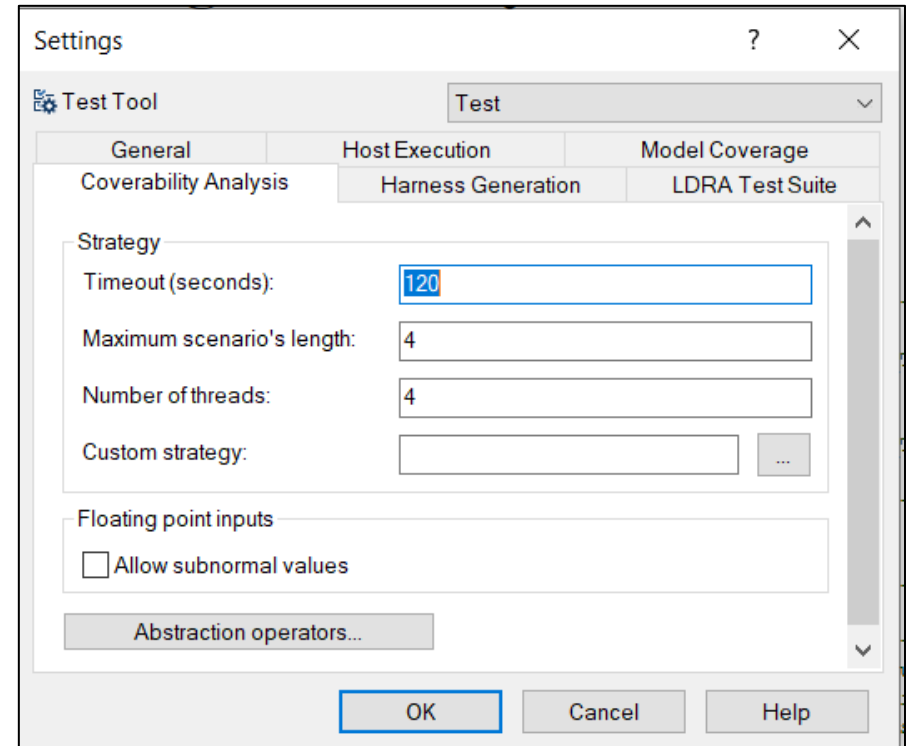
## timeout in seconds
-timeout 20

## maximum depth for BMC (bounded model checking)
-bmc_depth 7

## number of threads used by PSL
## * strictly negative value: PSL is run using a single core,
## * 0: PSL is run with the maximum of cores,
## * strictly positive value: PSL is run with the specified value, this value
## being bounded by the maximum number of cores on host
-threads 4

