MathWorks
AUTOMOTIVE CONFERENCE 2023
Korea

Streamline ASPICE Compliance for SYS and SWE Using Model-Based Design

Sean Ryu, MathWorks

MathWorks®

Key Takeaways

Model-Based Design and Model-Based Systems Engineering enable:

1. **Fast development and realization** of **system and software** architecture and design
2. **Early testing** to detect errors in designs and their realization
3. **Fast and efficient iterations**

→ Develop **high quality products** following an **efficient Automotive SPICE® compliant process**

Automotive SPICE® – Reference Model

ID:
SYS.5

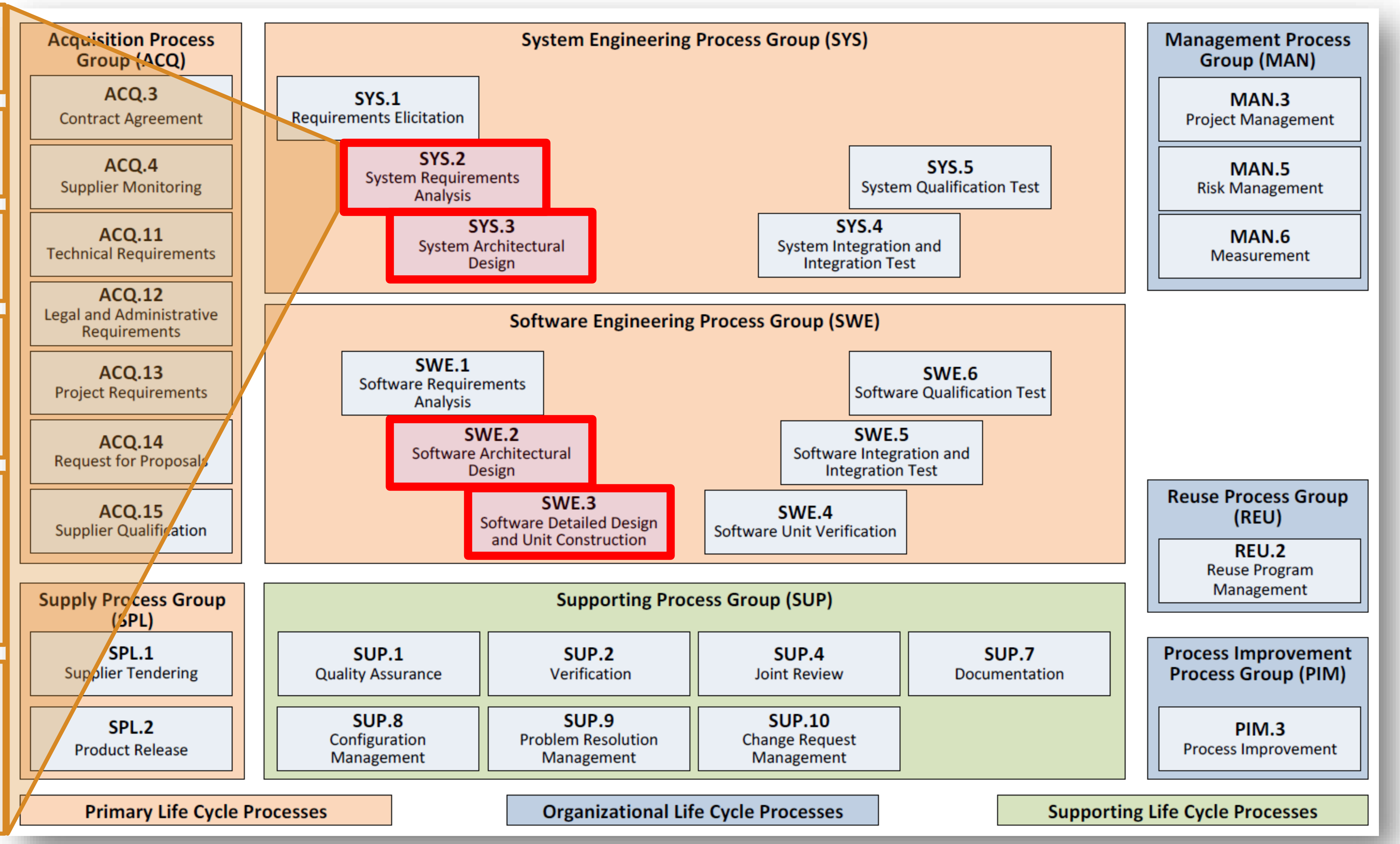
Name:
System Qualification Test

Purpose:

Outcomes:

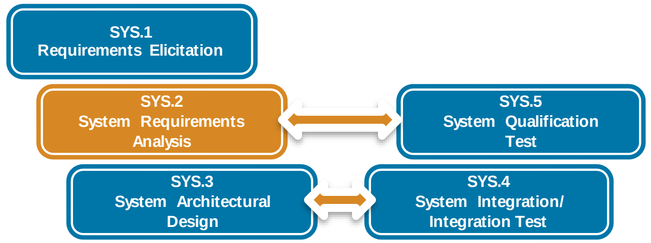
Base Practices:

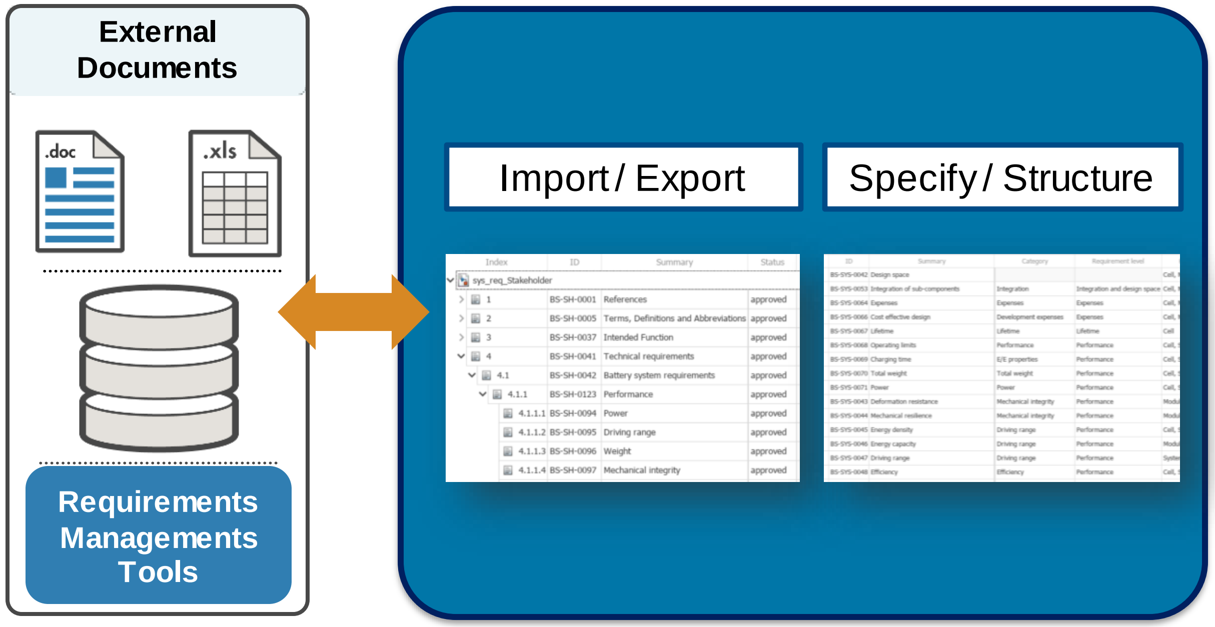
Output Work Products:



Acquisition Process Group (ACQ)
ACQ.3 Contract Agreement
ACQ.4 Supplier Monitoring
ACQ.11 Technical Requirements
ACQ.12 Legal and Administrative Requirements
ACQ.13 Project Requirements
ACQ.14 Request for Proposals
ACQ.15 Supplier Qualification
Supply Process Group (SPL)
SPL.1 Supplier Tendering
SPL.2 Product Release
System Engineering Process Group (SYS)
SYS.1 Requirements Elicitation
SYS.2 System Requirements Analysis
SYS.3 System Architectural Design
SYS.4 System Integration and Integration Test
SYS.5 System Qualification Test
Software Engineering Process Group (SWE)
SWE.1 Software Requirements Analysis
SWE.2 Software Architectural Design
SWE.3 Software Detailed Design and Unit Construction
SWE.4 Software Unit Verification
SWE.5 Software Integration and Integration Test
SWE.6 Software Qualification Test
Supporting Process Group (SUP)
SUP.1 Quality Assurance
SUP.2 Verification
SUP.4 Joint Review
SUP.7 Documentation
SUP.8 Configuration Management
SUP.9 Problem Resolution Management
SUP.10 Change Request Management
Management Process Group (MAN)
MAN.3 Project Management
MAN.5 Risk Management
MAN.6 Measurement
Reuse Process Group (REU)
REU.2 Reuse Program Management
Process Improvement Process Group (PIM)
PIM.3 Process Improvement
Primary Life Cycle Processes
Organizational Life Cycle Processes
Supporting Life Cycle Processes

SYS.2 System Requirements Analysis

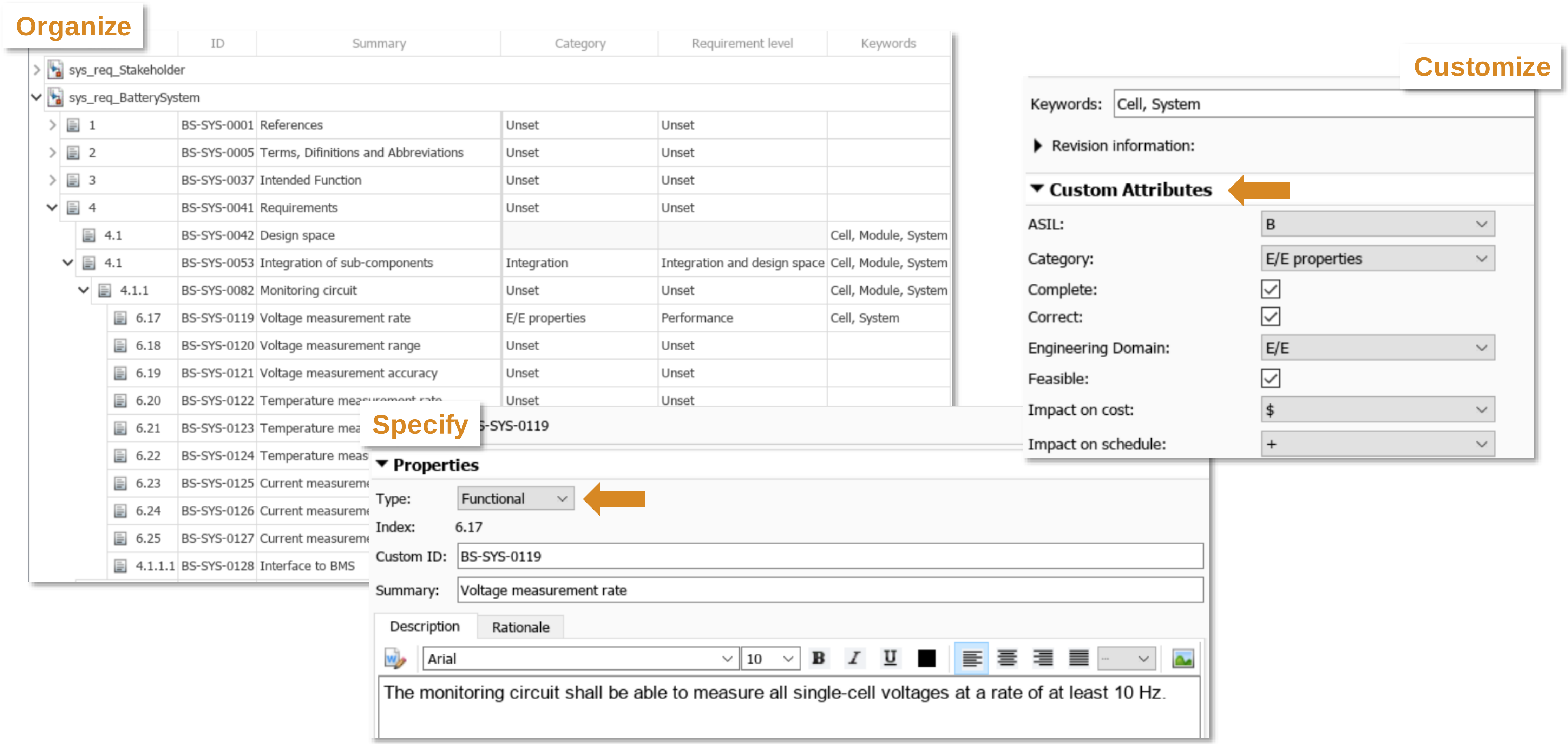

SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test



Index	ID	Summary	Status
sys_req_Stakeholder			
1	BS-SH-0001	References	approved
2	BS-SH-0005	Terms, Definitions and Abbreviations	approved
3	BS-SH-0037	Intended Function	approved
4	BS-SH-0041	Technical requirements	approved
4.1	BS-SH-0042	Battery system requirements	approved
4.1.1	BS-SH-0123	Performance	approved
4.1.1.1	BS-SH-0094	Power	approved
4.1.1.2	BS-SH-0095	Driving range	approved
4.1.1.3	BS-SH-0096	Weight	approved
4.1.1.4	BS-SH-0097	Mechanical integrity	approved

ID	Summary	Category	Requirement level	
BS-SYS-0042	Design space			Cell,
BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell,
BS-SYS-0064	Expenses	Expenses	Expenses	Cell,
BS-SYS-0066	Cost effective design	Development expenses	Expenses	Cell,
BS-SYS-0067	Lifetime	Lifetime	Lifetime	Cell
BS-SYS-0068	Operating limits	Performance	Performance	Cell,
BS-SYS-0069	Charging time	E/E properties	Performance	Cell,
BS-SYS-0070	Total weight	Total weight	Performance	Cell,
BS-SYS-0071	Power	Power	Performance	Cell,
BS-SYS-0043	Deformation resistance	Mechanical integrity	Performance	Modu
BS-SYS-0044	Mechanical resilience	Mechanical integrity	Performance	Modu
BS-SYS-0045	Energy density	Driving range	Performance	Cell,
BS-SYS-0046	Energy capacity	Driving range	Performance	Modu
BS-SYS-0047	Driving range	Driving range	Performance	Syste
BS-SYS-0048	Efficiency	Efficiency	Performance	Cell,

Organize, Specify and Customize Requirements with Requirements Toolbox



Organize

	ID	Summary	Category	Requirement level	Keywords
sys_req_Stakeholder					
sys_req_BatterySystem					
1	BS-SYS-0001	References	Unset	Unset	
2	BS-SYS-0005	Terms, Difinitions and Abbreviations	Unset	Unset	
3	BS-SYS-0037	Intended Function	Unset	Unset	
4	BS-SYS-0041	Requirements	Unset	Unset	
4.1	BS-SYS-0042	Design space			Cell, Module, System
4.1	BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell, Module, System
4.1.1	BS-SYS-0082	Monitoring circuit	Unset	Unset	Cell, Module, System
6.17	BS-SYS-0119	Voltage measurement rate	E/E properties	Performance	Cell, System
6.18	BS-SYS-0120	Voltage measurement range	Unset	Unset	
6.19	BS-SYS-0121	Voltage measurement accuracy	Unset	Unset	
6.20	BS-SYS-0122	Temperature measurement rate	Unset	Unset	
6.21	BS-SYS-0123	Temperature mea			
6.22	BS-SYS-0124	Temperature meas			
6.23	BS-SYS-0125	Current measureme			
6.24	BS-SYS-0126	Current measureme			
6.25	BS-SYS-0127	Current measureme			
4.1.1.1	BS-SYS-0128	Interface to BMS			

Specify

S-SYS-0119

Properties

Type: Functional

Index: 6.17

Custom ID: BS-SYS-0119

Summary: Voltage measurement rate

Description | Rationale

Arial 10

The monitoring circuit shall be able to measure all single-cell voltages at a rate of at least 10 Hz.

Customize

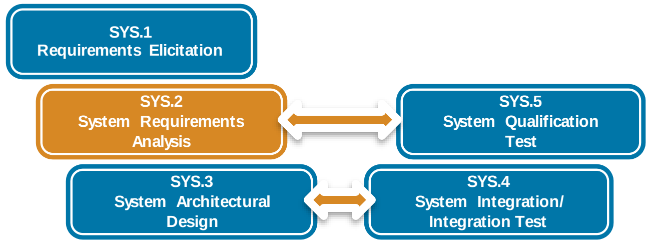
Keywords: Cell, System

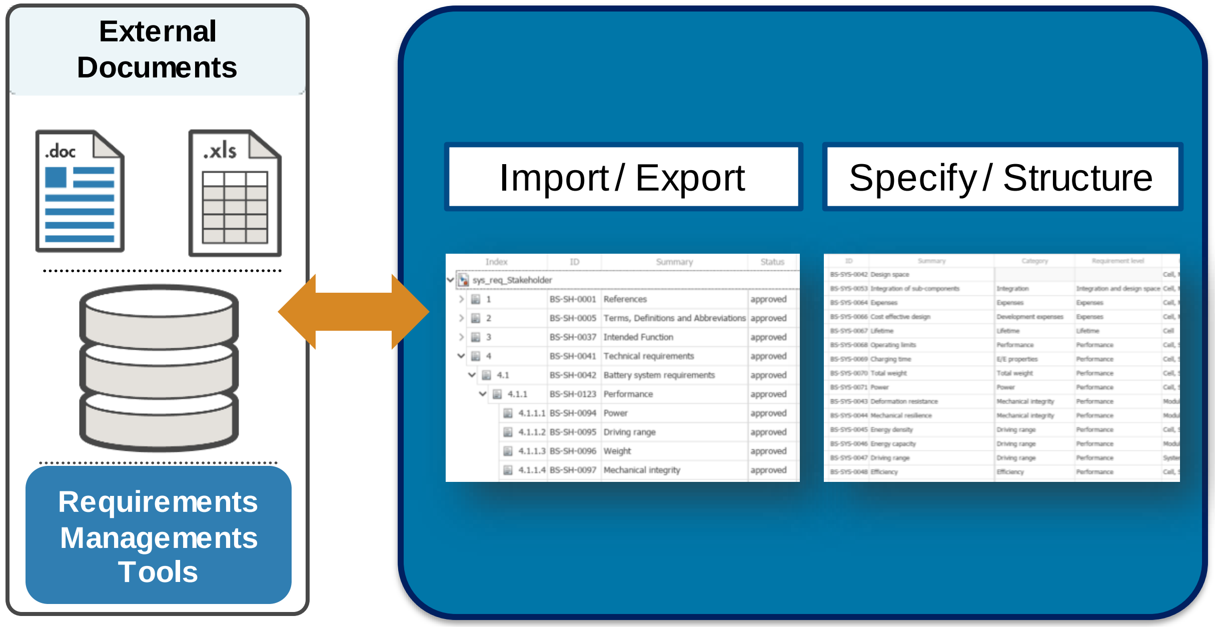
Revision information:

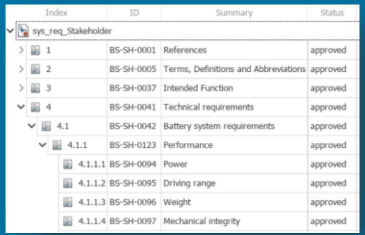
Custom Attributes

ASIL:	B
Category:	E/E properties
Complete:	✓
Correct:	✓
Engineering Domain:	E/E
Feasible:	✓
Impact on cost:	$
Impact on schedule:	+

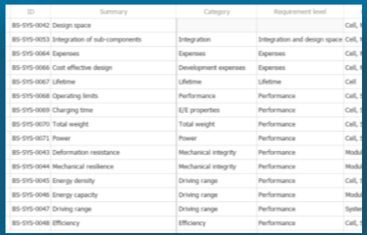
SYS.2 System Requirements Analysis


SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test



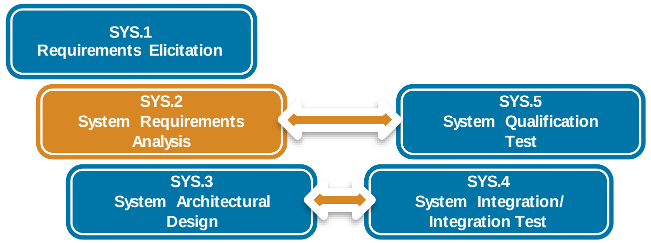


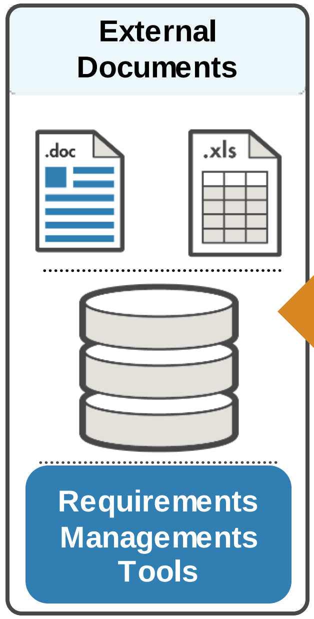
Index	ID	Summary	Status
sys_req_Stakeholder			
1	BS-SH-0001	References	approved
2	BS-SH-0005	Terms, Definitions and Abbreviations	approved
3	BS-SH-0037	Intended Function	approved
4	BS-SH-0041	Technical requirements	approved
4.1	BS-SH-0042	Battery system requirements	approved
4.1.1	BS-SH-0123	Performance	approved
4.1.1.1	BS-SH-0094	Power	approved
4.1.1.2	BS-SH-0095	Driving range	approved
4.1.1.3	BS-SH-0096	Weight	approved
4.1.1.4	BS-SH-0097	Mechanical integrity	approved



