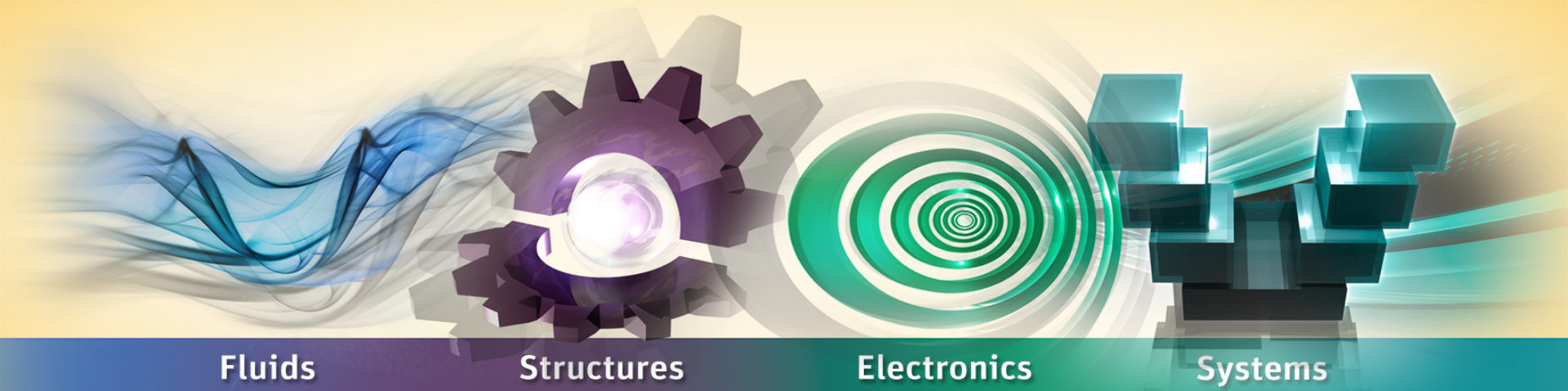
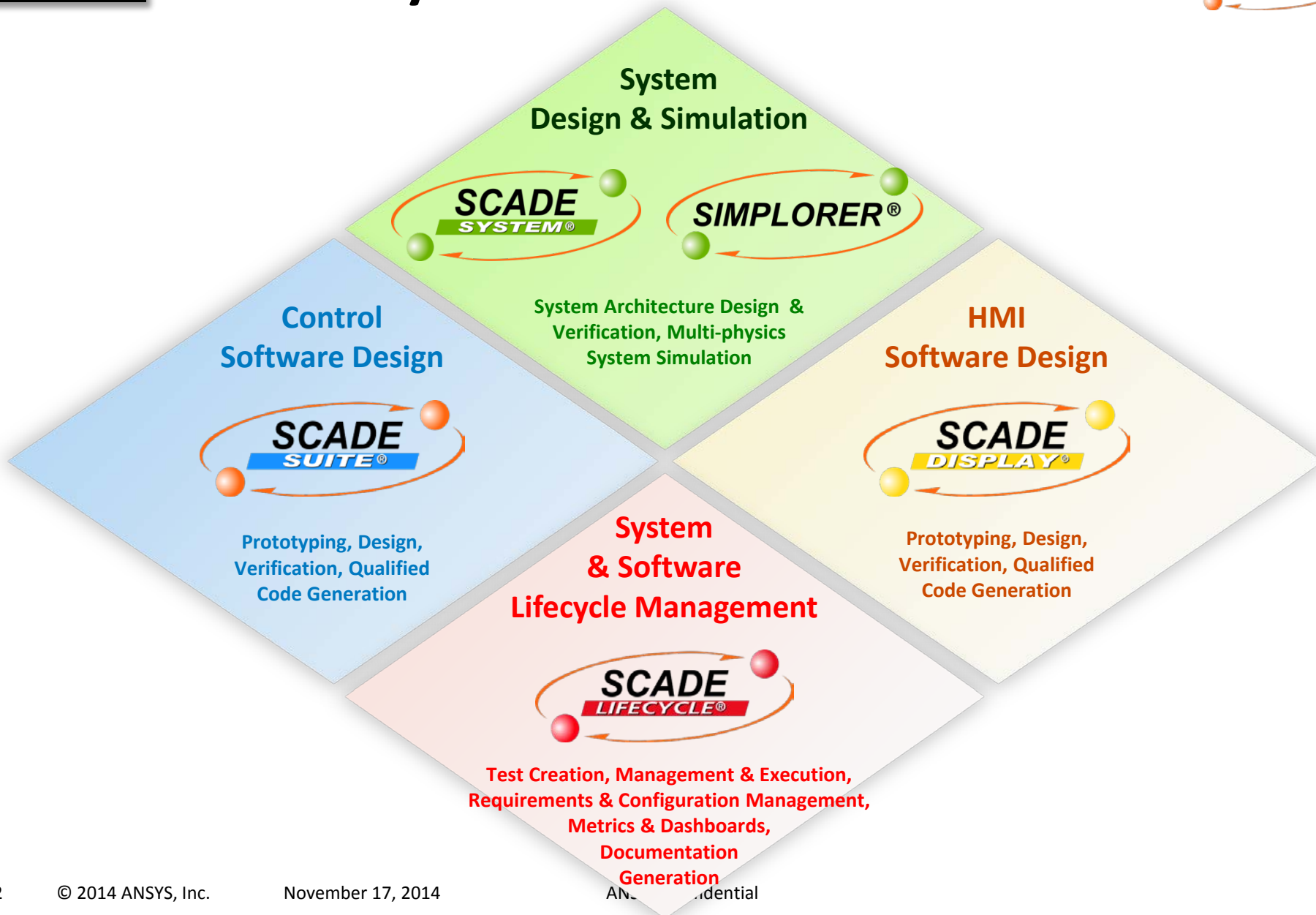


