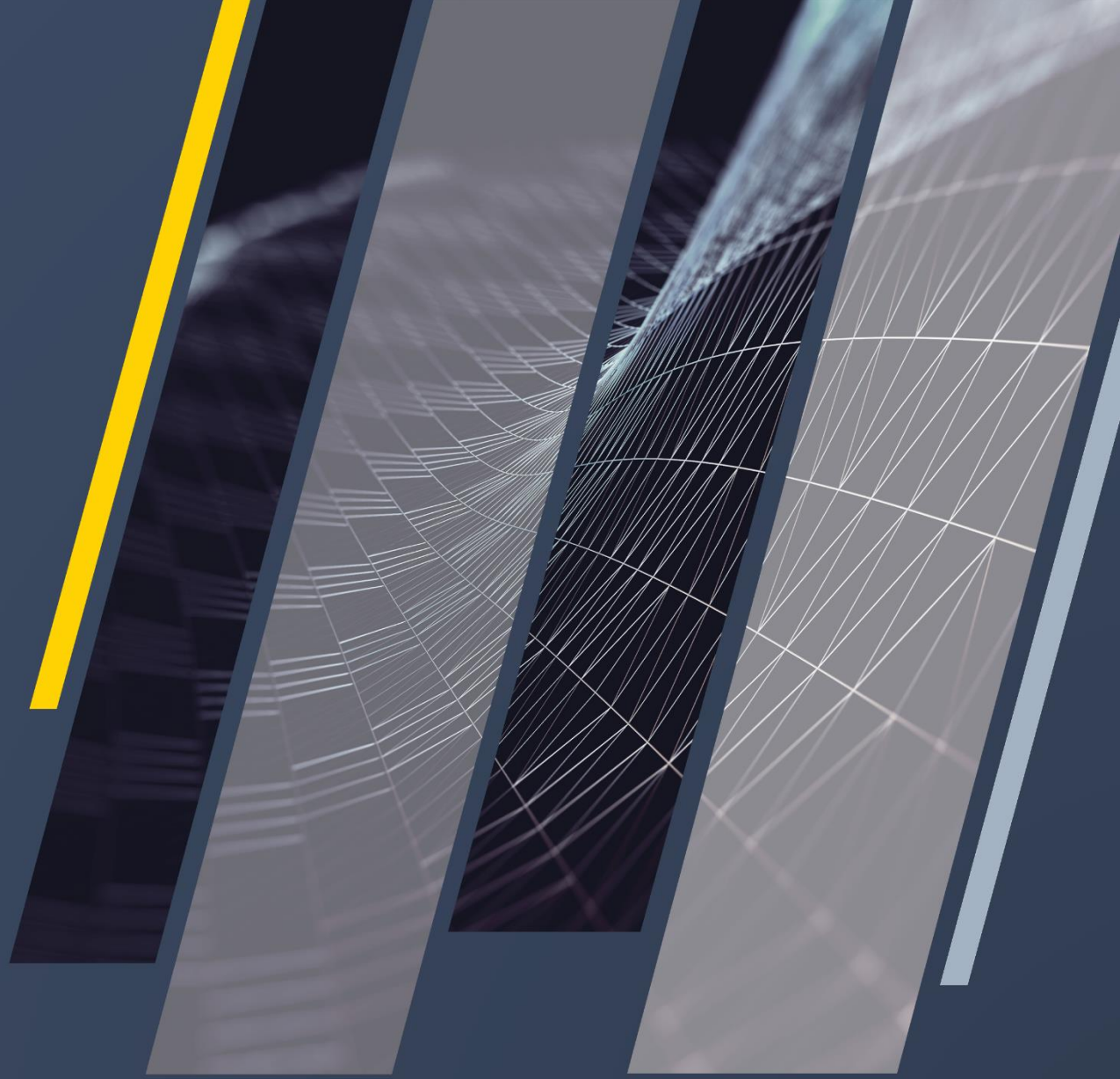


Model-Based System and Software Analysis and Development Tools

Gui Goretkin
Senior Application Engineer – ANSYS SCADE
e: guilherme.goretkin@ansys.com

February 2019

ANSYS offers the only true multi-physics simulation platform

Market Leader Across Individual Physics with Industry-Leading Platform

Structures

Fluids

Electromagnetics

Semiconductor
Power

Mission-critical
Embedded Software

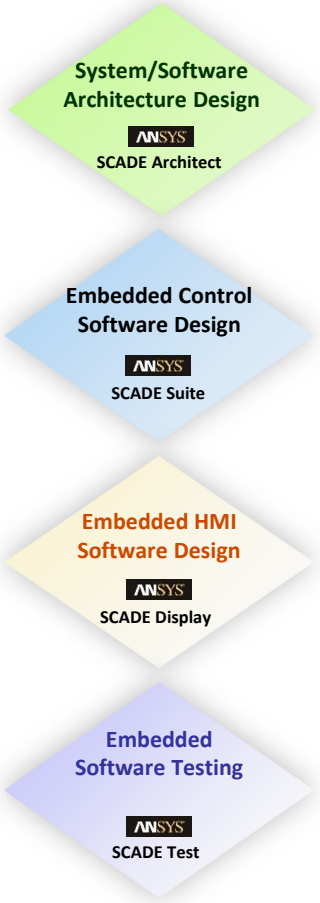
Optical

Platform

© 2019 ANSYS, Inc.

SCADE:
Safety Critical Application Development Environment

- ANSYS SCADE is a suite of integrated tools:

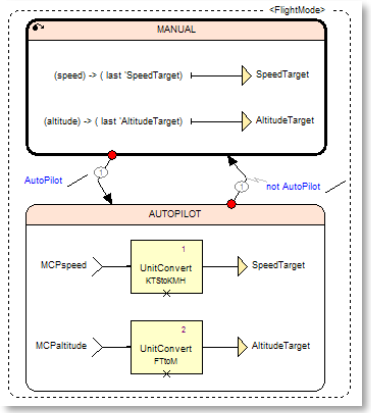


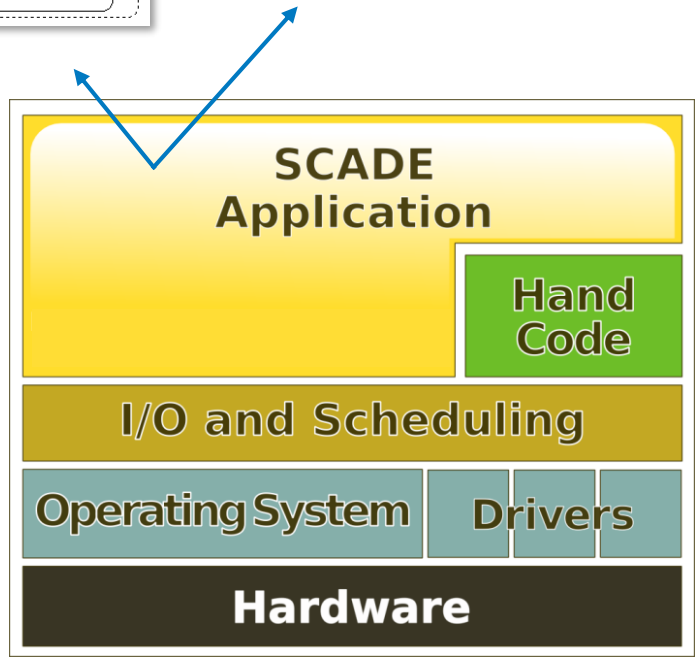
- SCADE Architect: SysML Engineering tool, extensible to support Domain Specific Languages (DSL) via a dedicated module named “Configurator”.
- SCADE Suite: Industry-proven solution dedicated to the development of safety critical embedded software. The SCADE Suite code generator is qualified according to DO-178C/DO-330 at TQL-1.
- SCADE Display: Model-based HMI software design solution, designed for displays with safety objectives. The SCADE Display code generator is qualified according to DO-178C/DO-330 at TQL-1.
- SCADE Test: Complete set of simulation, verification and validation tools.

© 2019 ANSYS, Inc.