ID	Summary	Category	Requirement level	
BS-SYS-0042	Design space			Cell,
BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell,
BS-SYS-0064	Expenses	Expenses	Expenses	Cell,
BS-SYS-0066	Cost effective design	Development expenses	Expenses	Cell,
BS-SYS-0067	Lifetime	Lifetime	Lifetime	Cell
BS-SYS-0068	Operating limits	Performance	Performance	Cell,
BS-SYS-0069	Charging time	E/E properties	Performance	Cell,
BS-SYS-0070	Total weight	Total weight	Performance	Cell,
BS-SYS-0071	Power	Power	Performance	Cell,
BS-SYS-0043	Deformation resistance	Mechanical integrity	Performance	Modu
BS-SYS-0044	Mechanical resilience	Mechanical integrity	Performance	Modu
BS-SYS-0045	Energy density	Driving range	Performance	Cell,
BS-SYS-0046	Energy capacity	Driving range	Performance	Modu
BS-SYS-0047	Driving range	Driving range	Performance	Syste
BS-SYS-0048	Efficiency	Efficiency	Performance	Cell,

SYS.2 System Requirements Analysis

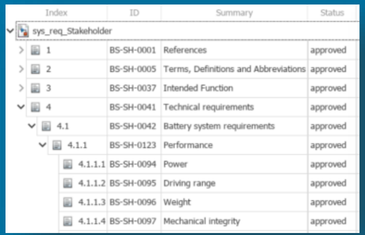

SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test



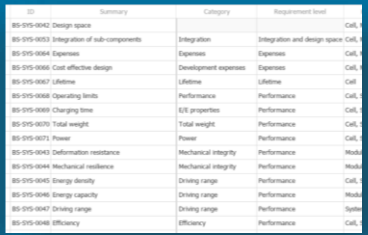
Import / Export

Specify / Structure

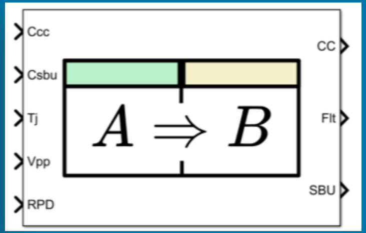
Analyze / Simulate



Index	ID	Summary	Status
sys_req_Stakeholder			
1	BS-SH-0001	References	approved
2	BS-SH-0005	Terms, Definitions and Abbreviations	approved
3	BS-SH-0037	Intended Function	approved
4	BS-SH-0041	Technical requirements	approved
4.1	BS-SH-0042	Battery system requirements	approved
4.1.1	BS-SH-0123	Performance	approved
4.1.1.1	BS-SH-0094	Power	approved
4.1.1.2	BS-SH-0095	Driving range	approved
4.1.1.3	BS-SH-0096	Weight	approved
4.1.1.4	BS-SH-0097	Mechanical integrity	approved

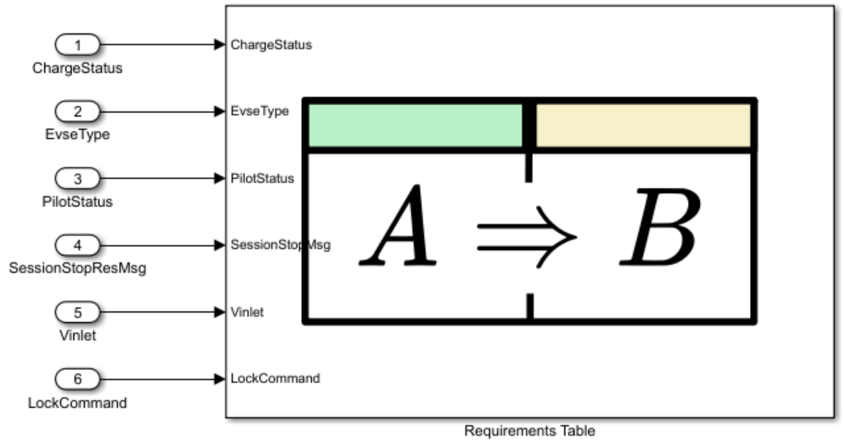


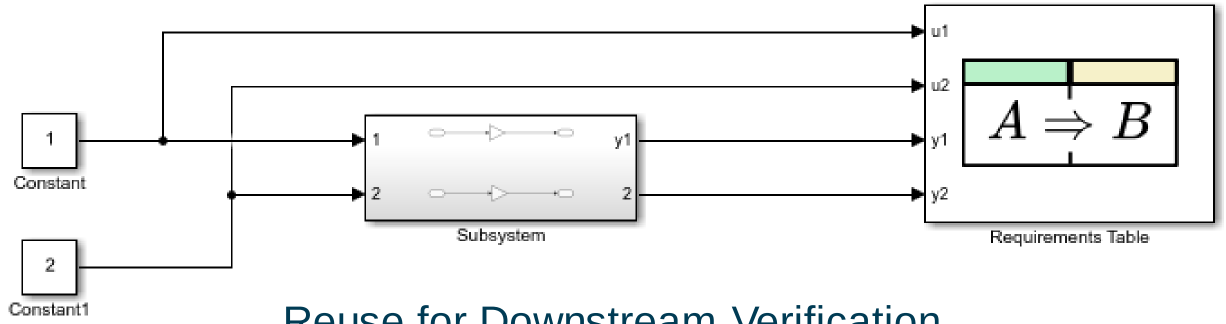
ID	Summary	Category	Requirement level	
BS-SYS-0042	Design space			Cell,
BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell,
BS-SYS-0064	Expenses	Expenses	Expenses	Cell,
BS-SYS-0066	Cost effective design	Development expenses	Expenses	Cell,
BS-SYS-0067	Lifetime	Lifetime	Lifetime	Cell
BS-SYS-0068	Operating limits	Performance	Performance	Cell,
BS-SYS-0069	Charging time	E/E properties	Performance	Cell,
BS-SYS-0070	Total weight	Total weight	Performance	Cell,
BS-SYS-0071	Power	Power	Performance	Cell,
BS-SYS-0043	Deformation resistance	Mechanical integrity	Performance	Modu
BS-SYS-0044	Mechanical resilience	Mechanical integrity	Performance	Modu
BS-SYS-0045	Energy density	Driving range	Performance	Cell,
BS-SYS-0046	Energy capacity	Driving range	Performance	Modu
BS-SYS-0047	Driving range	Driving range	Performance	Syste
BS-SYS-0048	Efficiency	Efficiency	Performance	Cell,



Analyze Logical Requirements with the Requirements Table R2022a

- Mathematically rigorous
- Executable through simulation
- Easier to author and maintain than text requirements
- Easier to analyze as requirement set grows
- Reusable for verification activities

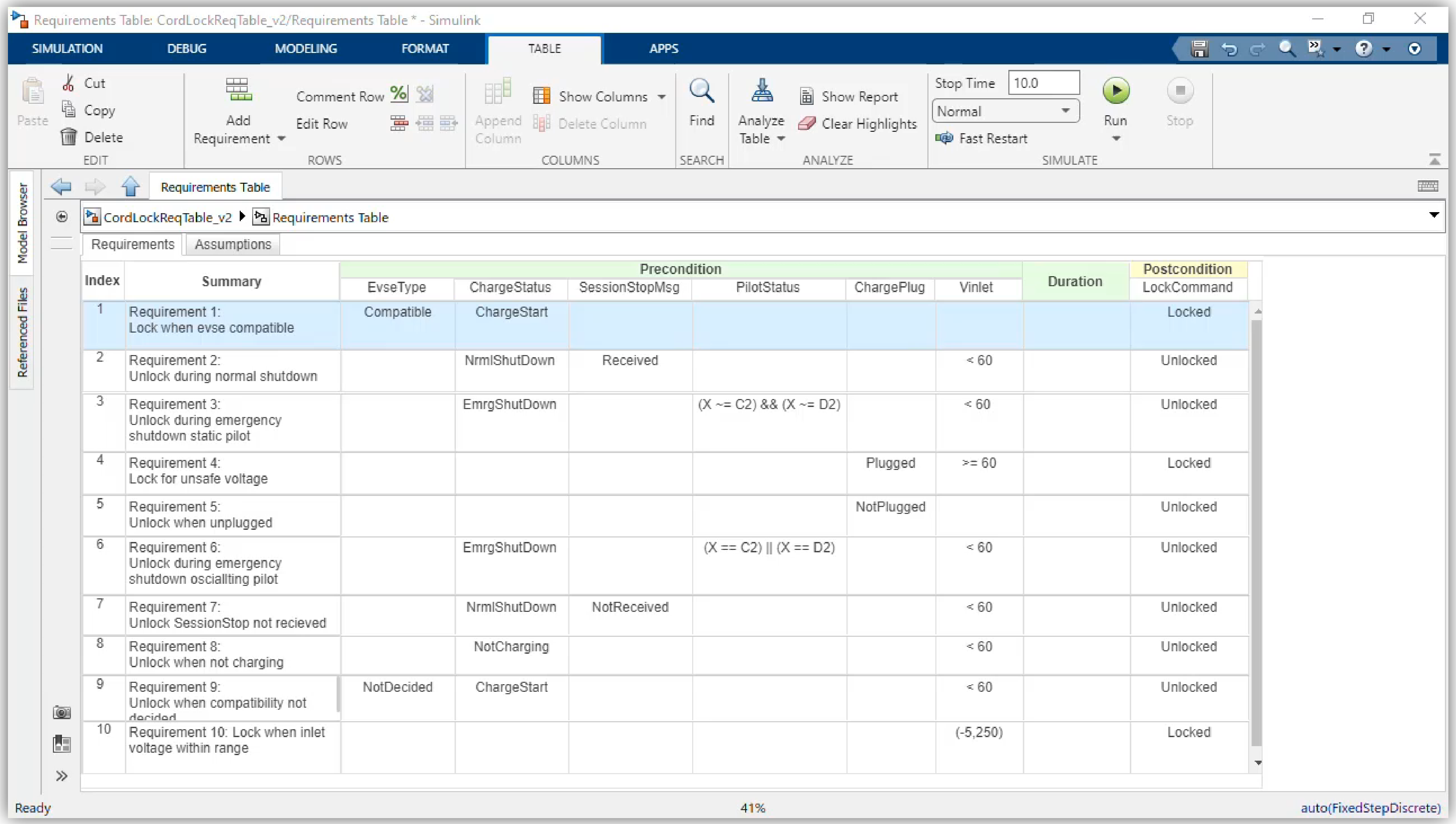




Reuse for Downstream Verification

15:50	Formal Requirements 작성방법과 Requirements-Based Test 를 위한 Test Case 생성 유재흥 부장, *매스웍스코리아*

Analyze Logical Requirements with the Requirements Table



Requirements Table: CordLockReqTable_v2/Requirements Table * - Simulink

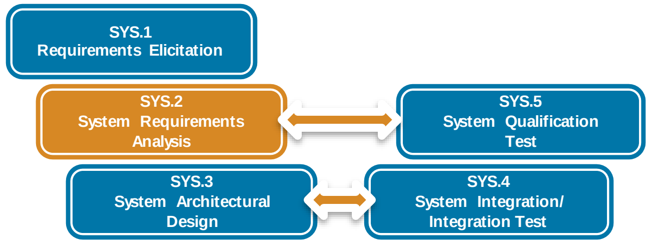
Stop Time 10.0

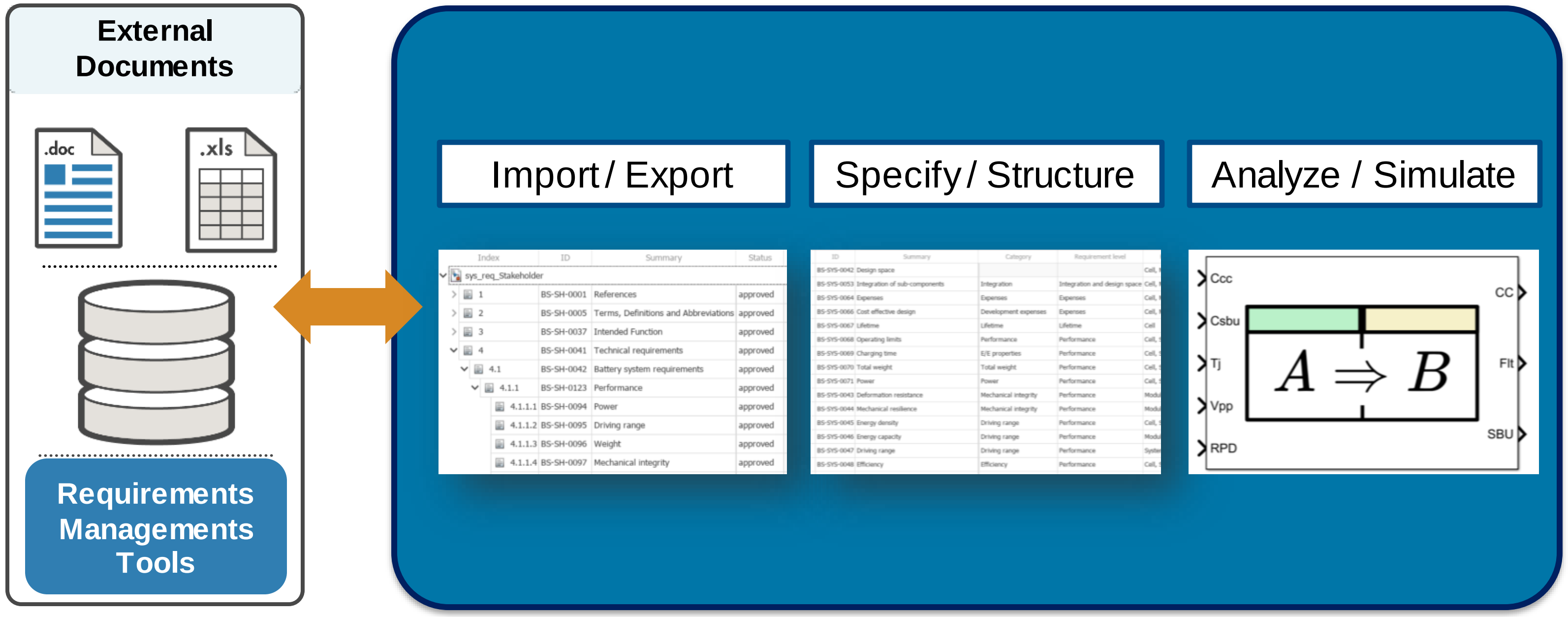
Normal

Index	Summary	Precondition						Duration	Postcondition
		EvseType	ChargeStatus	SessionStopMsg	PilotStatus	ChargePlug	VInlet		LockCommand
1	Requirement 1: Lock when evse compatible	Compatible	ChargeStart						Locked
2	Requirement 2: Unlock during normal shutdown		NrmlShutDown	Received			< 60		Unlocked
3	Requirement 3: Unlock during emergency shutdown static pilot		EmrgShutDown		(X ~= C2) && (X ~= D2)		< 60		Unlocked
4	Requirement 4: Lock for unsafe voltage					Plugged	>= 60		Locked
5	Requirement 5: Unlock when unplugged					NotPlugged			Unlocked
6	Requirement 6: Unlock during emergency shutdown oscialling pilot		EmrgShutDown		(X == C2) || (X == D2)		< 60		Unlocked
7	Requirement 7: Unlock SessionStop not recieved		NrmlShutDown	NotReceived			< 60		Unlocked
8	Requirement 8: Unlock when not charging		NotCharging				< 60		Unlocked
9	Requirement 9: Unlock when compatibility not decided	NotDecided	ChargeStart				< 60		Unlocked
10	Requirement 10: Lock when inlet voltage within range						(-5,250)		Locked

Ready 41% auto(FixedStepDiscrete)

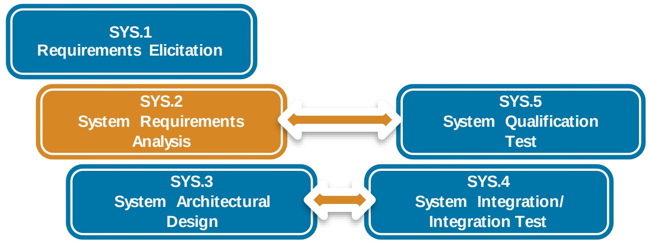
SYS.2 System Requirements Analysis


SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test

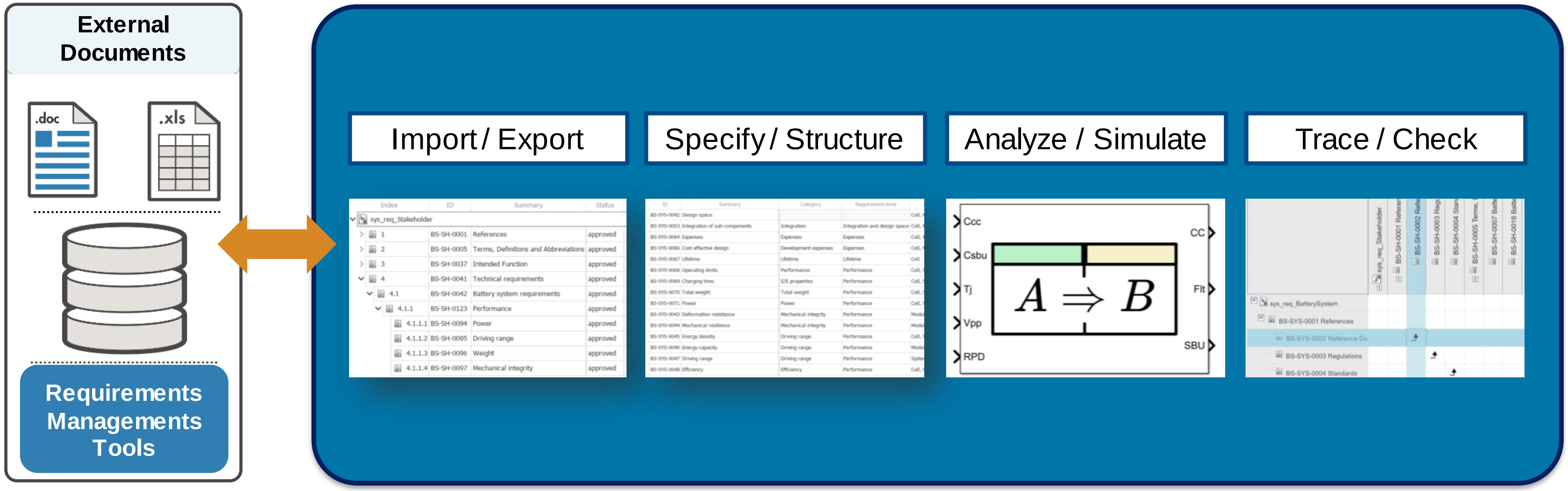


Index	ID	Summary	Status
sys_req_Stakeholder			
1	BS-SH-0001	References	approved
2	BS-SH-0005	Terms, Definitions and Abbreviations	approved
3	BS-SH-0037	Intended Function	approved
4	BS-SH-0041	Technical requirements	approved
4.1	BS-SH-0042	Battery system requirements	approved
4.1.1	BS-SH-0123	Performance	approved
4.1.1.1	BS-SH-0094	Power	approved
4.1.1.2	BS-SH-0095	Driving range	approved
4.1.1.3	BS-SH-0096	Weight	approved
4.1.1.4	BS-SH-0097	Mechanical integrity	approved

ID	Summary	Category	Requirement level	
BS-SYS-0042	Design space			Cell,
BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell,
BS-SYS-0064	Expenses	Expenses	Expenses	Cell,
BS-SYS-0066	Cost effective design	Development expenses	Expenses	Cell,
BS-SYS-0067	Lifetime	Lifetime	Lifetime	Cell
BS-SYS-0068	Operating limits	Performance	Performance	Cell,
BS-SYS-0069	Charging time	E/E properties	Performance	Cell,
BS-SYS-0070	Total weight	Total weight	Performance	Cell,
BS-SYS-0071	Power	Power	Performance	Cell,
BS-SYS-0043	Deformation resistance	Mechanical integrity	Performance	Modul
BS-SYS-0044	Mechanical resilience	Mechanical integrity	Performance	Modul
BS-SYS-0045	Energy density	Driving range	Performance	Cell,
BS-SYS-0046	Energy capacity	Driving range	Performance	Modul
BS-SYS-0047	Driving range	Driving range	Performance	Syste
BS-SYS-0048	Efficiency	Efficiency	Performance	Cell,


SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test

SYS.2 System Requirements Analysis



Index	ID	Summary	Status
sys_req_Stakeholder			
1	BS-SH-0001	References	approved
2	BS-SH-0005	Terms, Definitions and Abbreviations	approved
3	BS-SH-0037	Intended Function	approved
4	BS-SH-0041	Technical requirements	approved
4.1	BS-SH-0042	Battery system requirements	approved
4.1.1	BS-SH-0123	Performance	approved
4.1.1.1	BS-SH-0094	Power	approved
4.1.1.2	BS-SH-0095	Driving range	approved
4.1.1.3	BS-SH-0096	Weight	approved
4.1.1.4	BS-SH-0097	Mechanical integrity	approved

