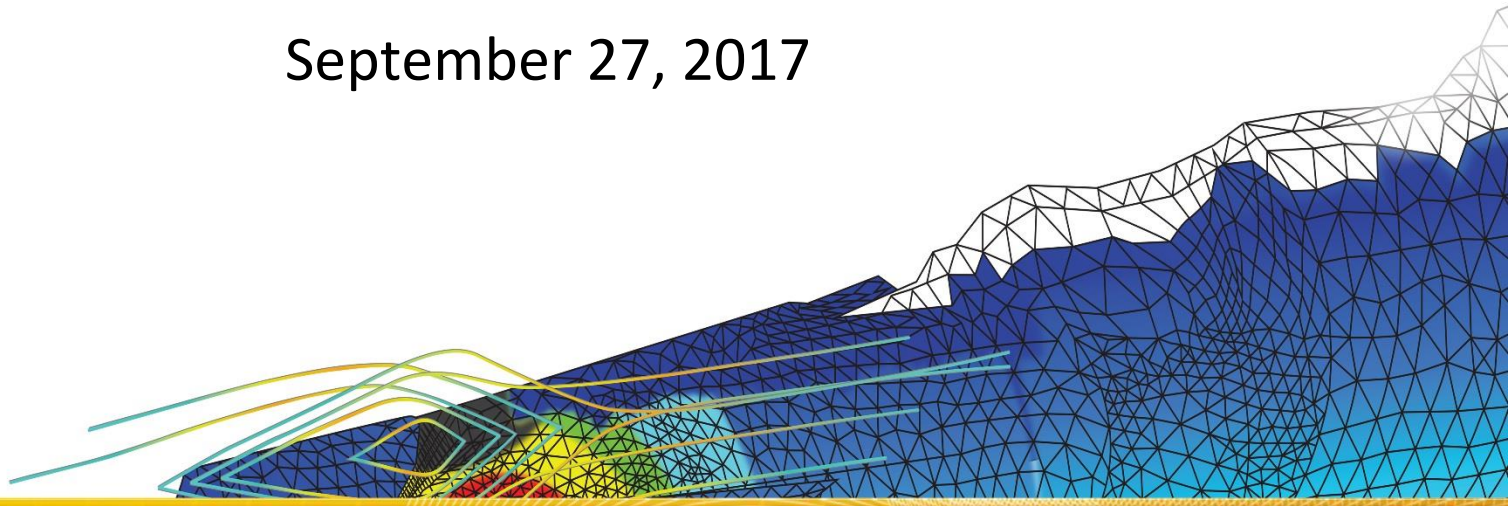
SCADE AADL Solution

Guilherme Goretkin

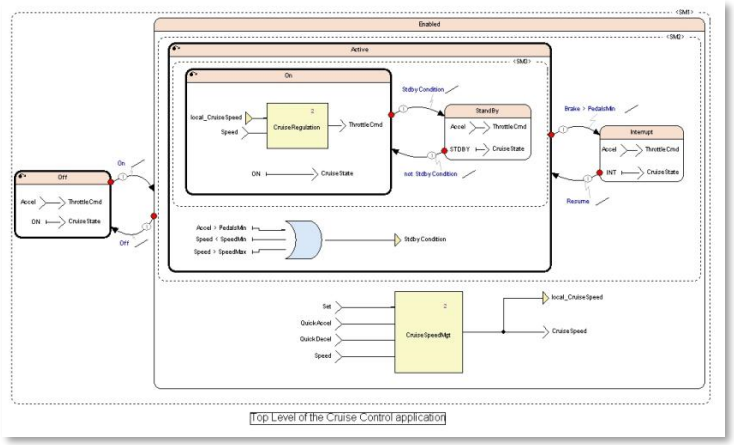
September 27, 2017



ANSYS SCADE Suite

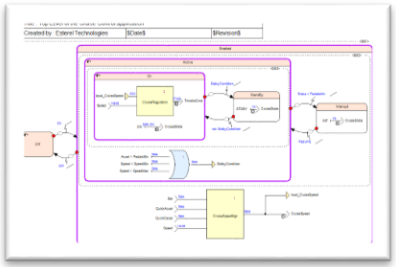
Embedded Control Software Design

Efficient modeling of controls, logic and algorithm designs within a single environment



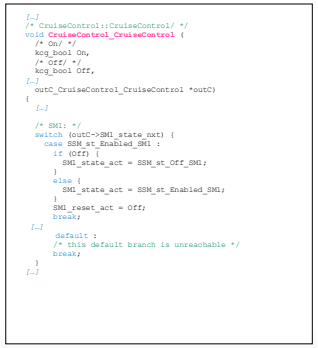
Integrated Suite for Prototyping, Modeling, Simulation, Verification, and Optimization

Efficient debugging and optimization of software models and code size, speed and performance



Qualified Code Generation

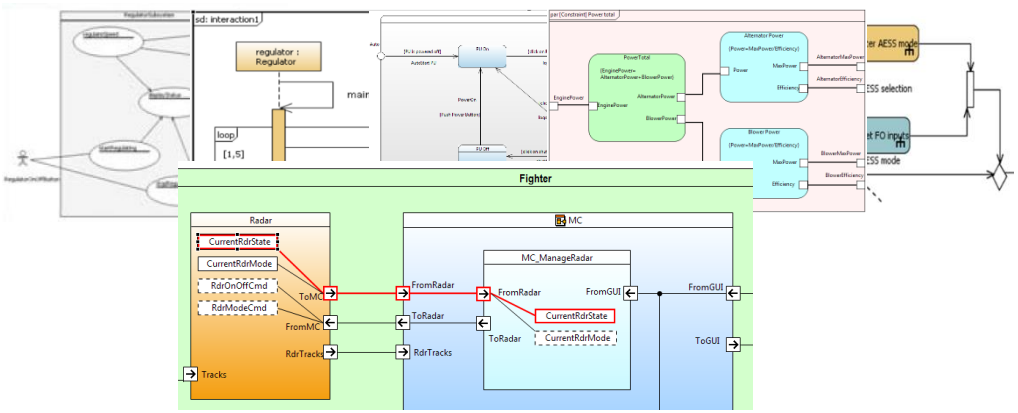
Automatic C and Ada qualified code generators (DO-178B/C, EN 50128, ISO 26262, IEC 61508)
Enables 80% embedded code production and testing cost reduction

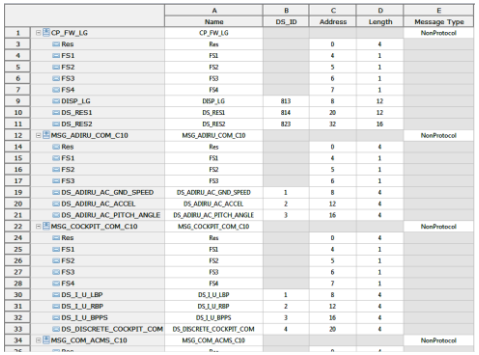


ANSYS SCADE Architect

Model-Based embedded systems architecture design

SysML standard based, focus on ease of use,
Data dictionaries and data propagation in architecture.





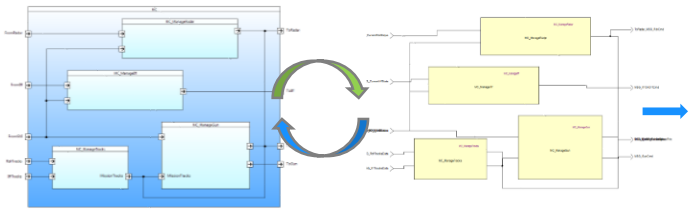
	A	B	C	D	E
	Name	DS_ID	Address	Length	Message Type
1	CP_FW_LG	CP_FW_LG			NonProtocol
3	Res	Res	0	4	
4	FS1	FS1	4	1	
5	FS2	FS2	5	1	
6	FS3	FS3	6	1	
7	FS4	FS4	7	1	
9	DISP_LG	DISP_LG	813	8	12
10	DS_RES1	DS_RES1	814	20	12
11	DS_RES2	DS_RES2	823	32	16
12	MSG_ADIRU_COM_C10	MSG_ADIRU_COM_C10			NonProtocol
14	Res	Res	0	4	
15	FS1	FS1	4	1	
16	FS2	FS2	5	1	
17	FS3	FS3	6	1	
19	DS_ADIRU_AC_GND_SPEED	DS_ADIRU_AC_GND_SPEED	1	8	4
20	DS_ADIRU_AC_ACCEL	DS_ADIRU_AC_ACCEL	2	12	4
21	DS_ADIRU_AC_PITCH_ANGLE	DS_ADIRU_AC_PITCH_ANGLE	3	16	4
22	MSG_COCKPIT_COM_C10	MSG_COCKPIT_COM_C10			NonProtocol
24	Res	Res	0	4	
25	FS1	FS1	4	1	
26	FS2	FS2	5	1	
27	FS3	FS3	6	1	
28	FS4	FS4	7	1	
30	DS_I_U_LBP	DS_I_U_LBP	1	8	4
31	DS_I_U_RBP	DS_I_U_RBP	2	12	4
32	DS_I_U_BPPS	DS_I_U_BPPS	3	16	4
33	DS_DISCRETE_COCKPIT_COM	DS_DISCRETE_COCKPIT_COM	4	20	4
34	MSG_COM_ACMS_C10	MSG_COM_ACMS_C10			NonProtocol

Interface Control Documents (ICD) production

Support of Domain Specific Language and hierarchical table with MS Excel import/export demonstrated through ready to use industry specific packages

Integrated workflow for software intensive systems design

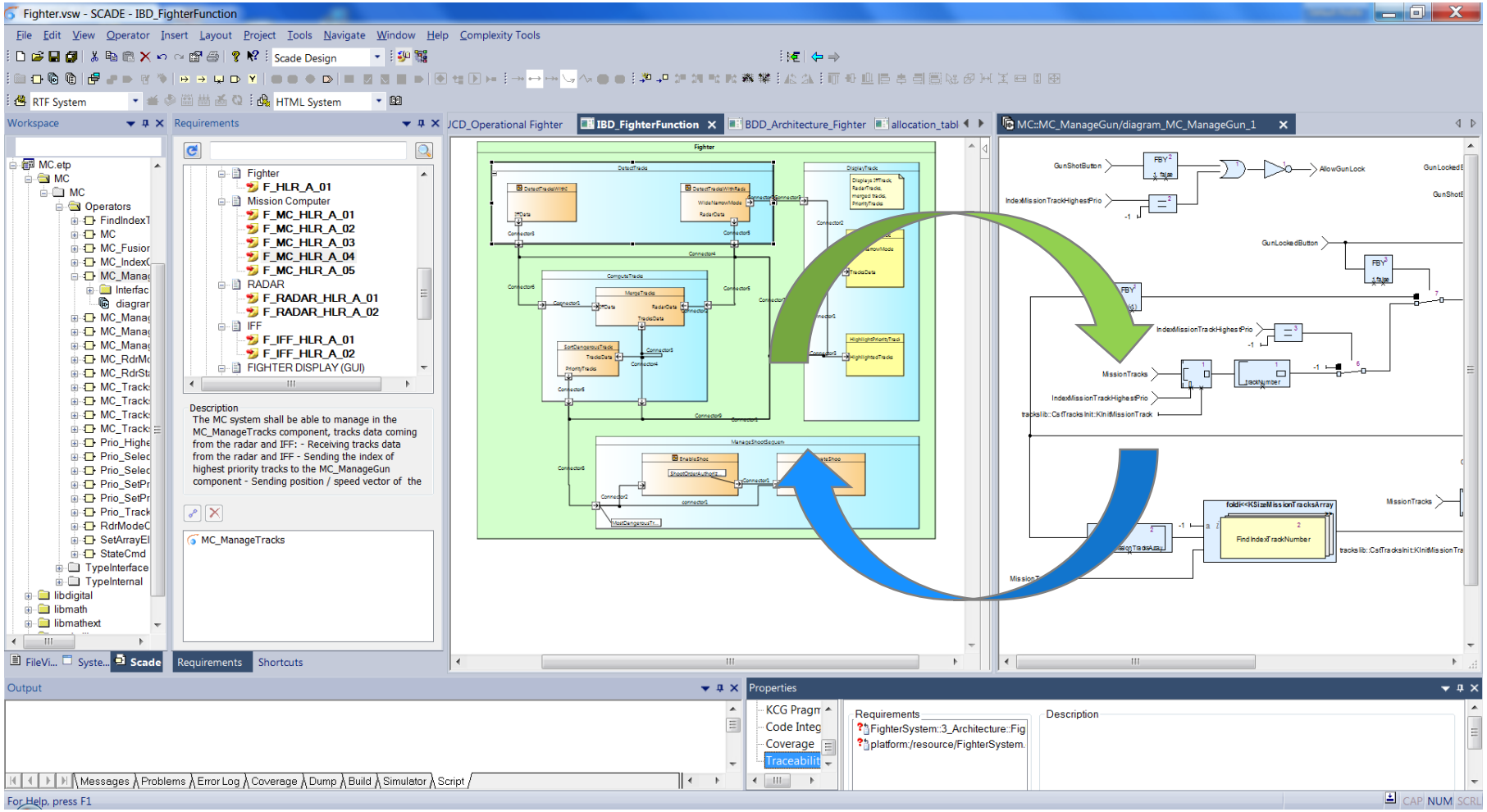
Synchronization with SCADE Suite designs for certified software development;
Supports industry engineering standards such as AUTOSAR, and US DoD FACE



```
/* Architecture::Regulation/ */
void Regulation_Architecture(
  /* ECU_Command/ */
  tECU_cmd_Architecture
    *ECU_Command,
  /* AccelPedal/ */
  tPercent_Architecture AccelPedal,
  /* Speed/ */
  tVehicleSpeed_Architecture Speed)
{
  kcg_float32 tmp;
  /* SM1:Regul:_L3/ */
  kcg_float32 _L3_Regul_SM1;
  /* SM1: */
  kcg_bool SM1_reset_act;
  /* SM1: */
  switch (outC->SM1_state_nxt) {
    case SSM_st_NotRegul_SM1 :
      SM1_reset_act =
        (*ECU_Command).Status ==
        ON_Architecture;
```

SCADE System Integration into SCADE Suite

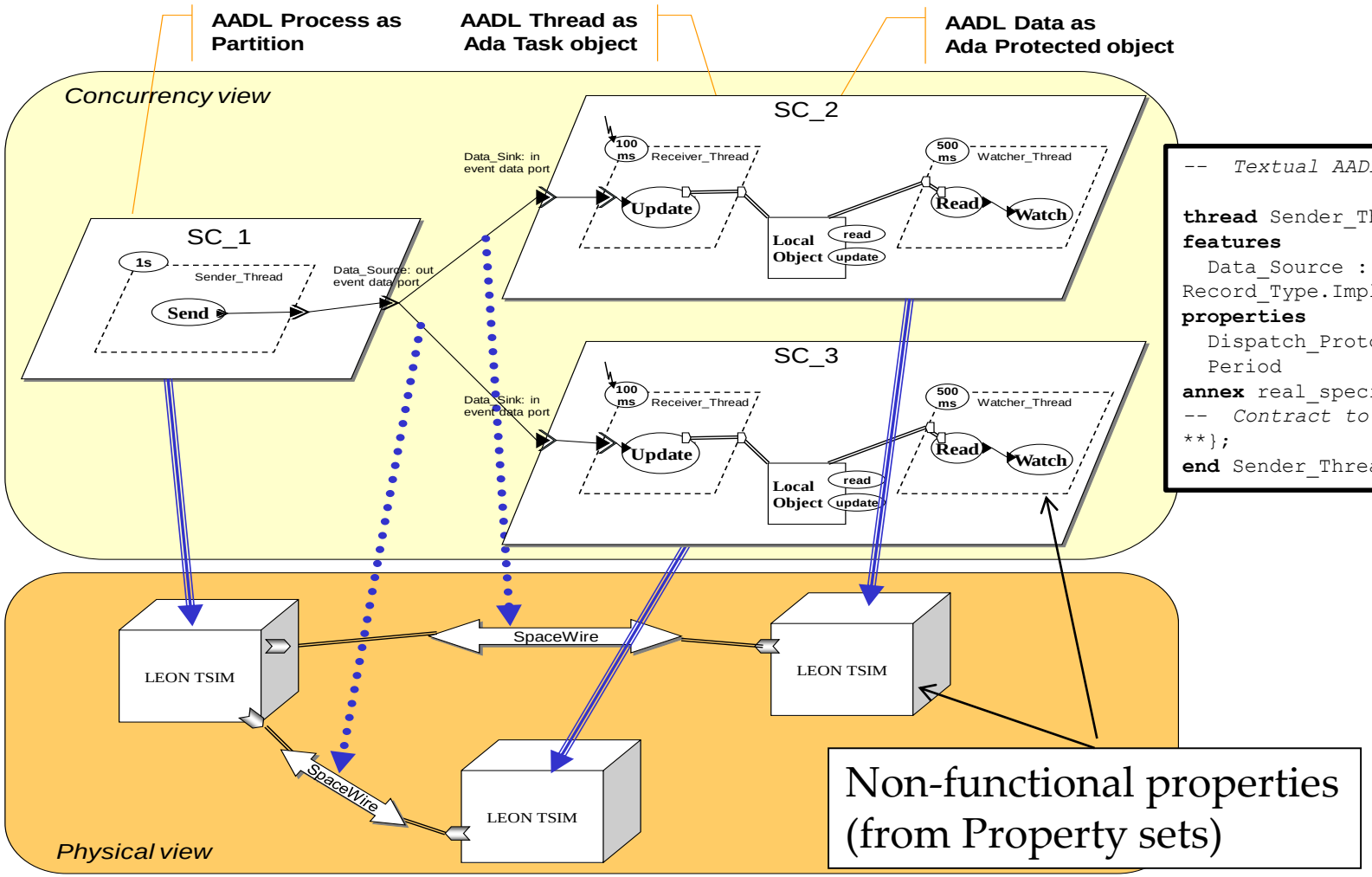
An Integrated Workflow for SW-intensive Systems