ANSYS SCADE for Model-Based Software Design

- Model-based design tool for safety and performance-critical embedded software systems
- Embedded controls, displays, HMI implementation
  - OSA solutions: FACE, ARINC 661
- Native requirements traceability
  - (JAMA, DOORS, etc)
- Portable & certified C/Ada code generation:
  - DO-178B/C up to DAL A - Aerospace/Defense
  - EN 50128 up to SIL 3/4 - Rail Transportation
  - IEC 61508 up to SIL 3 – Industrial & Energy
  - ISO 26262 certification up to ASIL D – Automotive
- Efficiency of development effort through automation
  - code generation, test scripting, report generator





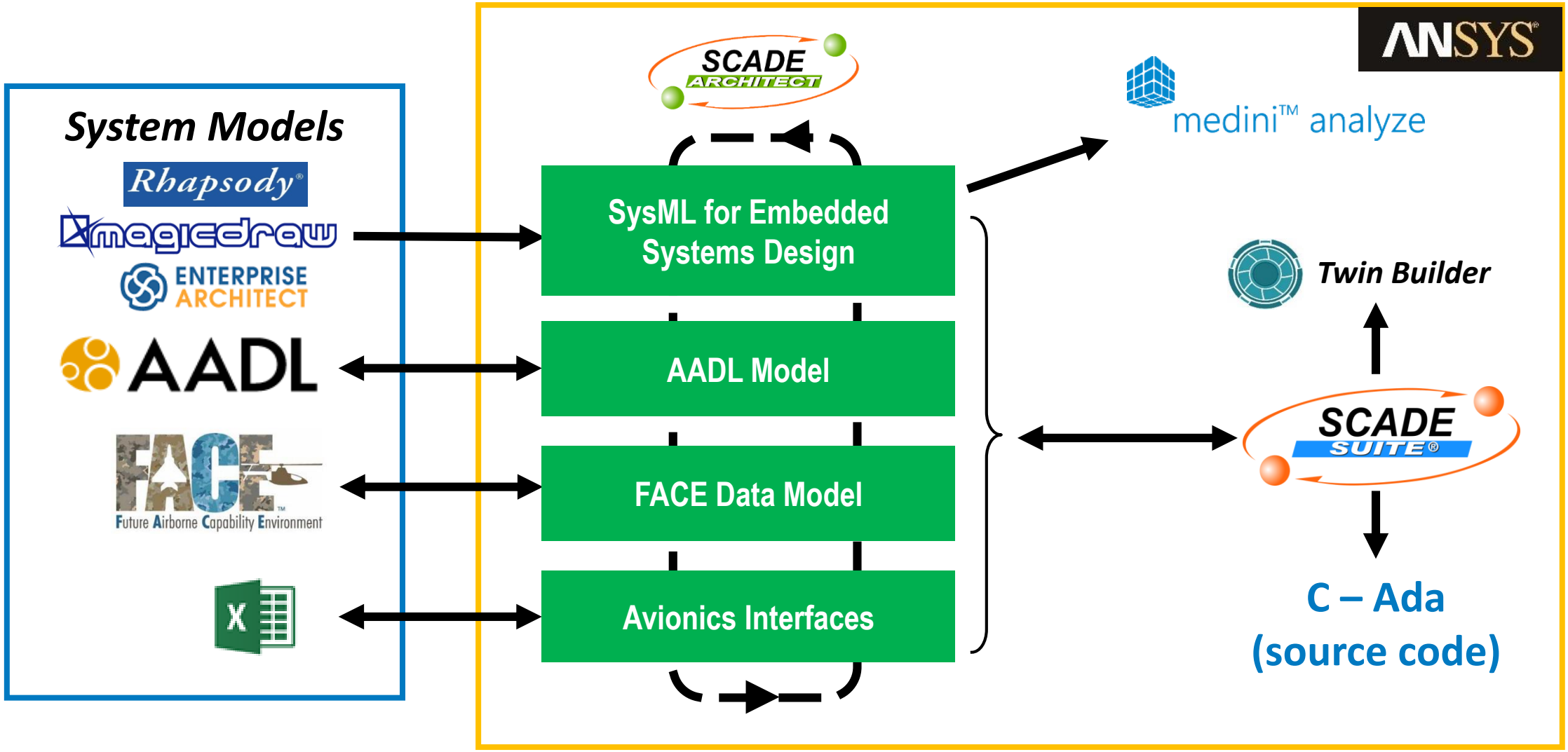
© 2019 ANSYS, Inc.

ANSYS SCADE Architect Differentiators

- Foundations: SysML Standard
  - ➔ Easier interoperability with other tools
  - ➔ Focus on Ease of use (hiding underlying UML profile, SysML simplifications)
- Powerful Tables / Model-Based ICD Management
  - ➔ Scalable comprehensive representations
  - ➔ Export information to Excel
  - ➔ Create SCADE Architect model from external data
- Domain Specific Language
  - ➔ Straightforward Customization for user-specific domain
  - ➔ Support of industrial standards: AADL, FACE, AUTOSAR (subset)
- System/Software Synchronization
  - ➔ Efficient bi-directional path with SCADE Suite for consistent System-SW designs

© 2019 ANSYS, Inc.

MBSE Workflows Capabilities supported by SCADE



© 2019 ANSYS, Inc.

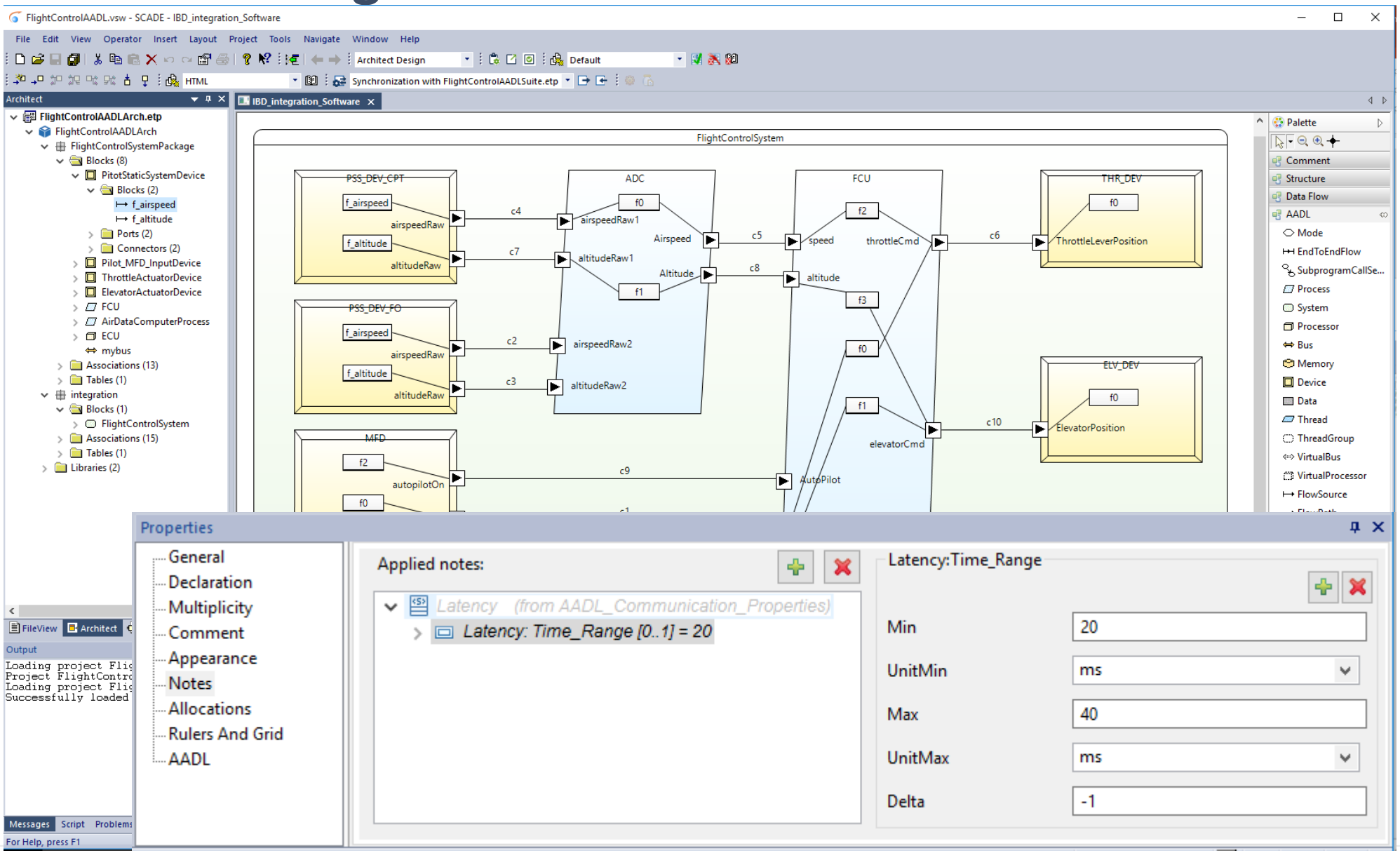
SCADE AADL Modeling Package

- AADL is an SAE International standard dedicated to real-time embedded systems
  - Modeling software and hardware resources for V&V
  - Powerful Property Sets extension concept