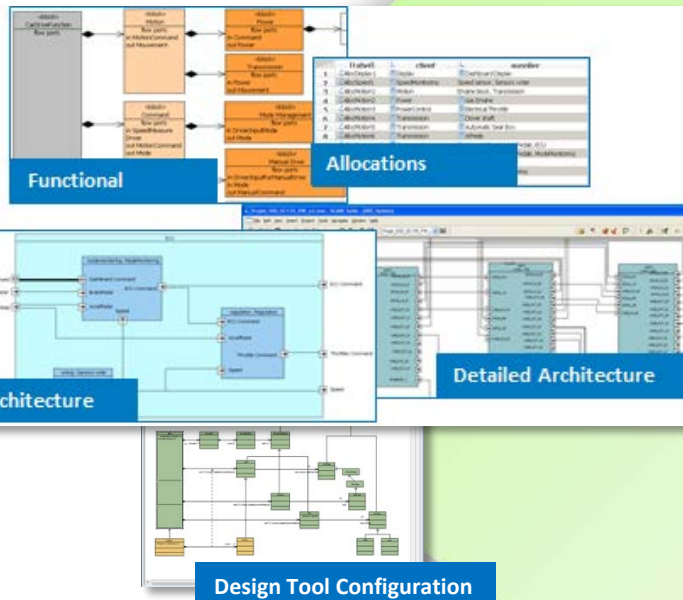
SCADE System Presentation







Model-Based System Design



Model Checks



**System / Software
Bi-directional Sync Up**



**System Model
Diff/Merge**



**Model Sharing
& IP Protection**

**DESIGN &
CONFIGURE**

VERIFY

**INTEGRATE &
COLLABORATE**

DESIGN Systems with SCADE System



Fluids

Structures

Electronics

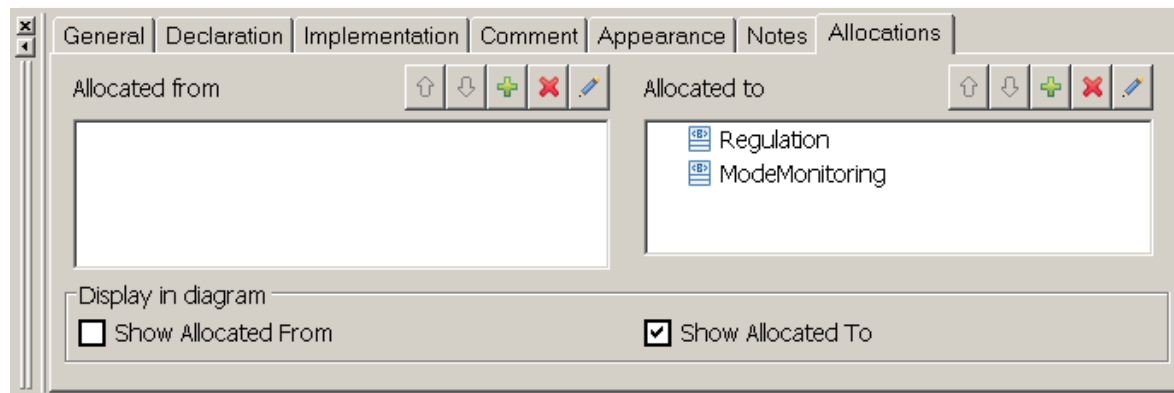
Systems



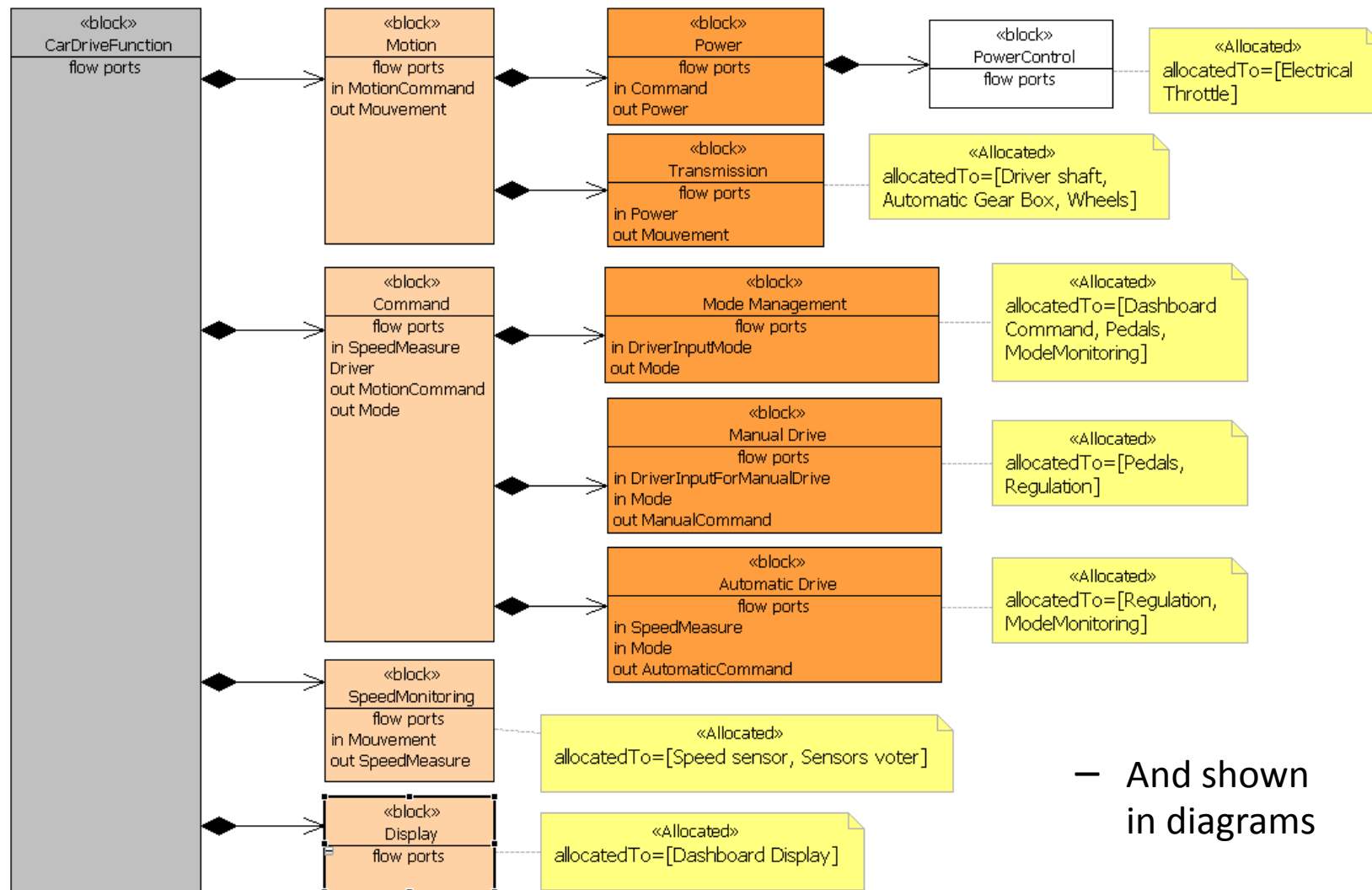
For any objects, e.g. functions allocated on architecture components

- Can be defined in allocation tables
- Or in the object properties

	Declared In	Name	Source	Target
0	FunctionsAllocation	AllocDisplay	Display	Dashboard Display
1	FunctionsAllocation	AllocPower	Power	Gas Engine
2	FunctionsAllocation	AllocPowerControl	PowerControl	Electrical Throttle
3	FunctionsAllocation	AllocTransmission1	Transmission	Driver shaft
4	FunctionsAllocation	AllocTransmission2	Transmission	Automatic Gear Box
5	FunctionsAllocation	AllocTransmission3	Transmission	Wheels
6	FunctionsAllocation	AllocCommand1	Command	Dashboard Command
7	FunctionsAllocation	AllocModeManagement1	Mode Management	Dashboard Command
8	FunctionsAllocation	AllocManualDrive	Manual Drive	Pedals
9	FunctionsAllocation	AllocAutomaticDrive1	Automatic Drive	Regulation
10	FunctionsAllocation	AllocSpeed1	SpeedMonitoring	Speed sensor
11	FunctionsAllocation	AllocSpeed2	SpeedMonitoring	Sensors voter
12	FunctionsAllocation	AllocMotion1	Motion	Engine block
13	FunctionsAllocation	AllocMotion2	Motion	Transmission