AADL:
Architecture Analysis & Design Language

- Standard promoted by SAE International, AS-2C committee, as AS-5506A
  - Version 1.0 published in 2004, v2 in 2009, v2.1 in 2012, v2.2 in 2017
  - Committee driven by inputs from the avionics and space industry
  - Academics drive analysis capability
- http://aadl.info list all resources around AADL
  - Public wiki : https://wiki.sei.cmu.edu/aadl/index.php/Main_Page
  - Include link to most research activities around AADL
- AADL is dedicated to real-time embedded domain
  - Modeling software and hardware resources for V&V
  - Extension & refinements concept to iterate down to generation
- Different representations
  - Graphical: high-level view of the system
  - Textual: to view all details
  - XML: to ease processing by 3rd party tool

AADL – outlook

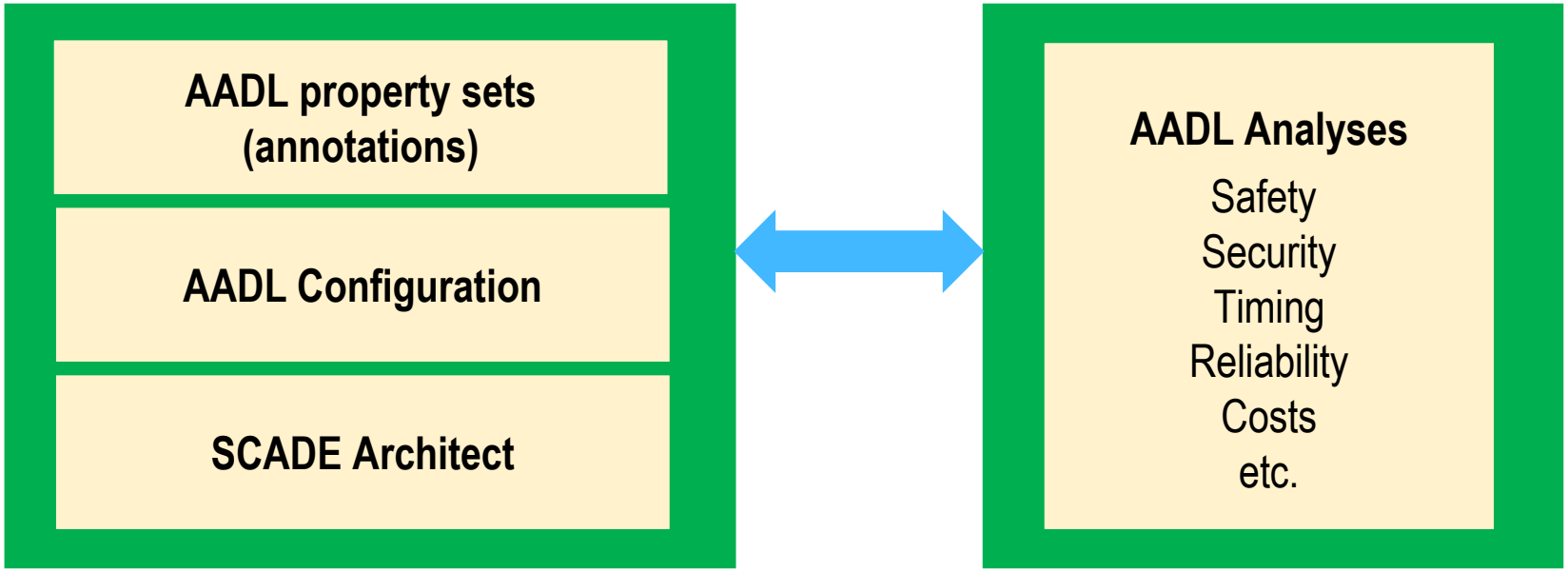



```
--  Textual AADL

thread Sender_Thread
features
  Data_Source : out event data port
Record_Type.Impl;
properties
  Dispatch_Protocol => Periodic;
  Period            => 1 Sec;
annex real_specification {**
--  Contract to be enforced
**};
end Sender_Thread;
```

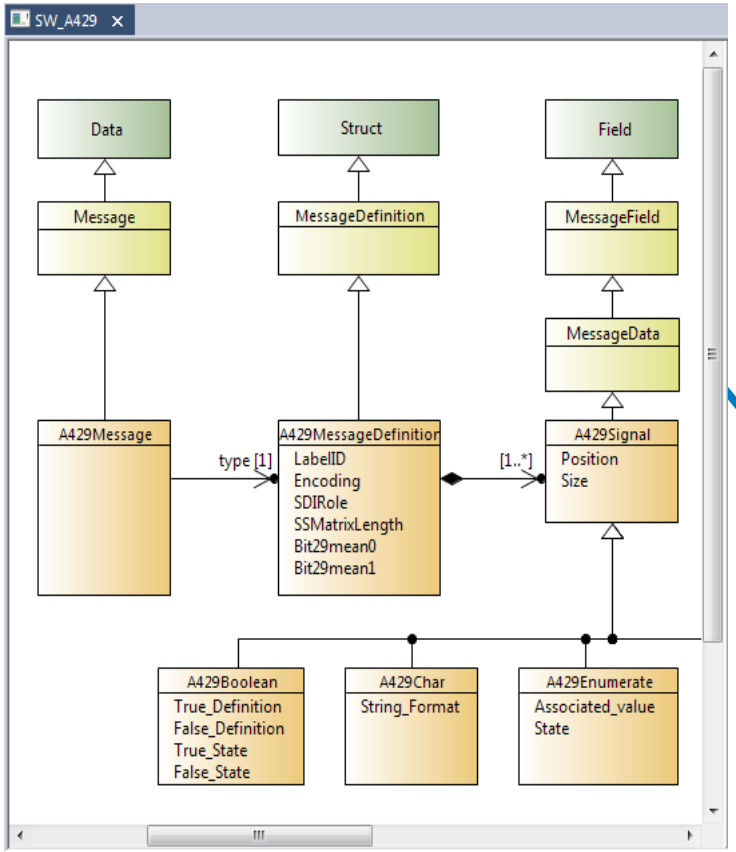
Non-functional properties (from Property sets)