- AADL Support with SCADE AADL Modeler + SCADE Avionics Package
  - Full compatibility with AADL v2.2 standard
    - Allows for legacy models import
    - Allows for export to third party analyzers
  - Easy to use
    - AADL expressiveness simplified: just concrete components
    - Nice graphical interface & diagrams
  - Benefit from SCADE tools ecosystem
    - Bi-directional synchro with SCADE Suite for SW component development, verification & certification
    - Traceability through SCADE ALM gateway
    - Same IDE as for SysML and FACE modeling (mixed designed supported)

© 2019 ANSYS, Inc.

SCADE Avionics Package: AADL



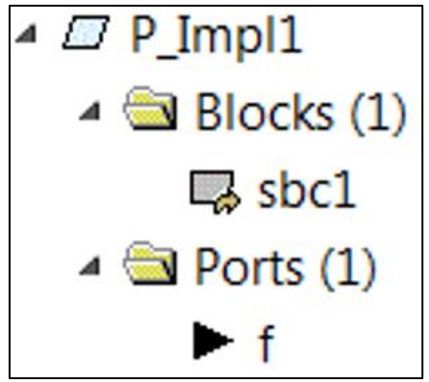
© 2019 ANSYS, Inc.

Import AADL files in ANSYS SCADE AADL

1. Merge component type and implementation in a single object

```
process P
  features
    f: in data port Base_Types::Unsigned_16;
end P;

process implementation P.Impl1
  subcomponents
    sbc1: data Base_Types::Unsigned_16;
end P.Impl1;
```

© 2019 ANSYS, Inc.

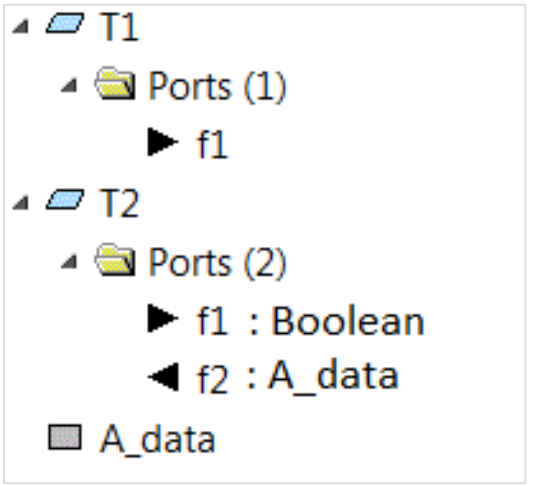
Import AADL files in ANSYS SCADE AADL

2. Inline inheritance
3. Resolve prototypes
4. Import abstract elements when they are refined to concrete ones

```
thread T1
  prototypes
    p: data;
  features
    f1: in data port p;
end T1;

thread T2 extends T1 (p => data Base_Types::Boolean)
  features
    f2 : out data port A;
end T2;

abstract A
end A;
```

© 2019 ANSYS, Inc.

Import AADL files in ANSYS SCADE AADL

5. Usage of SCADE Architect replication mechanism for immediate instantiation of components.

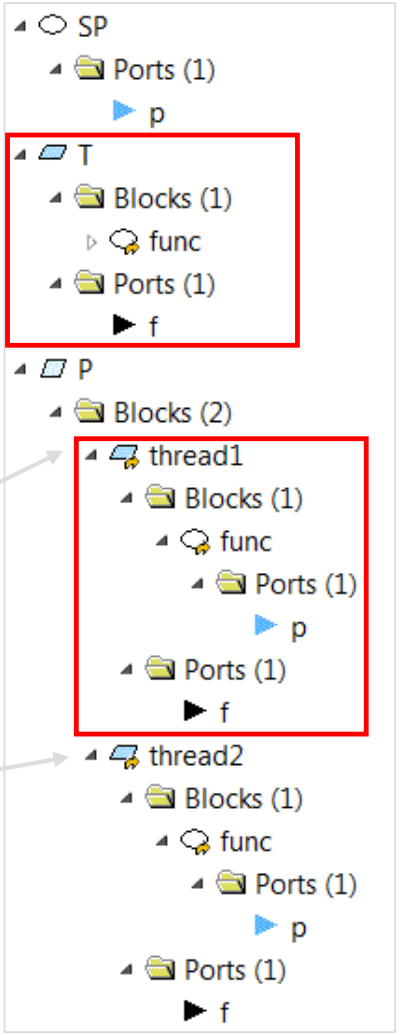
```
subprogram SP
  features
    p : in parameter Base_Types::Boolean;
end SP;

thread T
  features
    f: in data port Base_Types::Unsigned_16;
end T;

thread implementation T.impl
  subcomponents
    func: subprogram SP;
end T.impl;

process P
end P;

process implementation P.impl
  subcomponents
    thread1 : thread T.impl;
    thread2 : thread T.impl;
end P.impl;
```



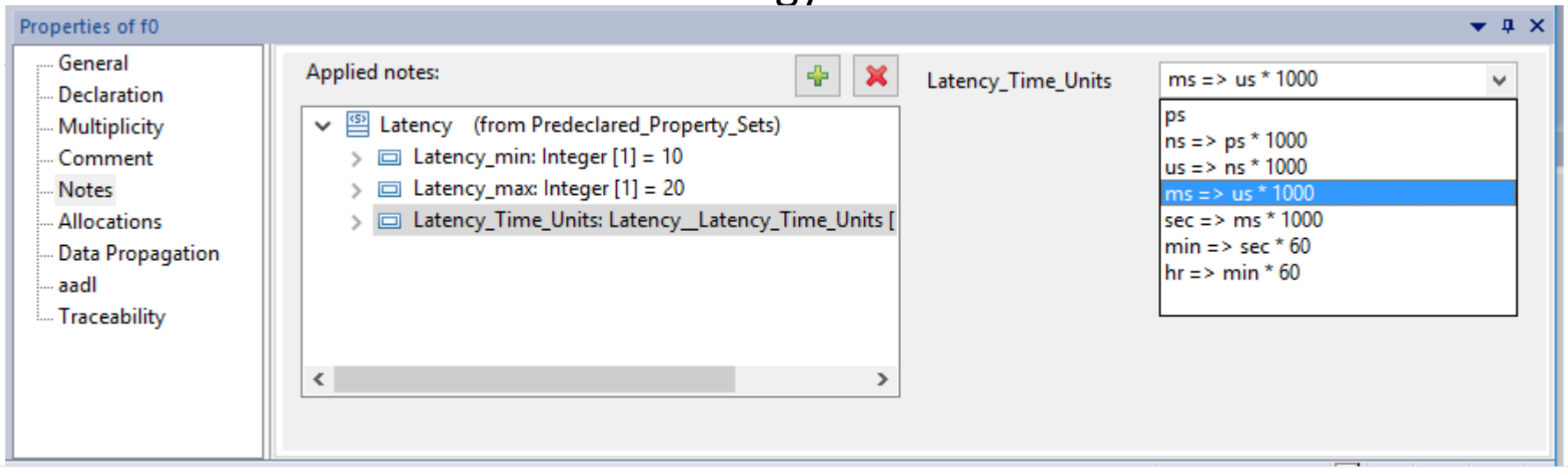
© 2019 ANSYS, Inc.