Allocations

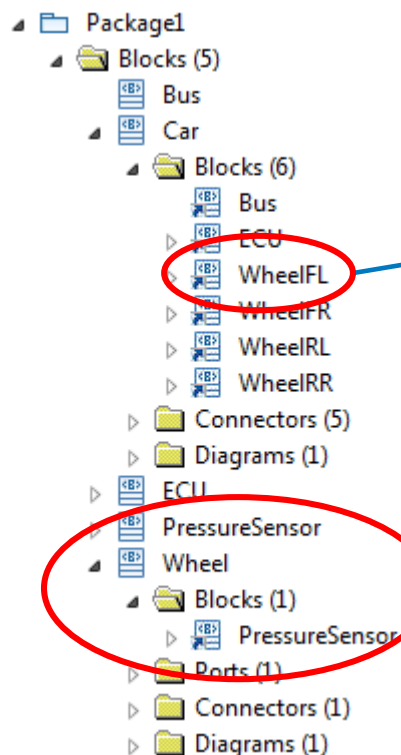


— And shown
in diagrams

Unlimited Component Reuse

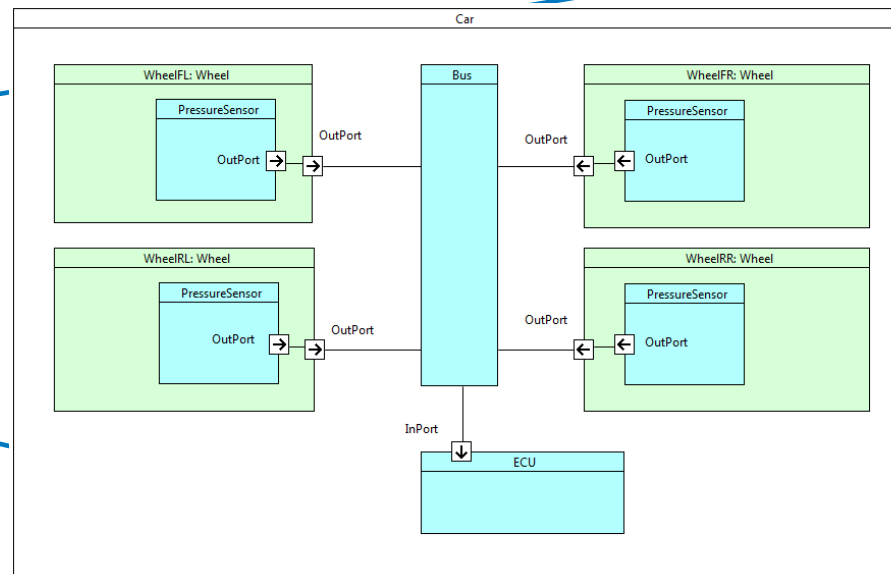
Multi-Instantiation Support

Traditional SysML tools

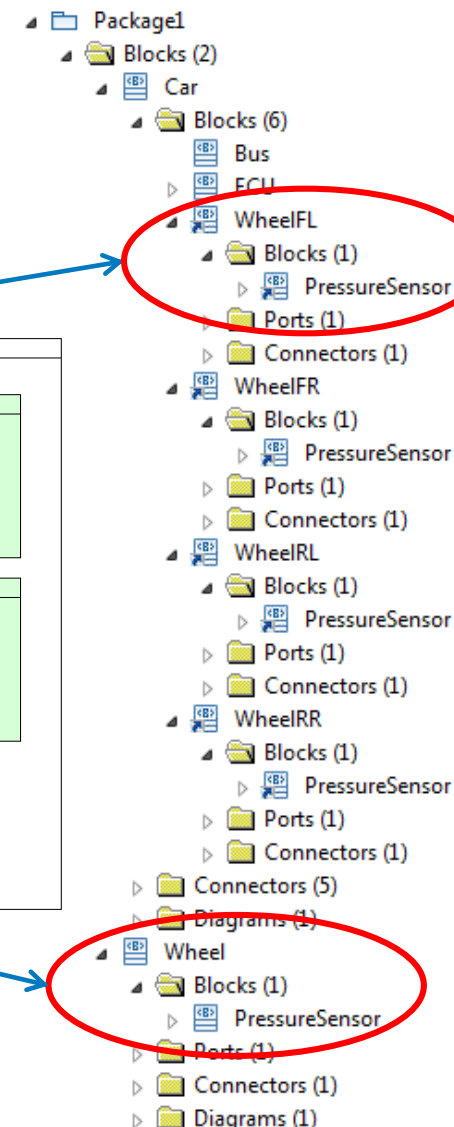


SCADE System

Each individual instance is **replicated**



Unique components are defined within other components



**Best practice in most industries
to manage the system interfaces: Data Dictionaries**

- Data attributes with various levels of refinements
- Large amount of data managed in data bases

Full support of Data Dictionaries in SCADE System

- Tables with customizable data attributes
- Import & export data dictionaries
- Link data with the functional and architecture models
- Data propagation management

Data Management

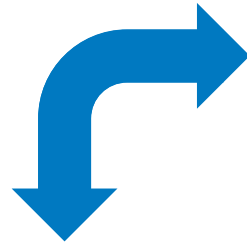
Import/export data dictionaries

Direct copy&paste from/to Excel

- Creates data,
- Binds names to existing information e.g. datatype

<Ctrl> v

<Ctrl> c

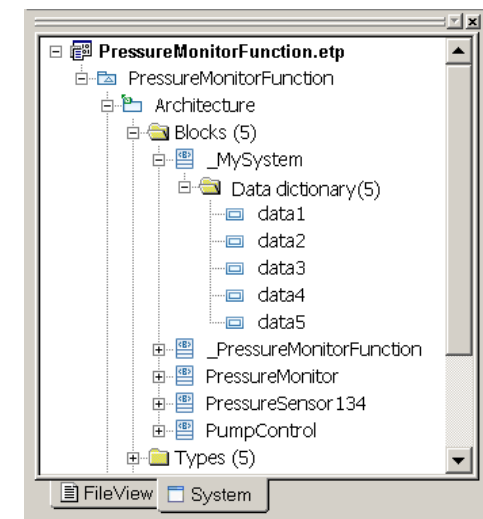


	A	B	C	D	E	F
1	data1	Pressure	0	250	Description data	
2	data2	Validity			Description boolean data	
3	data3	Bool			Description data	
4	data4	Real			Description boolean data	
5	data5	Pressure	0	100	Description data	
6						

Sheet1 Sheet2 Sheet3

Ready Average: 87,5 Count: 19 Sum: 350 100%

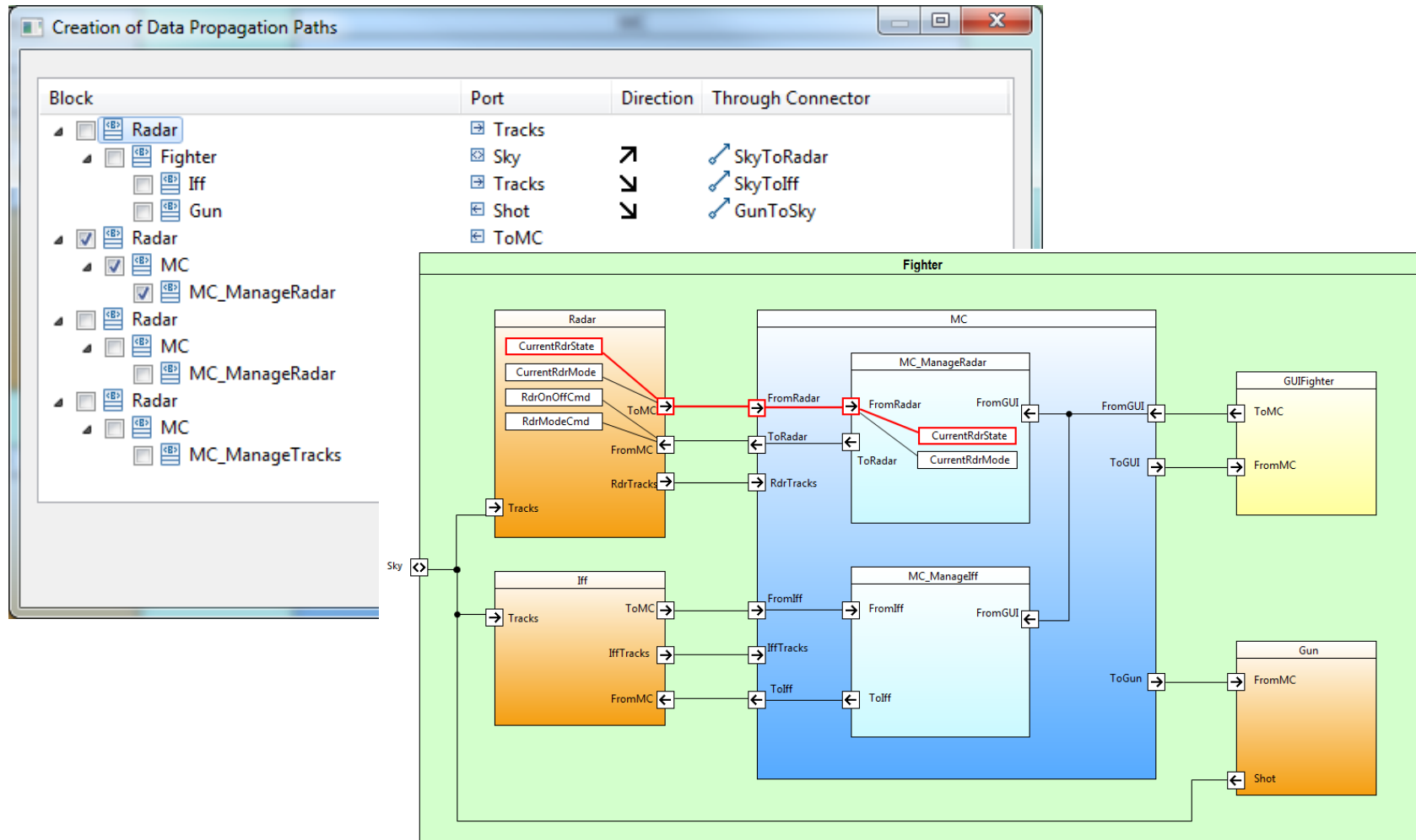
	Name	Type	Min	Max	Description
1	data1	Pressure	0	250	Description data
2	data2	Validity			Description boolean data
3	data3	real			Description data
4	data4	bool			Description boolean data
5	data5	Pressure	0	100	Description data



Data Management

Data Propagation Management


Data propagation between Blocks through Connectors



VERIFY & INTEGRATE Systems with SCADE System

A blue, wavy, translucent fluid flow visualization.