SCADE AADL Solution



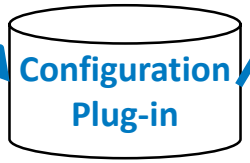
SCADE System Configuration Workflow

Specialist SCADE System Configurator

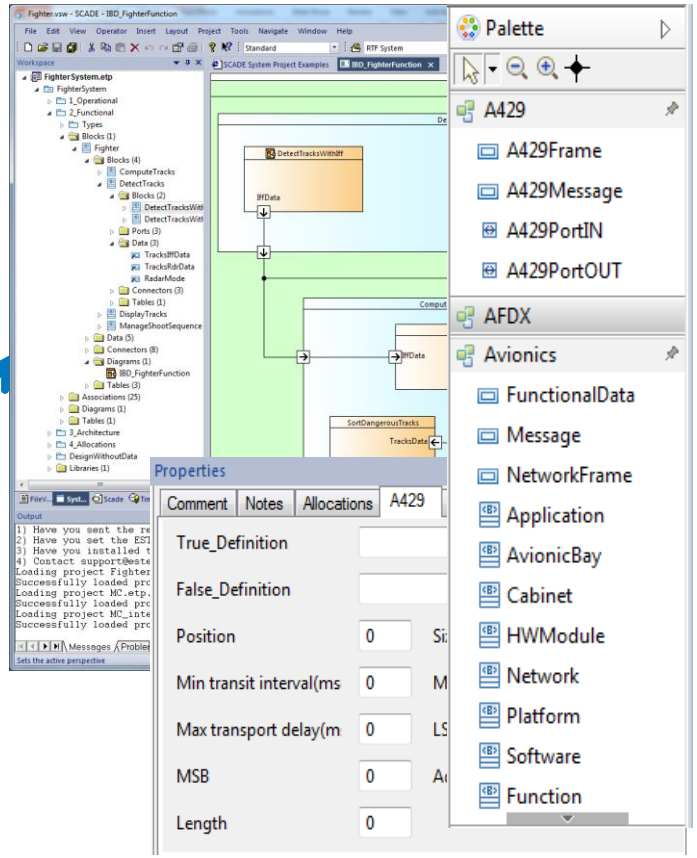


Generate

Deploy



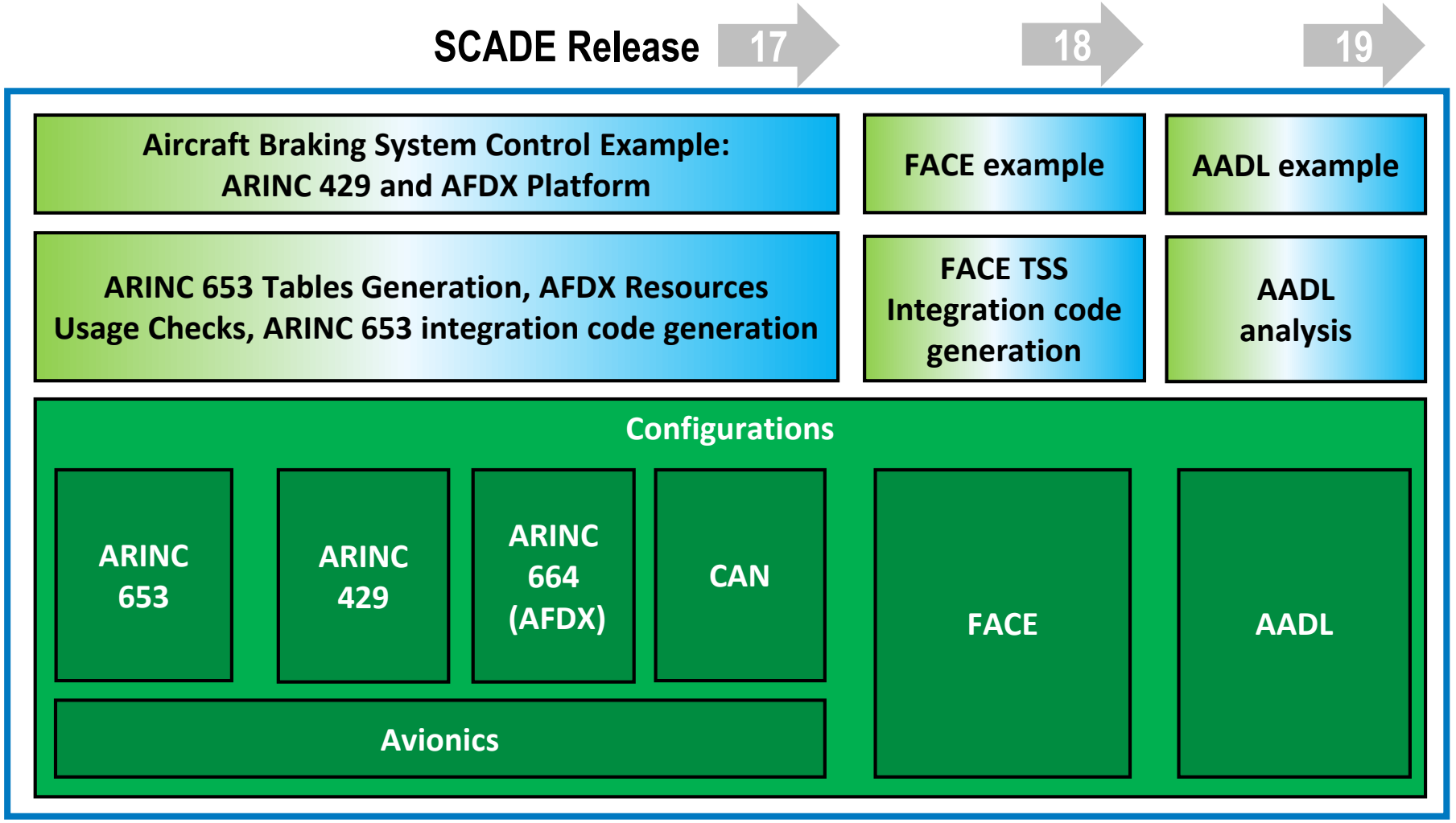
End-User SCADE System Modeler



Define customized object kinds, derived from SCADE System objects

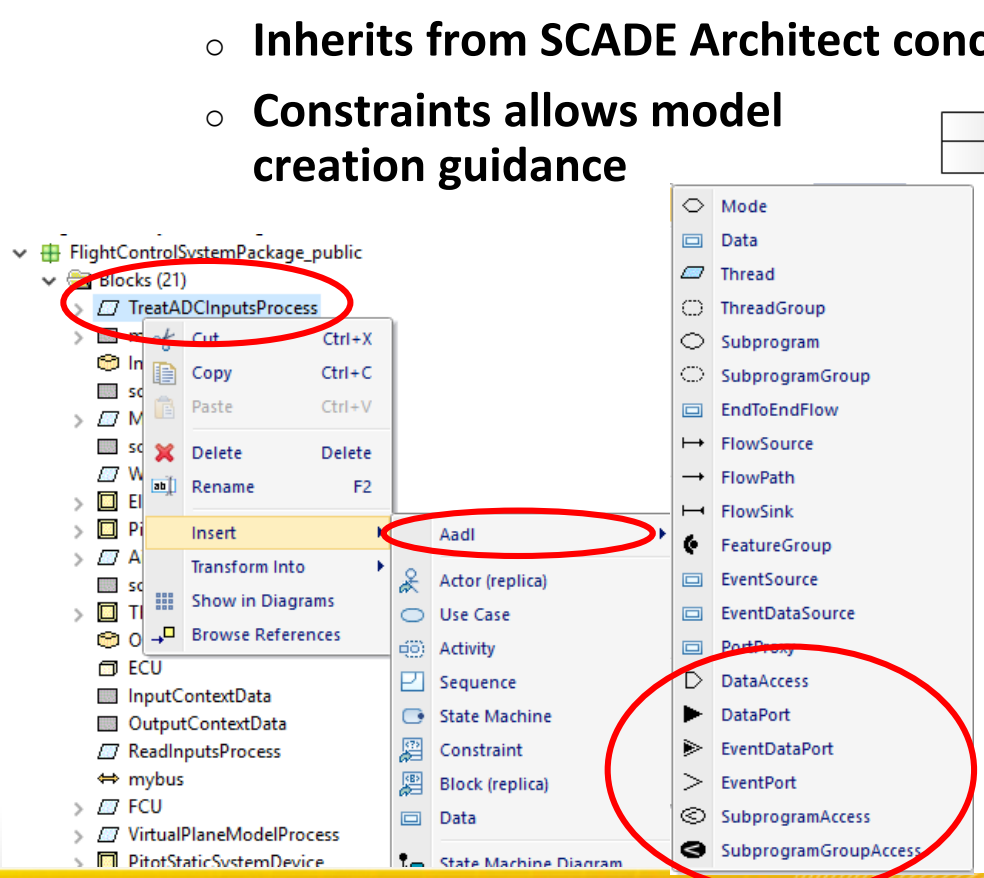
Domain specific modeler

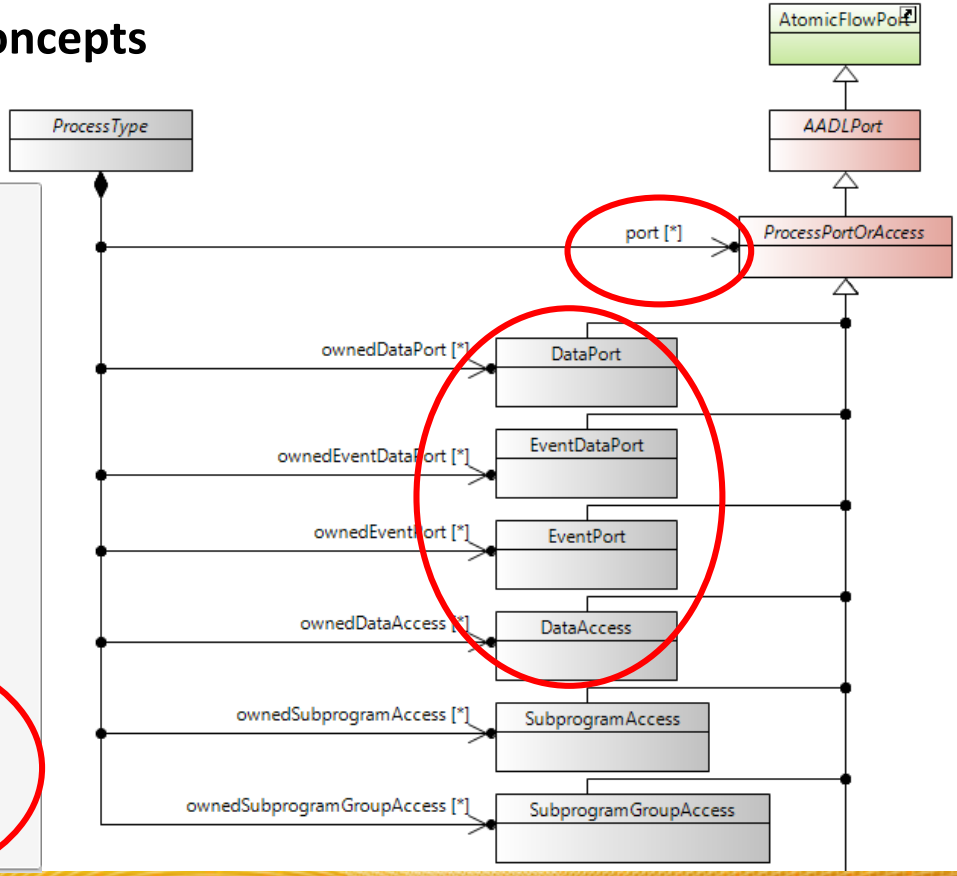
SCADE Avionics Package



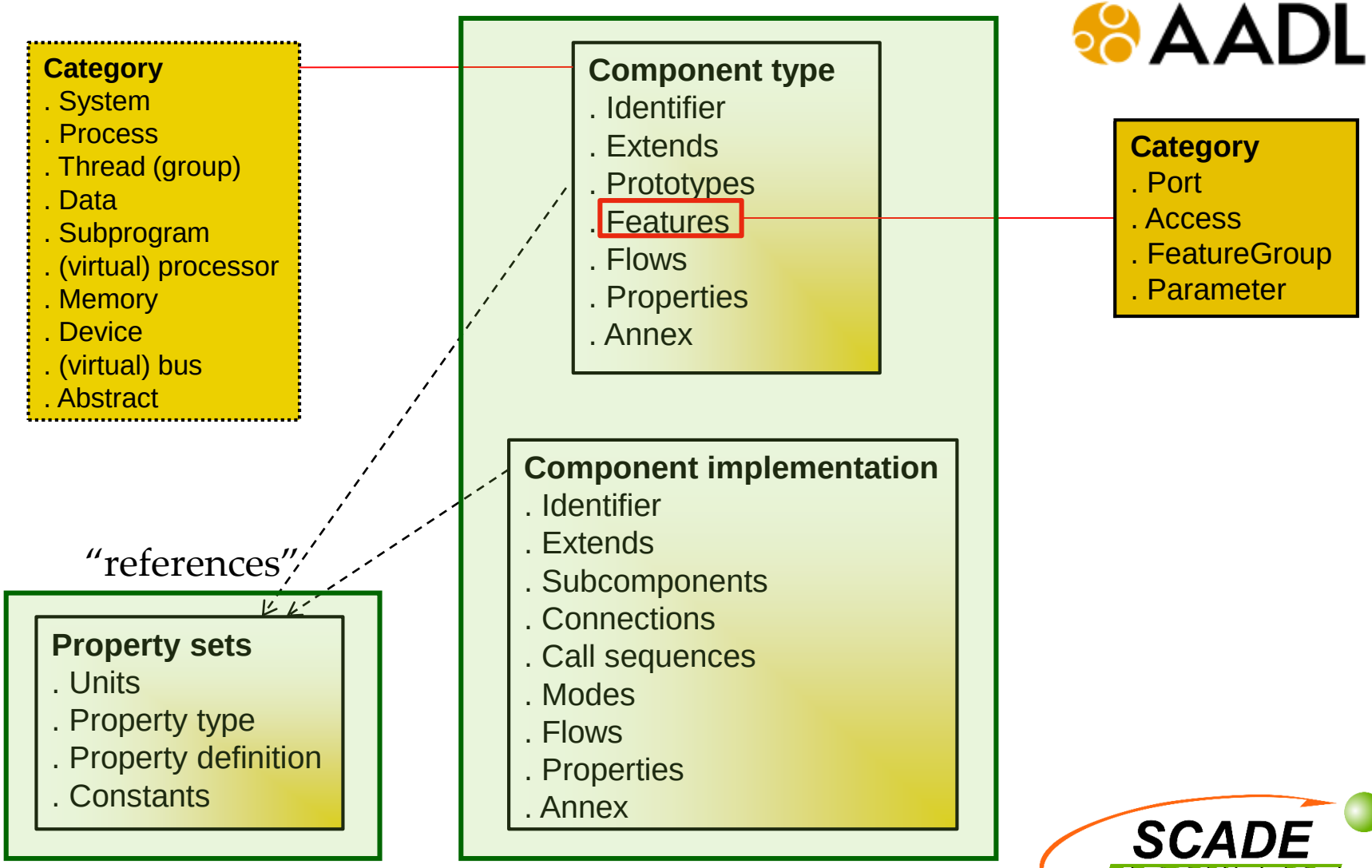
AADL meta model and SCADE Architect AADL configuration

- 1) AADL v2.2 meta model (ecore file)
  - ➔ Loaded in SCADE Architect Configurator
- 2) AADL concepts
  - Inherits from SCADE Architect concepts
  - Constraints allows model creation guidance





AADL model elements



AADL language expressiveness (& complexity)

- **AADL language**
  - **Object-oriented inheritance mechanism:**
    - *Prototypes* and *Abstract* components
      - later extended and refined into concrete category
    - *Component types* and *Component implementation*
      - An interface definition can have multiple implementations
      - But definition mandatory before specifying implementation
  - **Instantiation:**
    - *Component instances* are references to *component implementation*, that must be inlined for analysis
    - Inlining done as an explicit tool action in *OSATE* to get an instantiated model

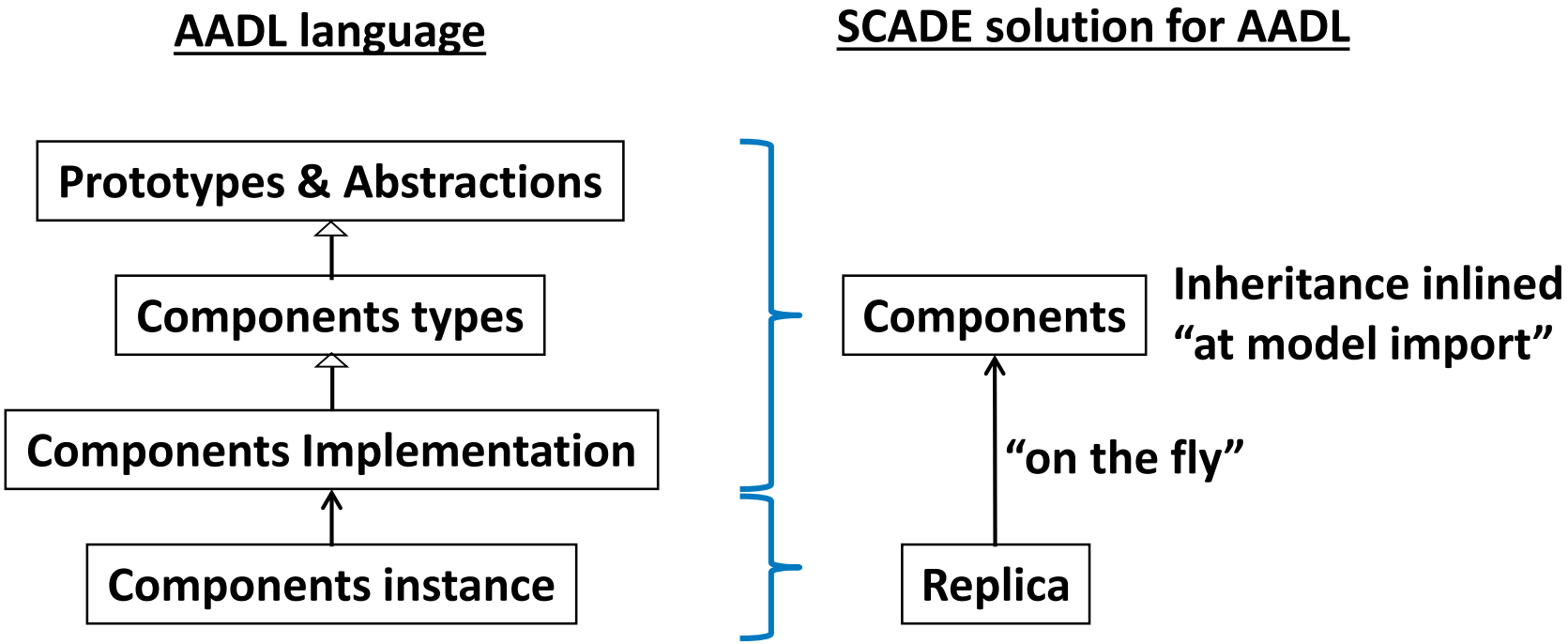
- **In SCADE: 2 simplifications**
  1. **AADL Abstraction & Inheritance inlining**
  2. **AADL instance based modeling**

SCADE solution for AADL

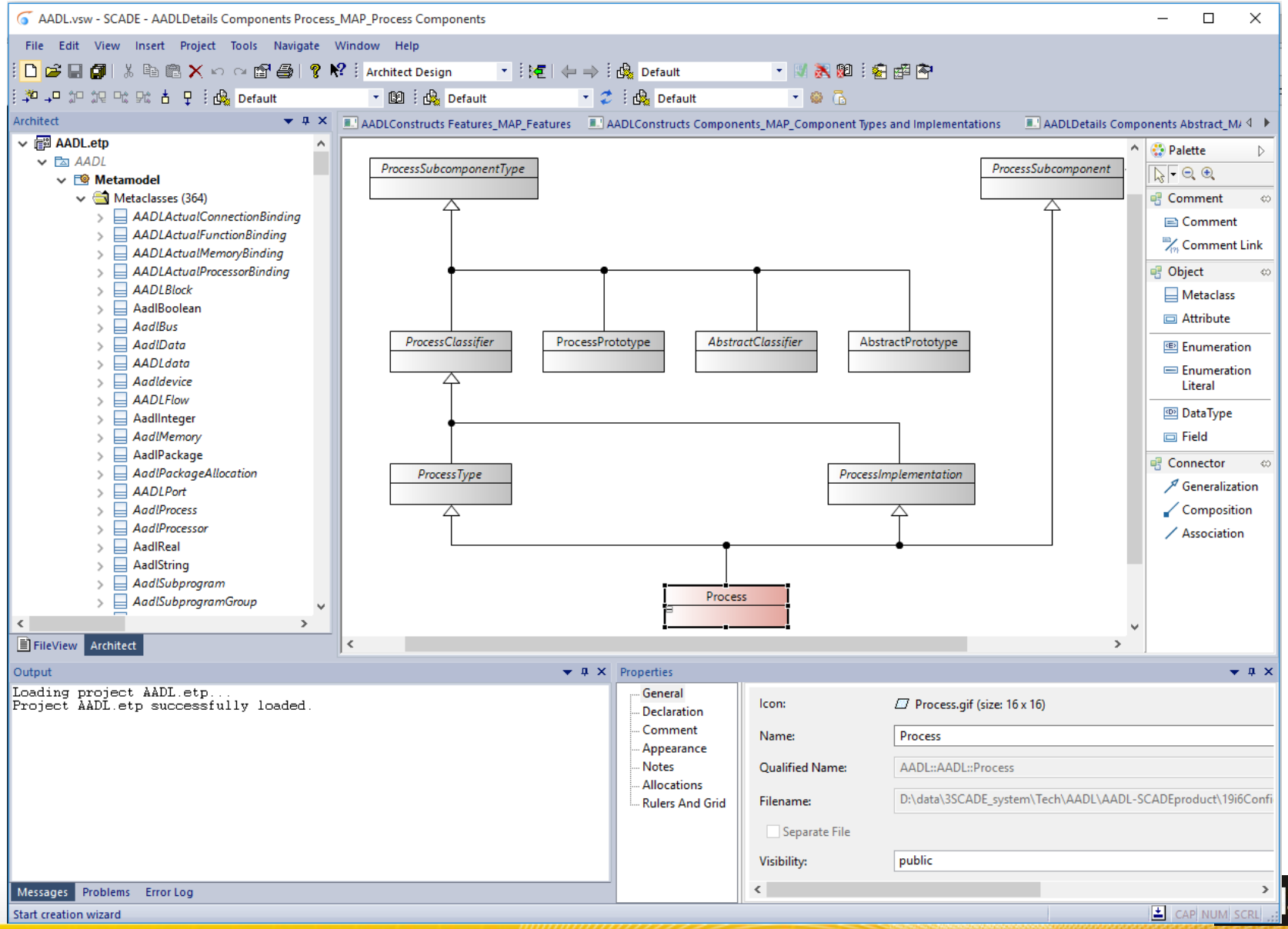
Instance based modeling

- **Benefit from SCADE Architect: Block Replica**
  - **The whole content of Block Definition is replicated in each instance (SysML parts)**
- **Support for AADL "instance based modeling"**
  - **AADL objects:**
    - "ProcessTypes" (interface only),
    - "ProcessImplementations" (content only),
    - "ProcessSubcomponents" (empty instances)
  - **Replaced by:**
    - "Process" definition: interface and full content, automatically replicated in each AADL "Process instance"
  - **Consequences**
    - Limitation: only "one Implementation per Type"
    - But much simpler model understanding for end user

AADL language expressiveness (& complexity)



SCADE AADL configuration



AADL Property sets

- **Property: Typed attribute, associated to components**
- **Property sets: group property definitions.**
  - **Property sets part of the standard, e.g. `Communication_Properties`**
  - **Or user-defined, e.g. for new analysis**

```
process MFDProcess
  features
    MCPaltitude: out data port scade_real;
    MCPspeed: out data port scade_real;
    AutoPilot: out data port scade_bool;
  flows
    f0: flow source MCPaltitude {Latency => 5 ms .. 10 ms;};
    f1: flow source MCPspeed {Latency => 5 ms .. 10 ms;};
    f2: flow source AutoPilot {Latency => 5 ms .. 10 ms;};
  properties
    Period => 25 ms;
end MFDProcess;
```

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;
```

```
--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);
-- …
```

```
-- …
flows
    f0: flow source MCPaltitude {Latency => 5 ms .. 10 ms;};
    f1: flow source MCPspeed {Latency => 5 ms .. 10 ms;};
    f2: flow source AutoPilot {Latency => 5 ms .. 10 ms;};
```

AADL Property sets

```
-- Predeclared_Property_Sets.aty
Notes-Types { Esterel-Technologies } DEFINITIONS ::=
BEGIN
Latency ::= SEQUENCE OF { SEQUENCE {
  annot_object OID, name STRING,
  information {
    Latency_min INTEGER,
    Latency_max INTEGER,
    Latency_Time_Units ENUM {NT_ENUM_VALUES {"ps", "ns => ps * 1000",
"us => ns * 1000", "ms => us * 1000", "sec => ms * 1000", "min => sec
* 60", "hr => min * 60"}} } } }
```