AADL Property sets

- SCADE annotations:
  - Typed attributes, associated to components
- Automated conversion

<property set>.aadl **<SCADE note types>.aty**

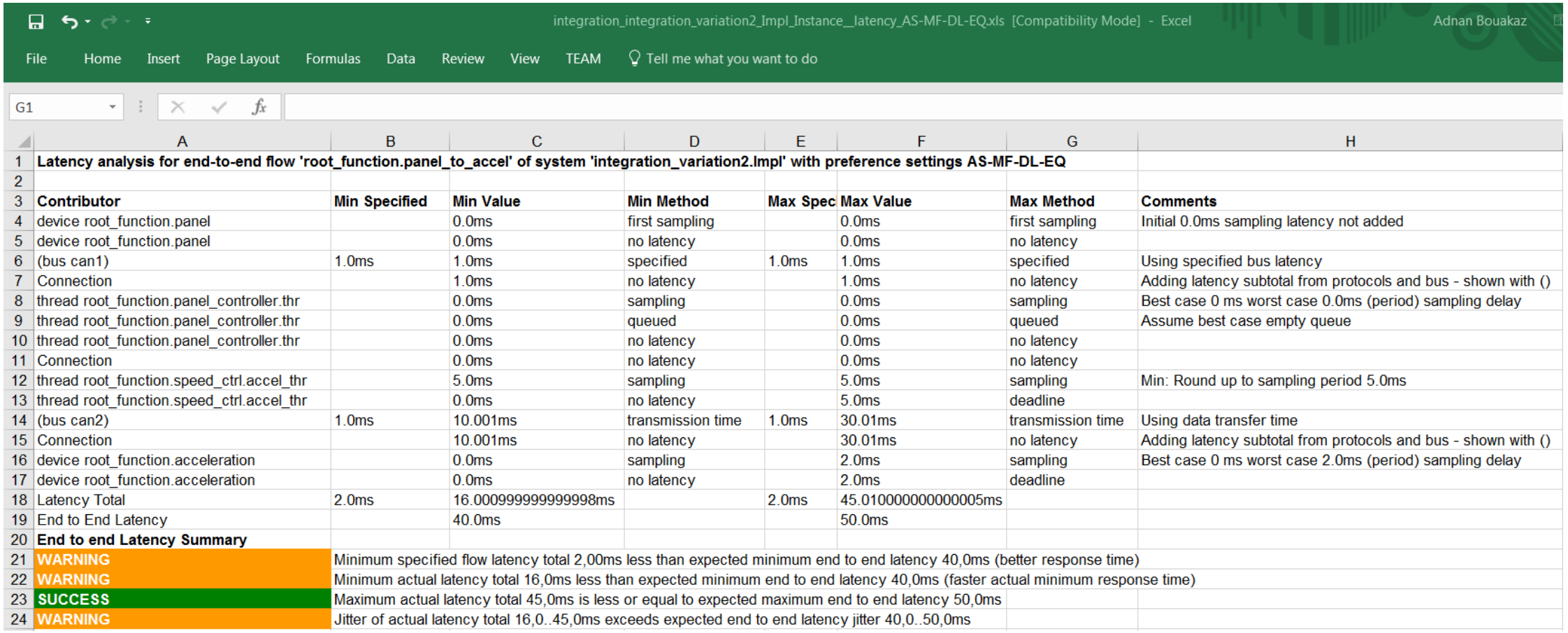
- Benefits
  - Reused SCADE IDE matured technology for structured annotations
  -



© 2019 ANSYS, Inc.

Case study

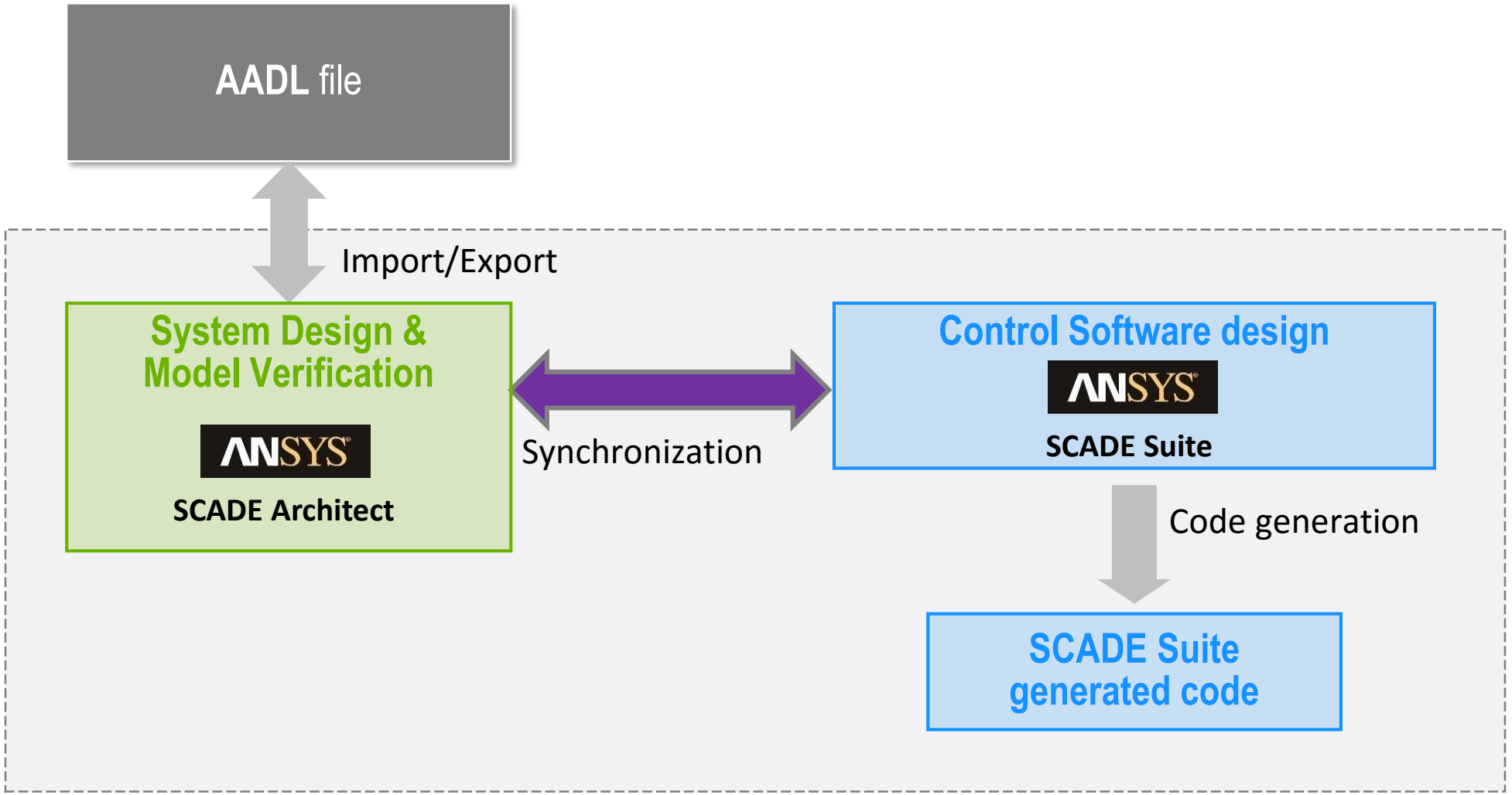
- Analysis example
  - Export self-driving car example from SCADE AADL to textual aadl file
  - Latency analysis result from Open Source tool OSATE