Fluids

A purple gear with a glowing white center, surrounded by a grey gear.

Structures

A green, concentric circular pattern resembling a lens or a target.

Electronics

A 3D stack of blue and black cubes with light rays emanating from them.

Systems



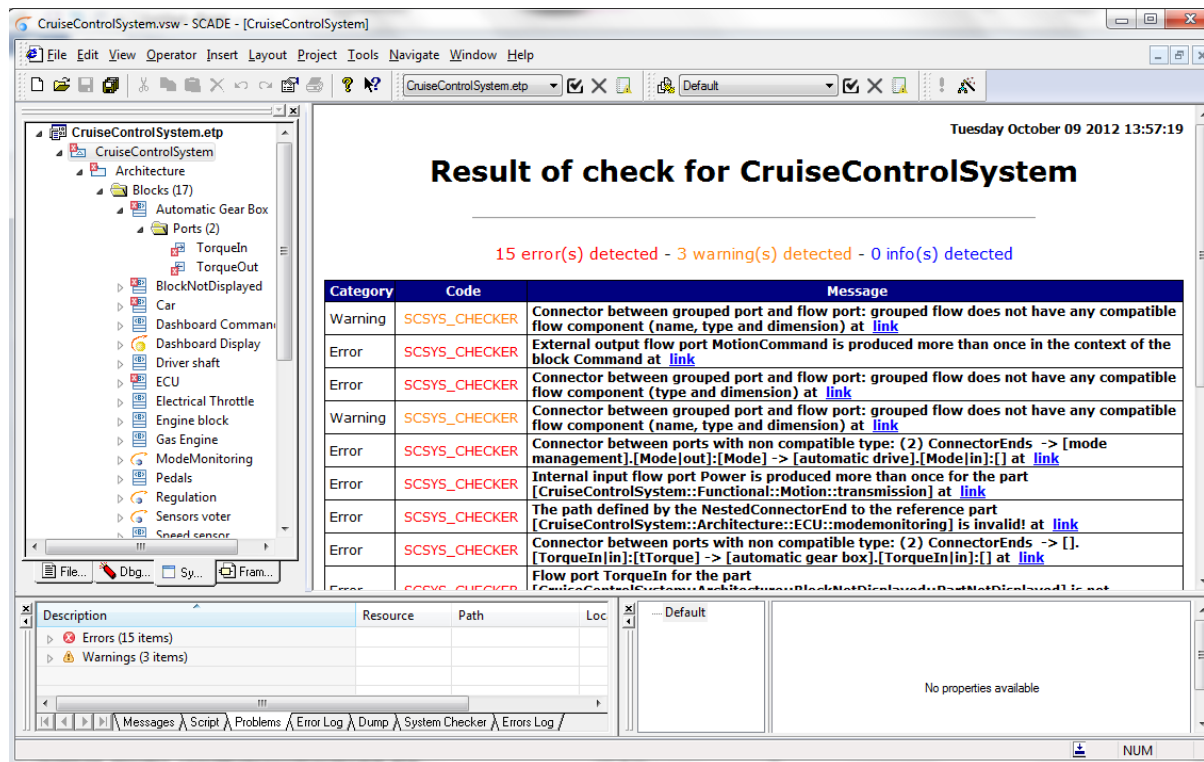
Checker

Automated System Design Rules Verification

Pre-existing set of rules

Easy customization of rules

- Rule example: ***“Are all data produced by exactly 1 component, and used at least once?”***



Result of check for CruiseControlSystem

15 error(s) detected - 3 warning(s) detected - 0 info(s) detected

Category	Code	Message
Warning	SCSYS_CHECKER	Connector between grouped port and flow port: grouped flow does not have any compatible flow component (name, type and dimension) at link
Error	SCSYS_CHECKER	External output flow port MotionCommand is produced more than once in the context of the block Command at link
Error	SCSYS_CHECKER	Connector between grouped port and flow port: grouped flow does not have any compatible flow component (type and dimension) at link
Warning	SCSYS_CHECKER	Connector between grouped port and flow port: grouped flow does not have any compatible flow component (name, type and dimension) at link
Error	SCSYS_CHECKER	Connector between ports with non compatible type: (2) ConnectorEnds -> [mode management].[Mode]out:[Mode] -> [automatic drive].[Mode]in:[] at link
Error	SCSYS_CHECKER	Internal input flow port Power is produced more than once for the part [CruiseControlSystem::Functional::Motion::transmission] at link
Error	SCSYS_CHECKER	The path defined by the NestedConnectorEnd to the reference part [CruiseControlSystem::Architecture::ECU::modemonitoring] is invalid! at link
Error	SCSYS_CHECKER	Connector between ports with non compatible type: (2) ConnectorEnds -> [TorqueIn]in:[] -> [automatic gear box].[TorqueIn]in:[] at link
Error	SCSYS_CHECKER	Flow port TorqueIn for the part [CruiseControlSystem::Architecture::BlockNotDisplayed::BlockNotDisplayed] is not

Errors (15 items)
Warnings (3 items)

Messages | Script | Problems | Error Log | Dump | System Checker | Errors Log

System Model API

Script Wizard

[Create script code](#) [Copy script code](#) [Eval script code](#) [All Classes](#)

Class Package

- Attributes

Defined in	Name	Type
	name	string
NamedElement	qualifiedName	string
	visibility	org.eclipse.uml2.uml.VisibilityKind

- Associations

Defined in	Multiplicity	Composition	Name	Distant class
Element	0+	<--	ownedComment	Comment
	0+		ownedElement	Element
	0..1		owner	Element
TemplateableElement	0..1	<--	ownedTemplateSignature	TemplateSignature
ParameterableElement	0+			
	0..1			
	0..1			
NamedElement	0+			
	0+			
	0+			
	0+			
	0+			
	0..1			
	0..1			
Namespace	0+			

Development of custom verifications, data import or export

- Wizard to help creating your first scripts
- TCL, Java/Eclipse™ and OCL technologies

Script1

```

proc ForTheName {name} {
    output "$name\n"
    return 1
}

proc ForEachNestedPackage {nestedPackage} {
    set vname [Get $nestedPackage]
    ForTheName $vname
}

foreach item $uml22 {
    MapRole $item nestedPackage
}

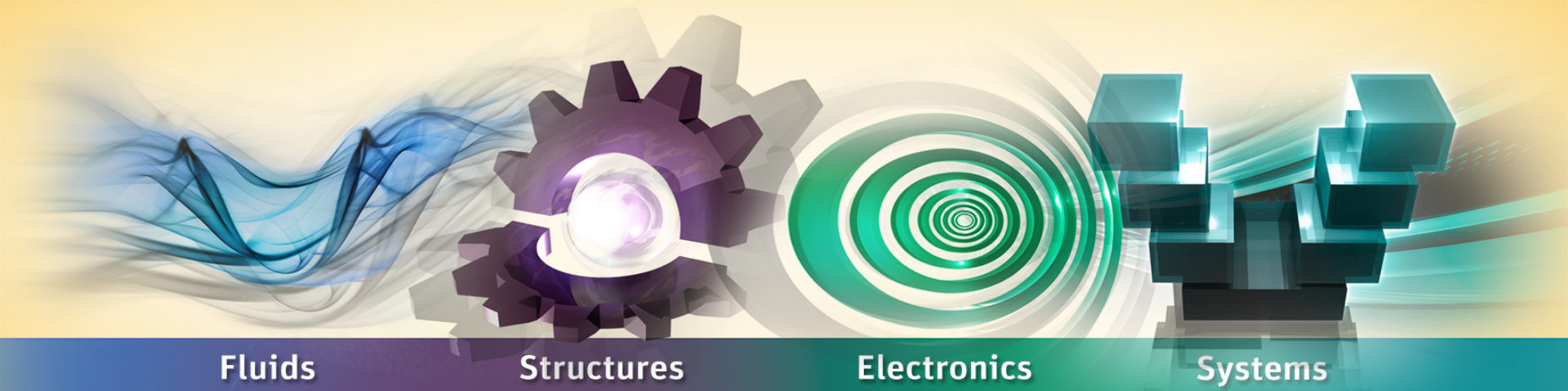
```