```
Annotation-Rules { Esterel-Technologies }  DEFINITIONS ::=
BEGIN
aadl-flow ::= {
  {Latency F 0 1},
  {Compute_Execution_Time F 0 1},
  {Period F 0 1},
  {Transmission_Time F 0 1},
  {Dispatch_Protocol F 0 1},
  {Data_Size F 0 1}
}
```

<configuration>-<meta class>

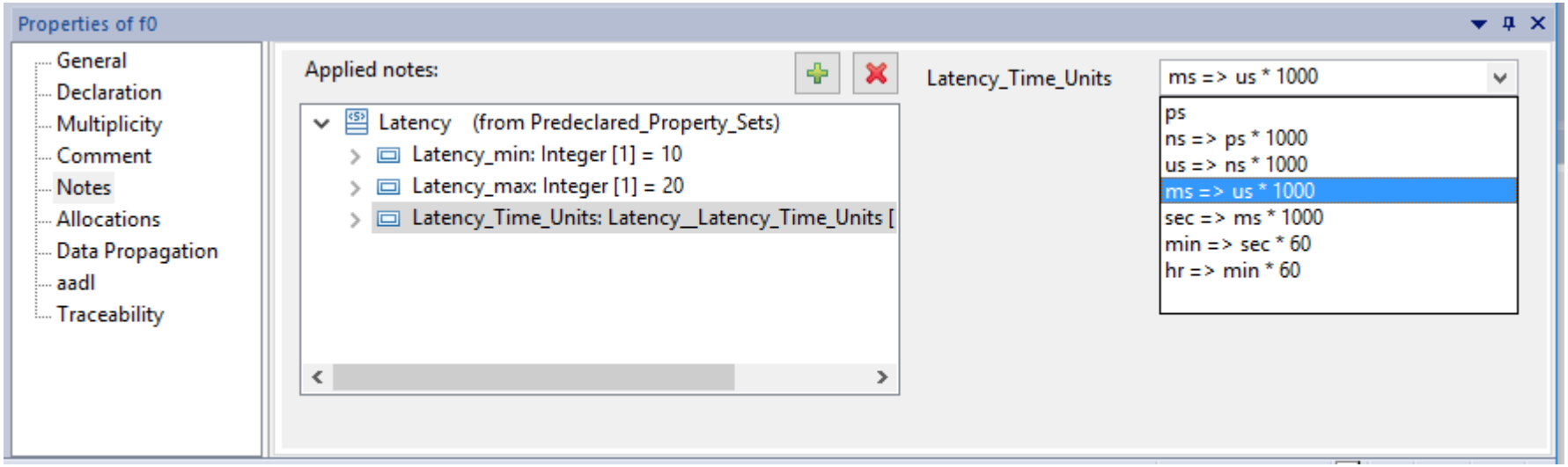
Notes-types allowed for this <meta class>

AADL Property sets

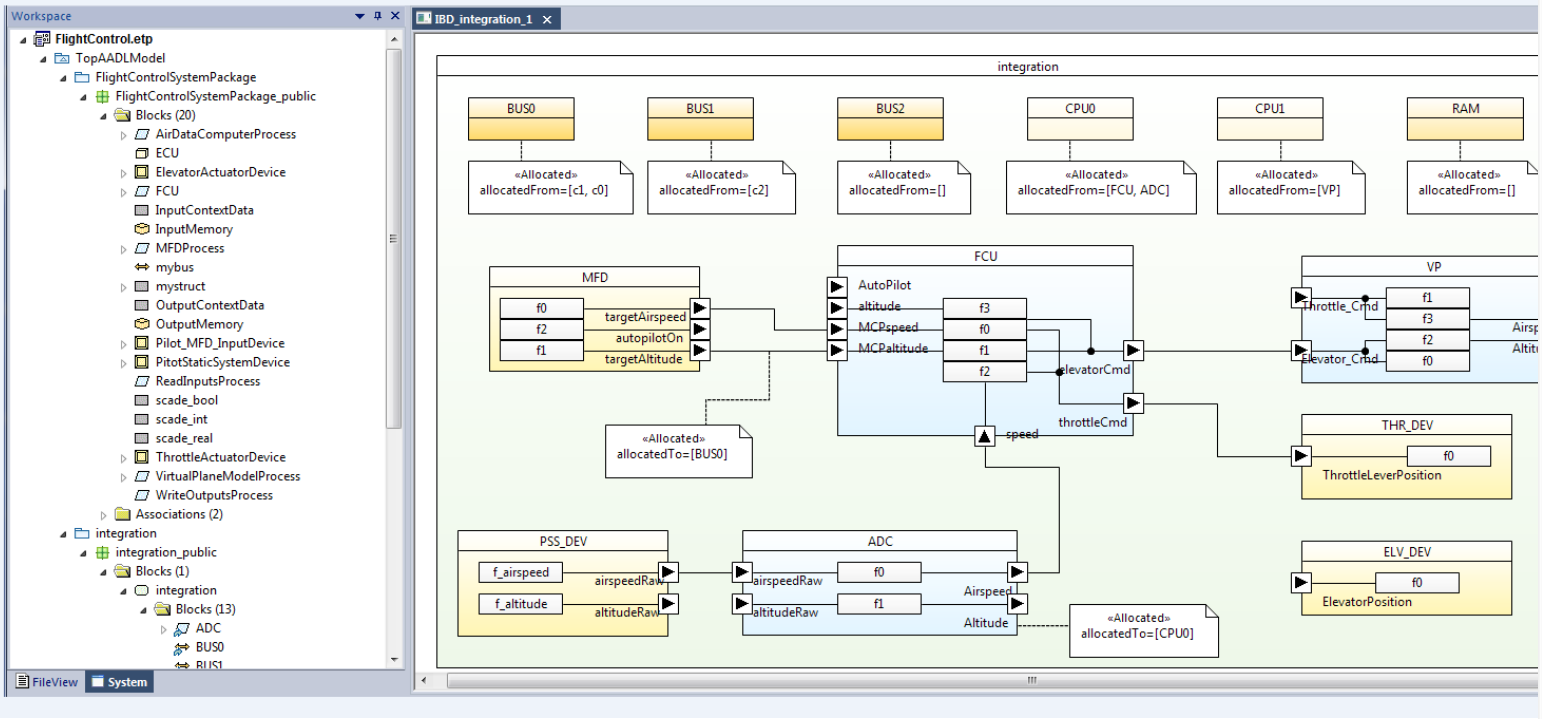
- **Automated conversion**

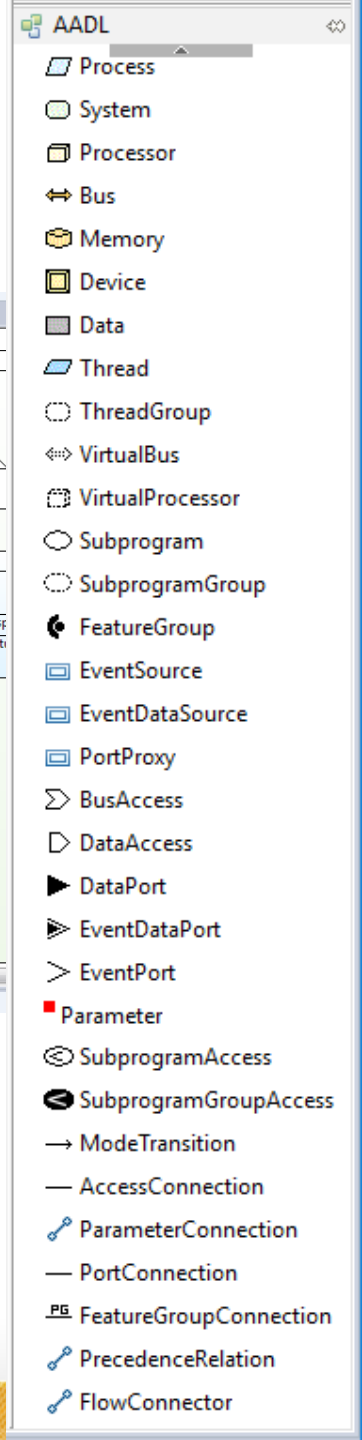
  <property set>.aadl → <SCADE note types>.aty

- **Benefits**
  - **Reused SCADE IDE matured technology**
  - **Automated GUI to set properties on objects in a model**



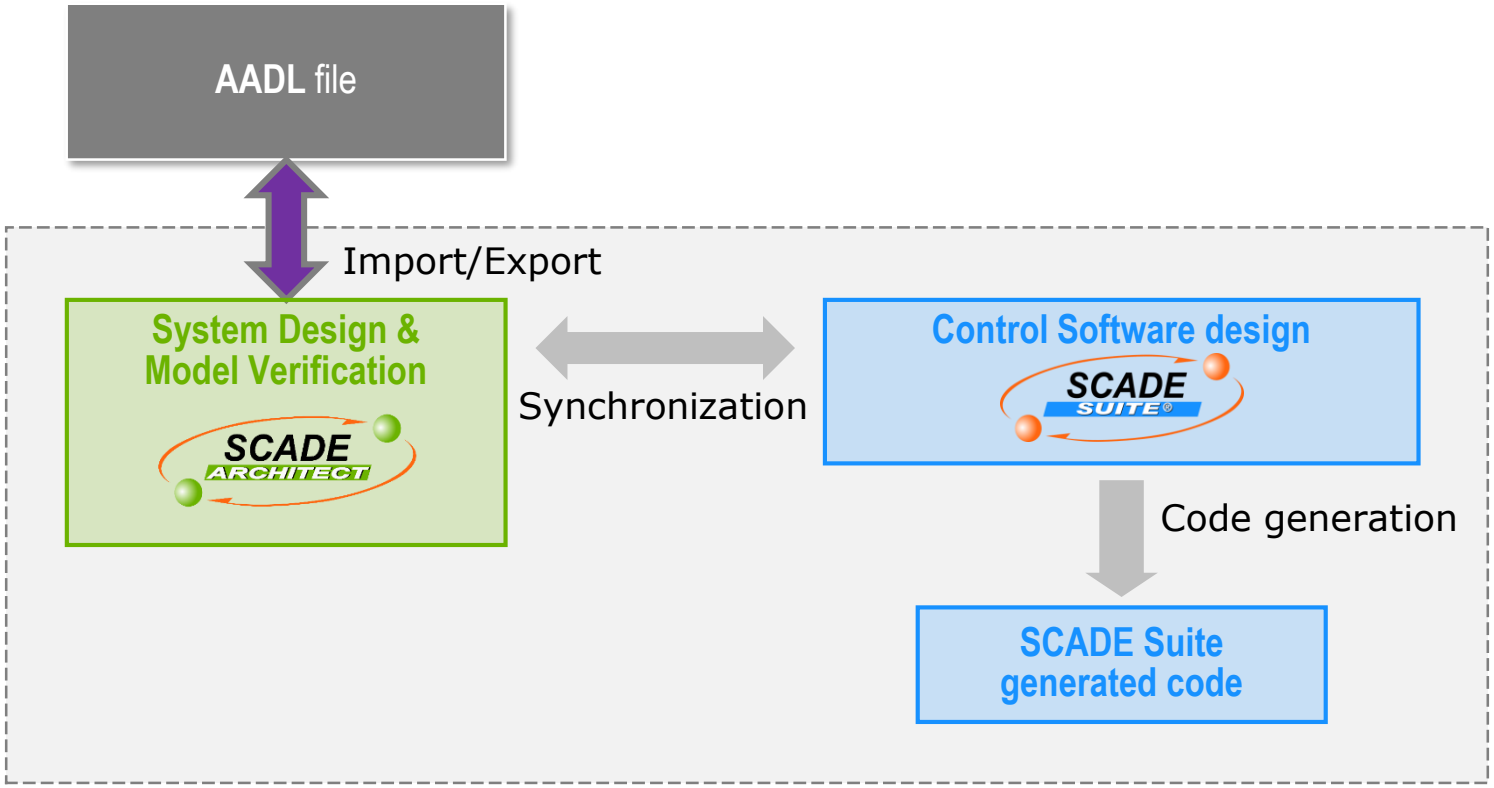
AADL example (Sept 2016)





- **Additional features expected in SCADE 19**
  - **Shapes in Diagram**

SCADE AADL solution: Workflow

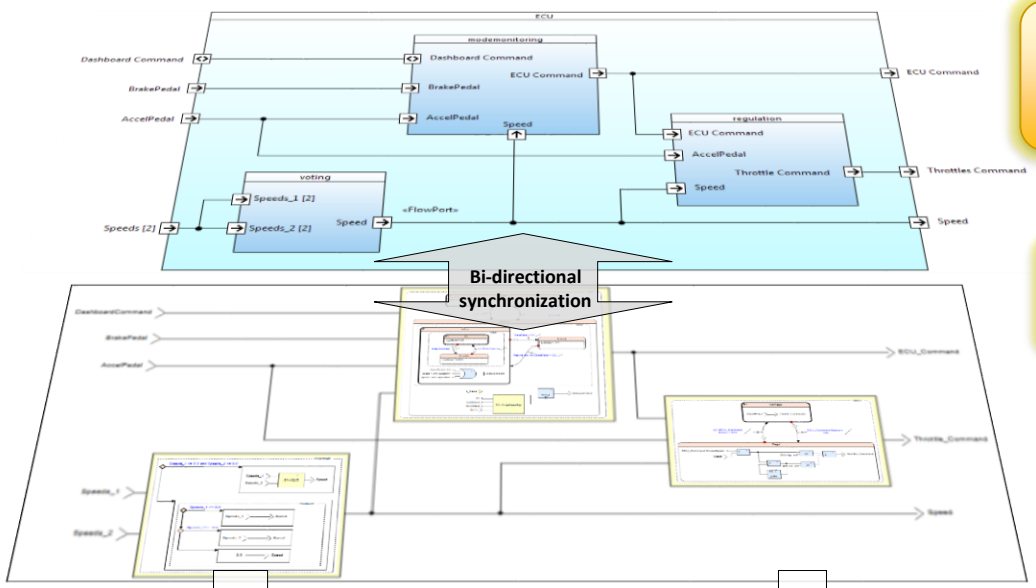


System/Software Architecture Synchronization

SW Architecture

SW Design

SW Coding



Seamless workflow from system architecture down to SW code generation

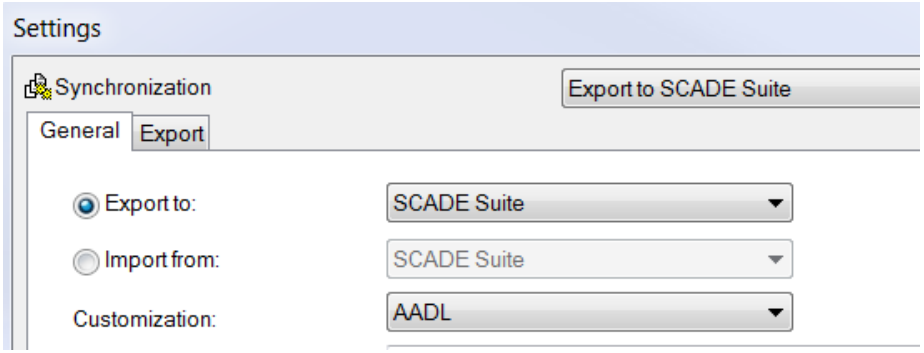
Reuse legacy SCADE Suite components in new architectures

Ensure design consistency with concurrent work & engineering iterations

Custom synchronizations
- for AUTOSAR models
- for FACE models
- For AADL models

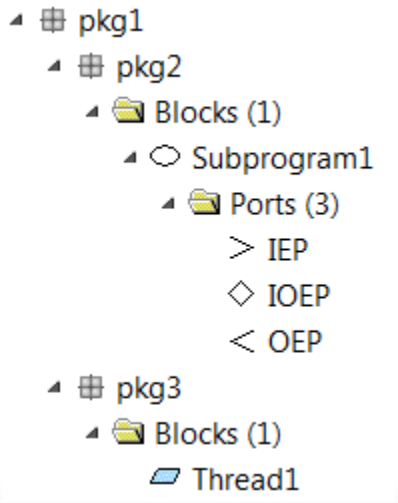
Synchronization SCADE AADL – SCADE Suite

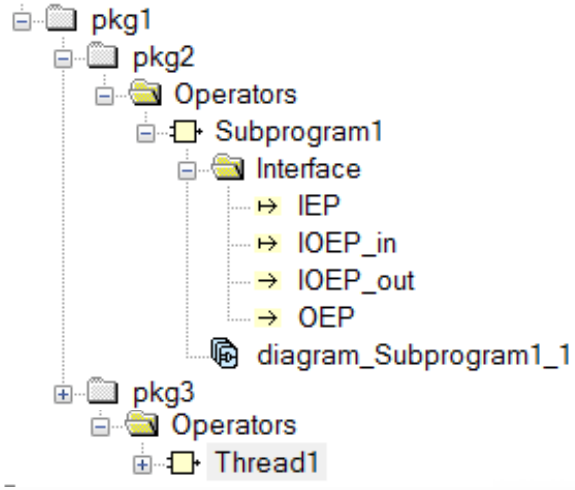
- **Customization of the SCADE Architect-SCADE Suite synchronization**



Synchronization SCADE AADL – SCADE Suite

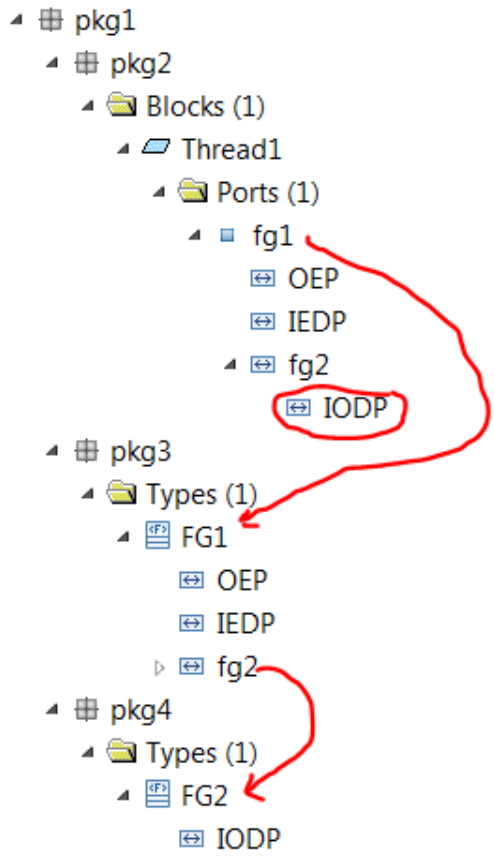
- **SCADE Suite ←→ AADL:**
  - **Selected operators ←→ choice to Thread and Subprogram (default Thread)**
  - **In/Out variable ←→ Port (default DataPort)**
    - DataPort → In/Out variable <type>
    - EventPort → In/Out boolean variable
    - EventDataPort → In/Out variable {EDP_Event:bool, EDP_Data:<type>}
    - Port ArrayDimension N → N In/Out variable

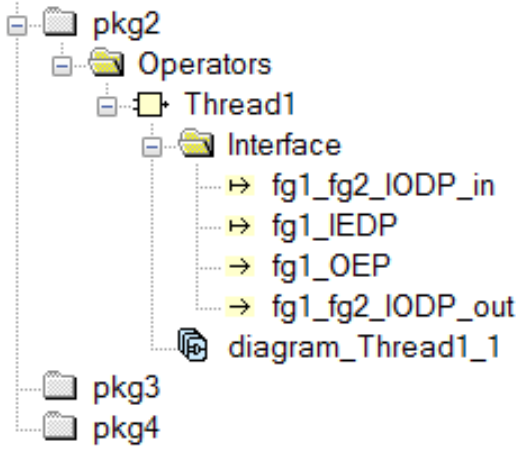


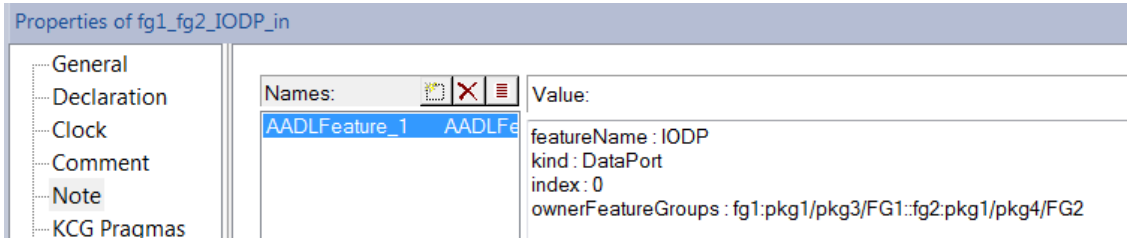


Synchronization SCADE AADL – SCADE Suite

FeatureGroup





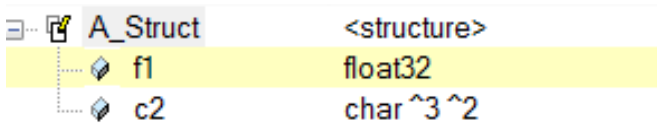


Synchronization SCADE AADL – SCADE Suite

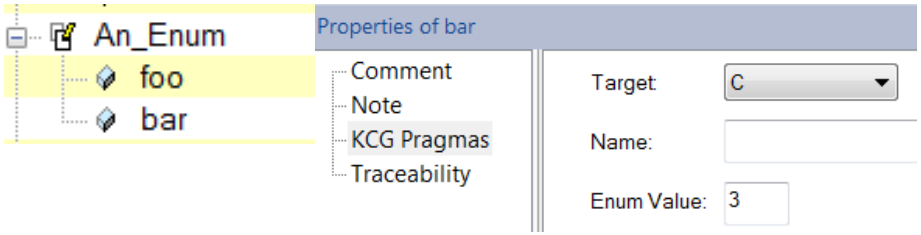
Datatypes

AADL base type	Suite primitive type
Boolean	bool
Float_64	float64
Unsigned_64	uint64
Character	char
Float_32	float32
Unsigned_32	uint32
Integer_64	int64
Integer_32	int32
Unsigned_8	uint8
Integer_16	int16
Unsigned_16	uint16
Integer_8	int8