	A	B	C	D	E	F	G	H
1	**Latency analysis for end-to-end flow 'root_function.panel_to_accel' of system 'integration_variation2.Impl' with preference settings AS-MF-DL-EQ**							
2								
3	**Contributor**	**Min Specified**	**Min Value**	**Min Method**	**Max Spec**	**Max Value**	**Max Method**	**Comments**
4	device root_function.panel		0.0ms	first sampling		0.0ms	first sampling	Initial 0.0ms sampling latency not added
5	device root_function.panel		0.0ms	no latency		0.0ms	no latency	
6	(bus can1)	1.0ms	1.0ms	specified	1.0ms	1.0ms	specified	Using specified bus latency
7	Connection		1.0ms	no latency		1.0ms	no latency	Adding latency subtotal from protocols and bus - shown with ()
8	thread root_function.panel_controller.thr		0.0ms	sampling		0.0ms	sampling	Best case 0 ms worst case 0.0ms (period) sampling delay
9	thread root_function.panel_controller.thr		0.0ms	queued		0.0ms	queued	Assume best case empty queue
10	thread root_function.panel_controller.thr		0.0ms	no latency		0.0ms	no latency	
11	Connection		0.0ms	no latency		0.0ms	no latency	
12	thread root_function.speed_ctrl.accel_thr		5.0ms	sampling		5.0ms	sampling	Min: Round up to sampling period 5.0ms
13	thread root_function.speed_ctrl.accel_thr		0.0ms	no latency		5.0ms	deadline	
14	(bus can2)	1.0ms	10.001ms	transmission time	1.0ms	30.01ms	transmission time	Using data transfer time
15	Connection		10.001ms	no latency		30.01ms	no latency	Adding latency subtotal from protocols and bus - shown with ()
16	device root_function.acceleration		0.0ms	sampling		2.0ms	sampling	Best case 0 ms worst case 2.0ms (period) sampling delay
17	device root_function.acceleration		0.0ms	no latency		2.0ms	deadline	
18	Latency Total	2.0ms	16.000999999999998ms		2.0ms	45.010000000000005ms		
19	End to End Latency		40.0ms			50.0ms		
20	**End to end Latency Summary**							
21	WARNING	Minimum specified flow latency total 2,00ms less than expected minimum end to end latency 40,0ms (better response time)						
22	WARNING	Minimum actual latency total 16,0ms less than expected minimum end to end latency 40,0ms (faster actual minimum response time)						
23	SUCCESS	Maximum actual latency total 45,0ms is less or equal to expected maximum end to end latency 50,0ms						
24	WARNING	Jitter of actual latency total 16,0..45,0ms exceeds expected end to end latency jitter 40,0..50,0ms						

© 2019 ANSYS, Inc.

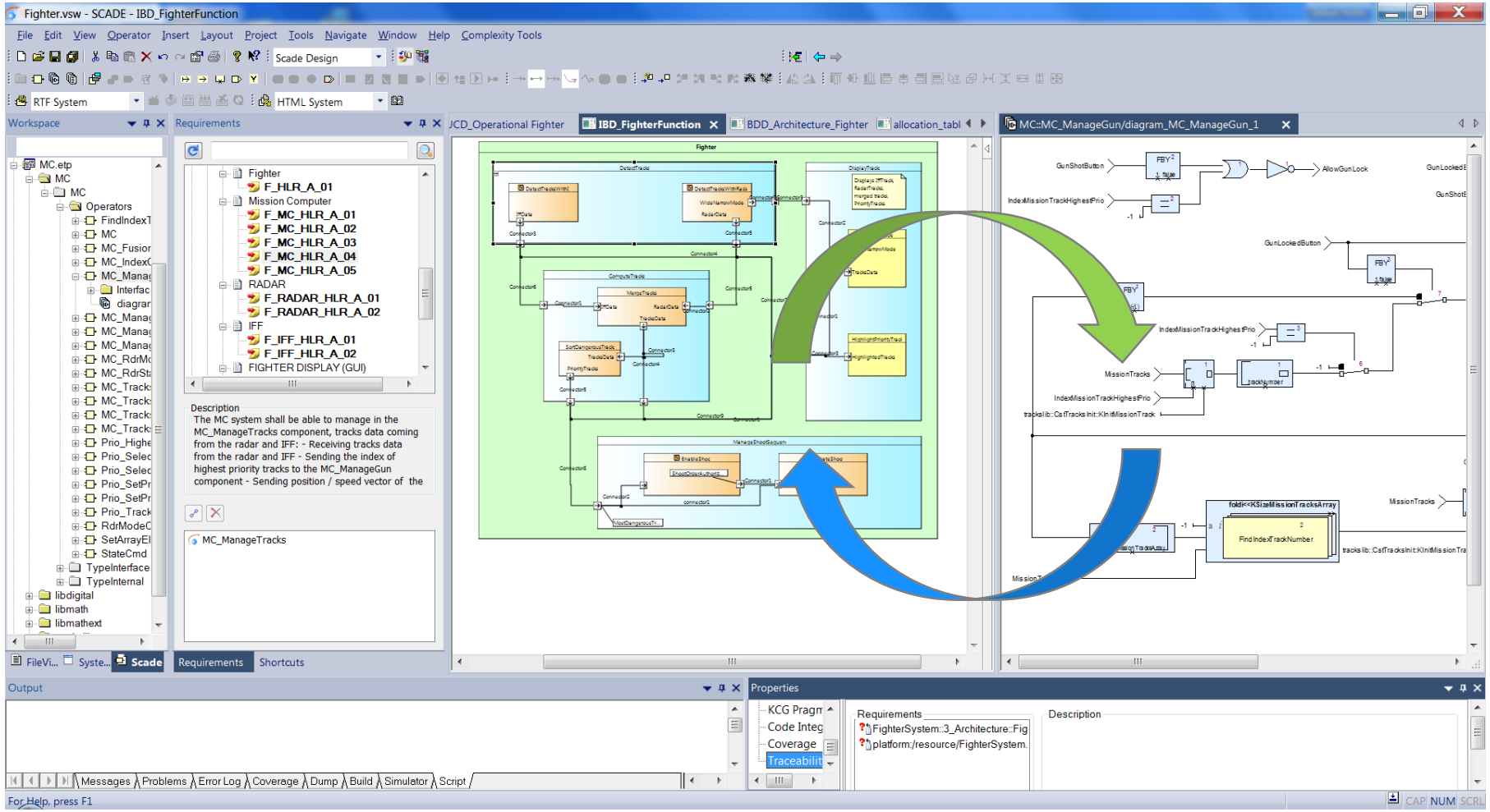
ANSYS SCADE AADL : Workflow



© 2019 ANSYS, Inc.

ANSYS SCADE Architect – ANSYS SCADE Suite Integration

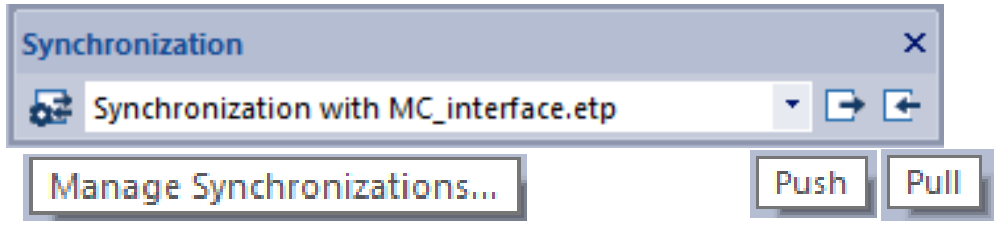
An Integrated Workflow for SW-intensive Systems