Output

Port	Comment
Driver	FlowPort "Driver" in Block "CarDriveFunction" is never produced or used
SpeedMeasure	FlowPort "SpeedMeasure" in Block "Command" is never produced or used
MotionCommand	FlowPort "MotionCommand" in Block "Command" is never produced or used
Mode	FlowPort "Mode" in Block "Command" is never produced or used
Driver	FlowPort "Driver" in Block "Command" is never produced or used
MotionCommand	FlowPort "MotionCommand" in Block "Motion" is never produced or used
Mouvement	FlowPort "Mouvement" in Block "Motion" is never produced or used
Command	FlowPort "Command" in Block "Power" is never produced or used
Power	FlowPort "Power" in Block "Power" is never produced or used

CustomCheck Script

INTEGRATE & COLLABORATE

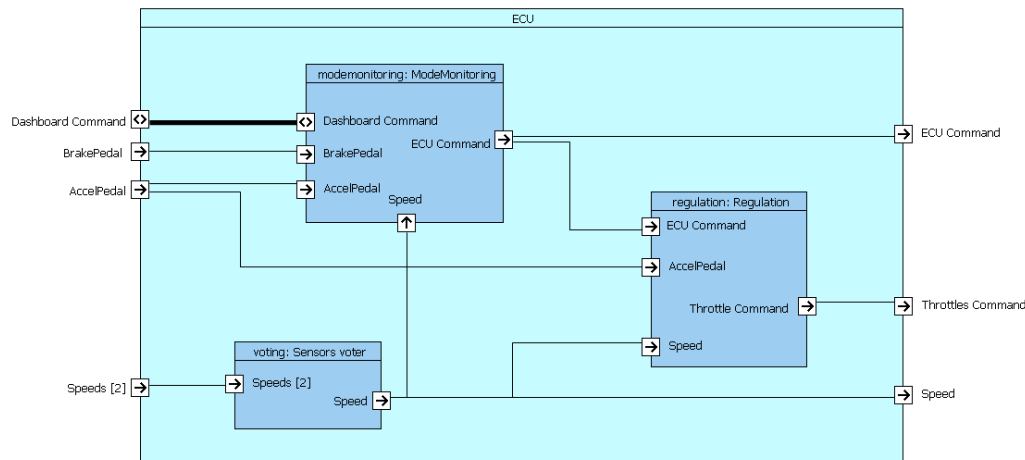


System – Software Collaboration

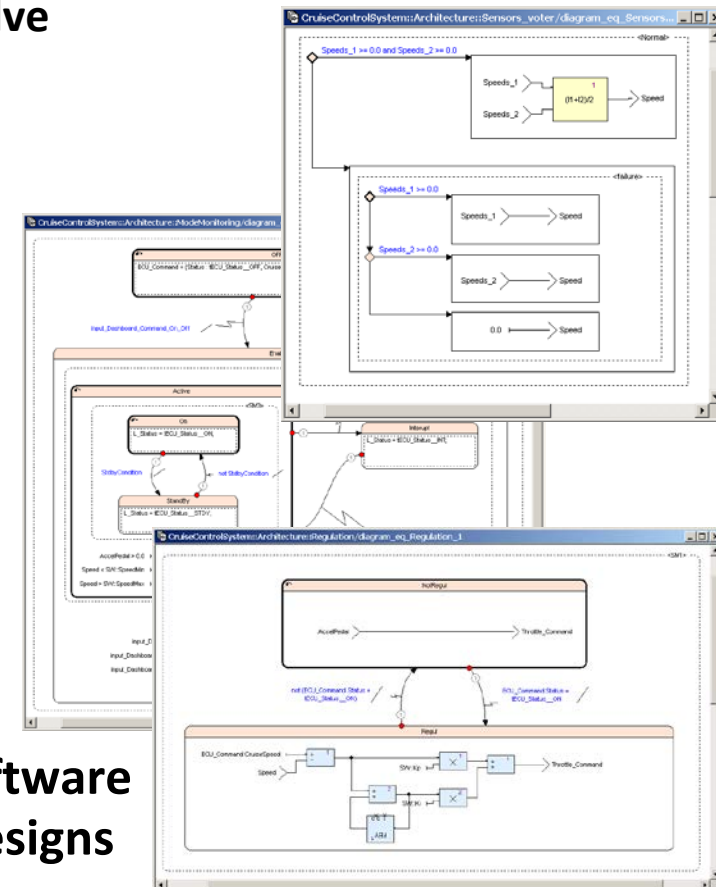
System – Software Models Synchronization