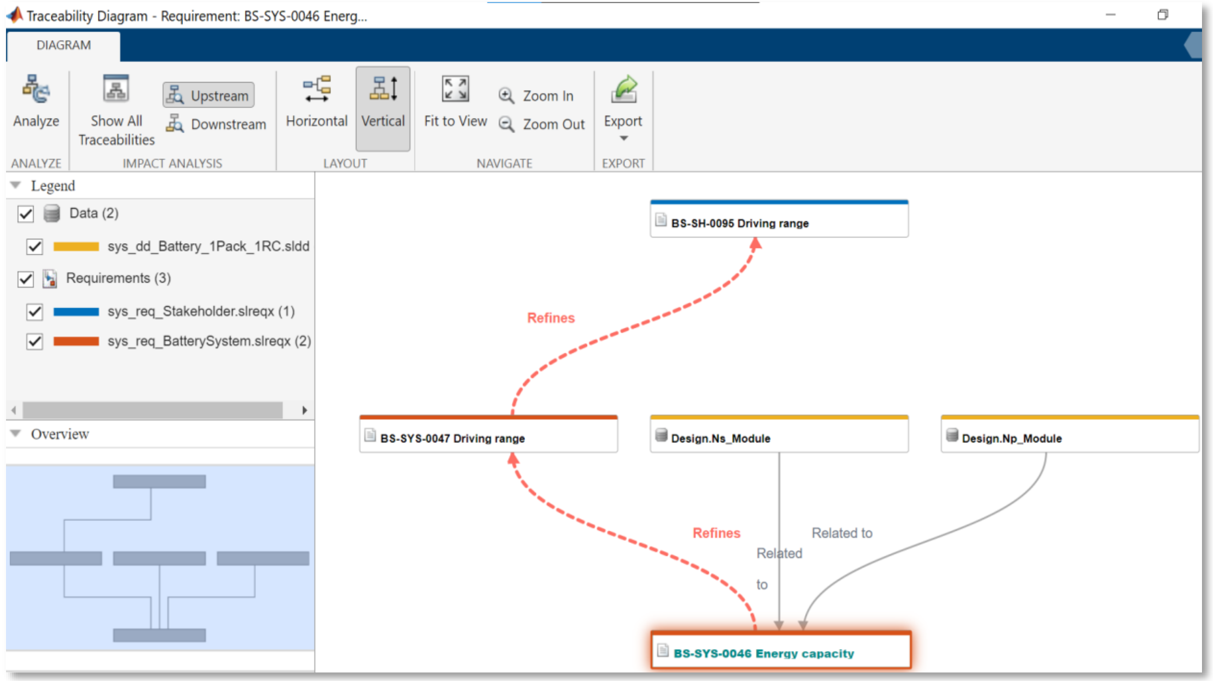
ID	Summary	Category	Requirement level	
BS-SYS-0042	Design space			Cell,
BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell,
BS-SYS-0064	Expenses	Expenses	Expenses	Cell,
BS-SYS-0066	Cost effective design	Development expenses	Expenses	Cell,
BS-SYS-0067	Lifetime	Lifetime	Lifetime	Cell
BS-SYS-0068	Operating limits	Performance	Performance	Cell,
BS-SYS-0069	Charging time	E/E properties	Performance	Cell,
BS-SYS-0070	Total weight	Total weight	Performance	Cell,
BS-SYS-0071	Power	Power	Performance	Cell,
BS-SYS-0043	Deformation resistance	Mechanical integrity	Performance	Modul
BS-SYS-0044	Mechanical resilience	Mechanical integrity	Performance	Modul
BS-SYS-0045	Energy density	Driving range	Performance	Cell,
BS-SYS-0046	Energy capacity	Driving range	Performance	Modul
BS-SYS-0047	Driving range	Driving range	Performance	Syster
BS-SYS-0048	Efficiency	Efficiency	Performance	Cell,

	sys_req_Stakeholder	BS-SH-0001 Referen	BS-SH-0002 Refe	BS-SH-0003 Regu	BS-SH-0004 Stan	BS-SH-0005 Terms,	BS-SH-0007 Batte	BS-SH-0018 Batte
sys_req_BatterySystem								
BS-SYS-0001 References								
BS-SYS-0002 Reference Do			↑					
BS-SYS-0003 Regulations				↑				
BS-SYS-0004 Standards					↑			

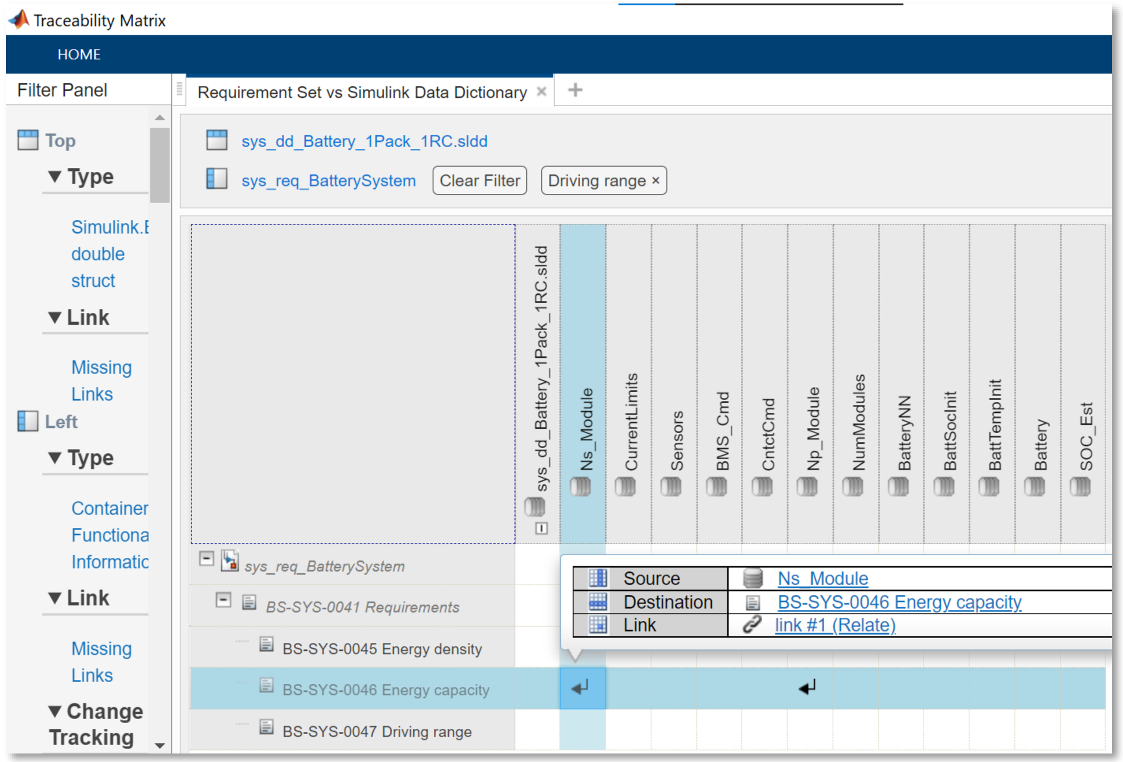
Use Traceability Diagrams and Matrixes to Check for Consistency and Completeness

Traceability Diagrams

Traceability Matrix



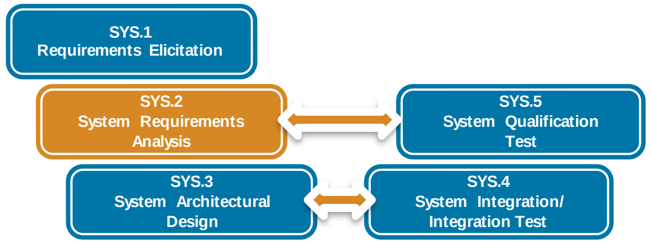
[Visualize Links with Traceability Diagrams]



Source	Ns_Module
Destination	BS-SYS-0046 Energy capacity
Link	link #1 (Relate)

[Track Requiement Links with a Traceability Matrix]

SYS.2 System Requirements Analysis

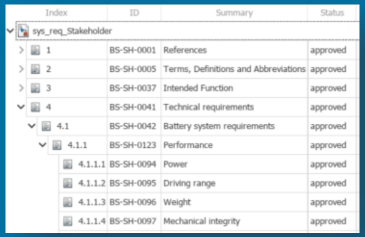

SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test

External Documents

.doc .xls

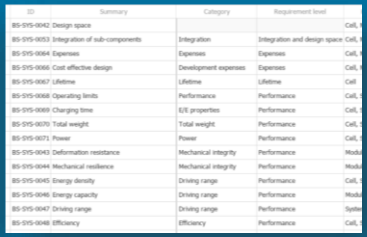
Requirements Managements Tools

Import / Export



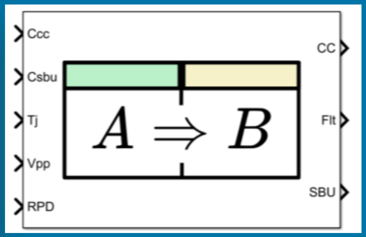
Index	ID	Summary	Status
sys_req_Stakeholder			
1	BS-SH-0001	References	approved
2	BS-SH-0005	Terms, Definitions and Abbreviations	approved
3	BS-SH-0037	Intended Function	approved
4	BS-SH-0041	Technical requirements	approved
4.1	BS-SH-0042	Battery system requirements	approved
4.1.1	BS-SH-0123	Performance	approved
4.1.1.1	BS-SH-0094	Power	approved
4.1.1.2	BS-SH-0095	Driving range	approved
4.1.1.3	BS-SH-0096	Weight	approved
4.1.1.4	BS-SH-0097	Mechanical integrity	approved

Specify / Structure

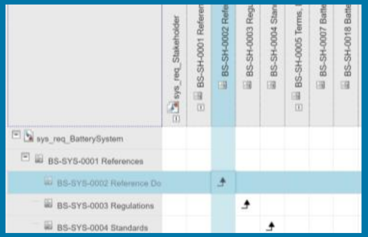


ID	Summary	Category	Requirement level	
BS-SYS-0042	Design space			Cell, M
BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell, M
BS-SYS-0064	Expenses	Expenses	Expenses	Cell, M
BS-SYS-0066	Cost effective design	Development expenses	Expenses	Cell, M
BS-SYS-0067	Lifetime	Lifetime	Lifetime	Cell
BS-SYS-0068	Operating limits	Performance	Performance	Cell, S
BS-SYS-0069	Charging time	E/E properties	Performance	Cell, S
BS-SYS-0070	Total weight	Total weight	Performance	Cell, S
BS-SYS-0071	Power	Power	Performance	Cell, S
BS-SYS-0043	Deformation resistance	Mechanical integrity	Performance	Modul
BS-SYS-0044	Mechanical resilience	Mechanical integrity	Performance	Modul
BS-SYS-0045	Energy density	Driving range	Performance	Cell, S
BS-SYS-0046	Energy capacity	Driving range	Performance	Modul
BS-SYS-0047	Driving range	Driving range	Performance	Syster
BS-SYS-0048	Efficiency	Efficiency	Performance	Cell, S

Analyze / Simulate



Trace / Check



	sys_req_Stakeholder	BS-SH-0001 Referen	BS-SH-0002 Refe	BS-SH-0003 Regu	BS-SH-0004 Stan	BS-SH-0005 Terms,	BS-SH-0007 Batte	BS-SH-0018 Batte
sys_req_BatterySystem								
BS-SYS-0001 References								
BS-SYS-0002 Reference Do			↑					
BS-SYS-0003 Regulations				↑				
BS-SYS-0004 Standards					↑			

Output Work Products

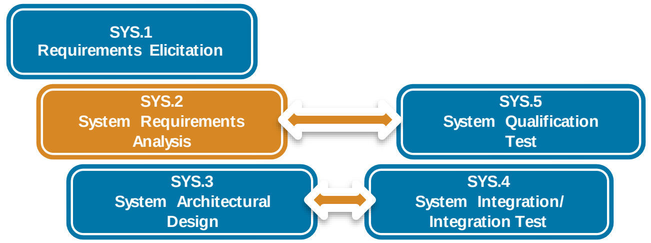
System requirements

Interfaces

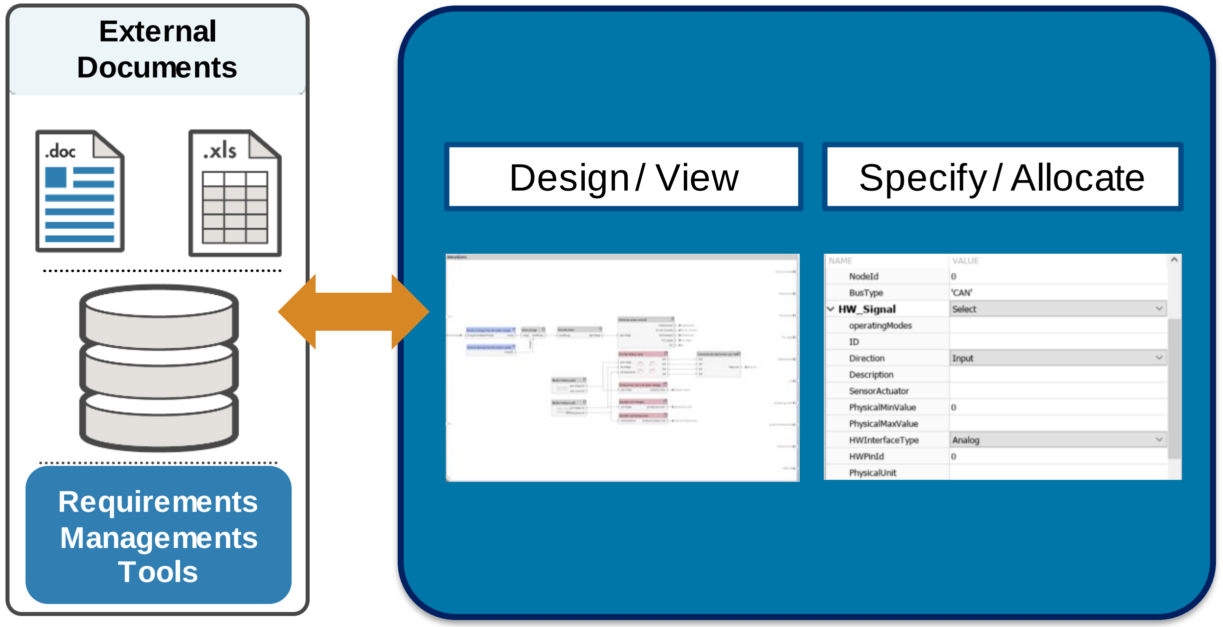
Traceability

Analysis report

Verification criteria


SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test

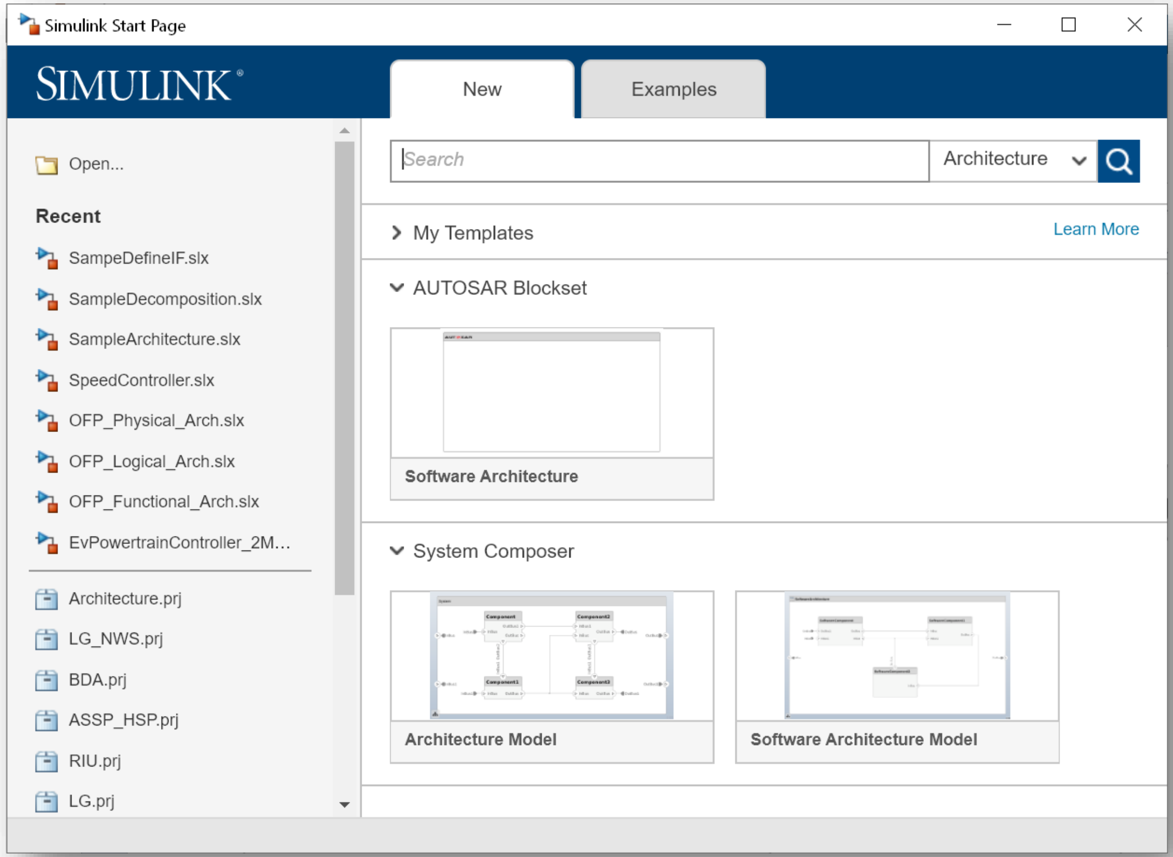
SYS.3 System Architectural Design



NAME	VALUE
NodeId	0
BusType	'CAN'
HW_Signal	Select
operatingModes	
ID	
Direction	Input
Description	
SensorActuator	
PhysicalMinValue	0
PhysicalMaxValue	
HWInterfaceType	Analog
HWPinId	0
PhysicalUnit	

Develop Architectural Design Models with System Composer

Default templates for architecture design



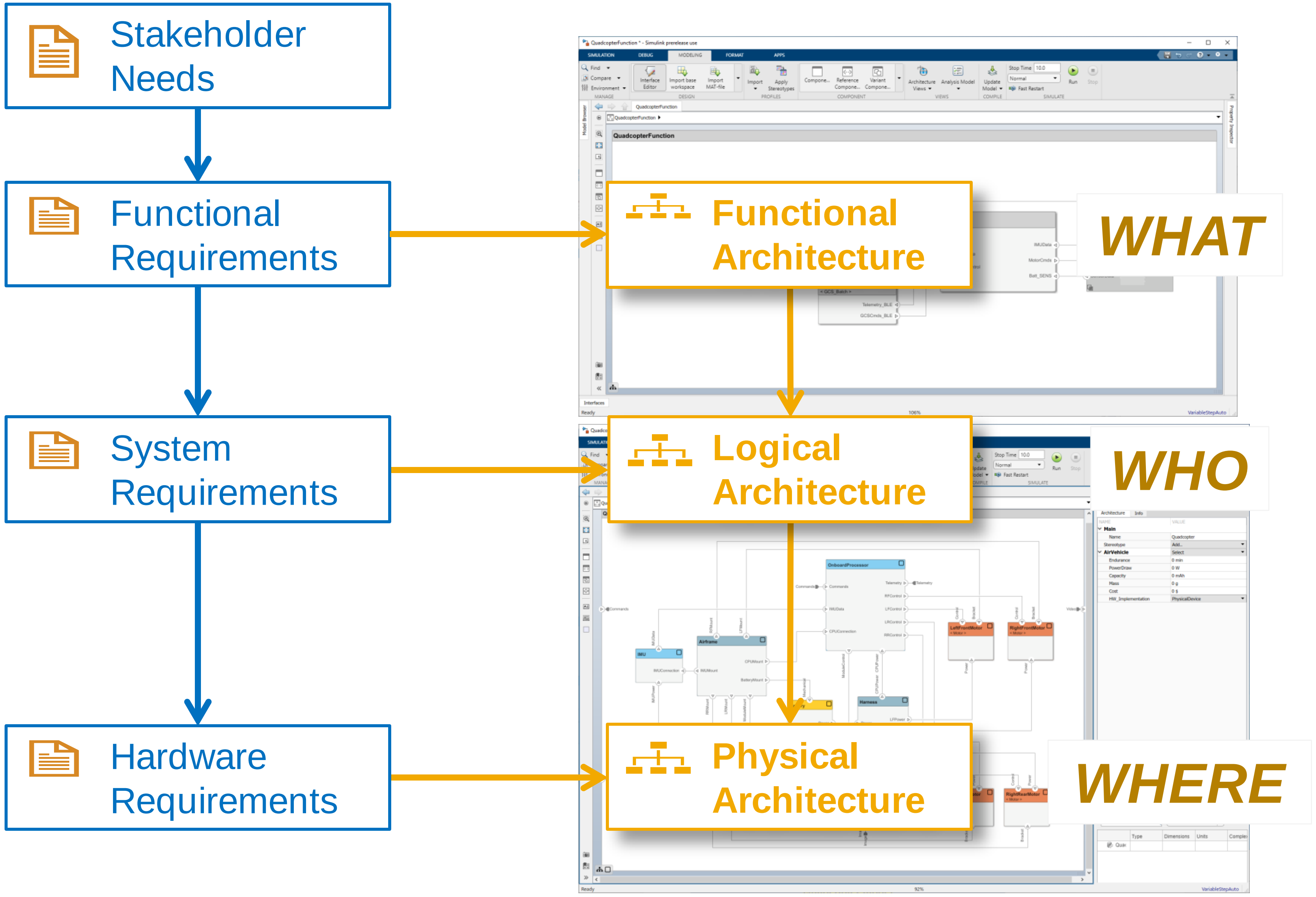
```
>> systemcomposer
```

- **System Architecture**
  - Enable the specification and analysis of architectures for model-based systems engineering and software architecture modeling
  - Behaviors can be captured in sequence diagram, state charts, or Simulink models