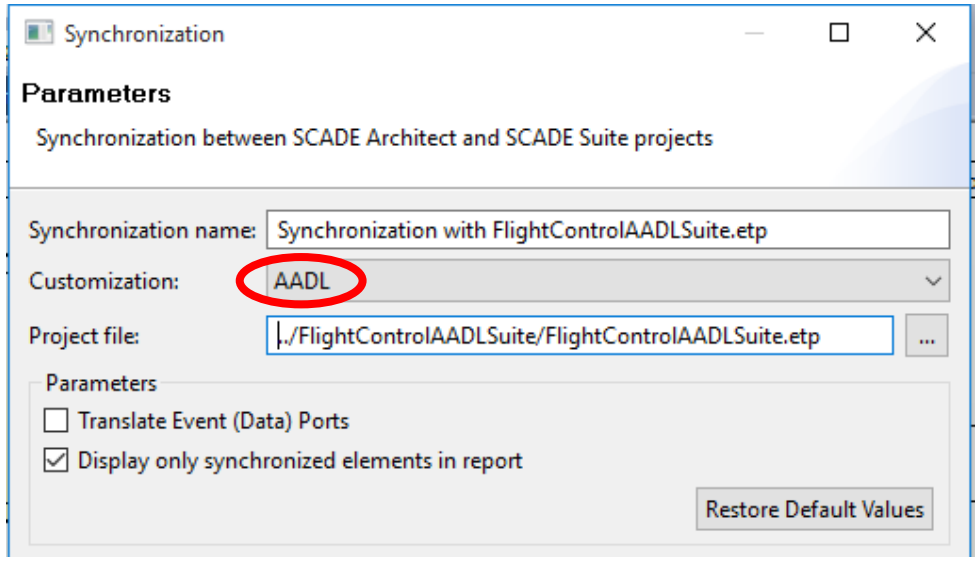


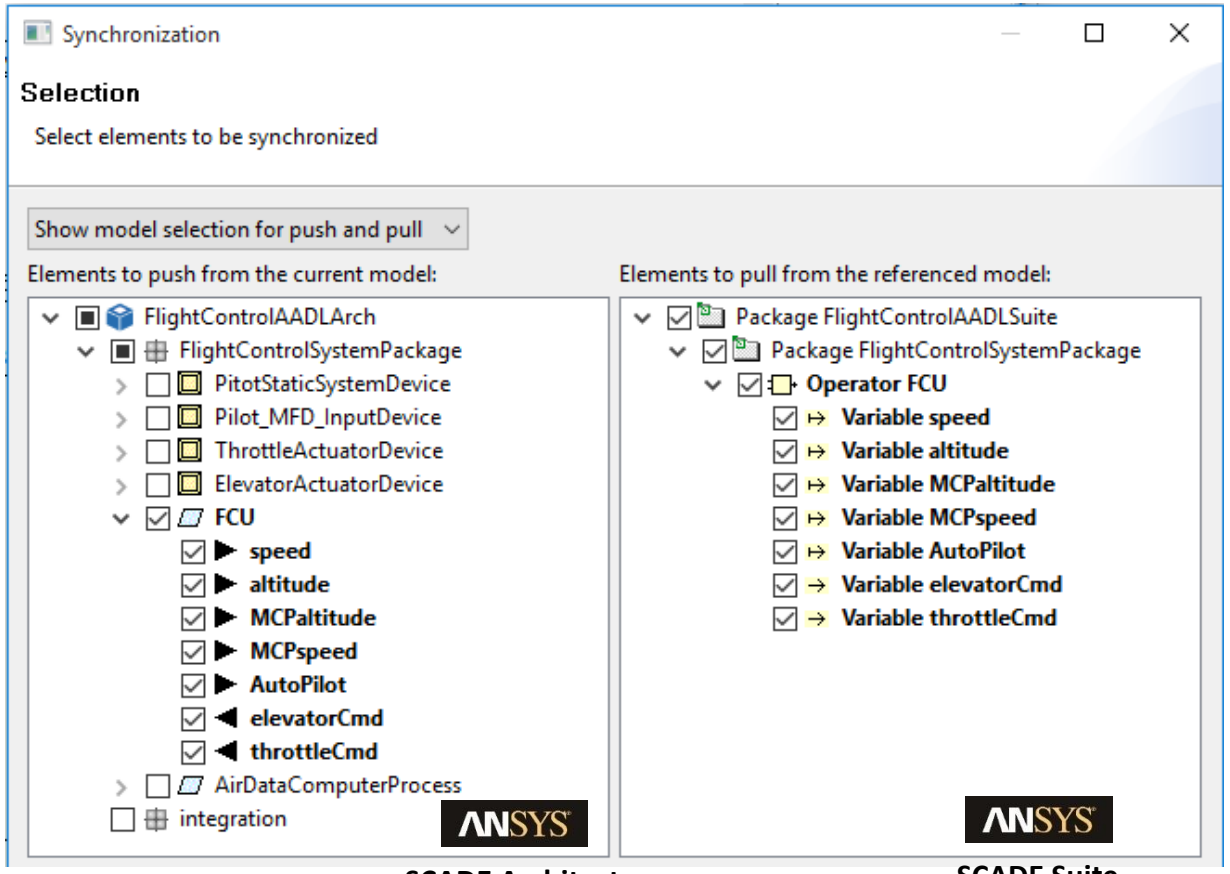
© 2019 ANSYS, Inc.

Synchronization ANSYS SCADE AADL – ANSYS SCADE Suite

- SCADE synchronization is customized for AADL models







SCADE Architect

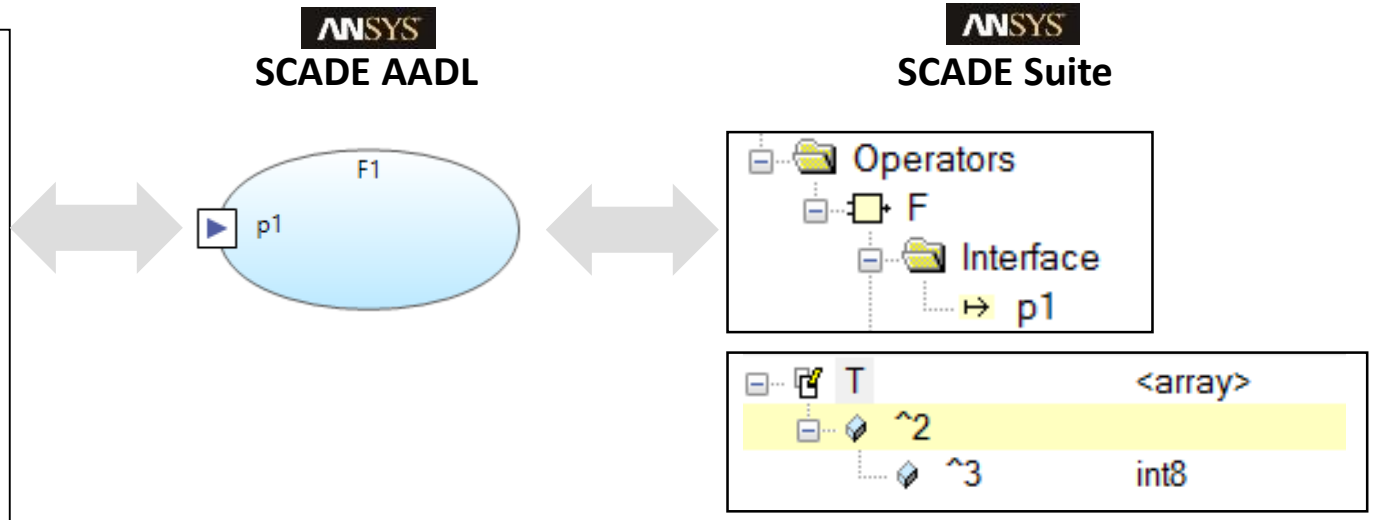
SCADE Suite

© 2019 ANSYS, Inc.

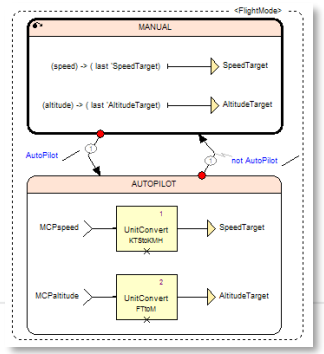
Synchronization ANSYS SCADE AADL – ANSYS SCADE Suite

- **Bi-directional synchronization**
  - AADL threads, devices and subprograms with SCADE Suite operators
  - AADL data with SCADE Suite datatypes

```
subprogram F
  features
    p1: in parameter T;
end F;
data T
  properties
    Data_Model::Data_Representation => Array;
    Data_Model::Base_Type => (classifier (Base_Types::Integer_8));
    Data_Model::Dimension => (2, 3);
end T;
```