- 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

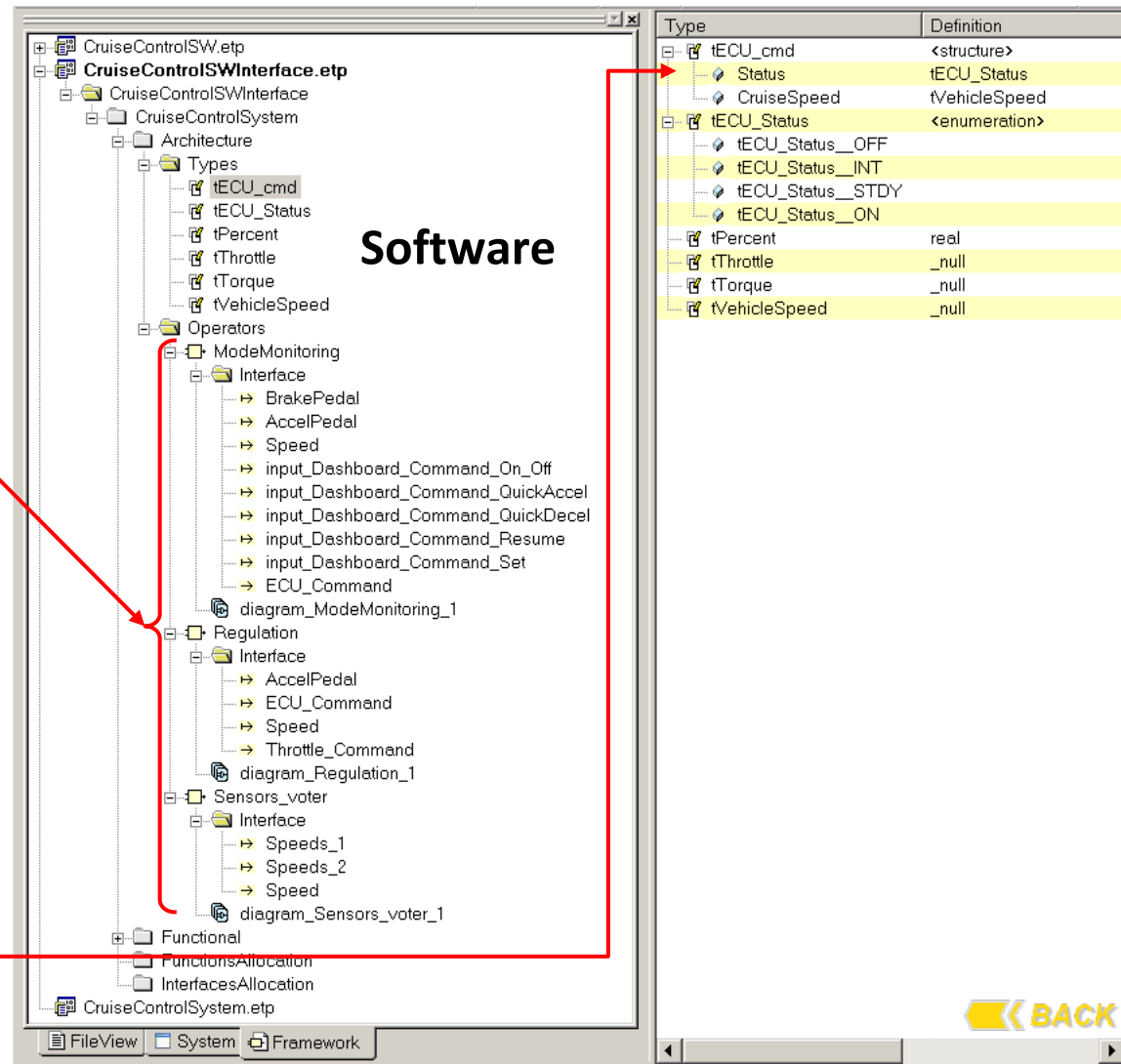
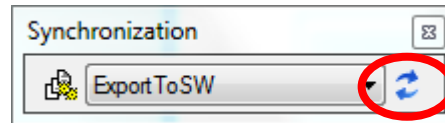
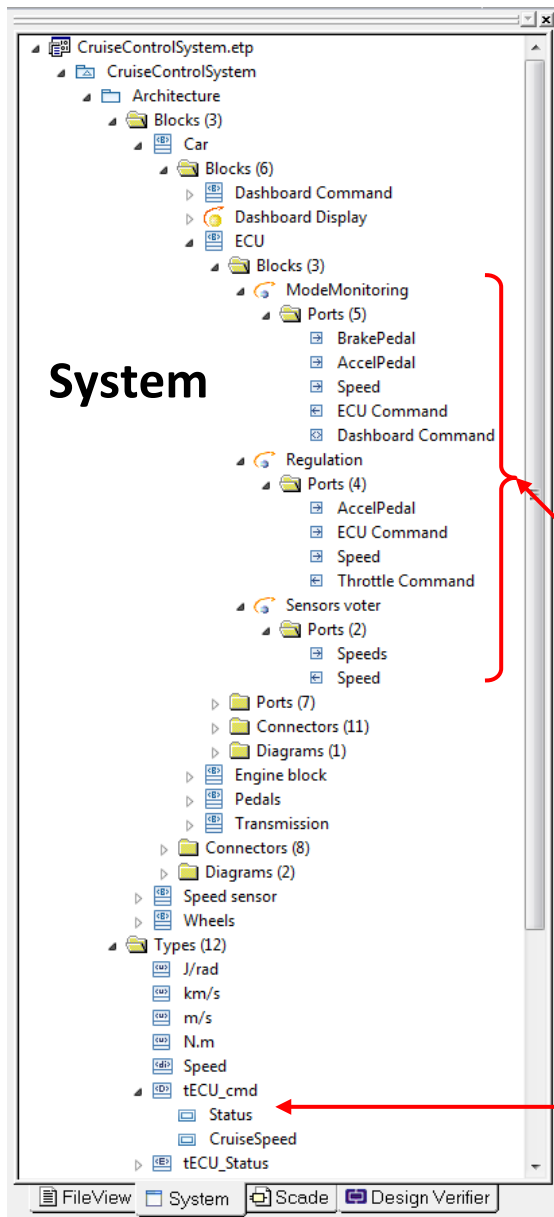
Interfaces described in SCADE System model



Software
designs



System – Software Collaboration

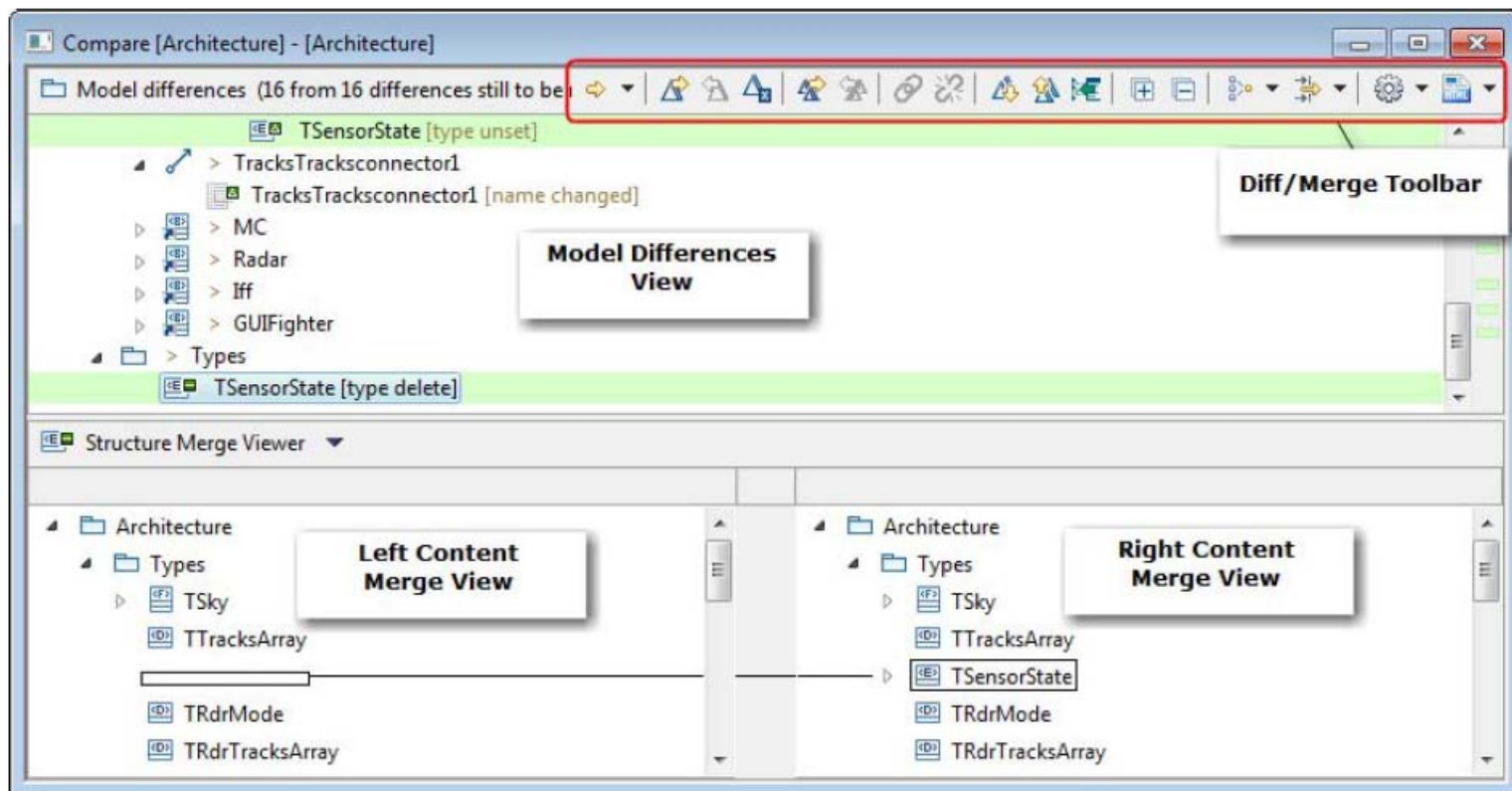


Type	Definition
tECU_cmd	<structure>
Status	tECU_Status
CruiseSpeed	tVehicleSpeed
tECU_Status	<enumeration>
tECU_Status__OFF	
tECU_Status__INT	
tECU_Status__STDY	
tECU_Status__ON	
tPercent	real
tThrottle	_null
tTorque	_null
tVehicleSpeed	_null

Model Diff/Merge

Compute, highlight and merge dependencies in 1 click

User control with “force match” and “exclude match” actions



Model Diff/Merge

Comprehensive report generation

- Differences *before* merge session
- Merge *actions*
- Differences *after* merge session

C:\test\diffclean\initial\System_diff_System.html

Difference report between (1) System and (2) System

(1) System : C:/test/diffclean/initial/initial.etp
(2) System : C:/test/diffclean/user1/user1.etp

.....