```
data A_Struct
  properties
    Data_Model::Data_Representation => Struct;
end A_Struct;
data implementation A_Struct.impl
  subcomponents
    f1 : data Base_Types::Float_32;
    c2 : data Base_Types::Character [2][3];
end A_Struct.impl;
```




```
data An_Enum
properties
  Data_Model::Data_Representation => Enum;
  Data_Model::Enumerators => ("foo", "bar");
  Data_Model::Representation => ("00", "11");
end An_E
```



SCADE AADL design and analysis flow

- **Using the AADL Configuration,**
  SCADE Architect becomes an AADL modeler
  - **Defines the architecture of the system in terms of processes, threads, processors, buses, etc.**

- **Using the AADL Annotations,**
  SCADE Architect supports AADL property sets
  - **The AADL model, with these information, can be read in tool such as OSATE or other AADL tools which performs analyses (schedulability, safety, etc.)**

- **Using the SCADE Architect-SCADE Suite Synchronization,**
  implement the software components with SCADE Suite

Conclusion: SCADE solution for AADL

- **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 synchronization with SCADE Suite for SW component development, verification & certification
  - Traceability through SCADE ALM gateway

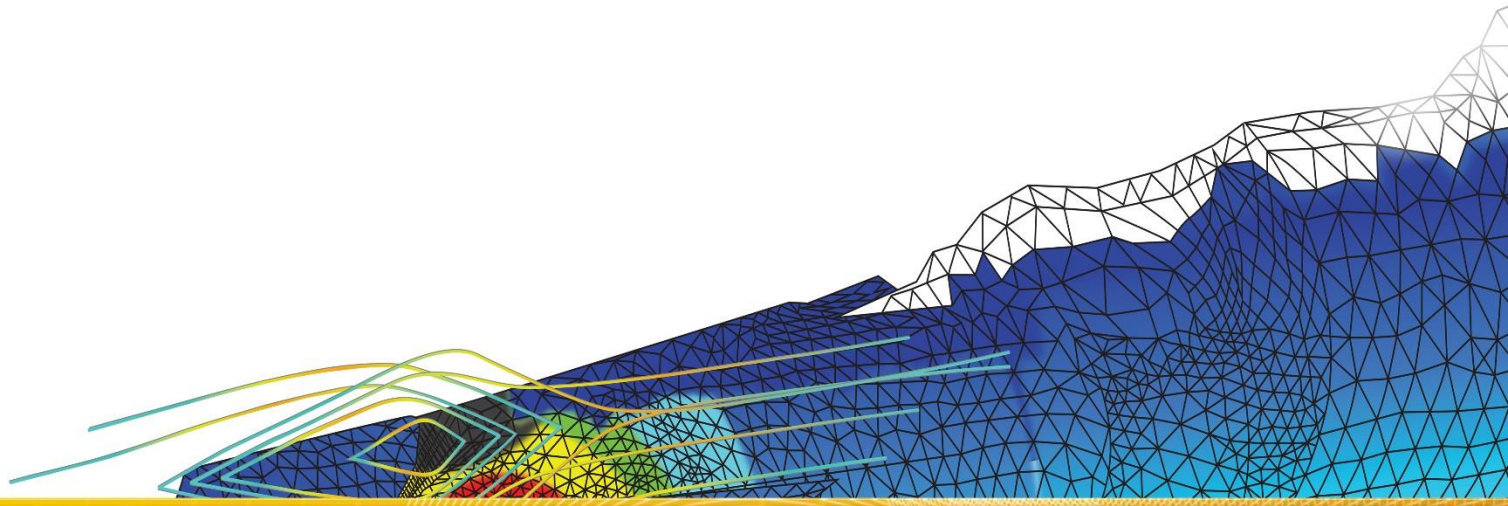
Next steps

- **Development plan**
  - **Finalize AADL core and property set ➔ by late September**
  - **Look & feel enhancement (shapes & palettes) ➔ by late October**
  - **Product: SCADE 19, by late January 2018**

- **Next features under study**
  - **Join usage of AADL, FACE and SCADE Avionics Configurations**
  - **Connect AADL error Annex to ANSYS medini analyze**

ANSYS®

ARINC 664 part 7 - AFDX



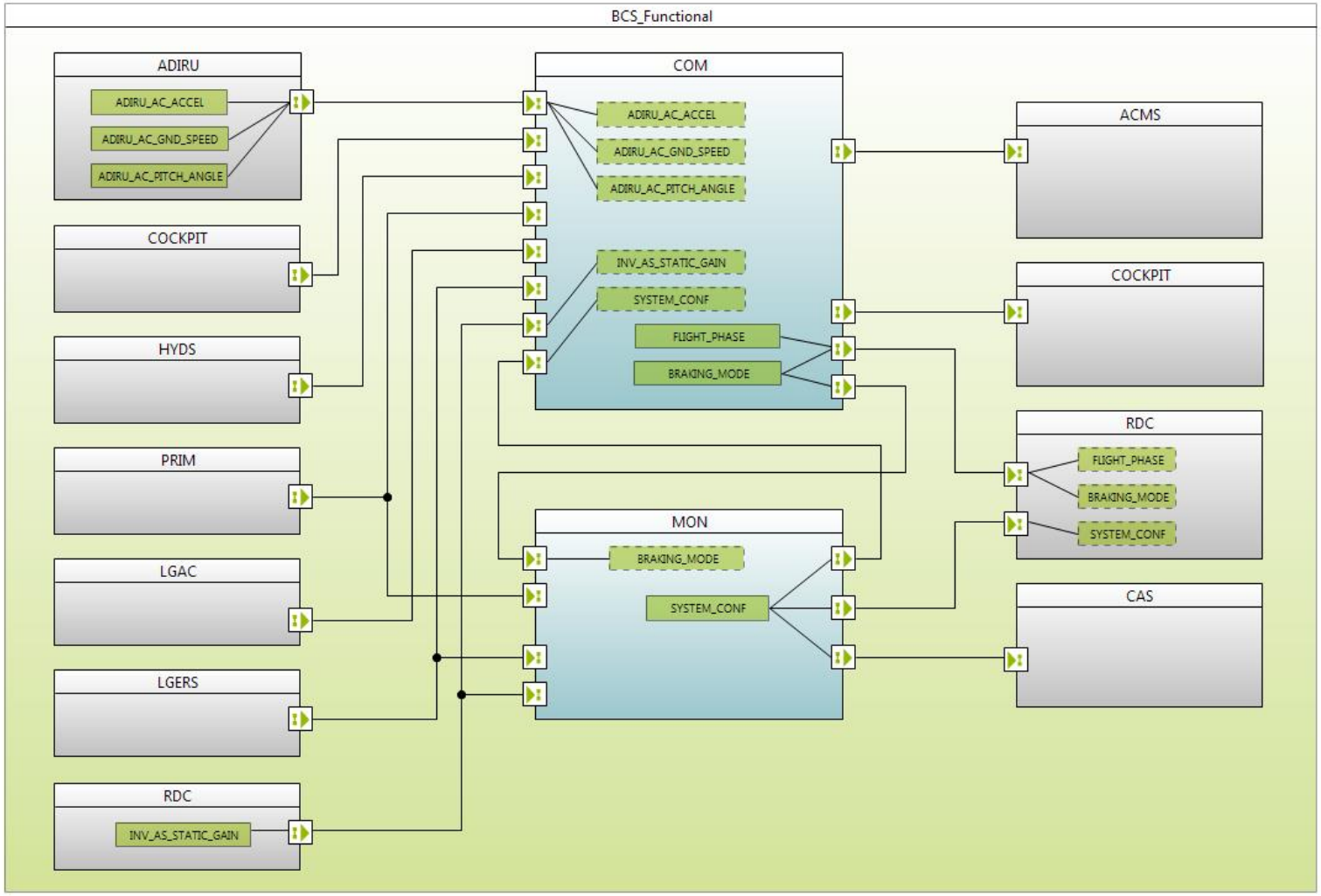
Communication Protocols

- **Avionics Full-Duplex Switched Ethernet (AFDX):**
  - **Done from UDP/IP/Ethernet protocol using a specific Ethernet switch**
  - **Network protocol, with active switches**
    - Have complex packets of data
    - VL (Virtual links) are seen as platform data
    - Software data use virtual links to circulate inside the network
  - **Specific implementation of ARINC 664 part 7 (Aircraft Data Network) for commercial aircraft**
  - **High Speed Commercial Ethernet with provisions for guaranteed**
    - 1GB/s, 100Mb/s, 10Mb/s implementations available