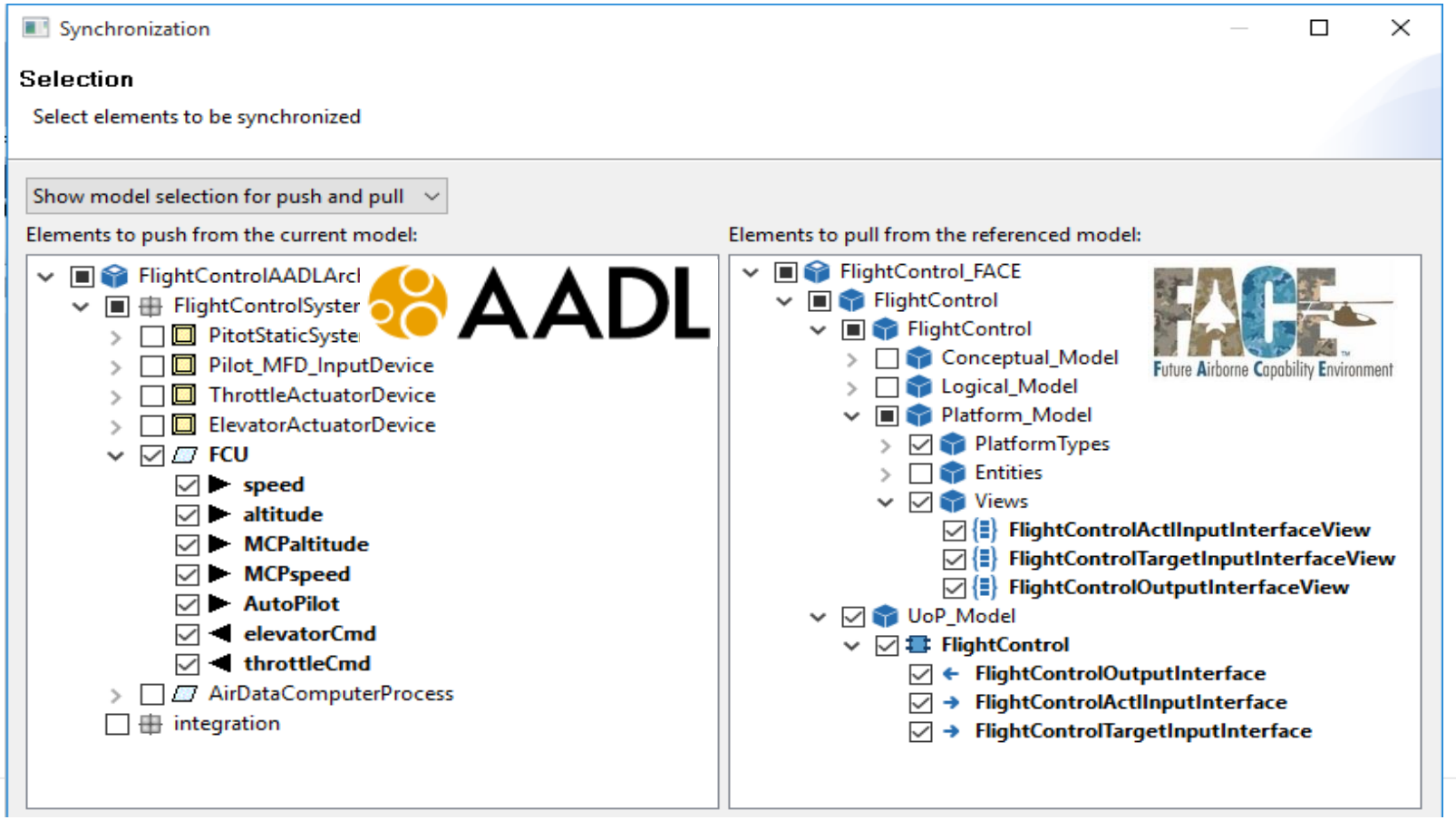
- **Behavior implementation in SCADE Suite**
  - Simulation, certified C/Ada code generation,
  - Test procedures and model coverage with SCADE Test



© 2019 ANSYS, Inc.

AADL – FACE : on-going work

- New AADL “FACE Annex” in progress by AADL Committee
- SCADE Architect AADL – FACE models synchronization



- Preliminary tool demo

© 2019 ANSYS, Inc.

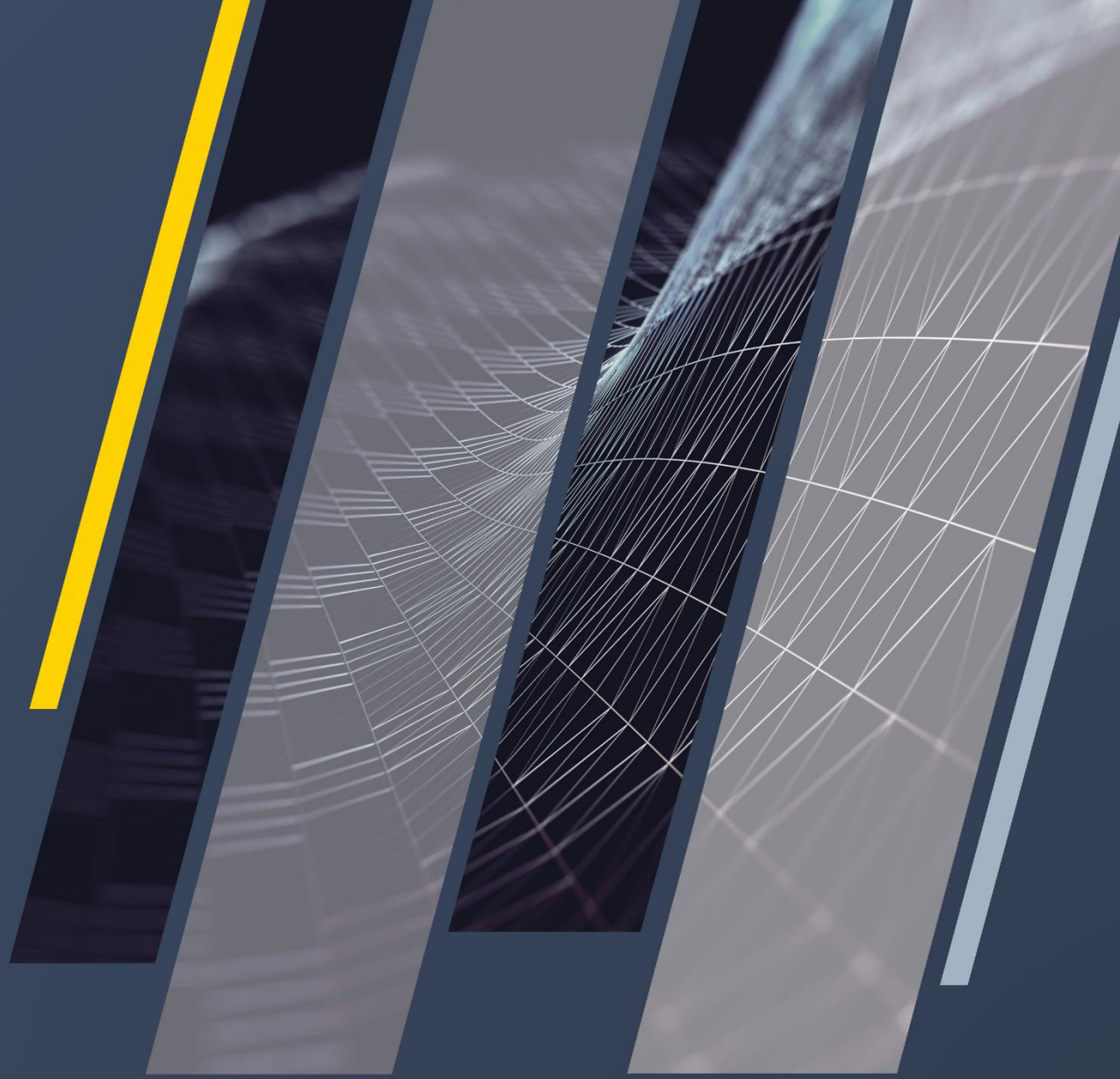
ANSYS SCADE AADL - CONCLUSION

- Full compatibility with AADL v2.2 standard
  - Allows for legacy models import; Allows for export to third party analyzers
- Easy to use
  - Nice graphical interface & diagrams; AADL expressiveness simplified
- Large ecosystem
  - Traceability to many requirements management tool through SCADE ALM gateway
  - Modeling SysML, AADL and FACE in the same IDE
  - Synchronization with SCADE Suite for SW component development, V&V, certification
  - Technical coordination with AdventiumLab for AADL models analysis

© 2019 ANSYS, Inc.