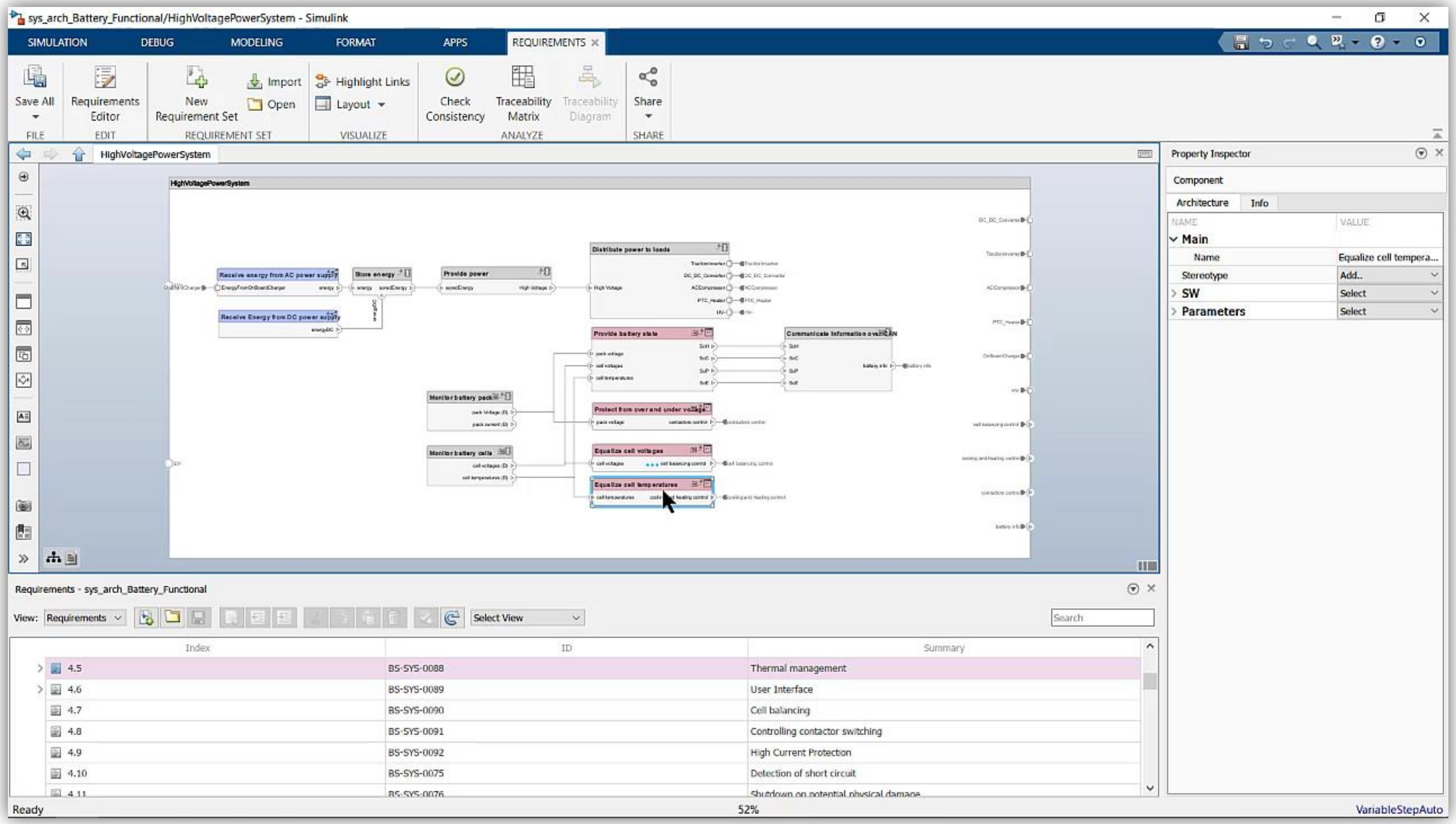
- **Software Architecture**
  - More compatible to Simulink model such as export-function, rate-based, or JMAAB models
  - Simulate and generate code from the architecture model as well as having features of System Composer to analyze architecture

- **AUTOSAR ASW Architecture (+ AUTOSAR Blockset)**
  - From the architecture model, adding and connecting AUTOSAR compositions and components, or import a composition from ARXML files
  - Export composition and component ARXML descriptions and generate component code (+ Embedded Coder)

Develop Architectural Design Models with System Composer



Develop Architectural Design Models with System Composer



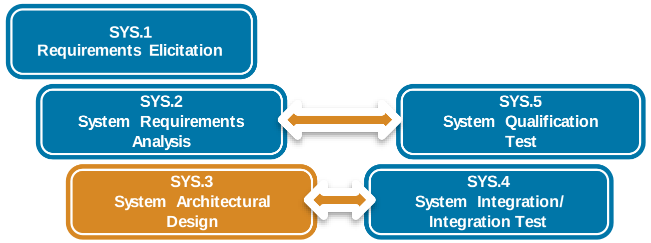
NAME	VALUE
Main	
Name	Equalize cell tempera...
Stereotype	Add..
SW	Select
Parameters	Select

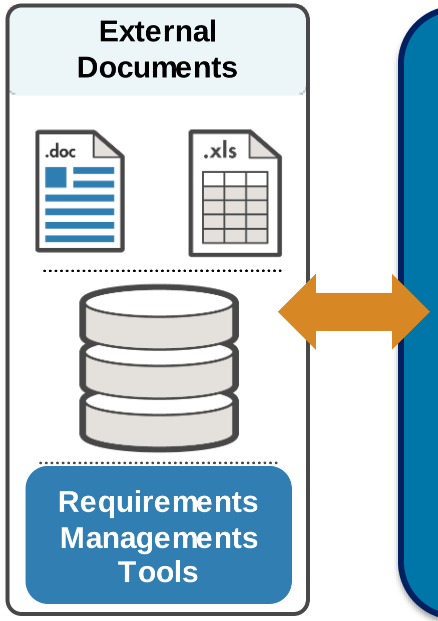
Requirements - sys_arch_Battery_Functional

View: Requirements | Select View | Search

Index	ID	Summary
4.5	BS-SYS-0088	Thermal management
4.6	BS-SYS-0089	User Interface
4.7	BS-SYS-0090	Cell balancing
4.8	BS-SYS-0091	Controlling contactor switching
4.9	BS-SYS-0092	High Current Protection
4.10	BS-SYS-0075	Detection of short circuit
4.11	BS-SYS-0076	Shutdown on potential physical damage

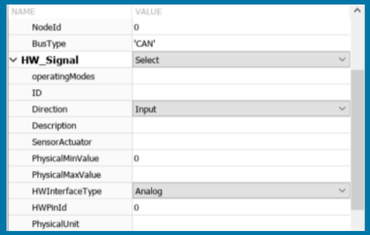
SYS.3 System Architectural Design


SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test



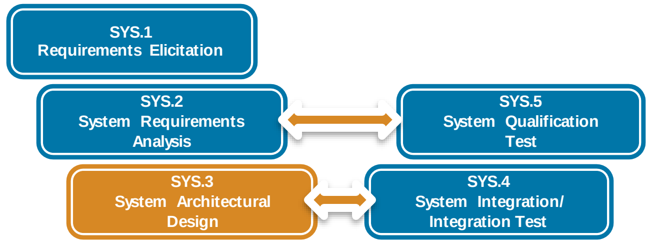
Design / View

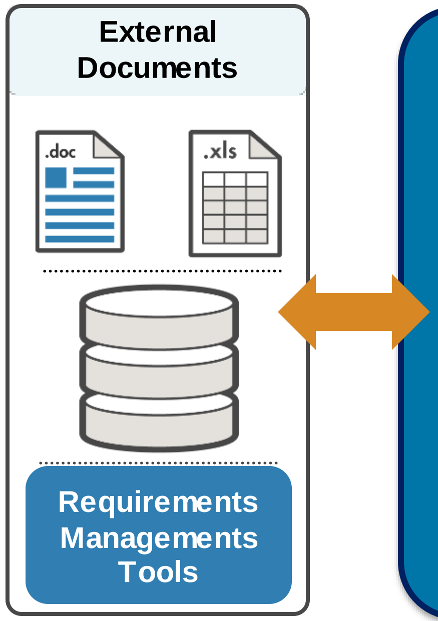
Specify / Allocate



NAME	VALUE
NodeId	0
BusType	'CAN'
HW_Signal	Select
operatingModes	
ID	
Direction	Input
Description	
SensorActuator	
PhysicalMinValue	0
PhysicalMaxValue	
HWInterfaceType	Analog
HWPinId	0
PhysicalUnit	

SYS.3 System Architectural Design

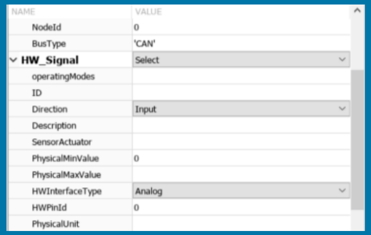

SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test



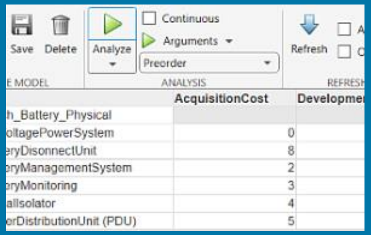
Design / View

Specify / Allocate

Analyze



NAME	VALUE
NodeId	0
BusType	'CAN'
HW_Signal	Select
operatingModes	
ID	
Direction	Input
Description	
SensorActuator	
PhysicalMinValue	0
PhysicalMaxValue	
HWInterfaceType	Analog
HWPinId	0
PhysicalUnit	



	AcquisitionCost	Developme
h_Battery_Physical		
oltagePowerSystem	0	
eryDisonnectUnit	8	
eryManagementSystem	2	
eryMonitoring	3	
allsolator	4	
erDistributionUnit (PDU)	5	

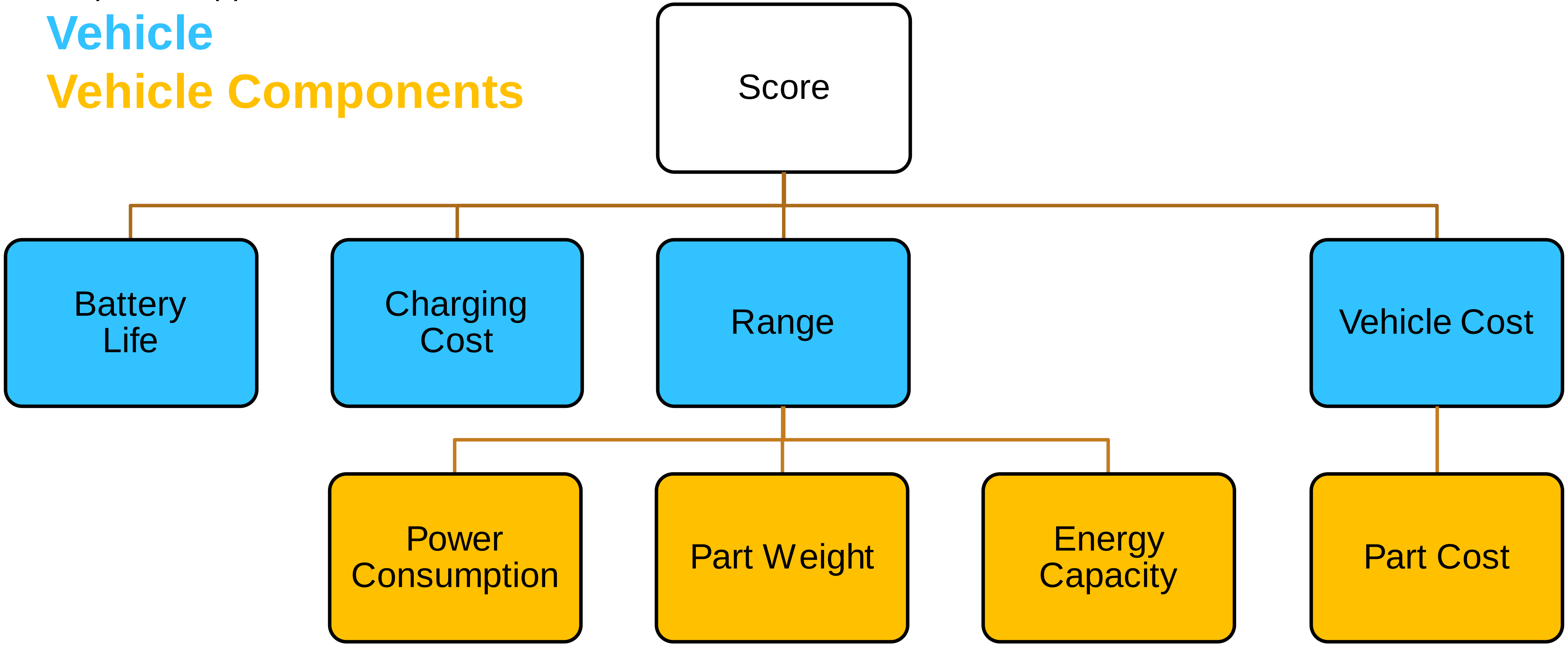
Analyze Architectural Design Models

Example

Properties applied to:

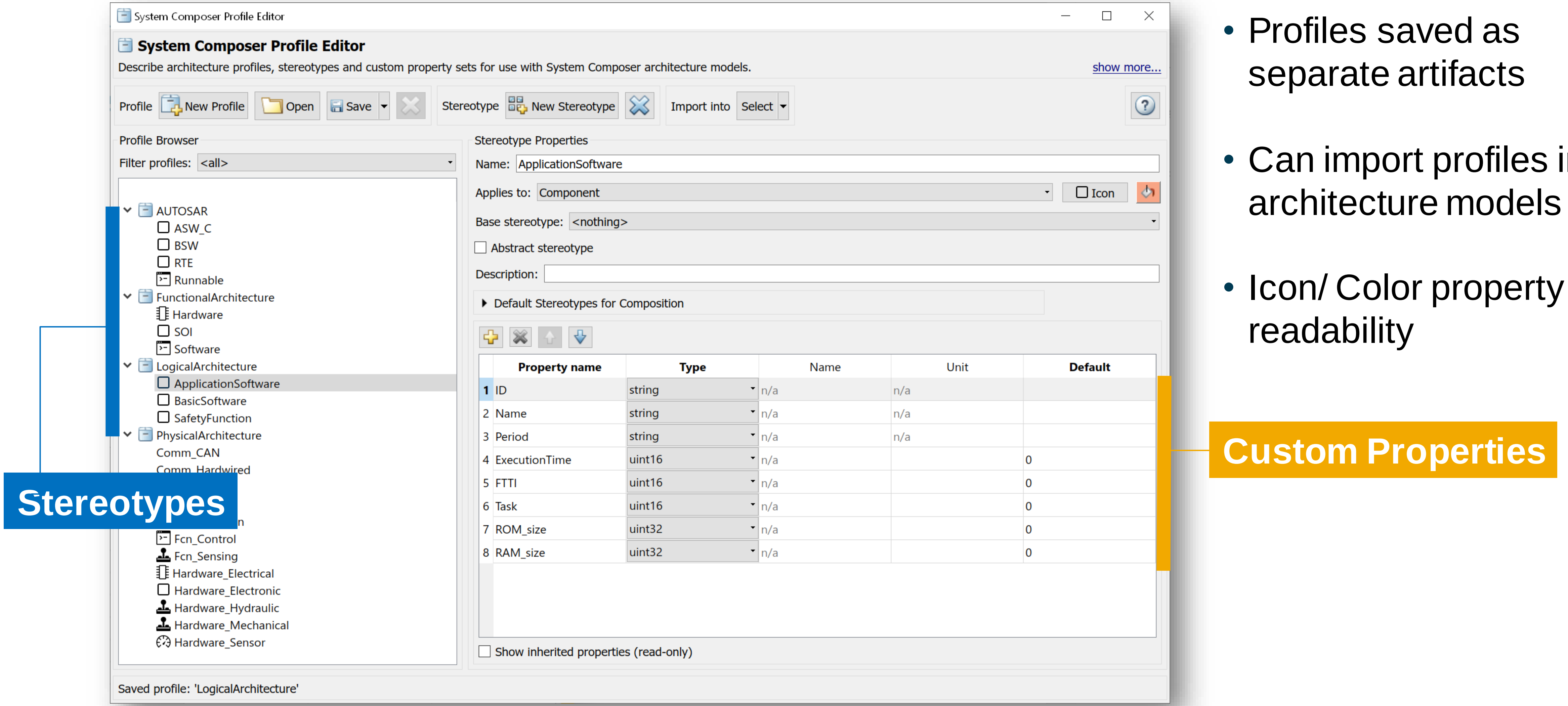
Vehicle

Vehicle Components



Analyze Architectural Design Models with Stereotypes

Define Stereotypes & Profiles using the Profile Editor


System Composer Profile Editor

System Composer Profile Editor

Describe architecture profiles, stereotypes and custom property sets for use with System Composer architecture models. show more...

Profile | New Profile | Open | Save | Stereotype | New Stereotype | Import into | Select

Profile Browser

Filter profiles: <all>

- AUTOSAR
  - ASW_C
  - BSW
  - RTE
  - Runnable
- FunctionalArchitecture
  - Hardware
  - SOI
  - Software
- LogicalArchitecture
  - ApplicationSoftware
  - BasicSoftware
  - SafetyFunction
- PhysicalArchitecture
  - Comm_CAN
  - Comm_Hardwired
  - Fcn_Control
  - Fcn_Sensing
  - Hardware_Electrical
  - Hardware_Electronic
  - Hardware_Hydraulic
  - Hardware_Mechanical
  - Hardware_Sensor

Stereotype Properties

Name: ApplicationSoftware

Applies to: Component

Base stereotype: <nothing>

Abstract stereotype

Description:

Default Stereotypes for Composition

	Property name	Type	Name	Unit	Default
1	ID	string	n/a	n/a	
2	Name	string	n/a	n/a	
3	Period	string	n/a	n/a	
4	ExecutionTime	uint16	n/a		0
5	FTTI	uint16	n/a		0
6	Task	uint16	n/a		0
7	ROM_size	uint32	n/a		0
8	RAM_size	uint32	n/a		0

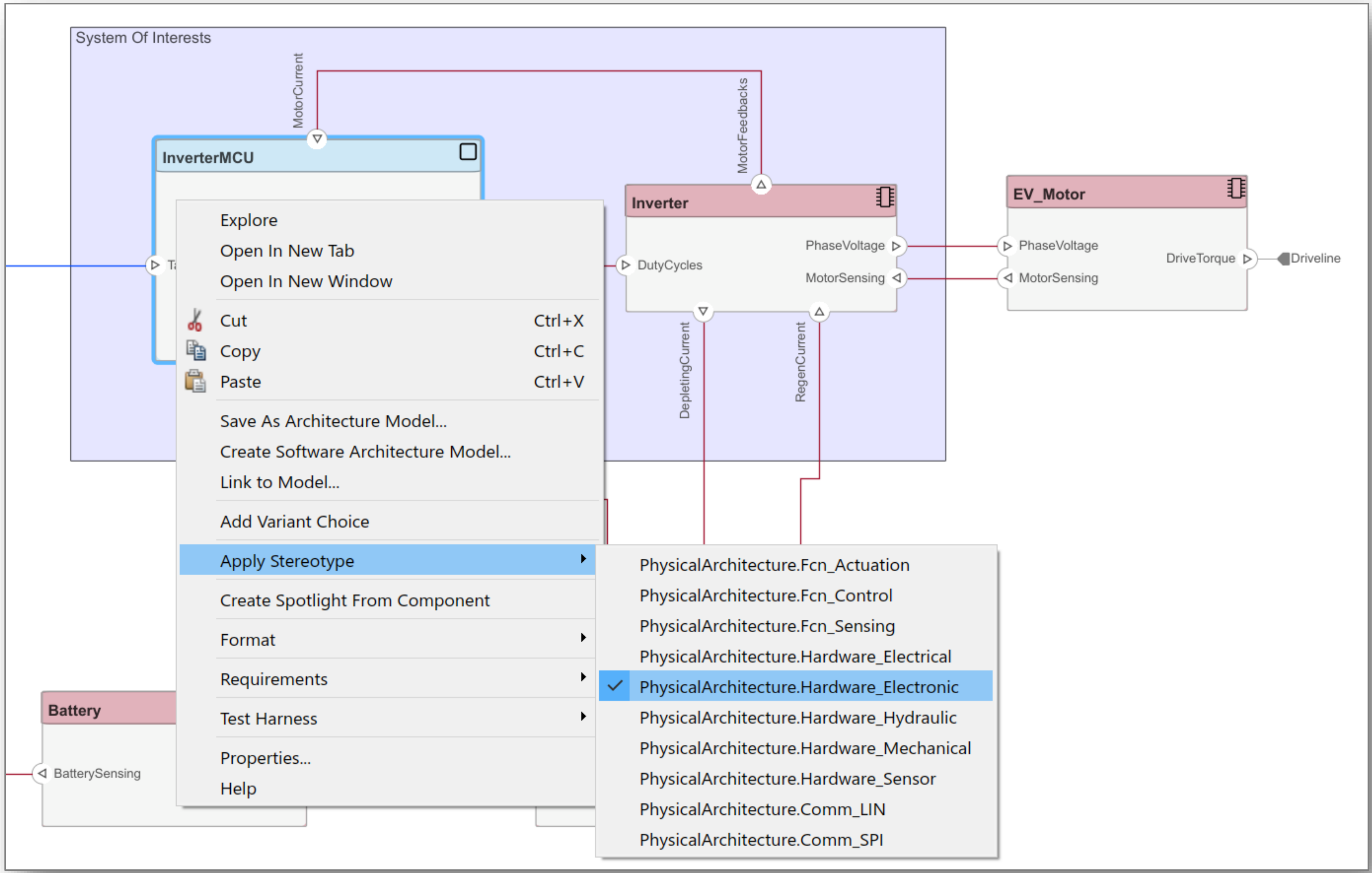
Show inherited properties (read-only)

Saved profile: 'LogicalArchitecture'

- Profiles saved as separate artifacts
- Can import profiles into architecture models
- Icon/ Color property for readability

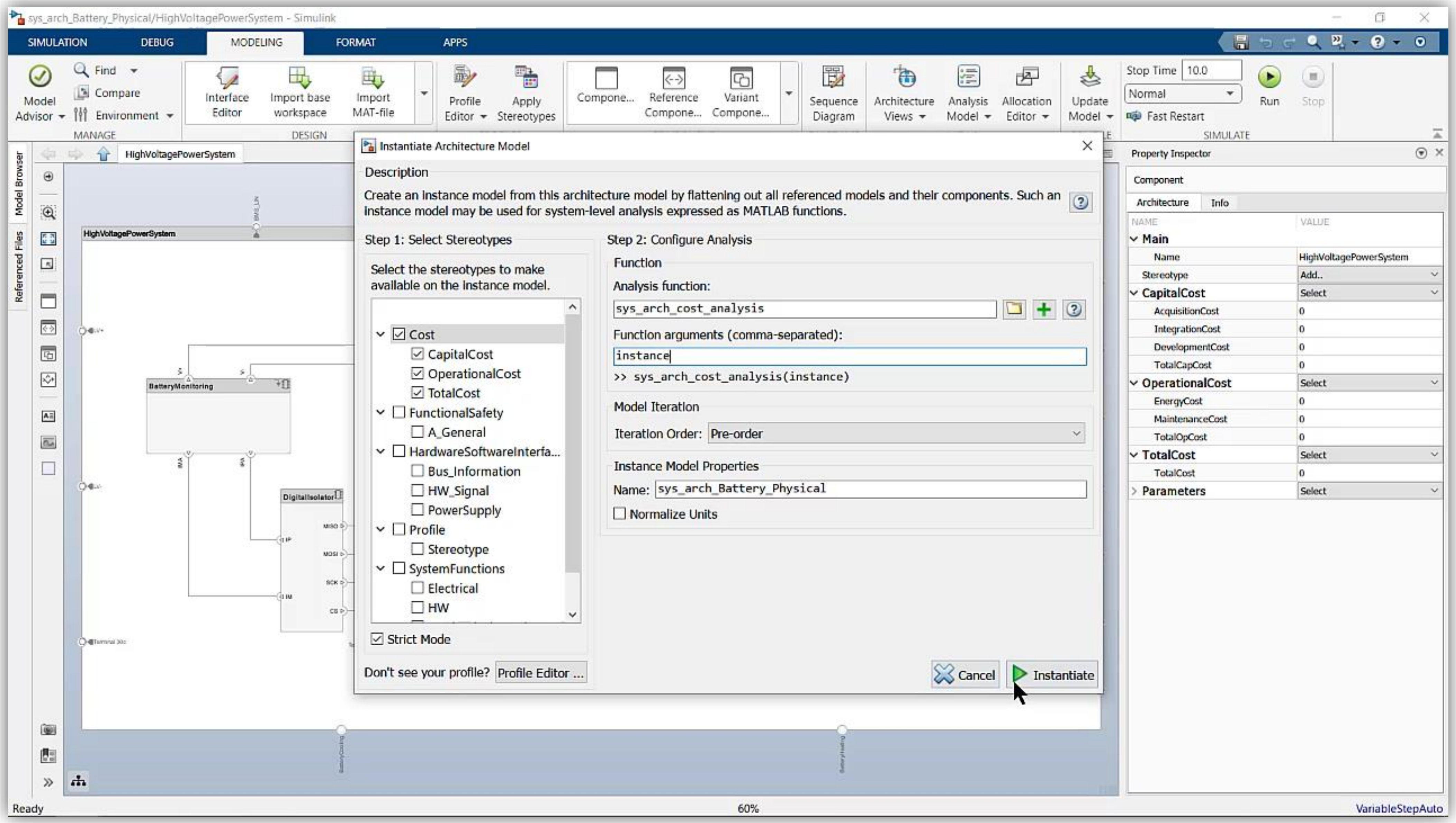
Analyze Architectural Design Models with Stereotypes

Apply Stereotype to Architecture


System Of Interests
MotorCurrent
MotorFeedbacks
InverterMCU
Inverter
DutyCycles
PhaseVoltage
MotorSensing
EV_Motor
PhaseVoltage
MotorSensing
DriveTorque
Driveline
DepletingCurrent
RegenCurrent
Battery
BatterySensing
Explore
Open In New Tab
Open In New Window
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Save As Architecture Model...
Create Software Architecture Model...
Link to Model...
Add Variant Choice
Apply Stereotype
Create Spotlight From Component
Format
Requirements
Test Harness
Properties...
Help
PhysicalArchitecture.Fcn_Actuation
PhysicalArchitecture.Fcn_Control
PhysicalArchitecture.Fcn_Sensing
PhysicalArchitecture.Hardware_Electrical
✓ PhysicalArchitecture.Hardware_Electronic
PhysicalArchitecture.Hardware_Hydraulic
PhysicalArchitecture.Hardware_Mechanical
PhysicalArchitecture.Hardware_Sensor
PhysicalArchitecture.Comm_LIN
PhysicalArchitecture.Comm_SPI

- Multiple stereotypes can be applied to a component

Analyze Architectural Design Models with System Composer



Instantiate Architecture Model

Description

Create an instance model from this architecture model by flattening out all referenced models and their components. Such an instance model may be used for system-level analysis expressed as MATLAB functions.