1 Changed System

(1) [initial::Architecture::System](#)
(2) [initial::Architecture::System](#)

Changed References	(1) System	(2) System
data value added	Data1	-

1.1 Changed Input1

(1) [initial::Architecture::System::Input1](#)
(2) [initial::Architecture::System::Input1](#)

Changed References	(1) System	(2) System
type	Array1	-

1.2 Added Data1

(1) [initial::Architecture::System::Data1](#)

End of Diff Report
2 differences found: 1 objects changed, 1 objects added, 0 objects deleted, 0 objects moved.

Collaboration Capabilities



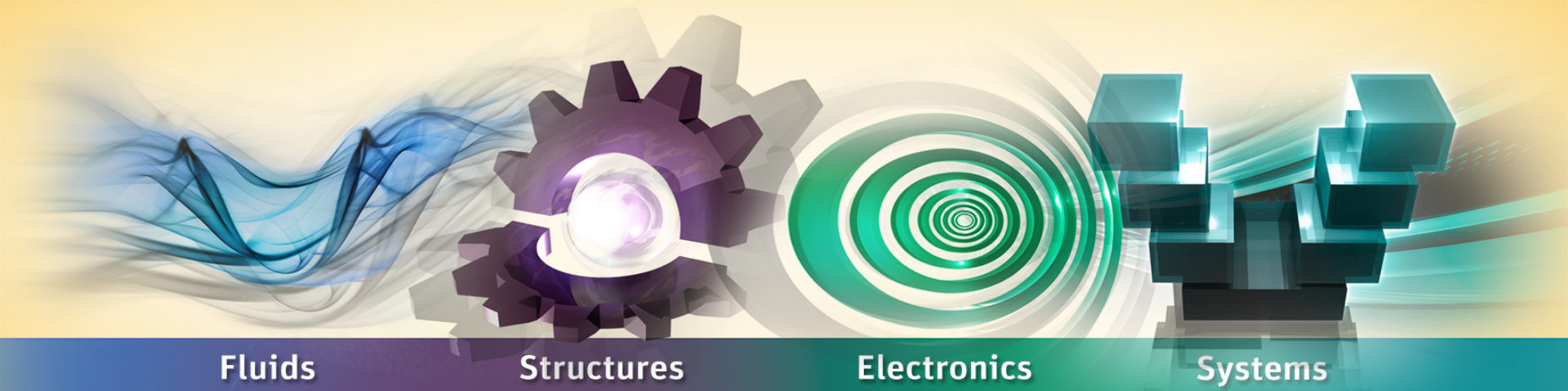
Model Component Export

- **Preserve IP** of other system design parts, e.g. to subcontract the SW development of one system block

Workflow example

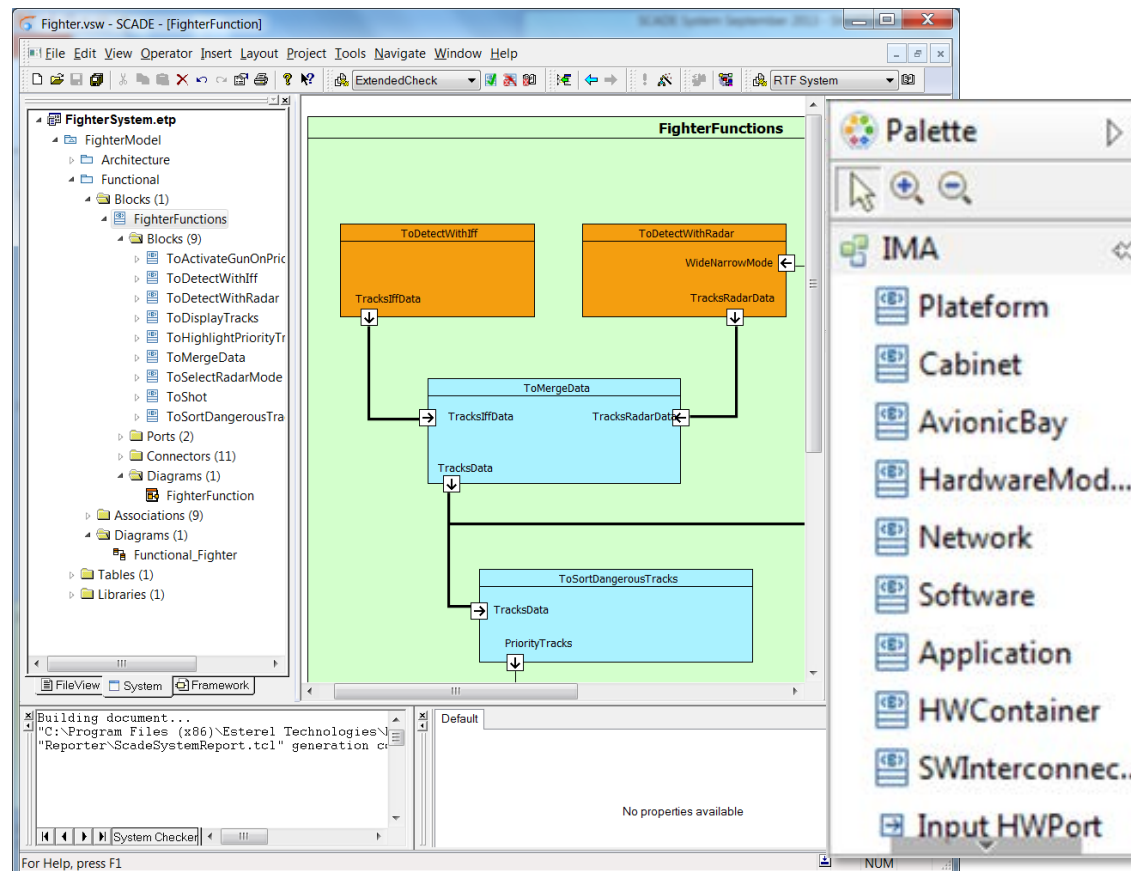
- **1/ Export subsystems**
 - Allow several teams working in parallel on the exported subsystems
- **2/ Reintegrate modifications**
 - Rely on Diff-Merge tool
- **3/ Report from reintegration**
- **4/ Iterate** as often as needed