## custom strategy
#-strategy " -solver current {inc=true} -bmc conj start 0 stop 5 -solver 5 -eqLemmas ind fu
```

Settings in 2022 R2



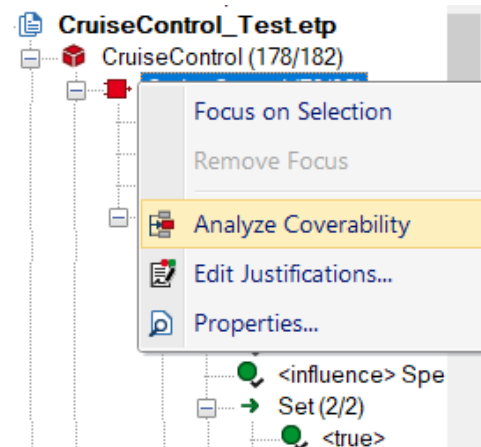
Settings in 2023 R1

Coverage Assistance – Improved GUI integration

- Coverability Analysis integrated in SCADE Test commands like others



- Available within contextual menu



Coverage Assistance – Support of Abstraction Operators

- Abstraction operators can be applied during Coverability analysis
- This abstraction is reported in the HTML Report

3.1. Assumptions

Object path	kind
FlightControl::FCU/MCPspeed	finite and not subnormal floating-point constraint
FlightControl::FCU/MCPaltitude	finite and not subnormal floating-point constraint
FlightControl::FCU/altitude	finite and not subnormal floating-point constraint
FlightControl::FCU/speed	finite and not subnormal floating-point constraint
FlightControl::FCU/A3=	assume constraint
FlightControl::FCU/A2=	assume constraint
FlightControl::FCU/A4=	assume constraint
FlightControl::FCU/A8=	assume constraint
FlightControl::FCU/A1=	assume constraint
FlightControl::FCU/A5=	assume constraint
FlightControl::FCU/A6=	assume constraint
FlightControl::FCU/A7=	assume constraint
Abstractions::AtanR_Abtract/A1=	assume constraint
Abstractions::SinR_Abtract/A1=	assume constraint

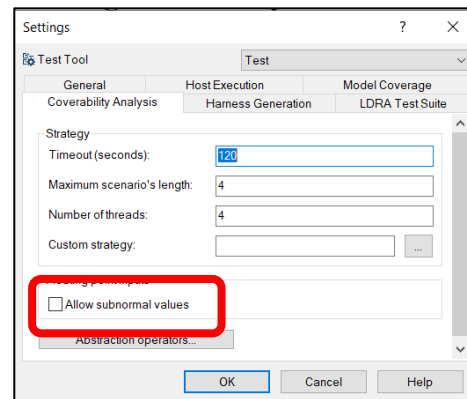
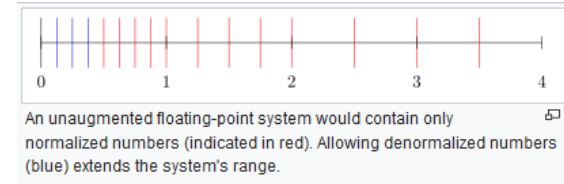
3.2. Abstractions

Concrete operator path	Abstract operator path
mathext::SinR	Abstractions::SinR_Abtract
mathext::AtanR	Abstractions::AtanR_Abtract

- *Note : Scenario provided may not reach expected Coverage point when provided abstraction is too far from actual behavior.*

Coverage Assistance – Subnormal values avoidance

- It is now possible to allow or avoid subnormal values
- In computer science, subnormal numbers are the subset of denormalized numbers (sometimes called denormals) that fill the underflow gap around zero in floating-point arithmetic. Any non-zero number with magnitude smaller than the smallest normal number is subnormal.
- In practice Subnormal values are values close to Zero
- Different processors may handle subnormal values directly in pipelined hardware, escape out to a microcode handler, or even issue a trap to the OS.



Requirements Traceability Improvements

Ansys

- New ALM Gateway API

- ALM Gateway provides now a new Python API Allowing to get access to traceability information
- Examples are provided here : v231\SCADE\SCADE\APIs\Python\examples\traceability
- There is one script example to remove pending links:

```
79 def print_links(trace_project: traceability.Project):
80     print("List of links")
81     print("(oid, req_id, status)")
82     for link in trace_project.traceability_links:
83         print(f"({link.source.identifier}, {link.target.identifier}, {link.status.name})")
84
85
86 def print_pending_links(trace_project: traceability.Project):
87     print("List of pending links")
88     print("(oid, req_id, status)")
89     for link in traceability.get_pending_links(trace_project):
90         print(f"({link.source.identifier}, {link.target.identifier}, {link.status.name})")
91
92
93 def print_traceable_elements(trace_project: traceability.Project):
94     print("List of TraceableElements")
95     for element in trace_project.traceable_elements:
96         print_element(element)
97
98
99 def remove_all_pending_links(project: traceability.Project):
100     for link in traceability.get_pending_links(project):
101         project.remove_link(link.source.identifier, link.target.identifier)
102
```

Thank You!

The Ansys logo consists of a yellow slanted bar followed by the word "Ansys" in a bold, black, sans-serif font.