Step 1: Select Stereotypes

Select the stereotypes to make available on the instance model.

- [✓] Cost
  - [✓] CapitalCost
  - [✓] OperationalCost
  - [✓] TotalCost
- [ ] FunctionalSafety
  - [ ] A_General
- [ ] HardwareSoftwareInterfa...
  - [ ] Bus_Information
  - [ ] HW_Signal
  - [ ] PowerSupply
- [ ] Profile
  - [ ] Stereotype
- [ ] SystemFunctions
  - [ ] Electrical
  - [ ] HW

[✓] Strict Mode

Don't see your profile? Profile Editor ...

Step 2: Configure Analysis

Function

Analysis function: `sys_arch_cost_analysis`

Function arguments (comma-separated): `instance`

```
>> sys_arch_cost_analysis(instance)
```

Model Iteration

Iteration Order: Pre-order

Instance Model Properties

Name: `sys_arch_Battery_Physical`

[ ] Normalize Units

Cancel | Instantiate

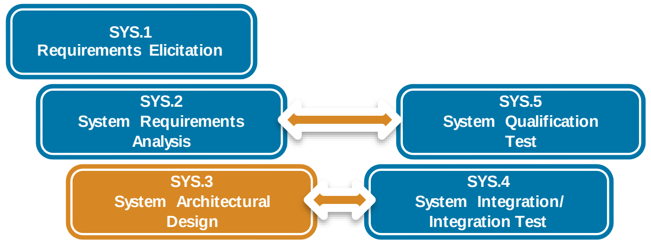
Property Inspector

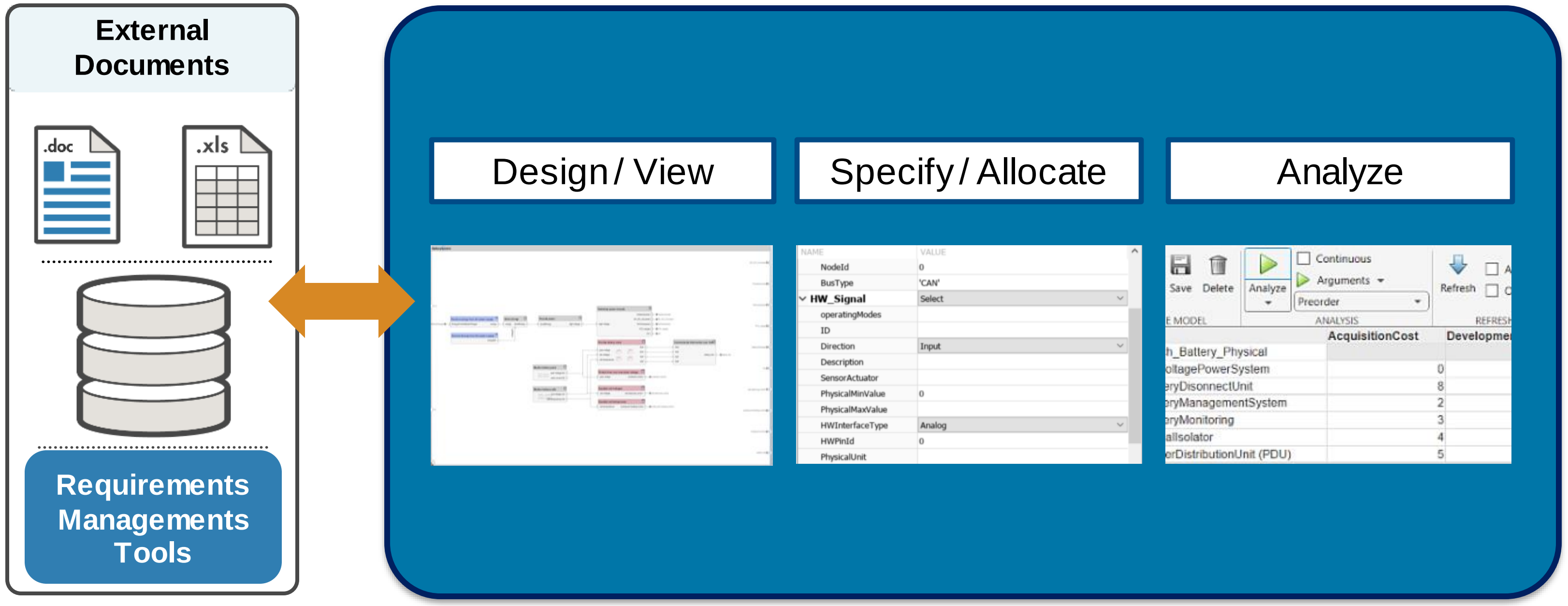
Component

Architecture | Info

NAME	VALUE
Main	
Name	HighVoltagePowerSystem
Stereotype	Add..
CapitalCost	Select
AcquisitionCost	0
IntegrationCost	0
DevelopmentCost	0
TotalCapCost	0
OperationalCost	Select
EnergyCost	0
MaintenanceCost	0
TotalOpCost	0
TotalCost	Select
TotalCost	0
Parameters	Select

SYS.3 System Architectural Design

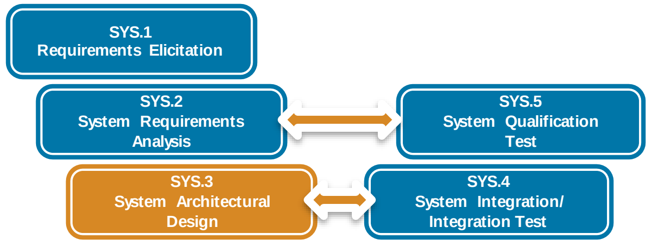

SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test

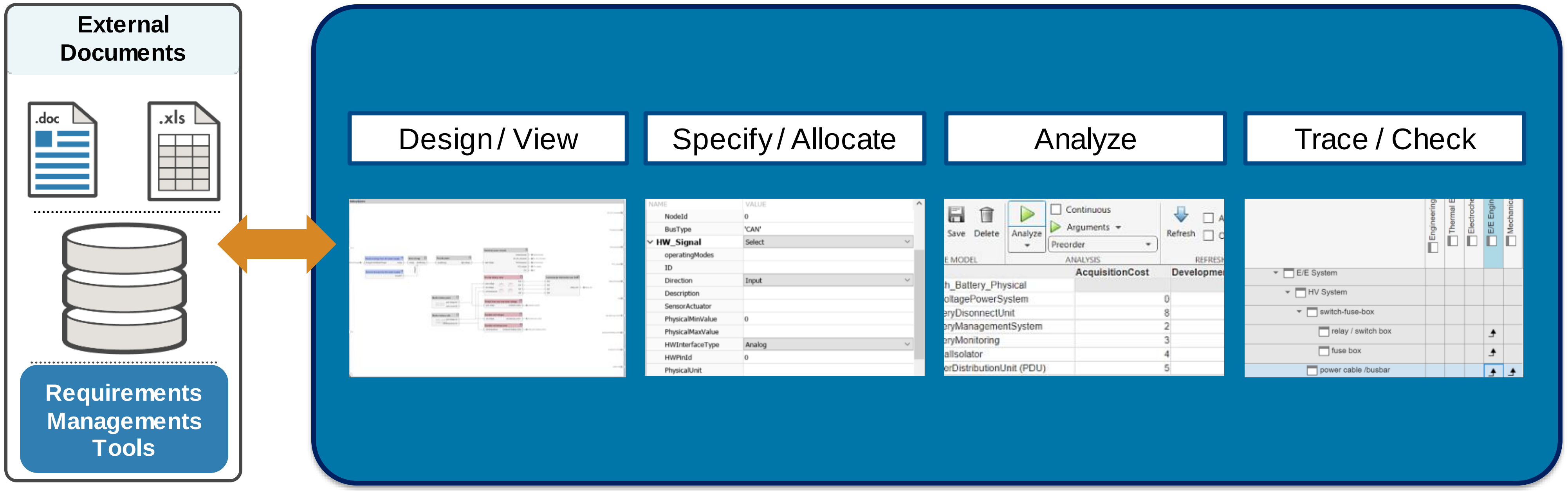


NAME	VALUE
NodeId	0
BusType	'CAN'
HW_Signal	Select
operatingModes	
ID	
Direction	Input
Description	
SensorActuator	
PhysicalMinValue	0
PhysicalMaxValue	
HWInterfaceType	Analog
HWPinId	0
PhysicalUnit	

	AcquisitionCost	Developme
h_Battery_Physical		
oltagePowerSystem	0	
eryDisonnectUnit	8	
eryManagementSystem	2	
eryMonitoring	3	
allsolator	4	
erDistributionUnit (PDU)	5	

SYS.3 System Architectural Design


SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test



NAME	VALUE
NodeId	0
BusType	'CAN'
HW_Signal	Select
operatingModes	
ID	
Direction	Input
Description	
SensorActuator	
PhysicalMinValue	0
PhysicalMaxValue	
HWInterfaceType	Analog
HWPinId	0
PhysicalUnit	

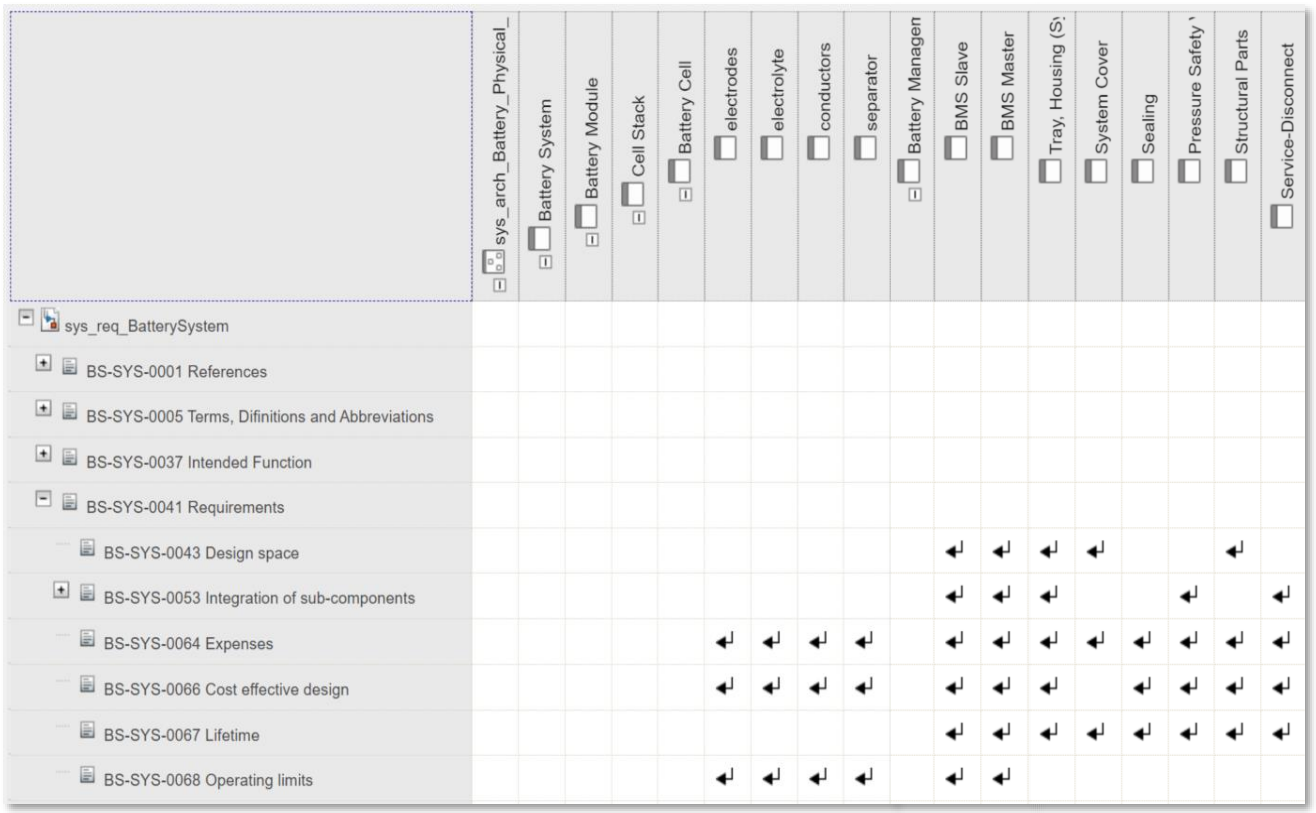
Save Delete Analyze Continuous Arguments Preorder Refresh
E MODEL ANALYSIS REFRESH

	AcquisitionCost	Developme
h_Battery_Physical		
ollagePowerSystem	0	
eryDisonnectUnit	8	
eryManagementSystem	2	
eryMonitoring	3	
allsolator	4	
erDistributionUnit (PDU)	5	

	Engineering	Thermal E	Electroche	E/E Engin	Mechanica
E/E System					
HV System					
switch-fuse-box					
relay / switch box				↑	
fuse box				↑	
power cable /busbar				↑	↑

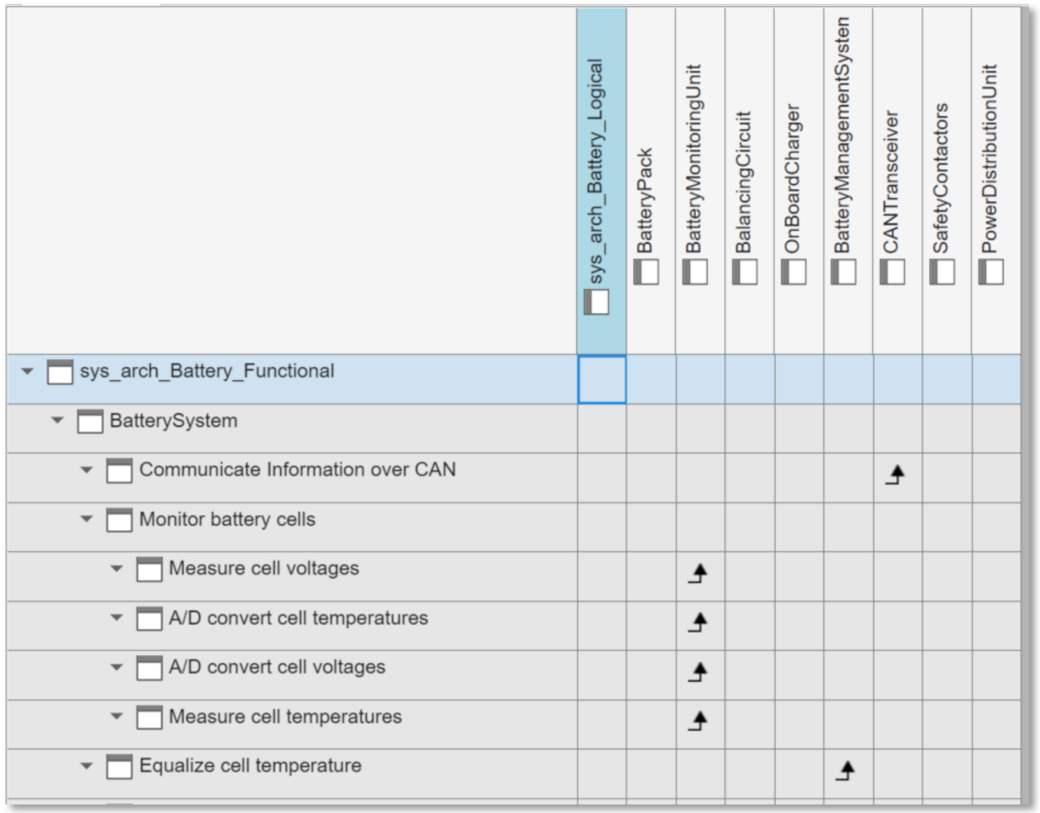
Ensure Consistency with Tool Support for Bidirectional Traceability

Requirements ↔ Architecture



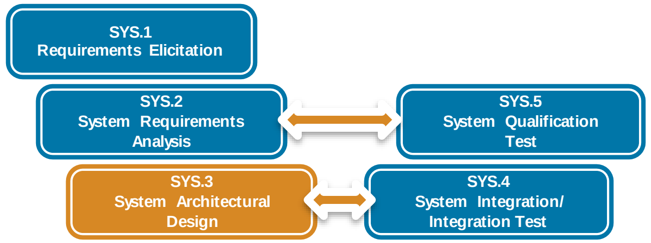
	sys_arch_Battery_Physical_	Battery System	Battery Module	Cell Stack	Battery Cell	electrodes	electrolyte	conductors	separator	Battery Managen	BMS Slave	BMS Master	Tray, Housing (S)	System Cover	Sealing	Pressure Safety	Structural Parts	Service-Disconnect
sys_req_BatterySystem																		
BS-SYS-0001 References																		
BS-SYS-0005 Terms, Difinitions and Abbreviations																		
BS-SYS-0037 Intended Function																		
BS-SYS-0041 Requirements																		
BS-SYS-0043 Design space											↵	↵	↵	↵			↵	
BS-SYS-0053 Integration of sub-components											↵	↵	↵			↵		↵
BS-SYS-0064 Expenses						↵	↵	↵	↵		↵	↵	↵	↵	↵	↵	↵	↵
BS-SYS-0066 Cost effective design						↵	↵	↵	↵		↵	↵	↵		↵	↵	↵	↵
BS-SYS-0067 Lifetime											↵	↵	↵	↵	↵	↵	↵	↵
BS-SYS-0068 Operating limits						↵	↵	↵	↵		↵	↵						

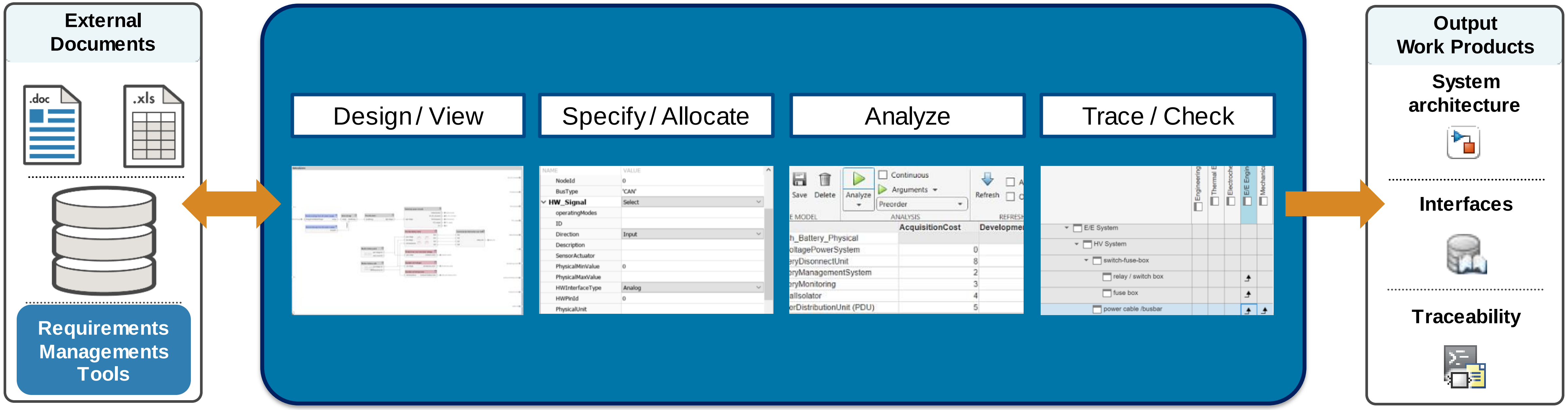
Architecture ↔ Architecture



	sys_arch_Battery_Logical	BatteryPack	BatteryMonitoringUnit	BalancingCircuit	OnBoardCharger	BatteryManagementSysten	CANTransceiver	SafetyContactors	PowerDistributionUnit
sys_arch_Battery_Functional									
BatterySystem									
Communicate Information over CAN							↑		
Monitor battery cells									
Measure cell voltages			↑						
A/D convert cell temperatures			↑						
A/D convert cell voltages			↑						
Measure cell temperatures			↑						
Equalize cell temperature						↑			