CUSTOMIZE the SCADE System Modeler

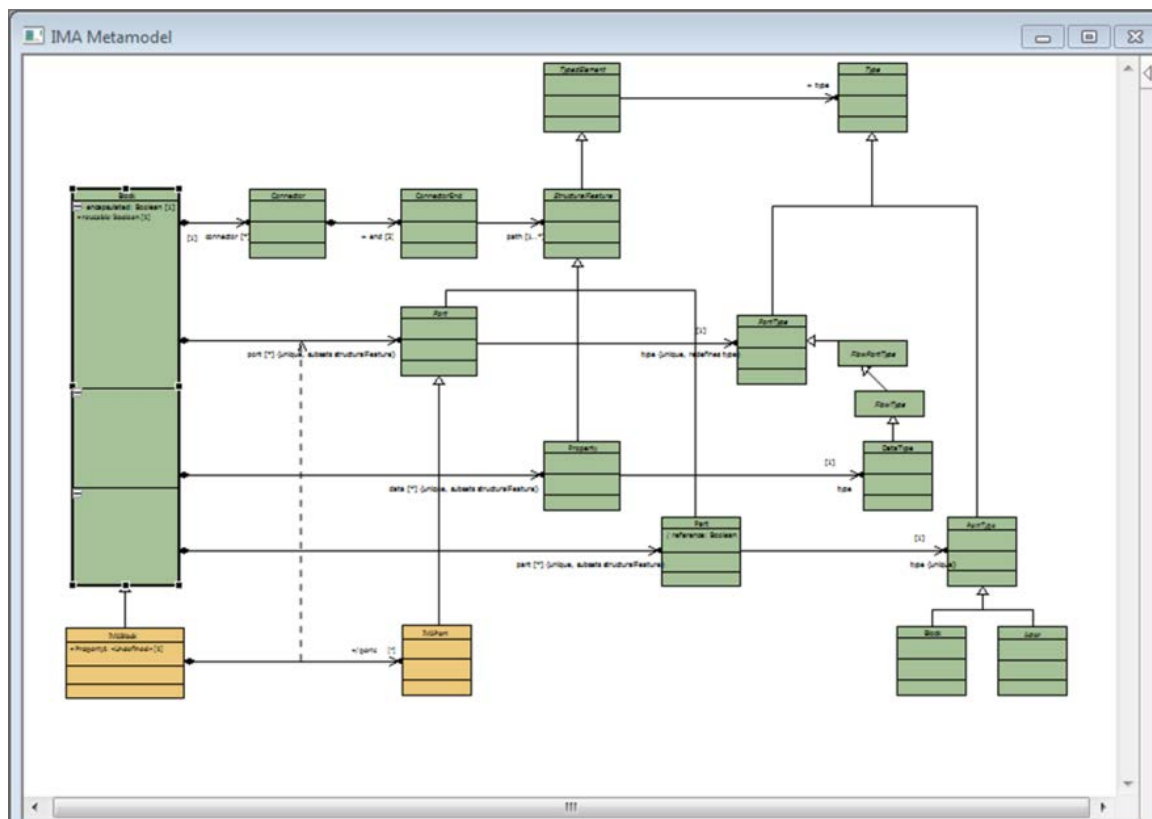


SCADE System Configurator

Customize SCADE System for a **Customer specific domain**,
as well as standard domains such as **IMA, AUTOSAR**, etc...



- **Define new object, derived from SCADE System objects**
 - **Function, Equipment, Bus, Port...**
- **Domain specific objects properties and inter-objects constraints**



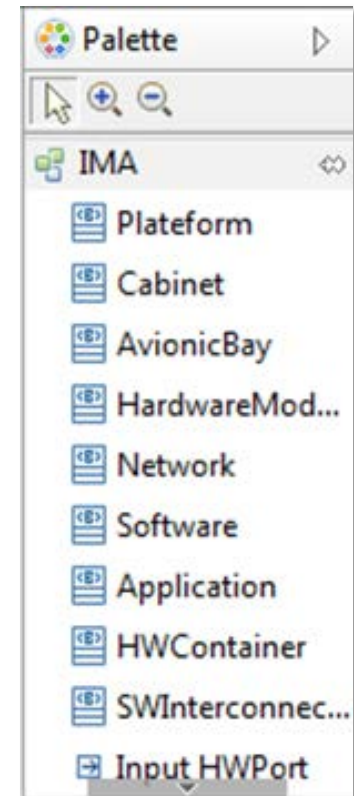
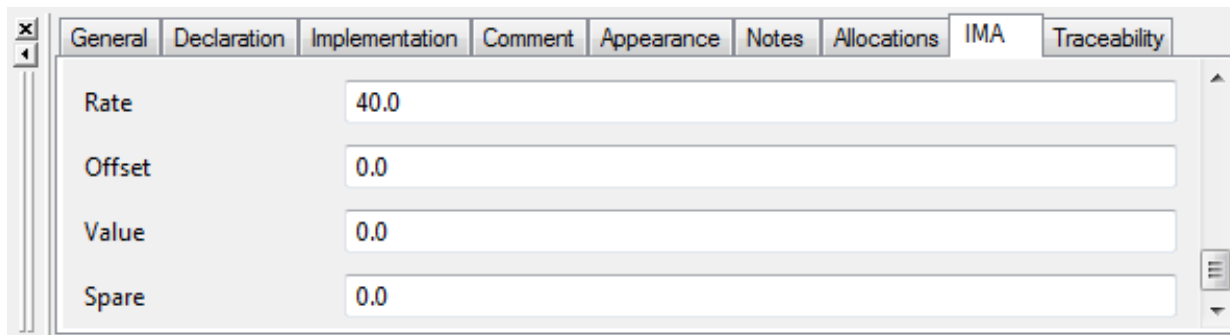
SCADE System configuration 2/2

Automatic packaging of the SCADE System Configuration

- SCADE System Configuration managed as a SCADE project
- Simple deployment on SCADE System end-user machine

System Modeler IDE customization

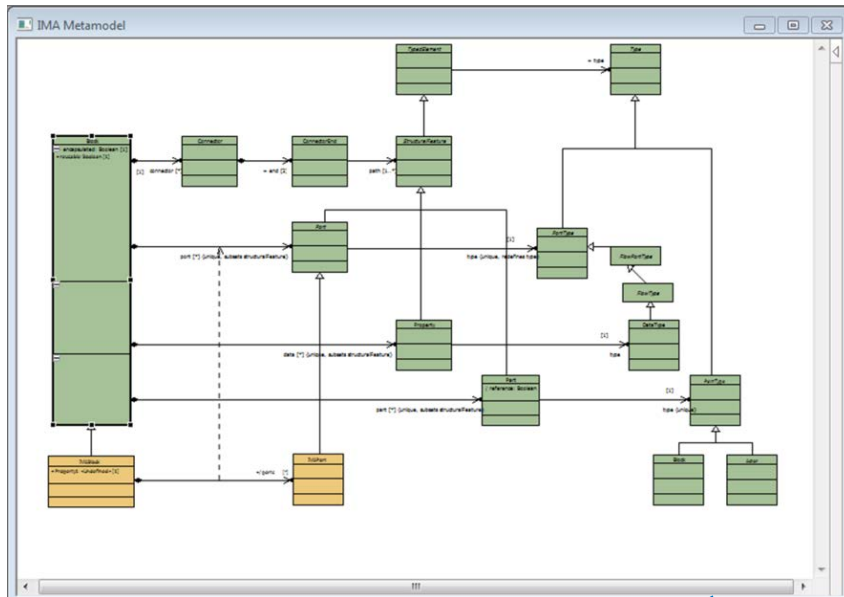
- Dedicated user interface palettes
- Dedicated property view



SCADE System configuration Workflow

Specialist

SCADE System Configurator



Define customized object kinds,
derived from SCADE System objects

Function

- prop1
- prop2

Equipment

- propX
- propY

Bus

- propZ

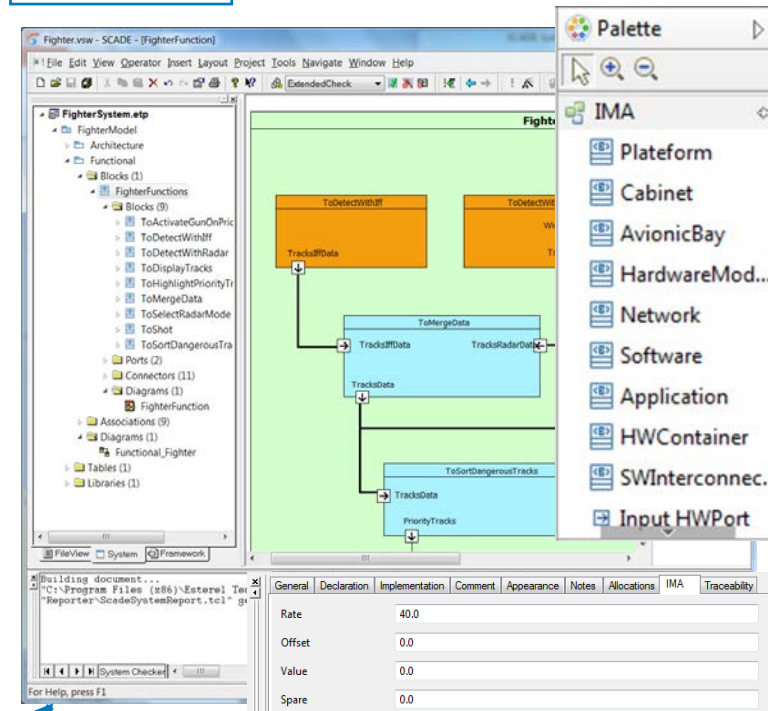
Generate

Deploy

Configuration
Plug-in

End-User

SCADE System Modeler



Domain specific modeler