  - **3 Types of Network elements**
    - End Systems (E/S)
    - Switches
    - Links

Functional Architecture Design

- **Represents functional dataflow communications (exchanges of data between functions)**
- **Manages data by import/export tables**
- **Propagates data in functional architecture**

Functional Architecture



Example

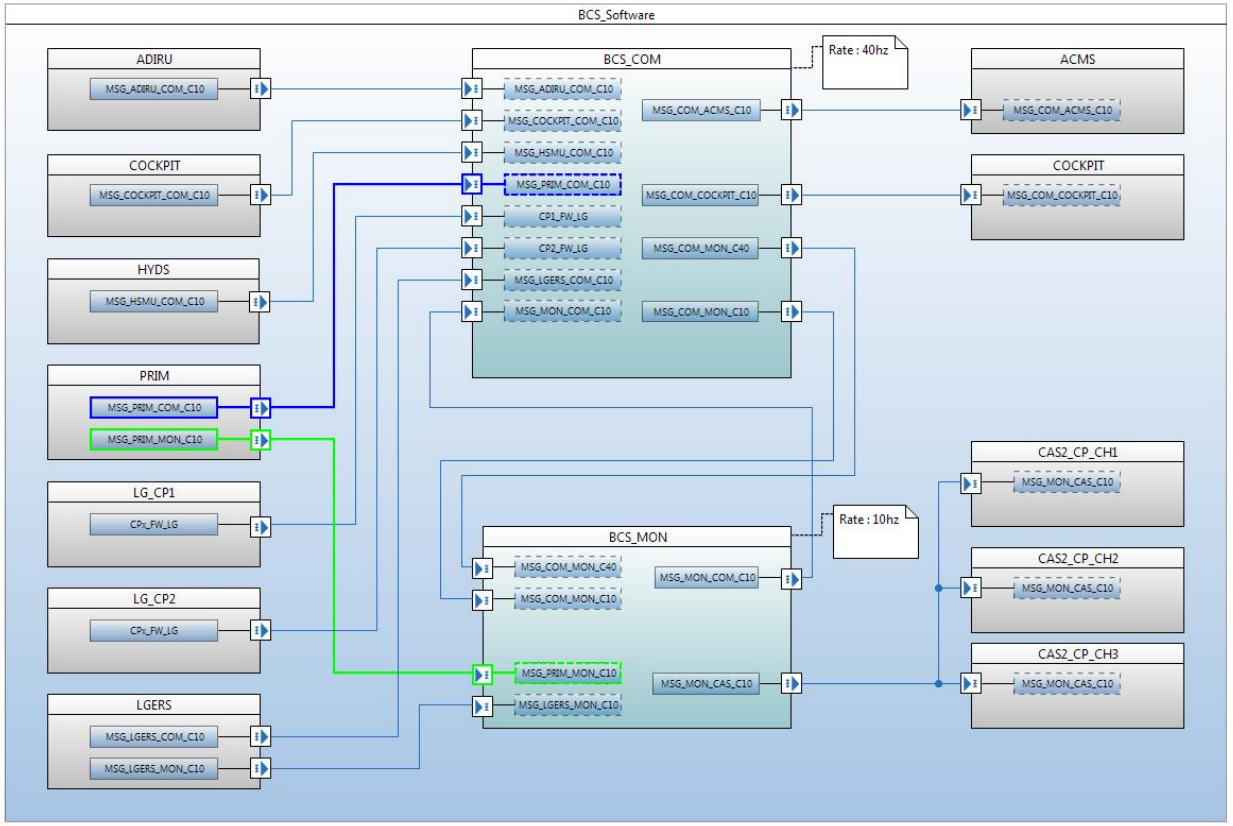
Software Architecture Design

- Gathers functions in Software components
- Defines ARINC 429, ARINC 664-P7 (AFDX), or CAN message exchanges between software components
- Propagates messages in logical architecture (software layer)
- Allocates functions to software components
- Allocates functional data onto ARINC 429, AFDX and CAN messages

Software Architecture

Example

Show AFDX messages exchanges between software components



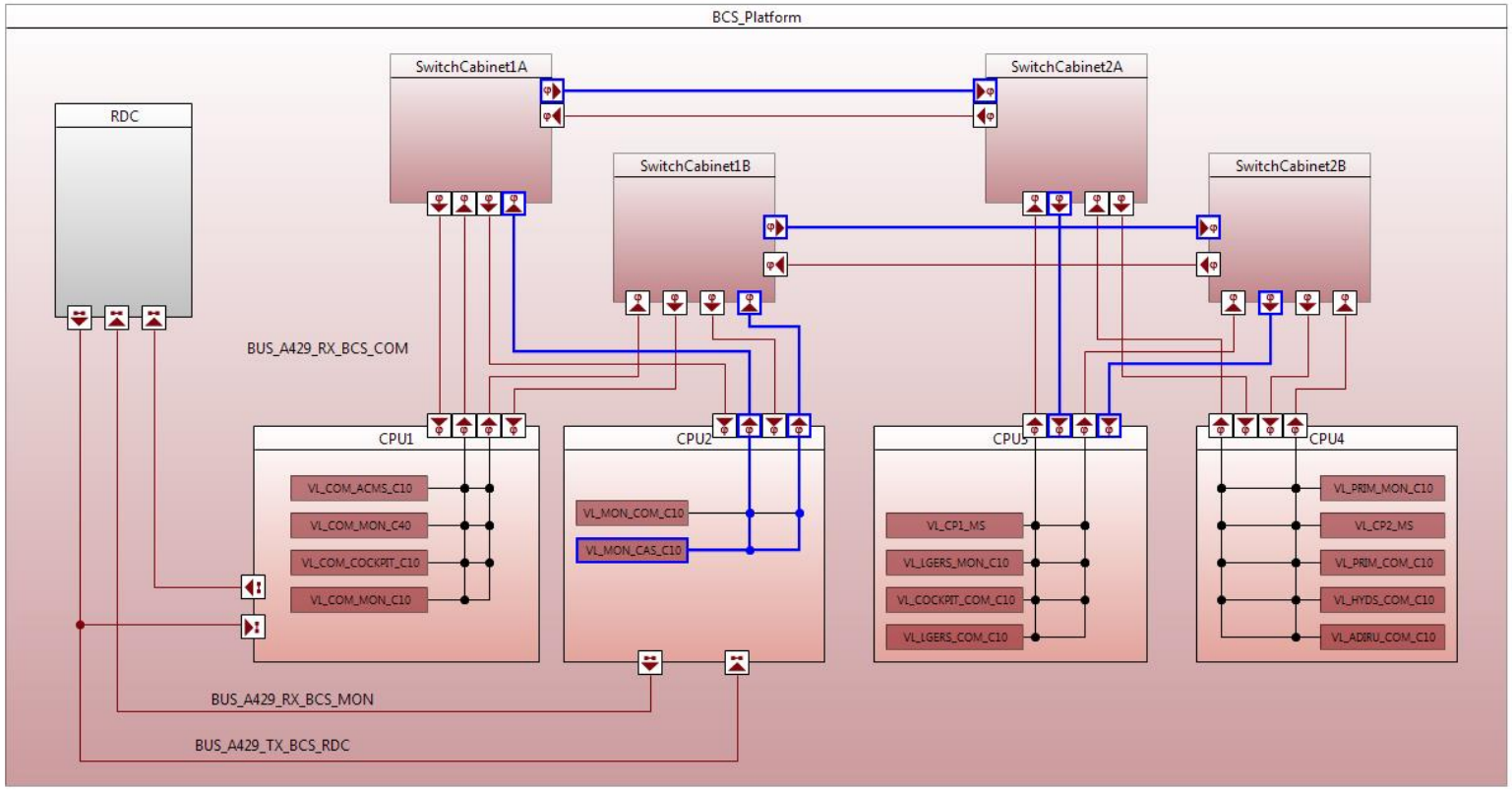
Dependent Platform Architecture Design

- Represents computing units, buses and switches
- Allocation of Software to hardware components
- Defines frames and virtual links (ARINC 664-P7 communication) by software data (messages) to propagate inside the network

Example

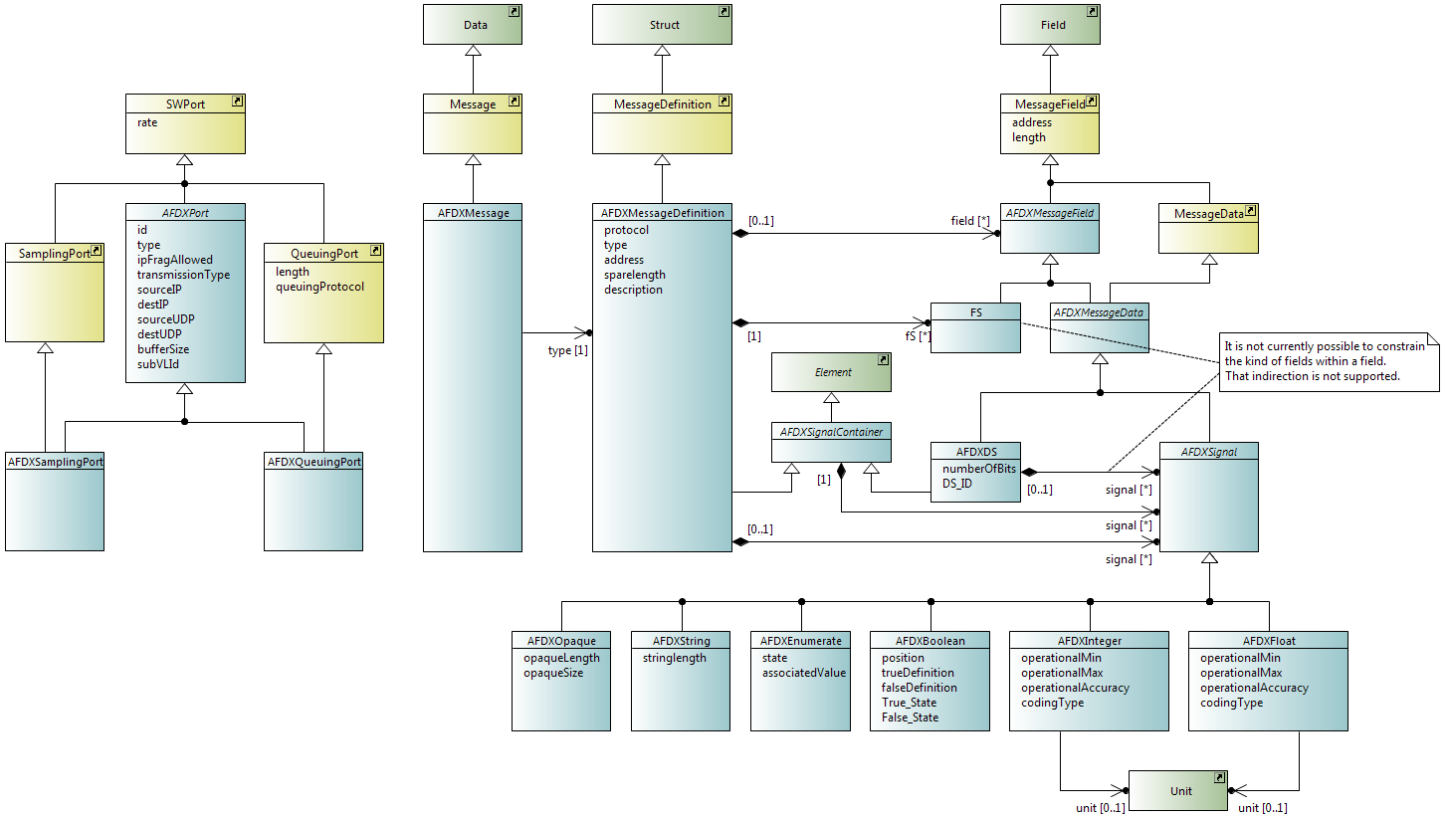
Dependent Platform Architecture

Show AFDX Switches and Virtual Links exchanges



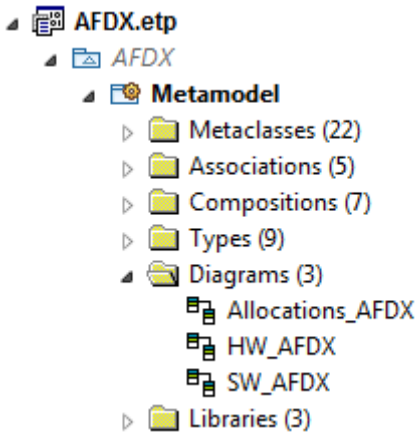
AFDX Configuration

- The metaclasses introduced by this configuration are
- displayed in blue color in metamodels



AFDX Configuration

- The SW_AFDX diagram shows the metaclasses defined for software architecture design
- The HW_AFDX diagram defines the elements for platform architecture design
- The Allocations_AFDX diagram shows the Allocation metaclass specializations between software and platform levels



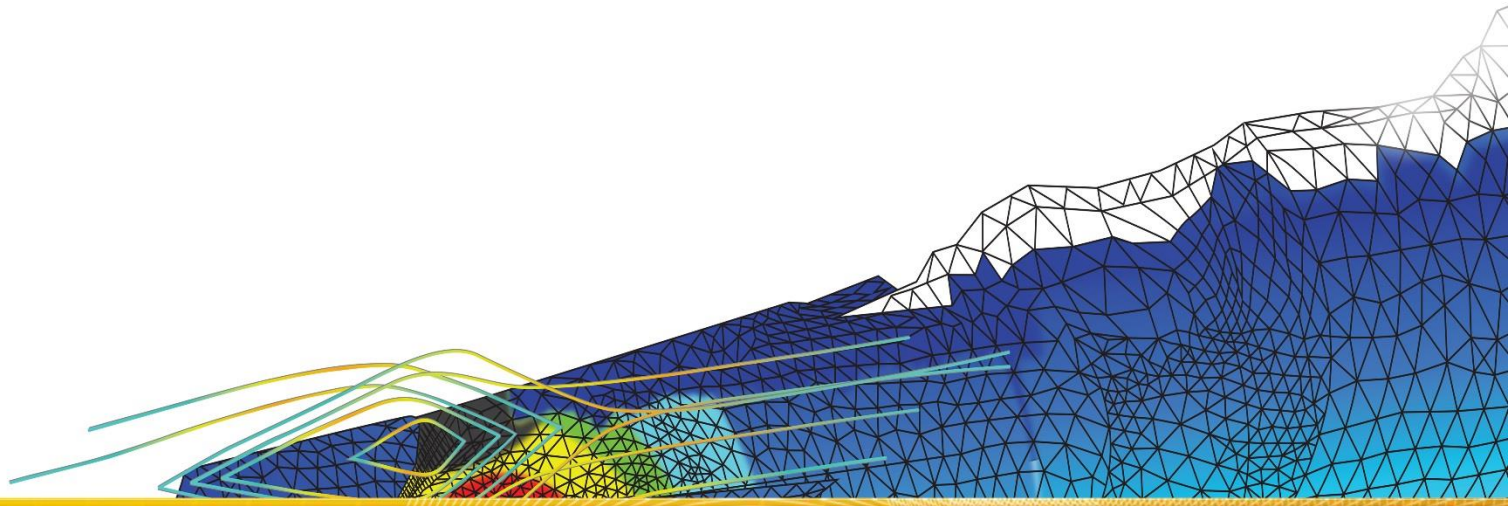
AFDX Configuration

- AFDX Data composed of Data Set (AFDXDS) that group data
  - Each DS associated to a Functional Status (FS), gives a hint on the validity of the group of data
  - Message can hold until 4 DS in the same time
  - Communications inside an AFDX network are often designed with only DS as base elements
  - FS cannot be allocated to functional data

- AFDXPort or AFDXMessage (i.e. software-level data) are mapped to VL (HWData) representing the Virtual Link paths between one emitter and N receivers

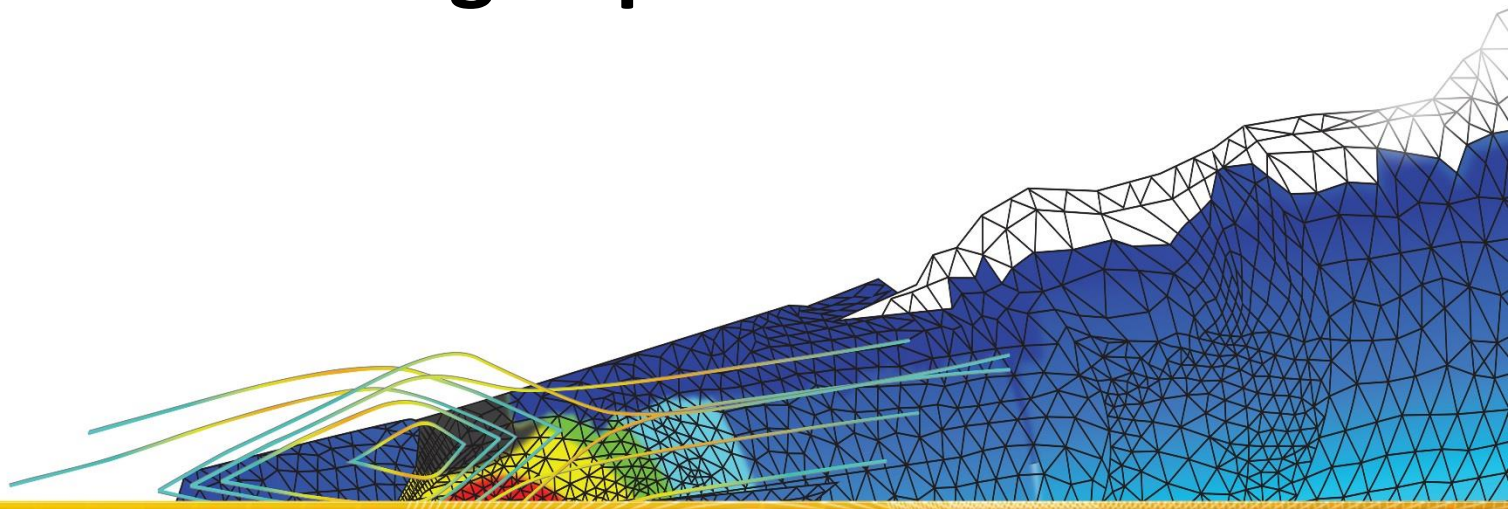
ANSYS®

Thank you!