SYS.3 System Architectural Design


SYS.1
Requirements Elicitation
SYS.2
System Requirements Analysis
SYS.5
System Qualification Test
SYS.3
System Architectural Design
SYS.4
System Integration/ Integration Test

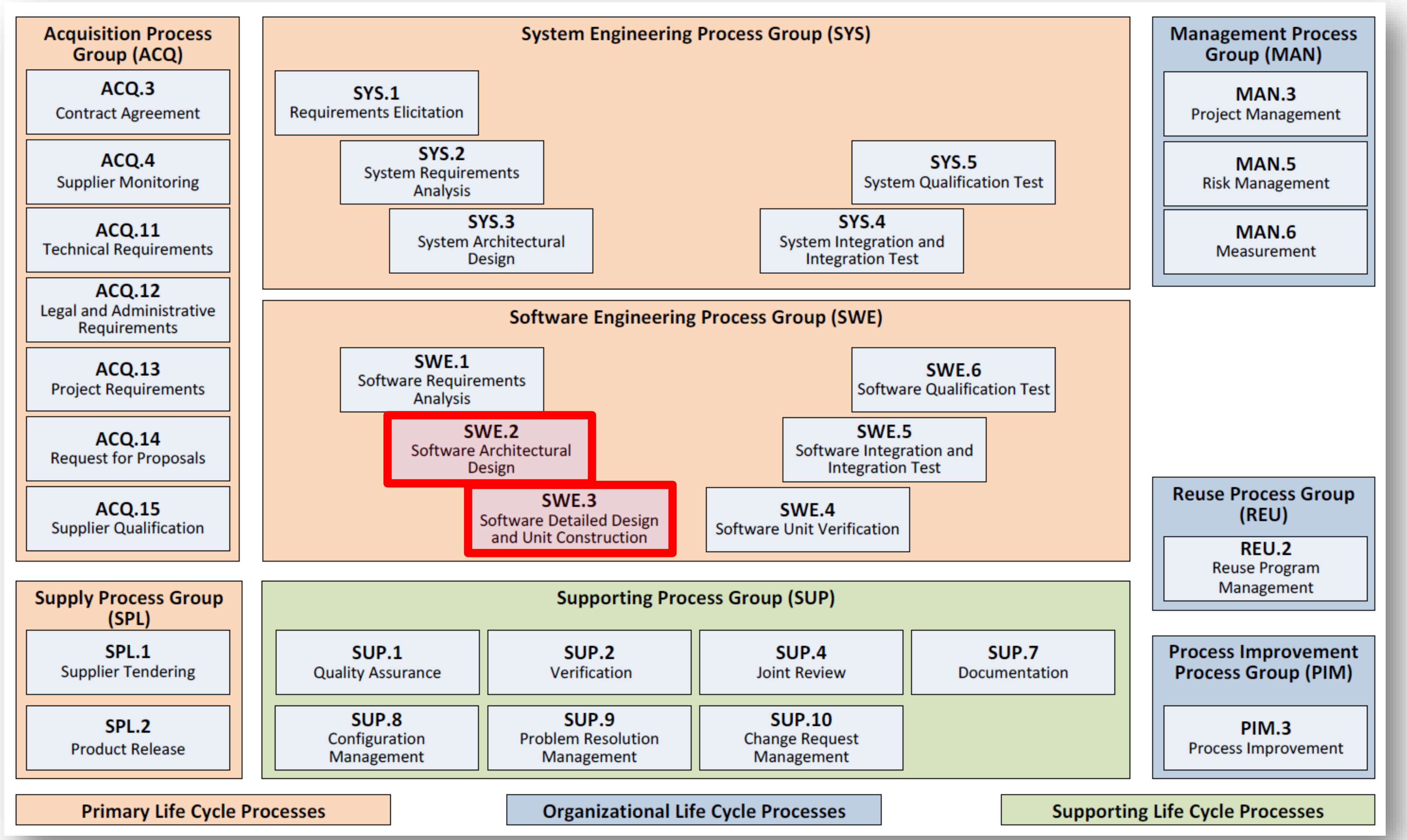


NAME	VALUE
NodeId	0
BusType	'CAN'
HW_Signal	Select
operatingModes	
ID	
Direction	Input
Description	
SensorActuator	
PhysicalMinValue	0
PhysicalMaxValue	
HWInterfaceType	Analog
HWPinId	0
PhysicalUnit	

	AcquisitionCost	Developme
h_Battery_Physical		
oltagePowerSystem	0	
eryDisonnectUnit	8	
eryManagementSystem	2	
eryMonitoring	3	
allsolator	4	
erDistributionUnit (PDU)	5	

	Engineering	Thermal E	Electroche	E/E Engin	Mechanica
E/E System					
HV System					
switch-fuse-box					
relay / switch box				↑	
fuse box				↑	
power cable /busbar				↑	↑

Automotive SPICE® – Reference Model



Acquisition Process Group (ACQ)
- **ACQ.3** Contract Agreement
- **ACQ.4** Supplier Monitoring
- **ACQ.11** Technical Requirements
- **ACQ.12** Legal and Administrative Requirements
- **ACQ.13** Project Requirements
- **ACQ.14** Request for Proposals
- **ACQ.15** Supplier Qualification

System Engineering Process Group (SYS)
- **SYS.1** Requirements Elicitation
- **SYS.2** System Requirements Analysis
- **SYS.3** System Architectural Design
- **SYS.4** System Integration and Integration Test
- **SYS.5** System Qualification Test

Software Engineering Process Group (SWE)
- **SWE.1** Software Requirements Analysis
- **SWE.2** Software Architectural Design
- **SWE.3** Software Detailed Design and Unit Construction
- **SWE.4** Software Unit Verification
- **SWE.5** Software Integration and Integration Test
- **SWE.6** Software Qualification Test

Management Process Group (MAN)
- **MAN.3** Project Management
- **MAN.5** Risk Management
- **MAN.6** Measurement

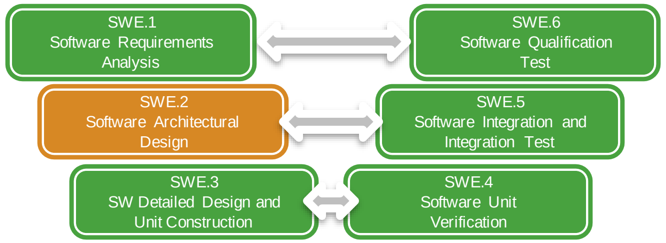
Reuse Process Group (REU)
- **REU.2** Reuse Program Management

Supply Process Group (SPL)
- **SPL.1** Supplier Tendering
- **SPL.2** Product Release

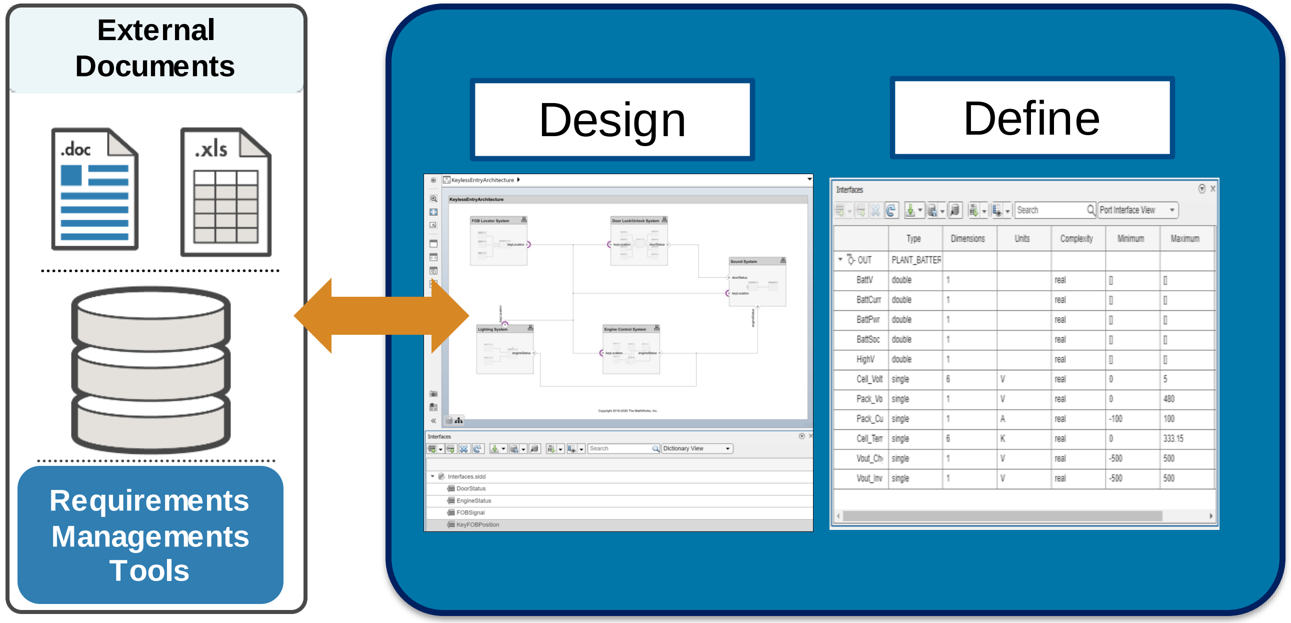
Supporting Process Group (SUP)
- **SUP.1** Quality Assurance
- **SUP.2** Verification
- **SUP.4** Joint Review
- **SUP.7** Documentation
- **SUP.8** Configuration Management
- **SUP.9** Problem Resolution Management
- **SUP.10** Change Request Management

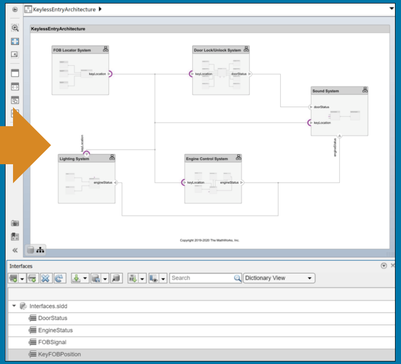
Process Improvement Process Group (PIM)
- **PIM.3** Process Improvement

Primary Life Cycle Processes | **Organizational Life Cycle Processes** | **Supporting Life Cycle Processes**

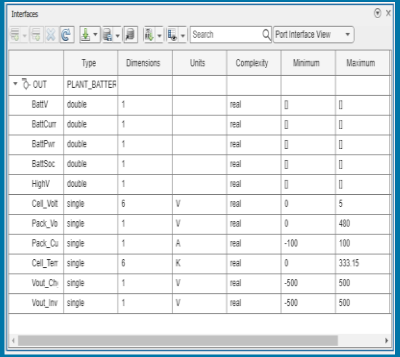

SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

SWE.2 Software Architectural Design



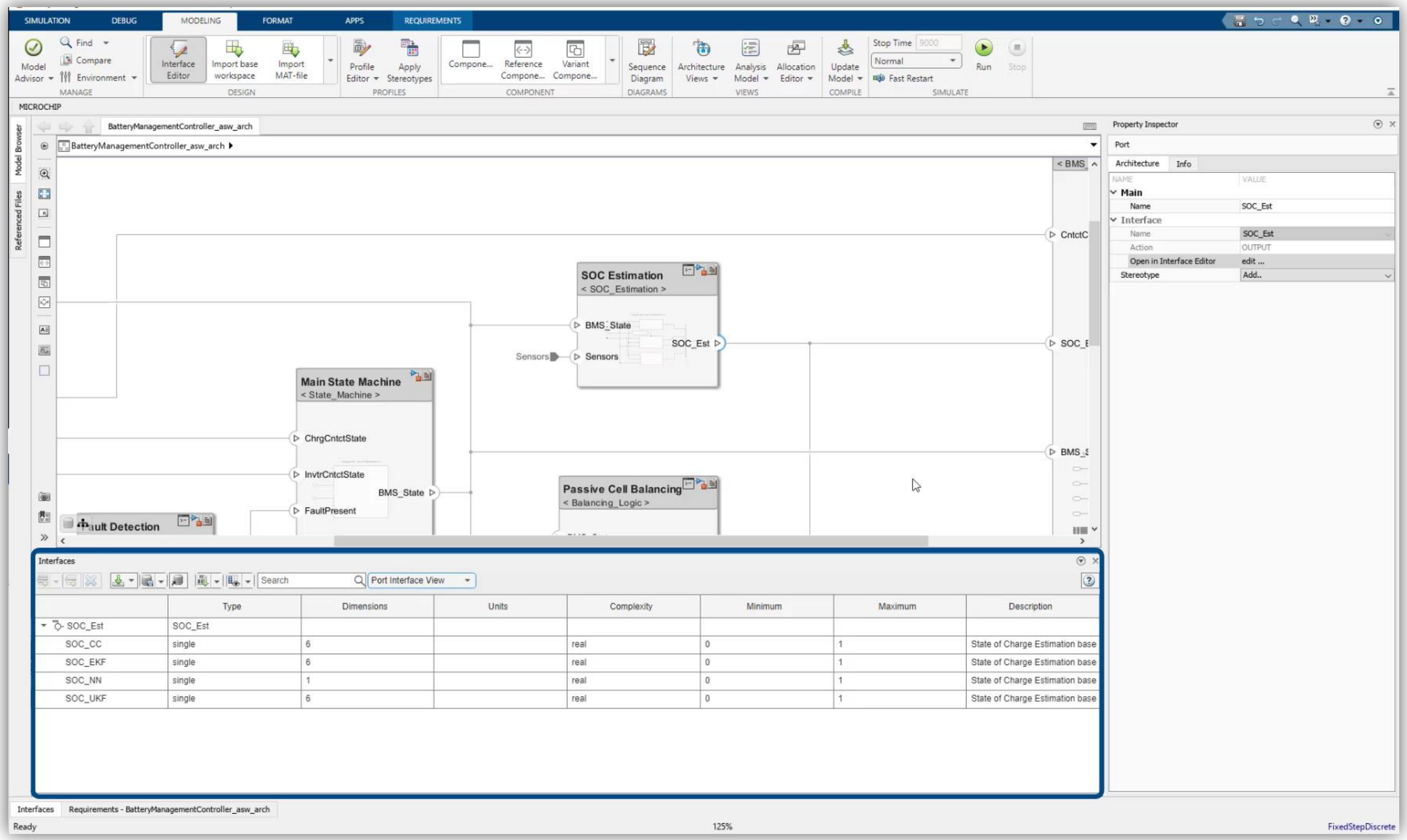


Interfaces.sldd
- DoorStatus
- EngineStatus
- FOBSignal
- KeyFOBPosition



	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Tem	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

Develop Software Architectural Design



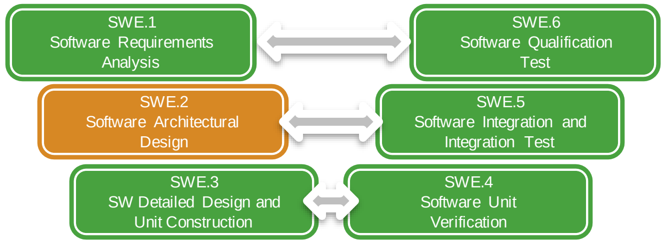
Property Inspector

Port

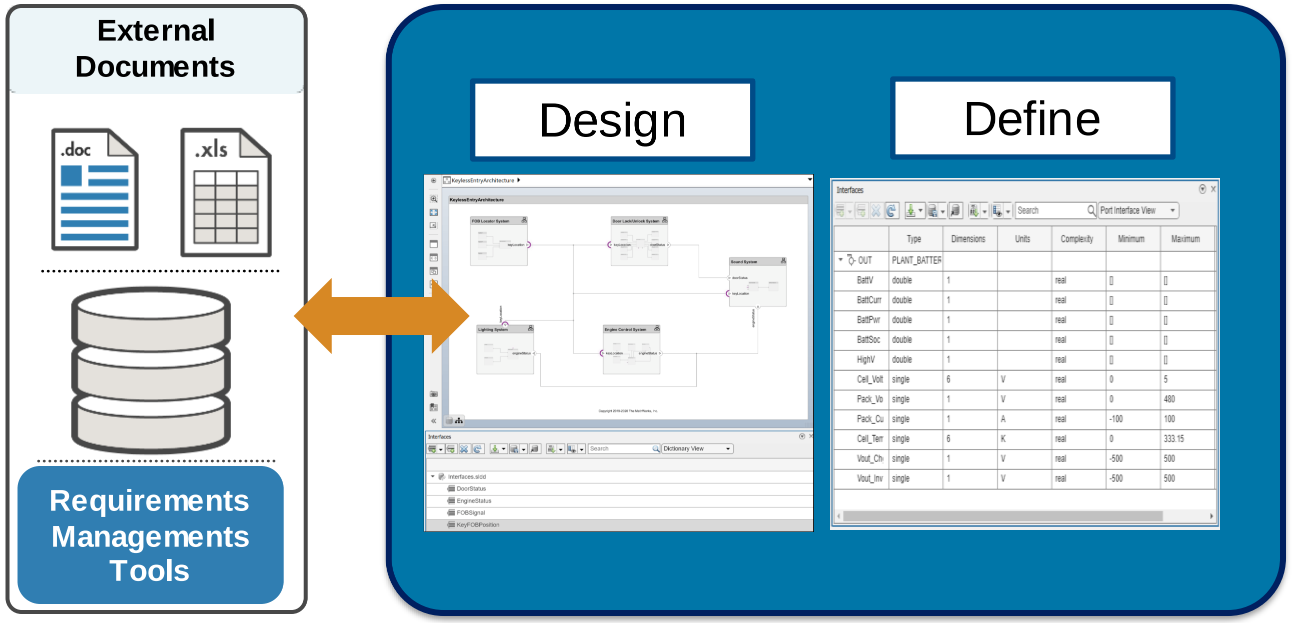
Architecture | Info

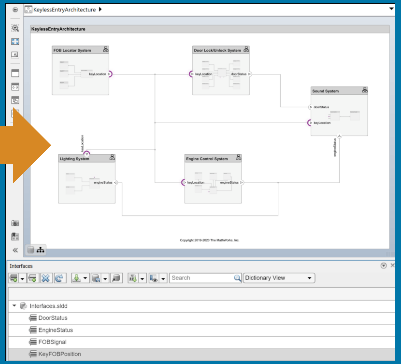
NAME	VALUE
Main	
Name	SOC_Est
Interface	
Name	SOC_Est
Action	OUTPUT
Open in Interface Editor	edit ...
Stereotype	Add..