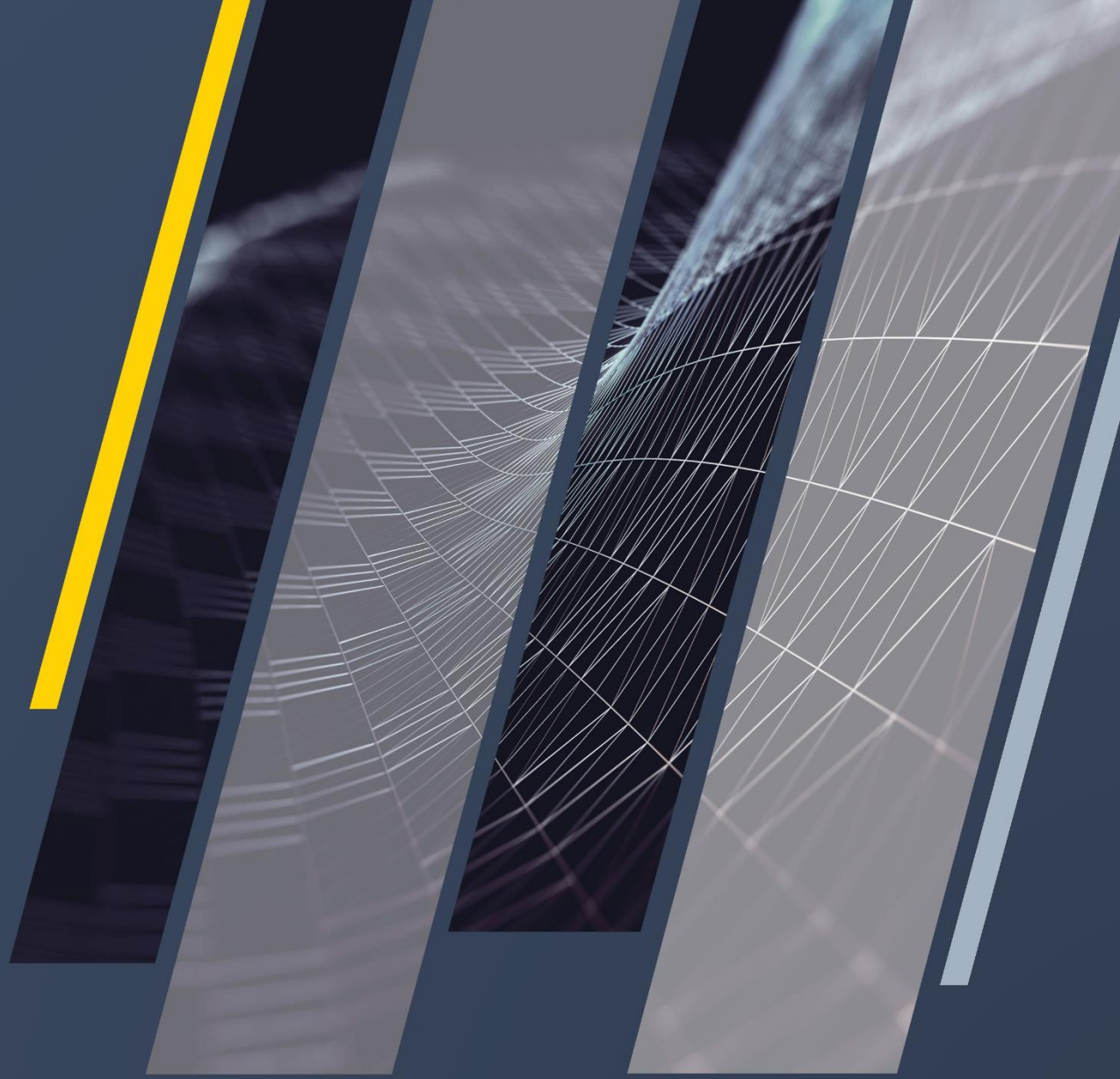
ANSYS®

Thank you



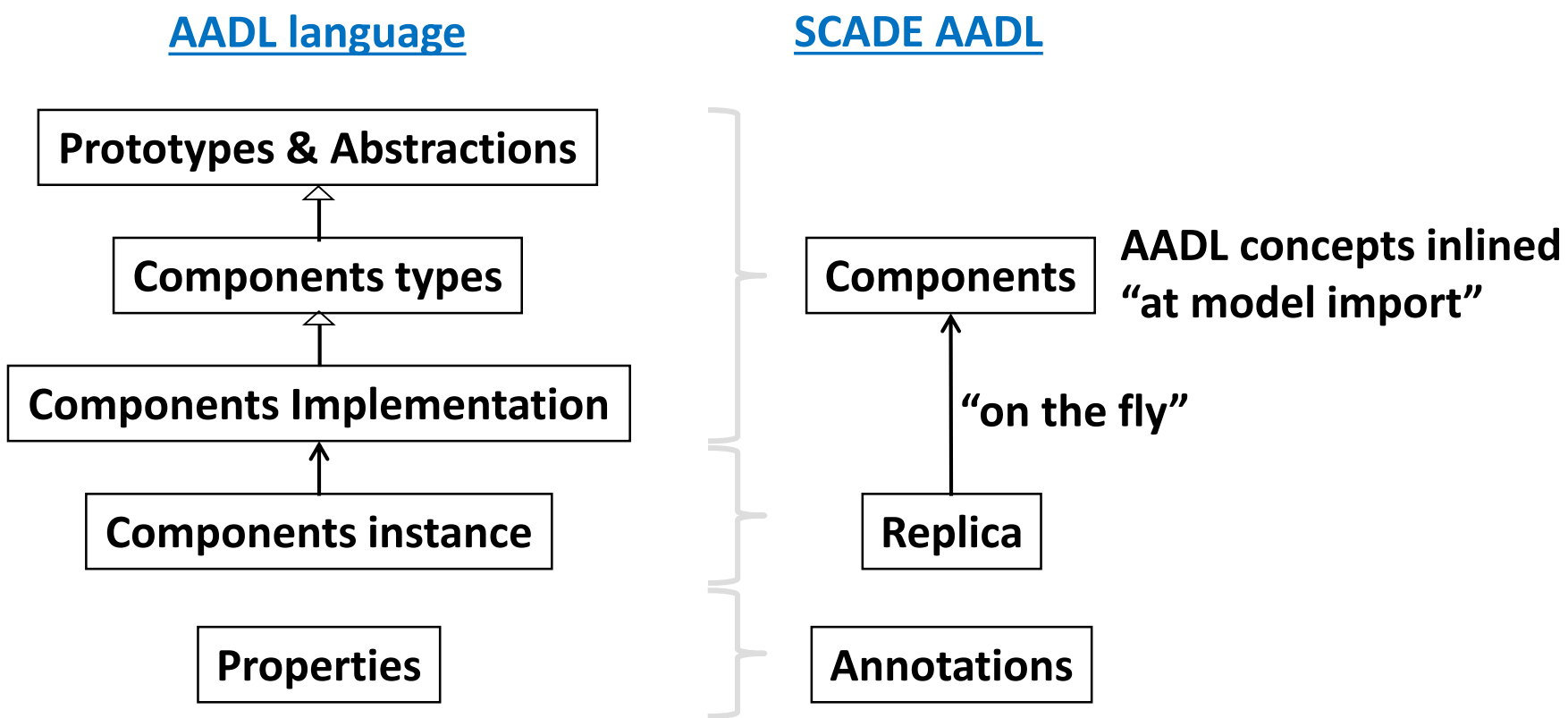
ANSYS®

Backup



ANSYS SCADE solution for AADL

- **Support for AADL “instance based modeling”: much simpler model understanding**



© 2019 ANSYS, Inc.

AADL Property sets

```
property set AADL_Projects
is Time_Units: type units (
    ps,
    ns  => ps  * 1000,
    us  => ns  * 1000,
    ms  => us  * 1000,
    sec => ms  * 1000,
    min => sec * 60,
    hr  => min * 60);
-- …
end AADL_Projects;
```

- Property definition
  - Typed attribute, associated to components
- Property sets: group of property definitions
  - Part of the standard
  - User-defined

```
--AADL2
--SAE Aerospace Standard AS5506B
--Appendix A: Predeclared Property Sets

property set Communication_Properties is
 Time: type aadlinteger units Time_Units;
 Time_Range: type range of Time;
 Latency: Time Range
    applies to (flow, connection, virtual
bus, bus, processor, virtual processor,
device, system, feature, memory);
```

- Model properties

```
process MFDProcess
  features
    MCPaltitude: out data port scade_real;
  flows
    f0: flow source MCPaltitude {
      Latency => 5 ms .. 10 ms;};
  properties
    Period => 25 ms;
end MFDProcess;
```

© 2019 ANSYS, Inc.

Case study

A simple self-driving car example. “AADL In Practice”, Julien Delange: http://www.aadl-book.com