ANSYS®

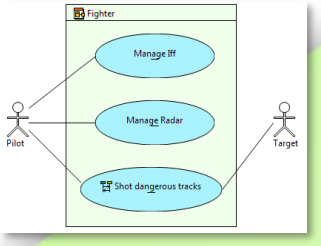
SCADE Architect Domain-Specific Modeling Capabilities



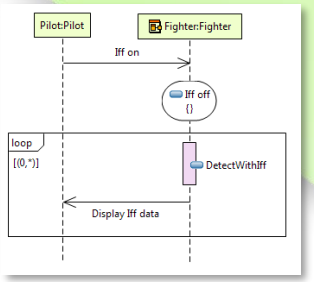
SCADE Architect

SCADE ARCHITECT

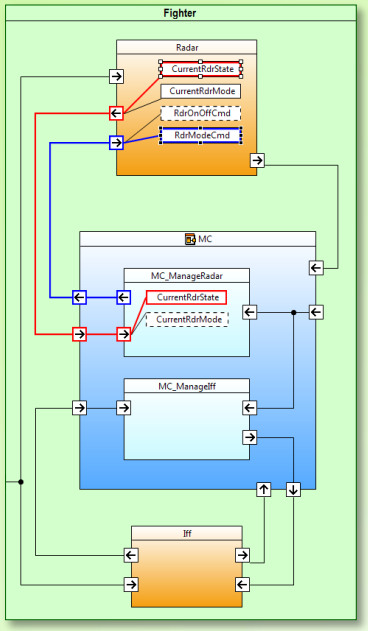
Embedded System Design



Operational Requirements Analysis



ANALYZE



Architecture Design & Data Propagation

DESIGN

System Model Checks

System Model Diff/Merge

VERIFY

System / Software Bi-directional Sync Up

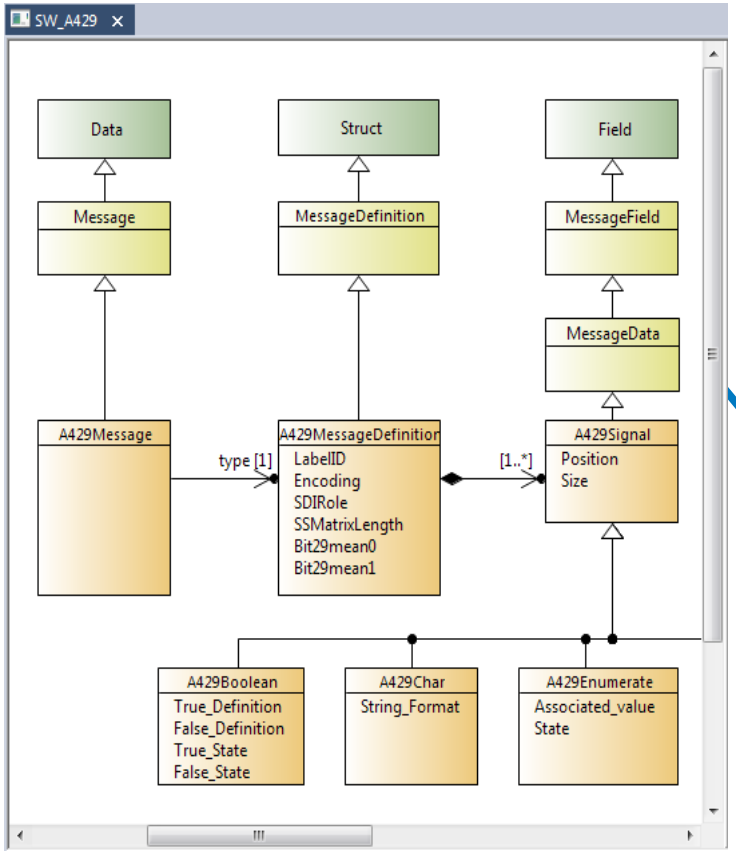
ICD Generation

GENERATE

CONFIGURE

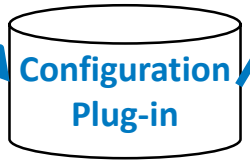
SCADE System Configuration Workflow

Specialist SCADE System Configurator

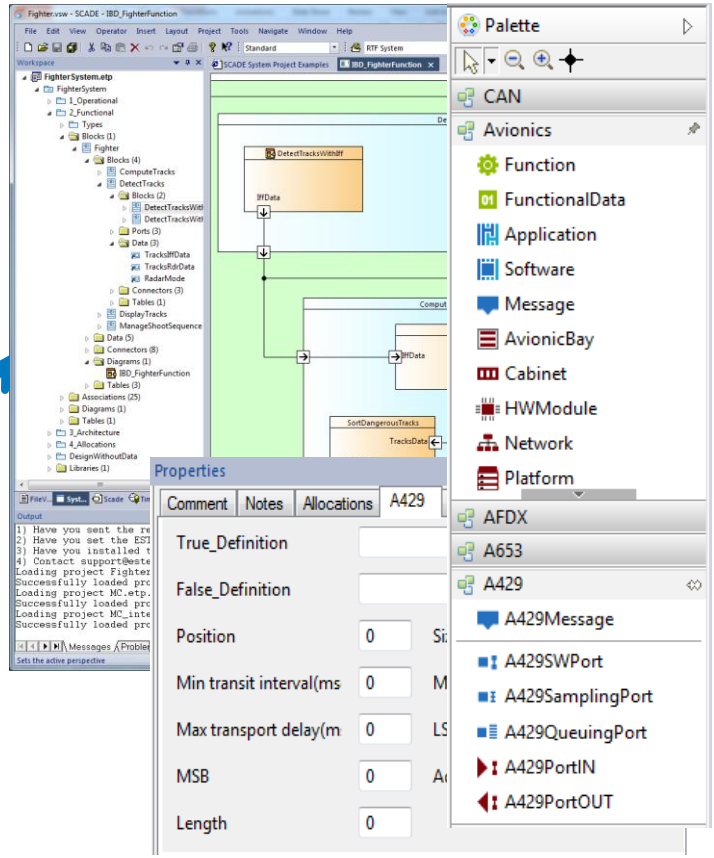


Generate

Deploy



End-User SCADE System Modeler

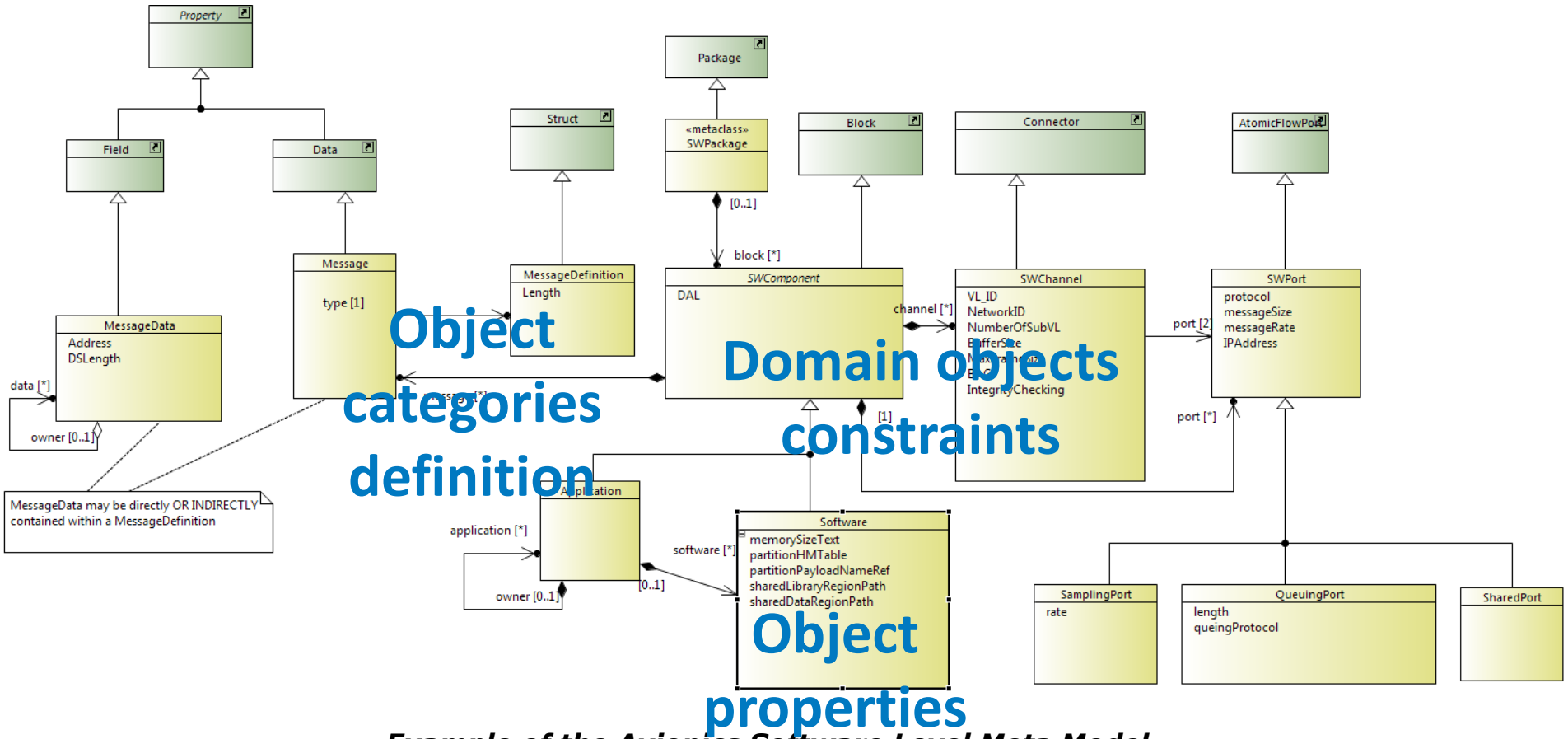


Define customized object kinds, derived from SCADE System objects

Domain specific modeler

Domain definition

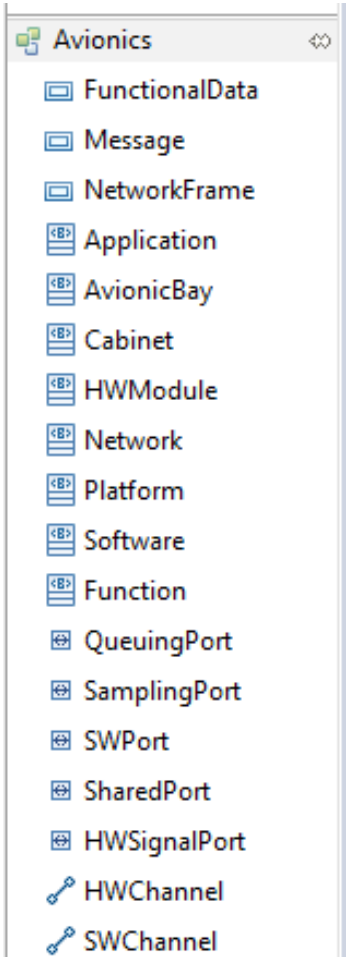
Allows for automated IDE customization

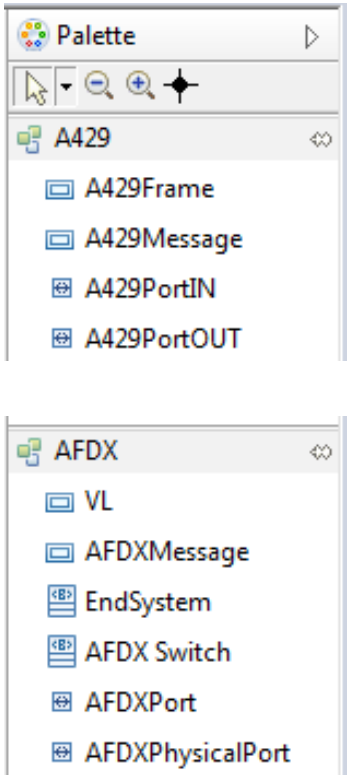


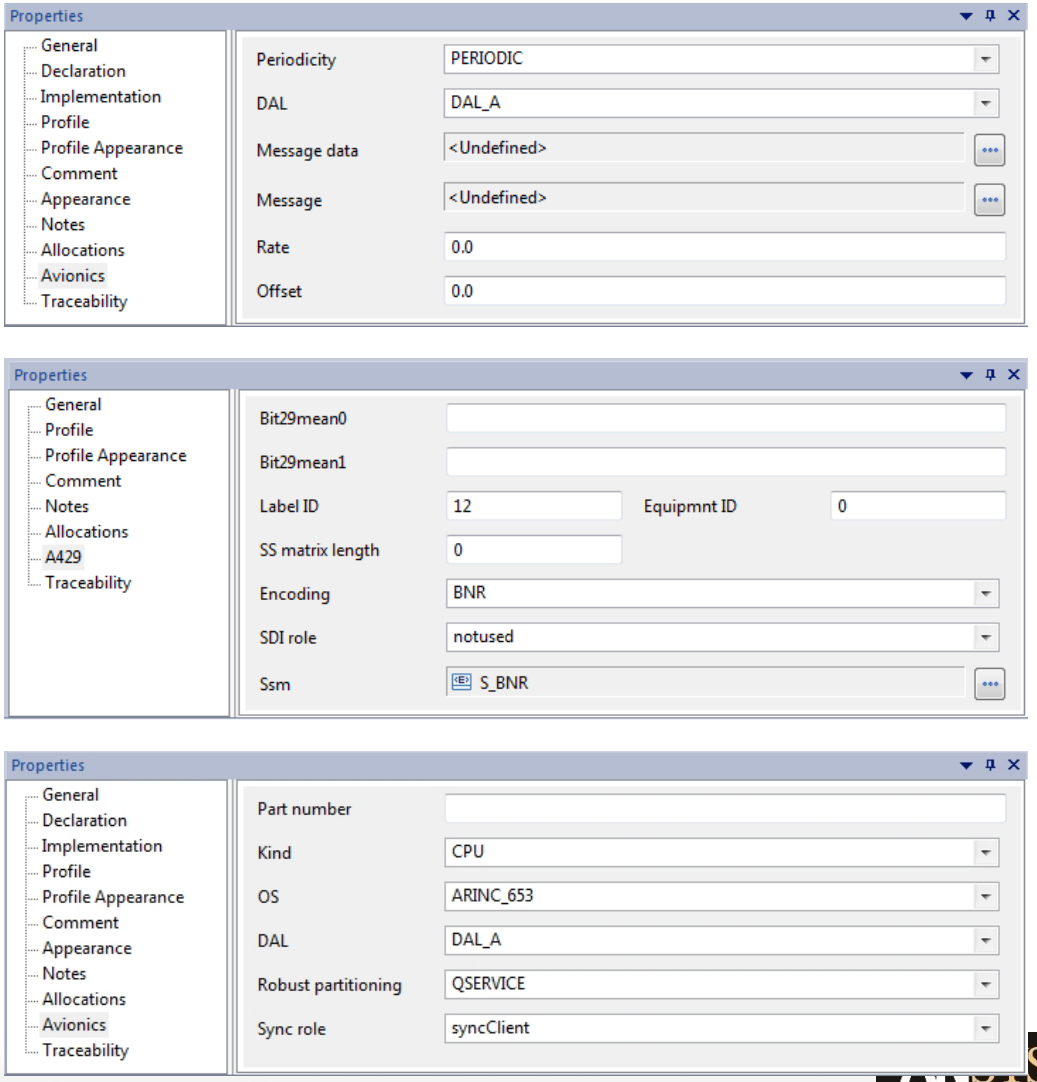
Example of the Avionics Software Level Meta Model

Customized User Interface

Domain oriented creation palettes and object properties

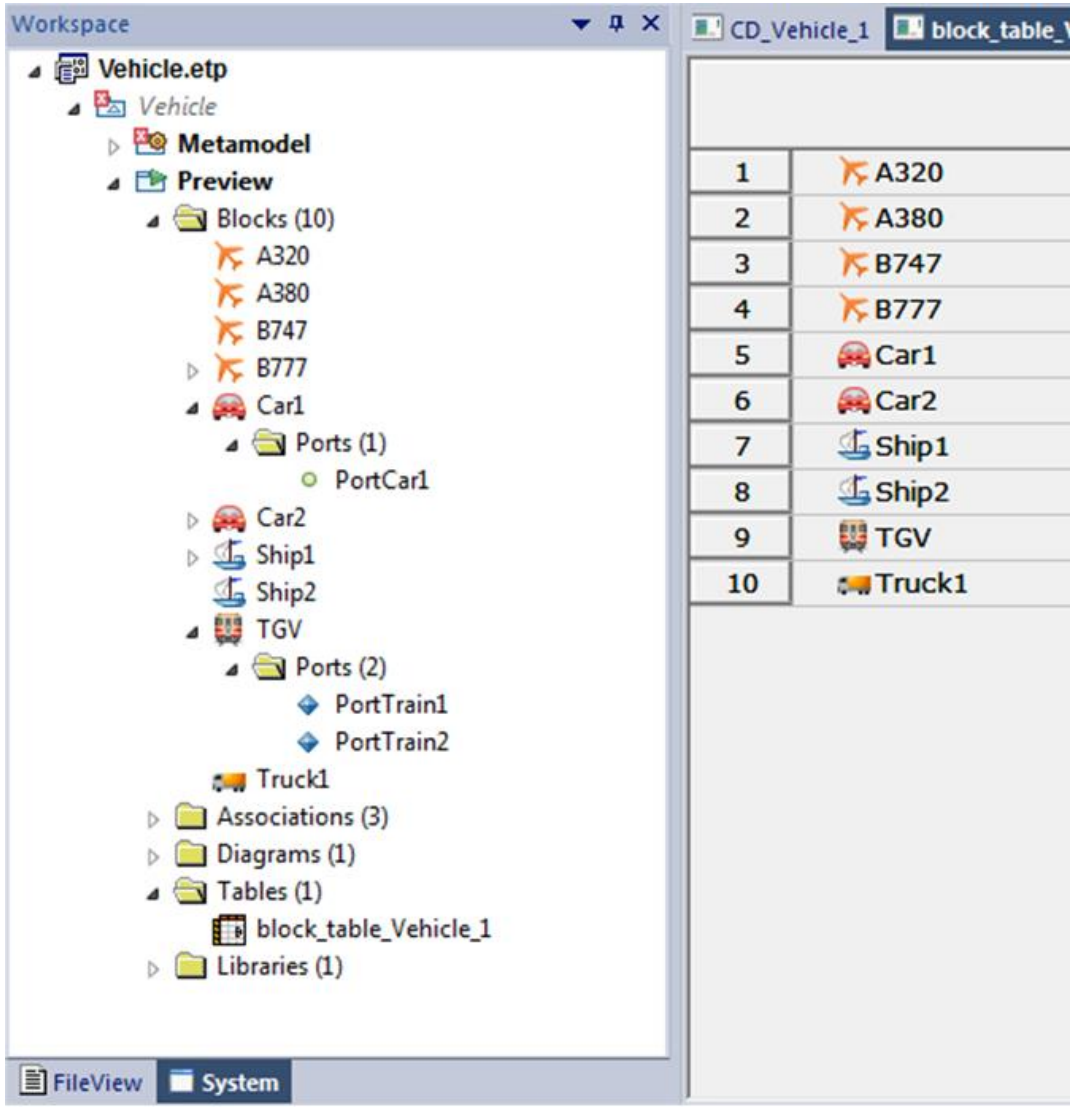




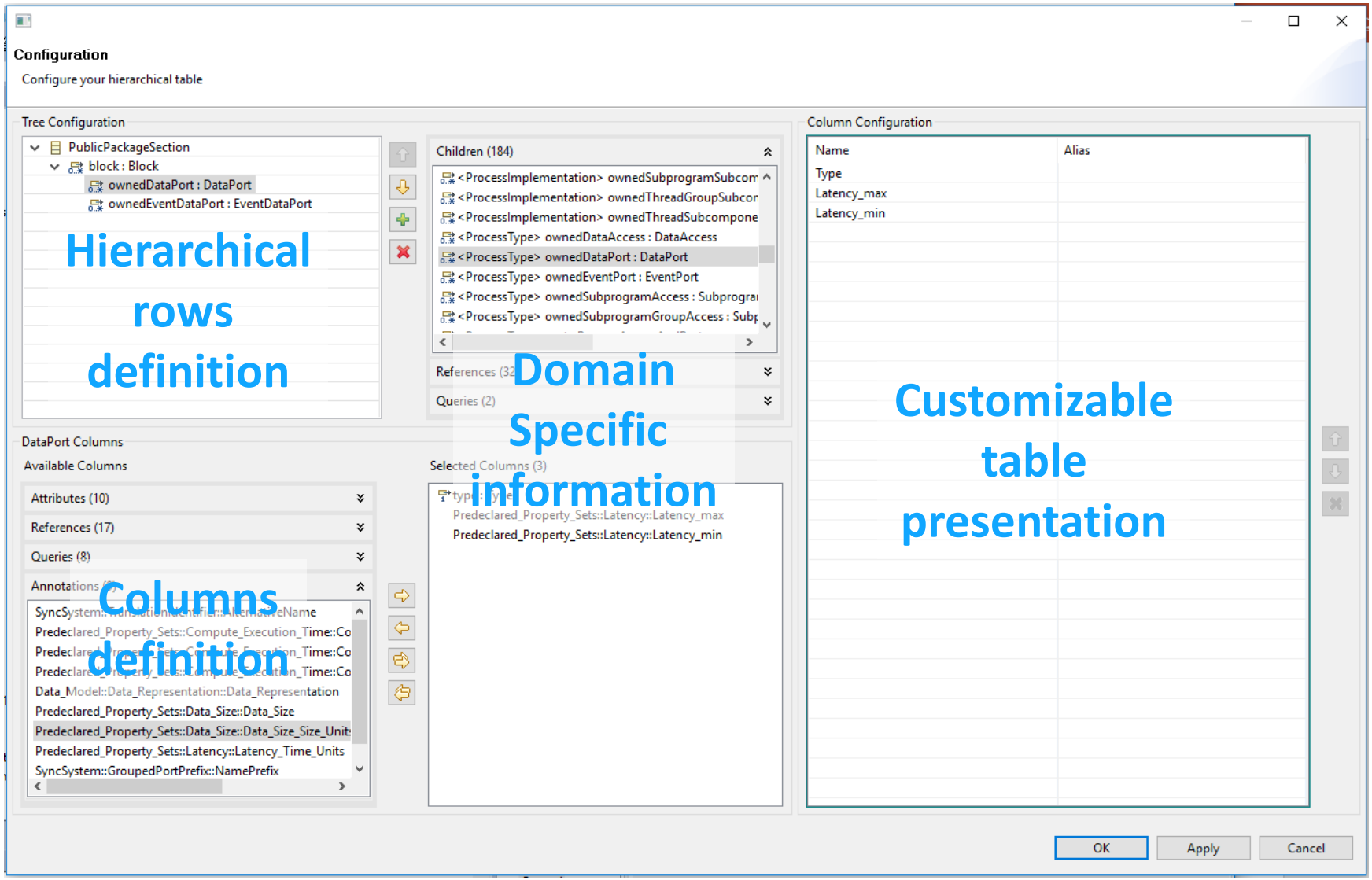


SCADE System Configurator: Custom Icons

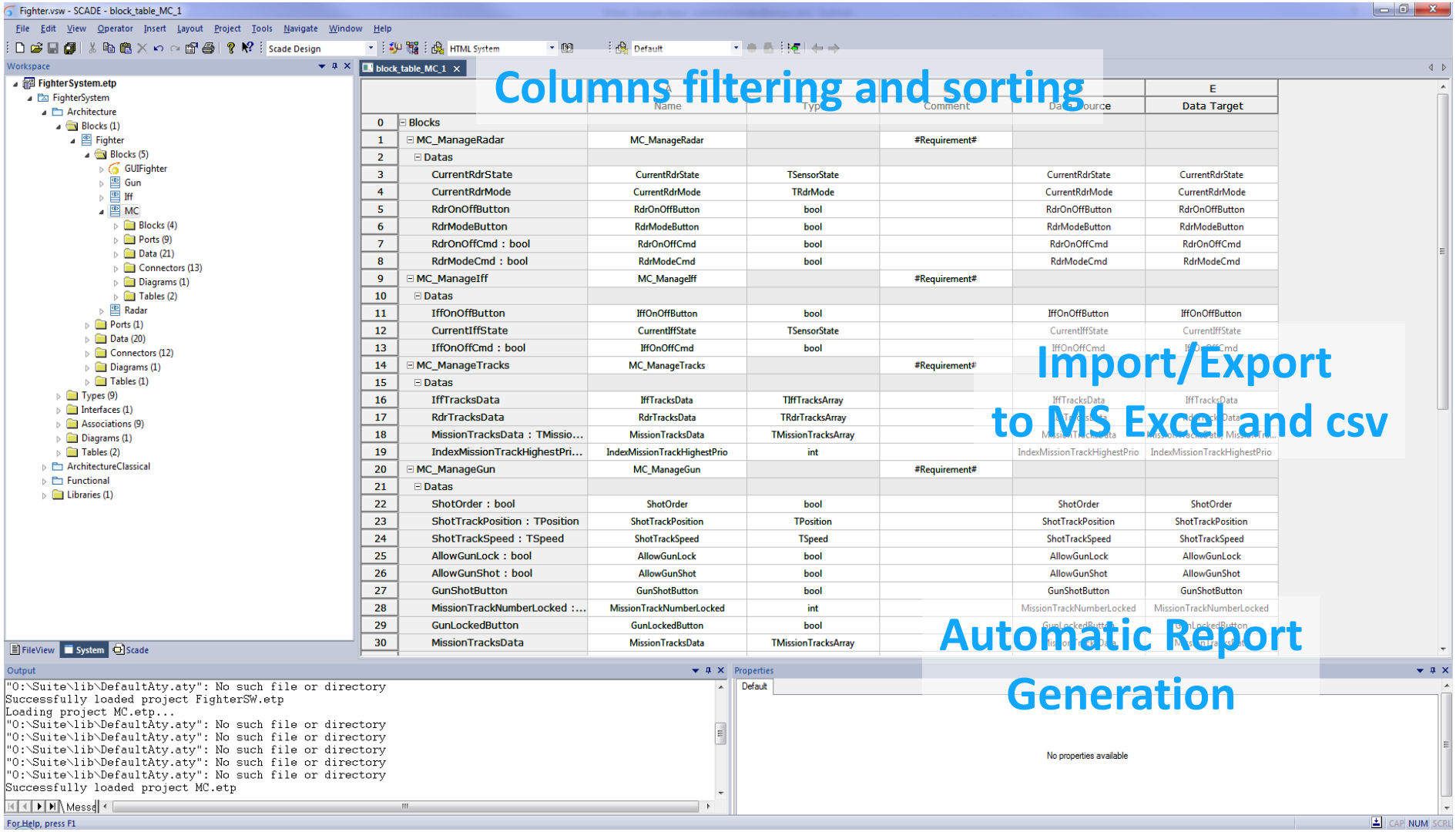
- **Custom objects icons support in:**
  - **System view**
  - **Tables**
  - **Creation palettes and menus**



Hierarchical Tables



Hierarchical Tables



	A Name	Type	Comment	D Data Source	E Data Target
0	Blocks				
1	MC_ManageRadar	MC_ManageRadar		#Requirement#	
2	Datas				
3	CurrentRdrState	CurrentRdrState	TSensorState		CurrentRdrState	CurrentRdrState
4	CurrentRdrMode	CurrentRdrMode	TRdrMode		CurrentRdrMode	CurrentRdrMode
5	RdrOnOffButton	RdrOnOffButton	bool		RdrOnOffButton	RdrOnOffButton
6	RdrModeButton	RdrModeButton	bool		RdrModeButton	RdrModeButton
7	RdrOnOffCmd : bool	RdrOnOffCmd	bool		RdrOnOffCmd	RdrOnOffCmd
8	RdrModeCmd : bool	RdrModeCmd	bool		RdrModeCmd	RdrModeCmd
9	MC_ManageIff	MC_ManageIff		#Requirement#	
10	Datas				
11	IffOnOffButton	IffOnOffButton	bool		IffOnOffButton	IffOnOffButton
12	CurrentIffState	CurrentIffState	TSensorState		CurrentIffState	CurrentIffState
13	IffOnOffCmd : bool	IffOnOffCmd	bool		IffOnOffCmd	IffOnOffCmd