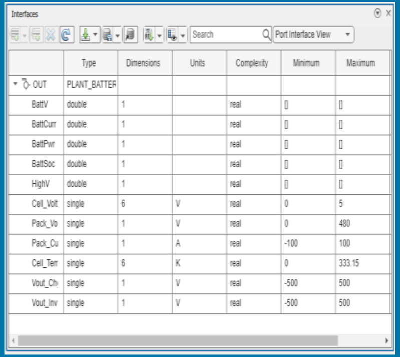
	Type	Dimensions	Units	Complexity	Minimum	Maximum	Description
SOC_Est	SOC_Est						
SOC_CC	single	6		real	0	1	State of Charge Estimation base
SOC_EKF	single	6		real	0	1	State of Charge Estimation base
SOC_NN	single	1		real	0	1	State of Charge Estimation base
SOC_UKF	single	6		real	0	1	State of Charge Estimation base


SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

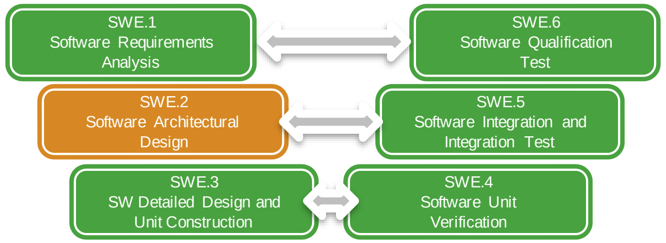
SWE.2 Software Architectural Design



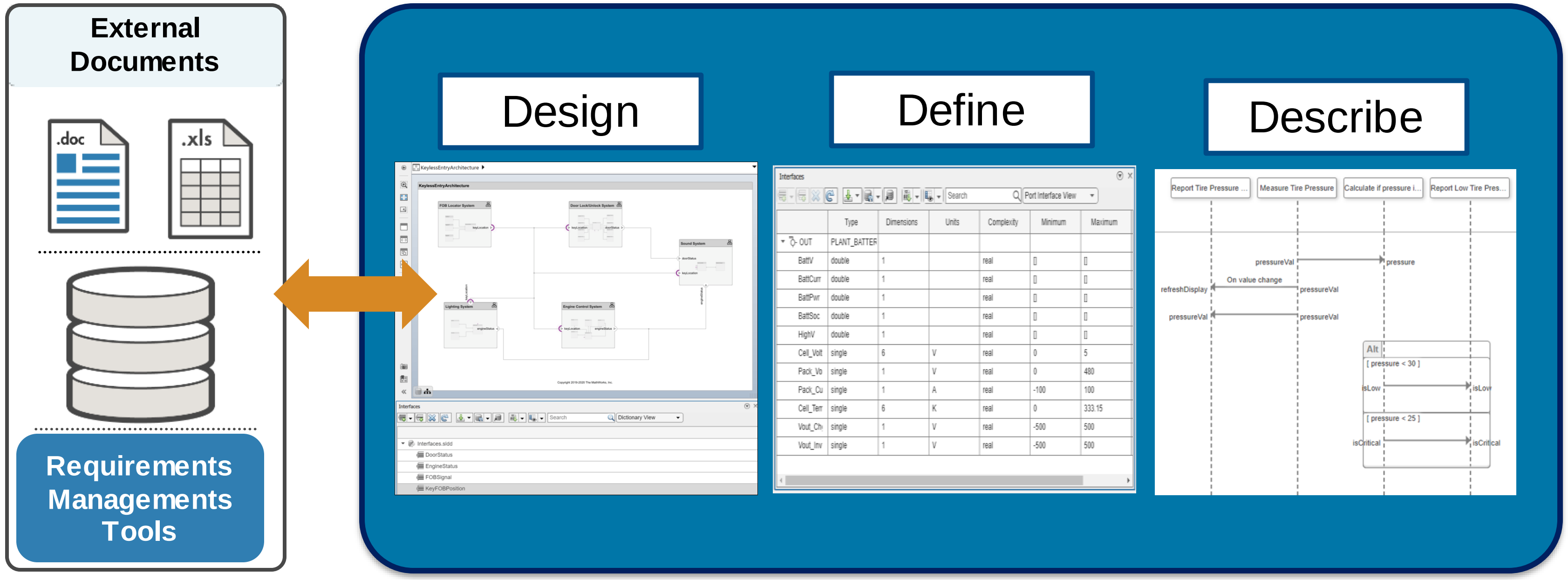




	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Terr	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500


SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

SWE.2 Software Architectural Design



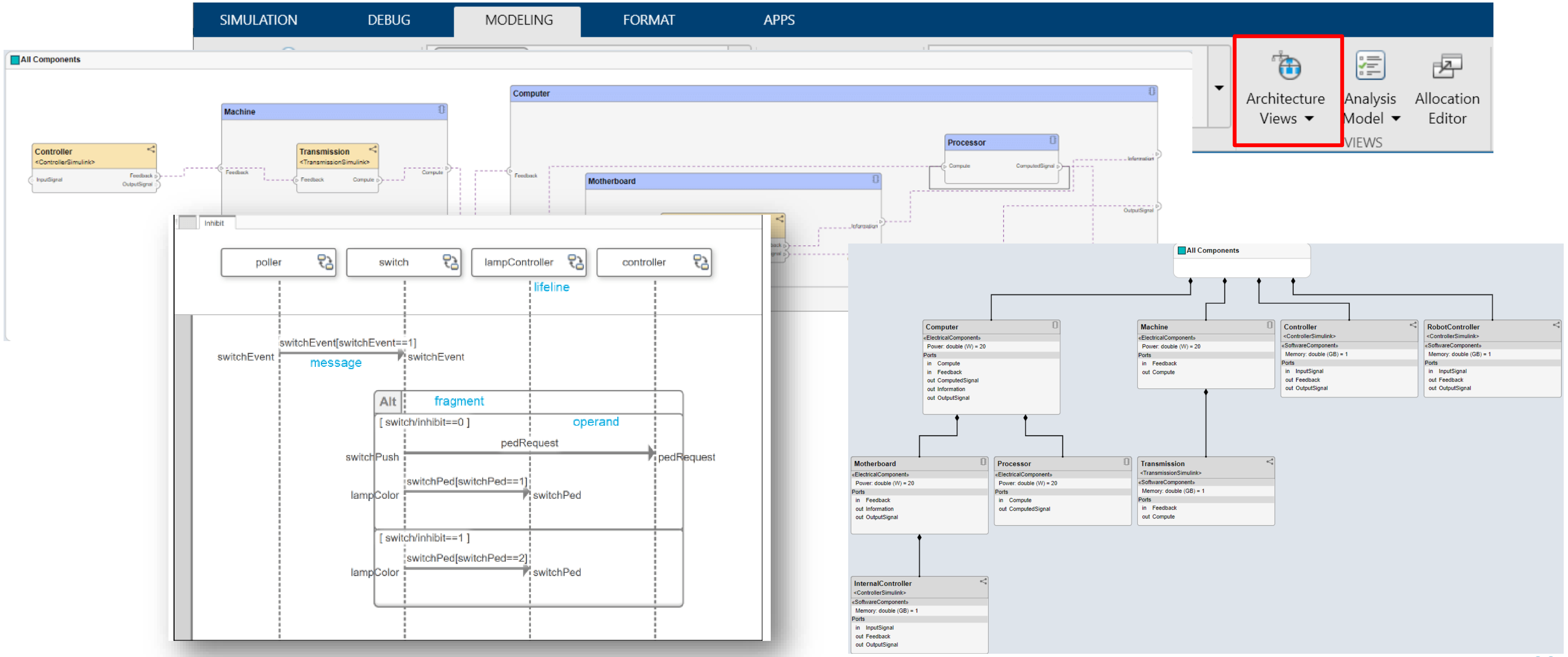
Interfaces

Search | Port Interface View

	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Ten	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

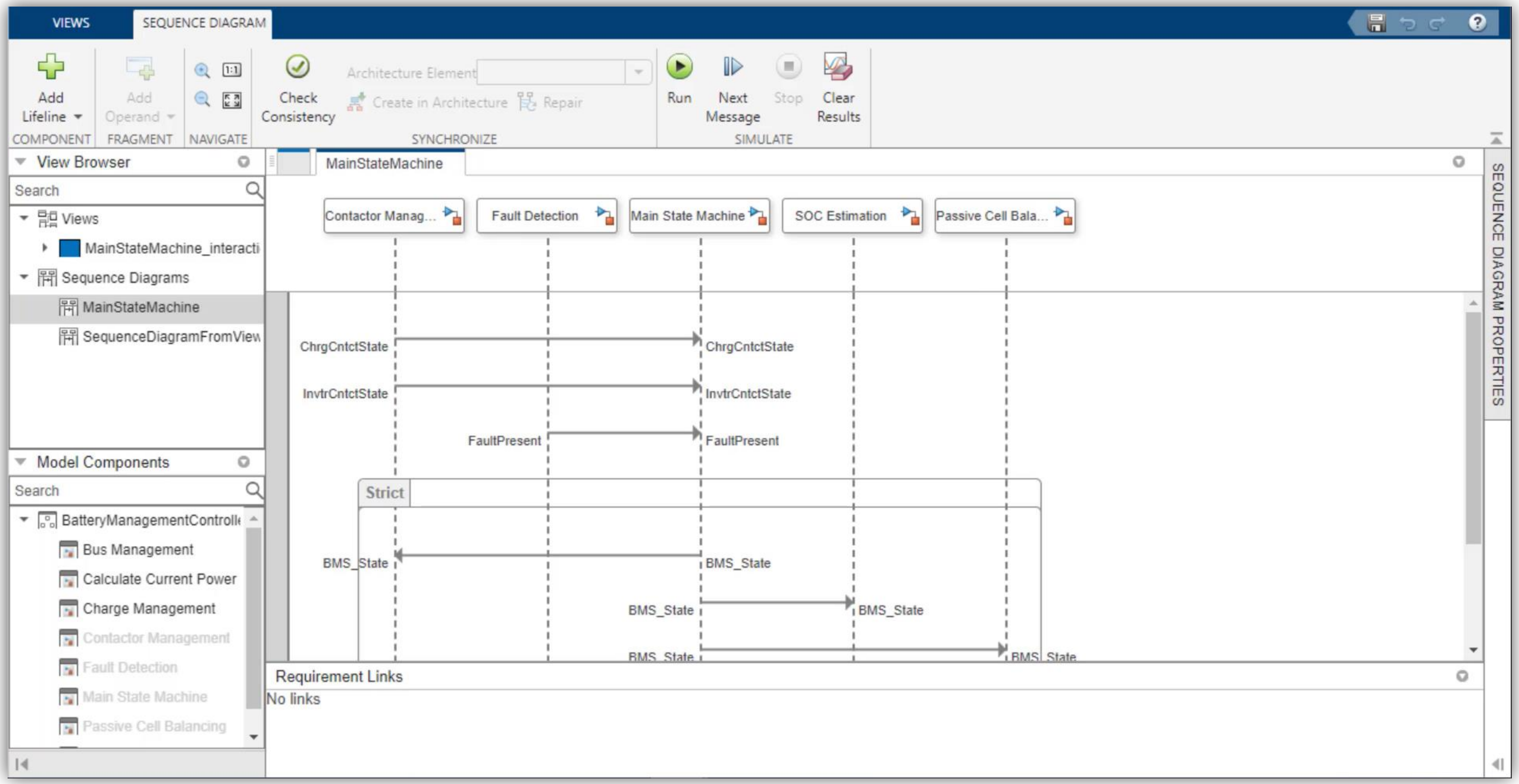
Describe Dynamic Behavior from Architecture Custom View

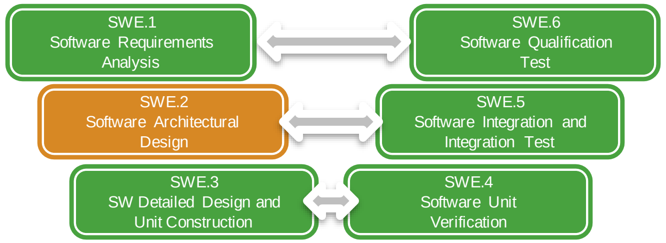
- Create a custom subset of components from architecture models by filtering model elements based on criteria such as stereotypes, properties, and requirement links



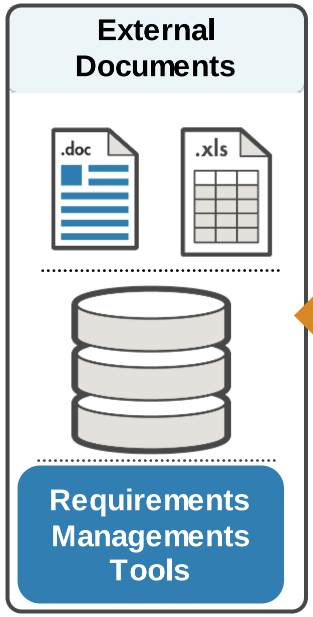
Describe System Behavior Using Sequence Diagrams

Describe Dynamic Behavior with Sequence Diagram

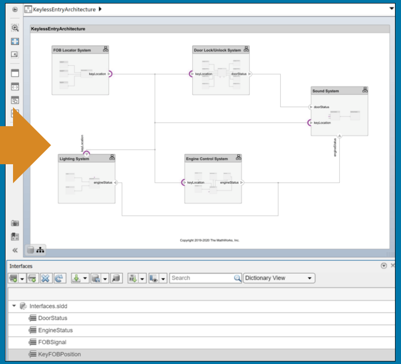



SWE.1 Software Requirements Analysis
SWE.6 Software Qualification Test
SWE.2 Software Architectural Design
SWE.5 Software Integration and Integration Test
SWE.3 SW Detailed Design and Unit Construction
SWE.4 Software Unit Verification

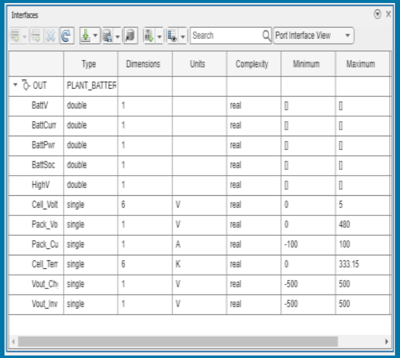
SWE.2 Software Architectural Design



Design

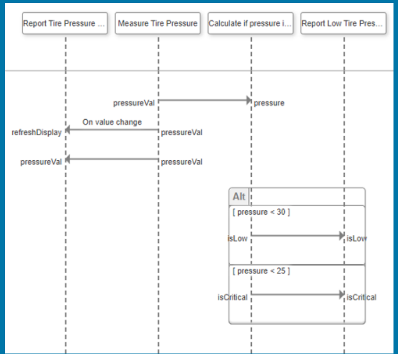


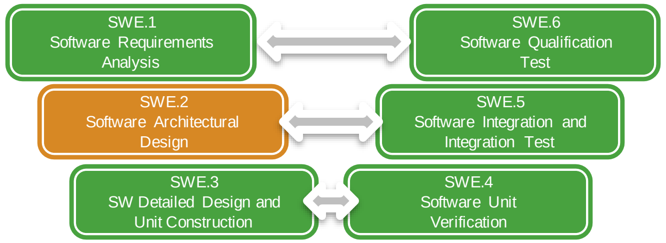
Define



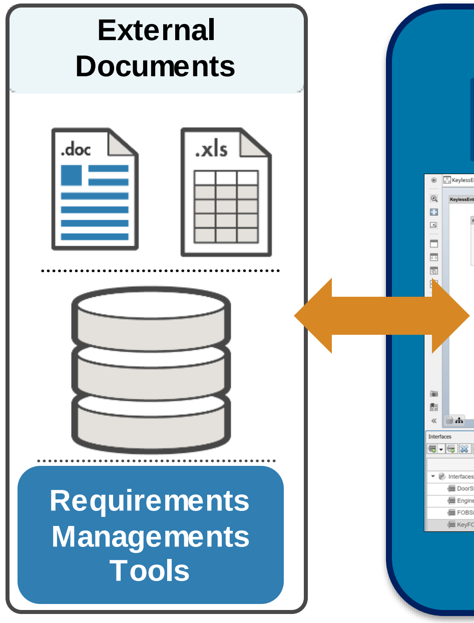
	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Tem	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

Describe

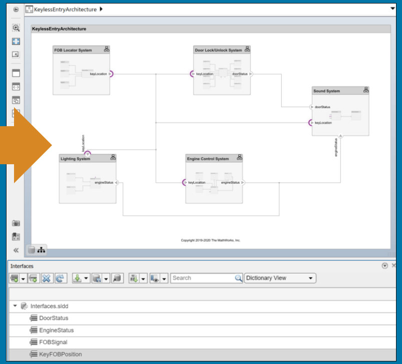



SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

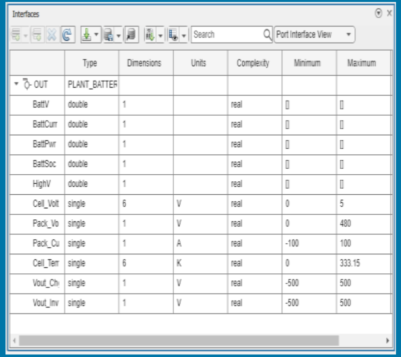
SWE.2 Software Architectural Design



Design



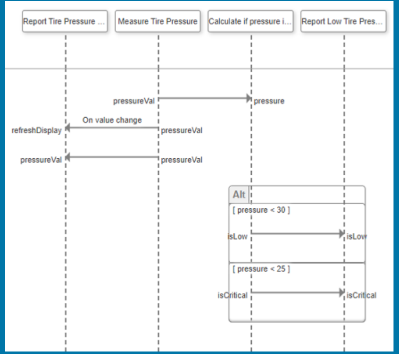
Define



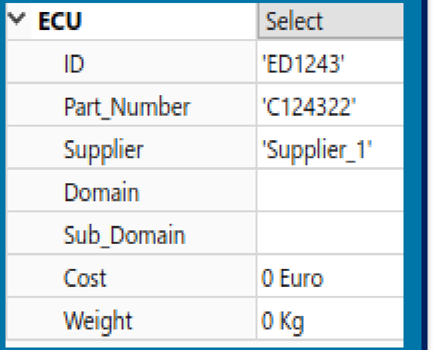
Interfaces

	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Ter	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

Describe

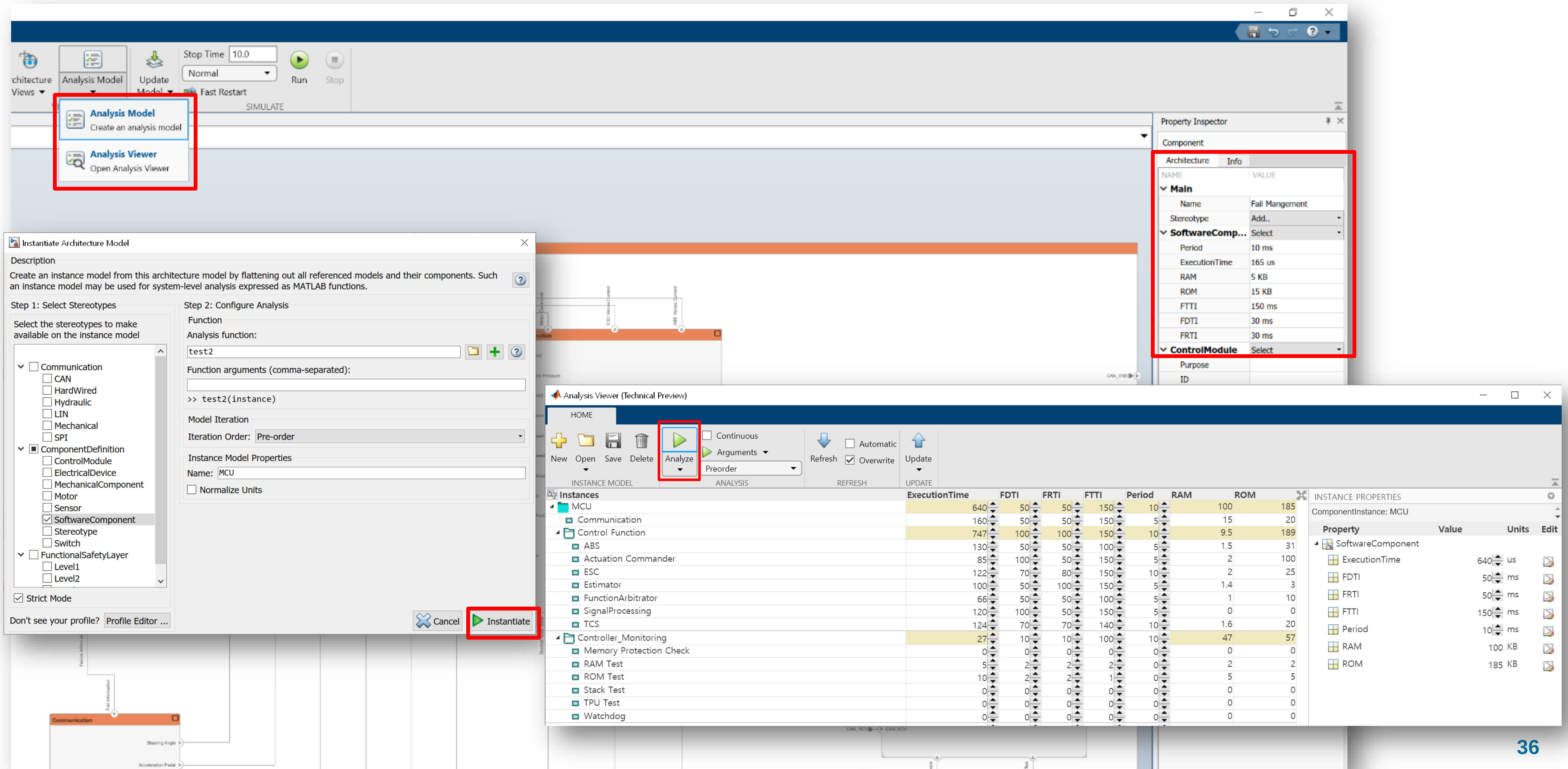


Evaluate



ECU	Select
ID	'ED1243'
Part_Number	'C124322'
Supplier	'Supplier_1'
Domain	
Sub_Domain	
Cost	0 Euro
Weight	0 Kg

Software Architecture Analysis using Stereotypes



Property Inspector

Component

Architecture | Info

NAME	VALUE
Main	
Name	Fail Mangement
Stereotype	Add..
SoftwareComp...	Select
Period	10 ms
ExecutionTime	165 us
RAM	5 KB
ROM	15 KB
FTTI	150 ms
FDTI	30 ms
FRTI	30 ms
ControlModule	Select
Purpose	
ID	

Instantiate Architecture Model

Description

Create an instance model from this architecture model by flattening out all referenced models and their components. Such an instance model may be used for system-level analysis expressed as MATLAB functions.

Step 1: Select Stereotypes

Select the stereotypes to make available on the instance model

- ☐ Communication
  - ☐ CAN
  - ☐ HardWired
  - ☐ Hydraulic
  - ☐ LIN
  - ☐ Mechanical
  - ☐ SPI
- ▣ ComponentDefinition
  - ☐ ControlModule
  - ☐ ElectricalDevice
  - ☐ MechanicalComponent
  - ☐ Motor
  - ☐ Sensor
  - ☑ SoftwareComponent
  - ☐ Stereotype
  - ☐ Switch
- ☐ FunctionalSafetyLayer
  - ☐ Level1
  - ☐ Level2

☑ Strict Mode

Don't see your profile? Profile Editor ...