14	MC_ManageTracks	MC_ManageTracks		#Requirement#	
15	Datas				
16	IffTracksData	IffTracksData	TIffTracksArray		IffTracksData	IffTracksData
17	RdrTracksData	RdrTracksData	TRdrTracksArray		RdrTracksData	RdrTracksData
18	MissionTracksData : TMissio...	MissionTracksData	TMissionTracksArray		MissionTracksData	MissionTracksData
19	IndexMissionTrackHighestPri...	IndexMissionTrackHighestPrio	int		IndexMissionTrackHighestPrio	IndexMissionTrackHighestPrio
20	MC_ManageGun	MC_ManageGun		#Requirement#	
21	Datas				
22	ShotOrder : bool	ShotOrder	bool		ShotOrder	ShotOrder
23	ShotTrackPosition : TPosition	ShotTrackPosition	TPosition		ShotTrackPosition	ShotTrackPosition
24	ShotTrackSpeed : TSpeed	ShotTrackSpeed	TSpeed		ShotTrackSpeed	ShotTrackSpeed
25	AllowGunLock : bool	AllowGunLock	bool		AllowGunLock	AllowGunLock
26	AllowGunShot : bool	AllowGunShot	bool		AllowGunShot	AllowGunShot
27	GunShotButton	GunShotButton	bool		GunShotButton	GunShotButton
28	MissionTrackNumberLocked :...	MissionTrackNumberLocked	int		MissionTrackNumberLocked	MissionTrackNumberLocked
29	GunLockedButton	GunLockedButton	bool		GunLockedButton	GunLockedButton
30	MissionTracksData	MissionTracksData	TMissionTracksArray		MissionTracksData	MissionTracksData

Benefits

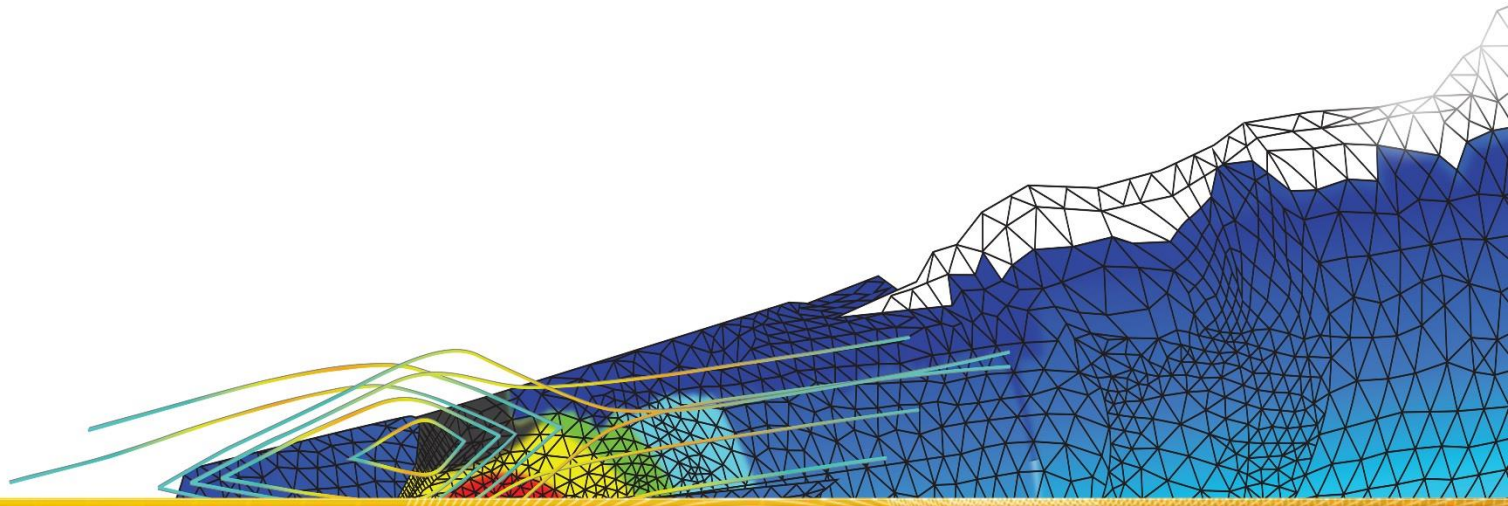
- Clear roles to facilitate MBSE deployment and ramp-up

- Method & Tools specialists
  - Simple and powerful graphical tool to express domain objects, properties, and inter-objects constraints
  - Configuration managed as a SCADE project
  - Simple deployment on SCADE Architect end-user machines

- System designers
  - Guidance in IDE for the domain objects
  - Complexity of the underlying SysML language is hidden
  - Graphical styles automatically set to domain objects
  - Customized tabular views directly available

ANSYS®

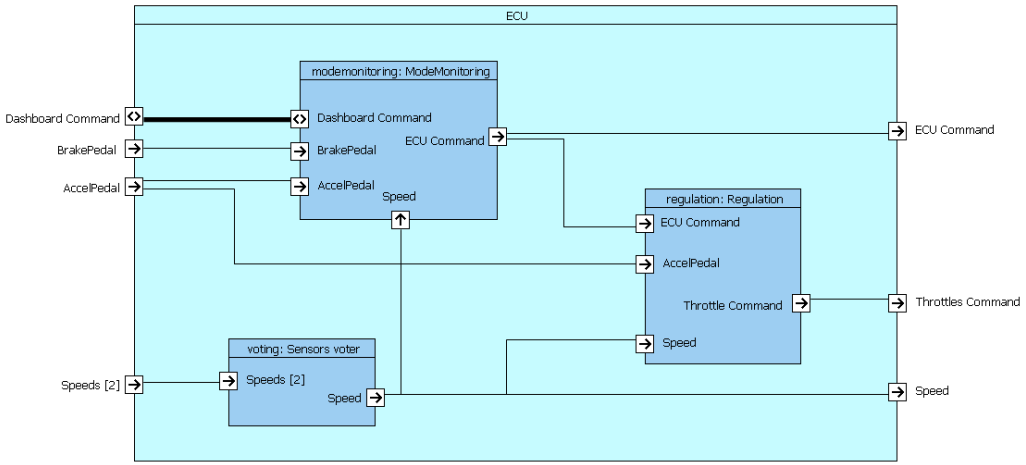
SCADE
System – SW synchronization

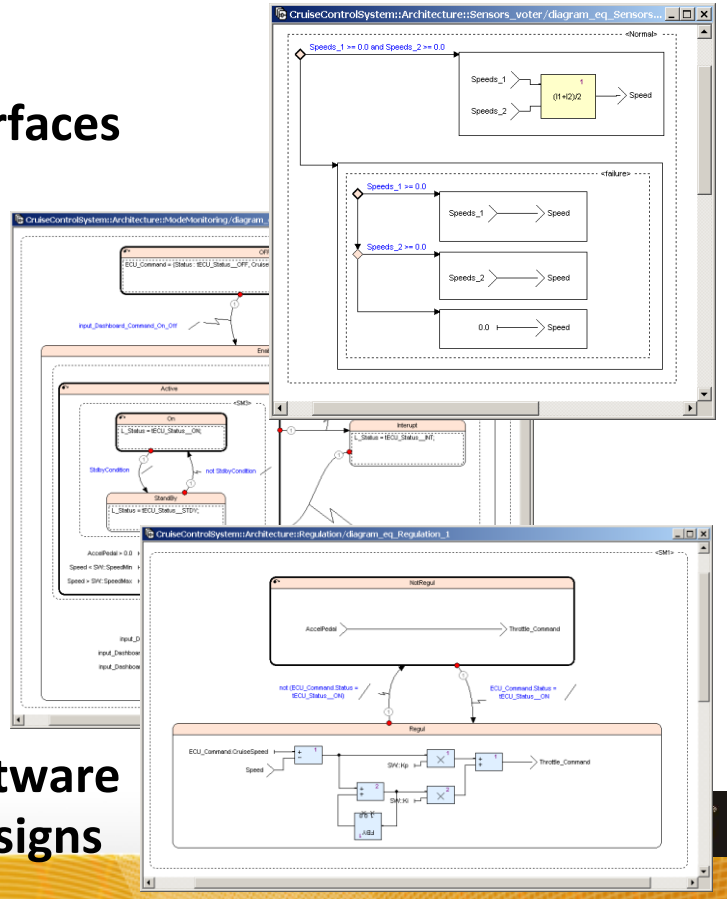


System – Software Collaboration

- System – Software Models Synchronization
  - Avoid duplication of efforts and inconsistencies between system structural models and software behavioral models
  - System design and Software components evolve independently
  - On-demand re-synchronization of interfaces

Interfaces described in SCADE System model





Software designs

System – Software Collaboration