Step 2: Configure Analysis

Function

Analysis function: test2

Function arguments (comma-separated):

```
>> test2(instance)
```

Model Iteration

Iteration Order: Pre-order

Instance Model Properties

Name: MCU

☐ Normalize Units

Cancel | Instantiate

Analysis Viewer (Technical Preview)

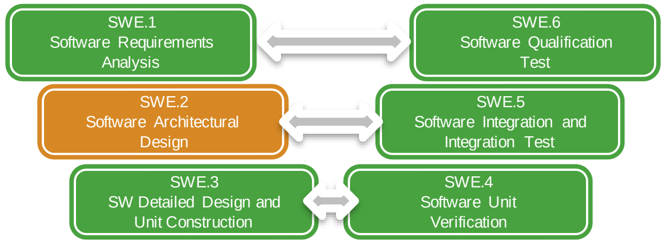
Instances	ExecutionTime	FDTI	FRTI	FTTI	Period	RAM	ROM
MCU	640	50	50	150	10	100	185
Communication	160	50	50	150	5	15	20
Control Function	747	100	100	150	10	9.5	189
ABS	130	50	50	100	5	1.5	31
Actuation Commander	85	100	50	150	5	2	100
ESC	122	70	80	150	10	2	25
Estimator	100	50	100	150	5	1.4	3
FunctionArbitrator	66	50	50	100	5	1	10
SignalProcessing	120	100	50	150	5	0	0
TCS	124	70	70	140	10	1.6	20
Controller_Monitoring	27	10	10	100	10	47	57
Memory Protection Check	0	0	0	0	0	0	0
RAM Test	5	2	2	2	0	2	2
ROM Test	10	2	2	1	0	5	5
Stack Test	0	0	0	0	0	0	0
TPU Test	0	0	0	0	0	0	0
Watchdog	0	0	0	0	0	0	0

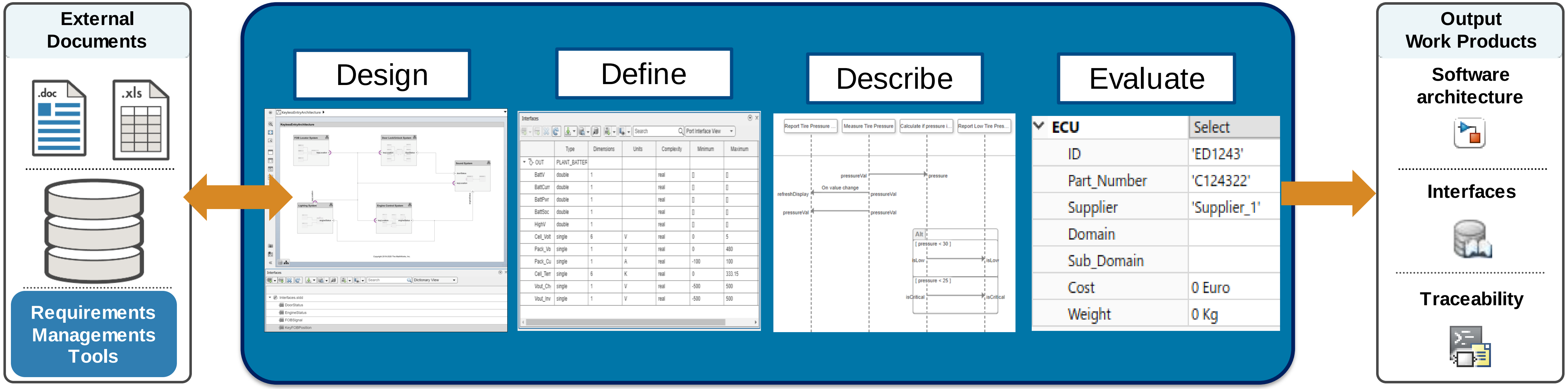
INSTANCE PROPERTIES

ComponentInstance: MCU

Property	Value	Units	Edit
SoftwareComponent			
ExecutionTime	640	us	
FDTI	50	ms	
FRTI	50	ms	
FTTI	150	ms	
Period	10	ms	
RAM	100	KB	
ROM	185	KB	

SWE.2 Software Architectural Design


SWE.1 Software Requirements Analysis
SWE.6 Software Qualification Test
SWE.2 Software Architectural Design
SWE.5 Software Integration and Integration Test
SWE.3 SW Detailed Design and Unit Construction
SWE.4 Software Unit Verification

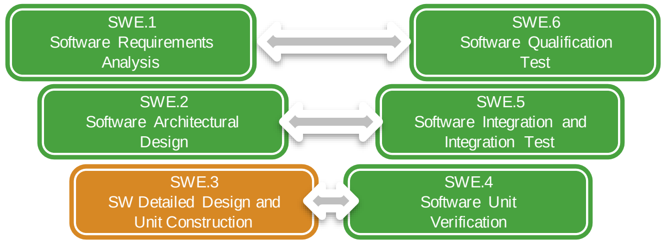


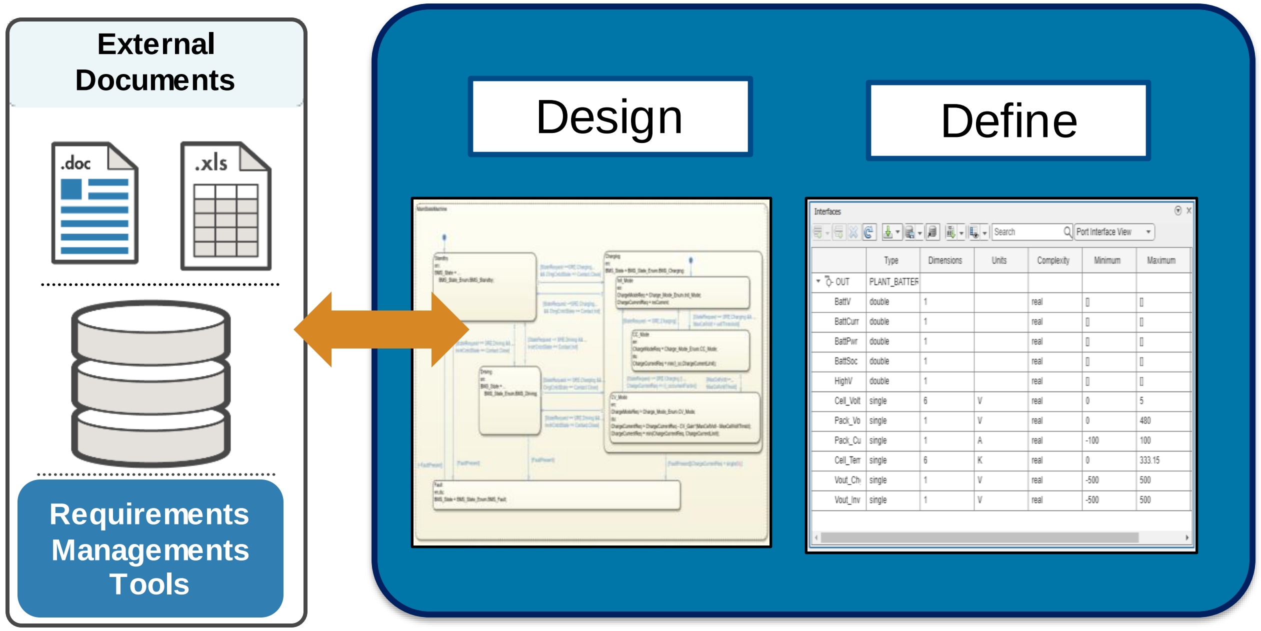
Interfaces — Port Interface View

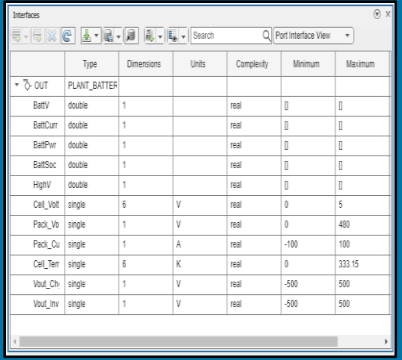
	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Ter	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

ECU	Select
ID	'ED1243'
Part_Number	'C124322'
Supplier	'Supplier_1'
Domain	
Sub_Domain	
Cost	0 Euro
Weight	0 Kg

SWE.3 SW Detailed Design and Unit Construction


SWE.1 Software Requirements Analysis
SWE.6 Software Qualification Test
SWE.2 Software Architectural Design
SWE.5 Software Integration and Integration Test
SWE.3 SW Detailed Design and Unit Construction
SWE.4 Software Unit Verification



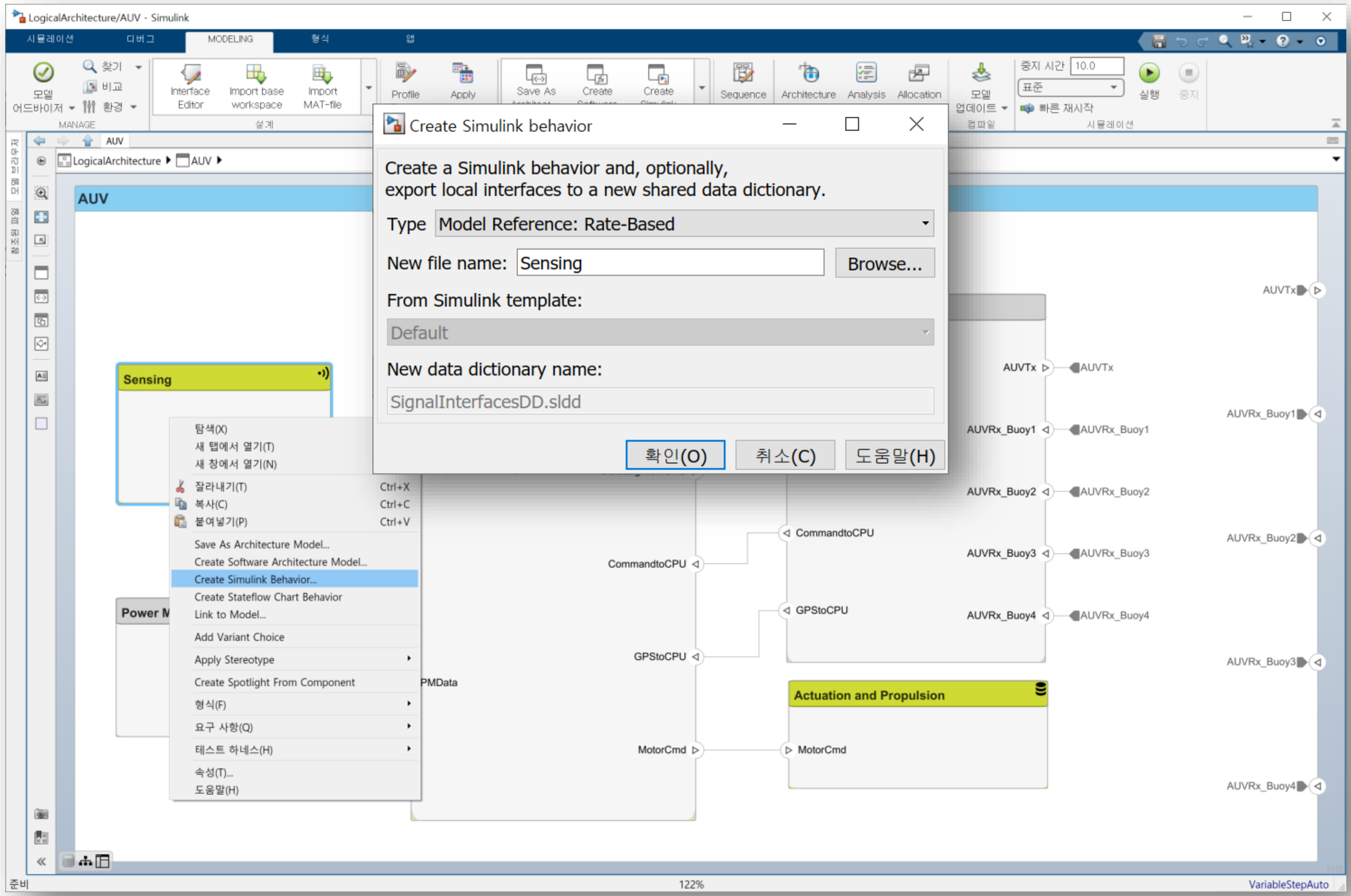

Interfaces

Search | Port Interface View

	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Ten	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

Develop Software Unit and Detailed Design

- Detailed design from architecture



Create Simulink behavior

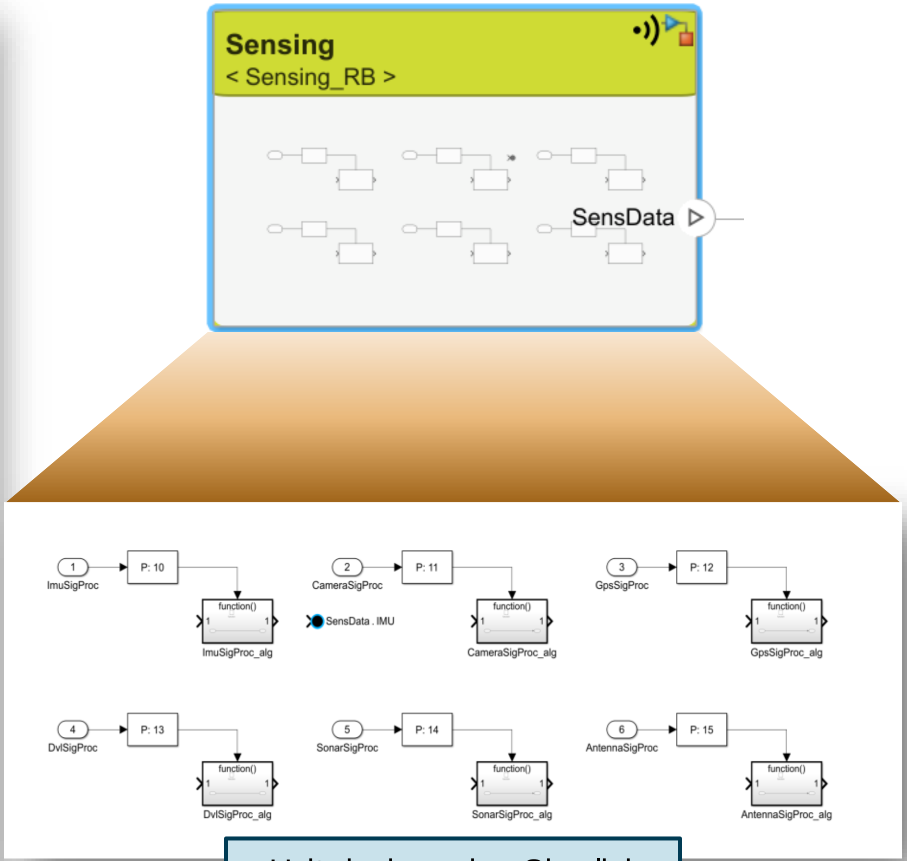
Create a Simulink behavior and, optionally, export local interfaces to a new shared data dictionary.

Type: Model Reference: Rate-Based

New file name: Sensing

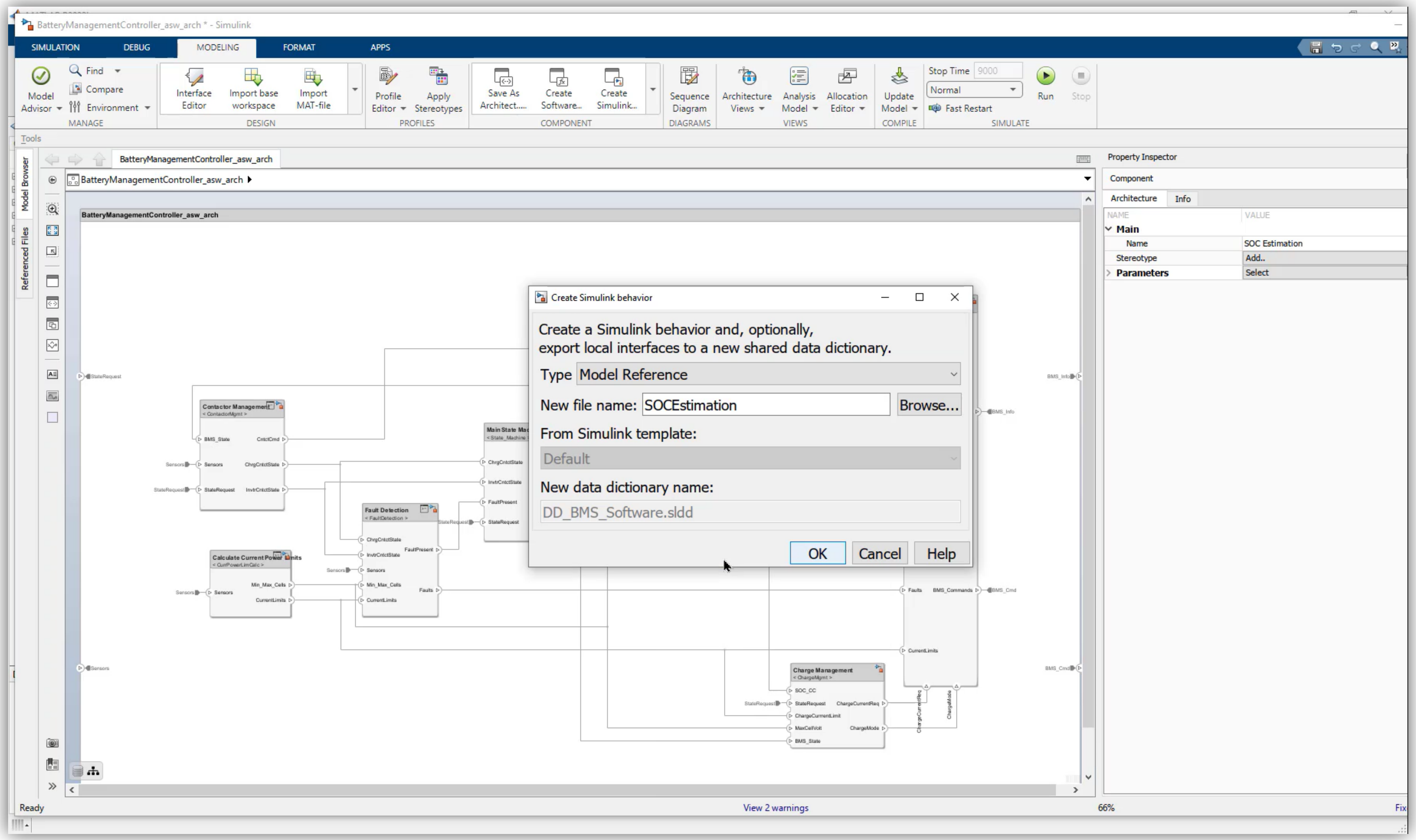
From Simulink template: Default

New data dictionary name: SignalInterfacesDD.sldd



Unit design using Simulink

Develop Software Unit and Detailed Design



Create Simulink behavior

Create a Simulink behavior and, optionally, export local interfaces to a new shared data dictionary.

Type: Model Reference

New file name: SOCEstimation

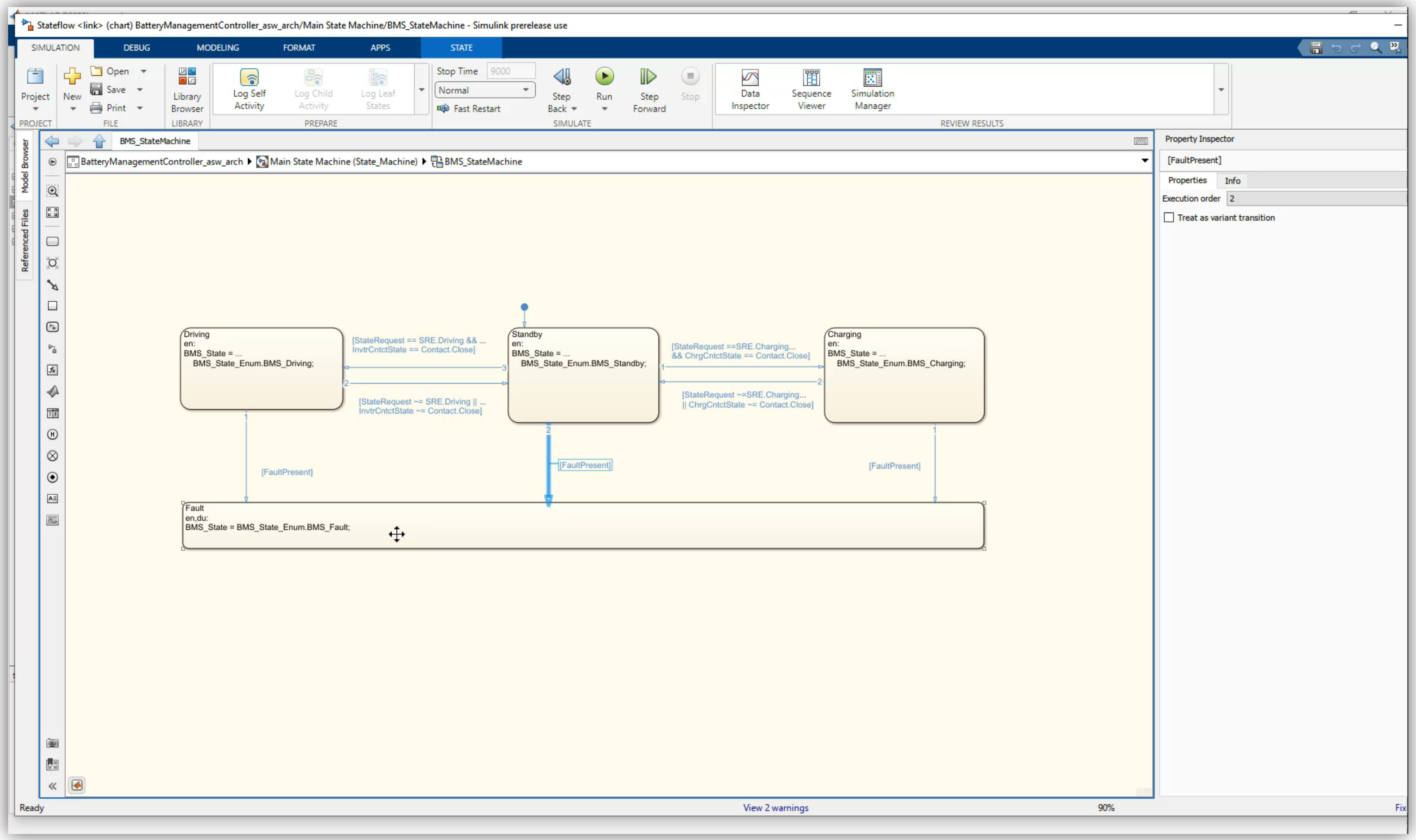
From Simulink template: Default

New data dictionary name: DD_BMS_Software.sldd

Property Inspector — Component — Architecture

NAME	VALUE
Main	
Name	SOC Estimation
Stereotype	Add..
Parameters	Select

Develop Software Unit and Detailed Design



Execution order 2
Treat as variant transition

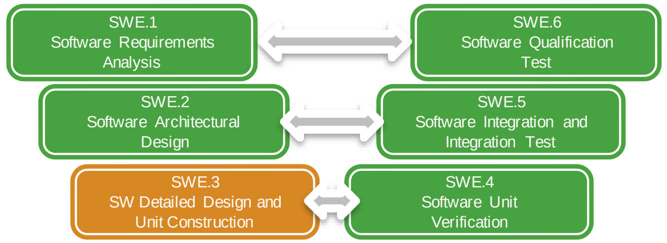
```
Driving
en:
BMS_State = ...
    BMS_State_Enum.BMS_Driving;
```

```
Standby
en:
BMS_State = ...
    BMS_State_Enum.BMS_Standby;
```

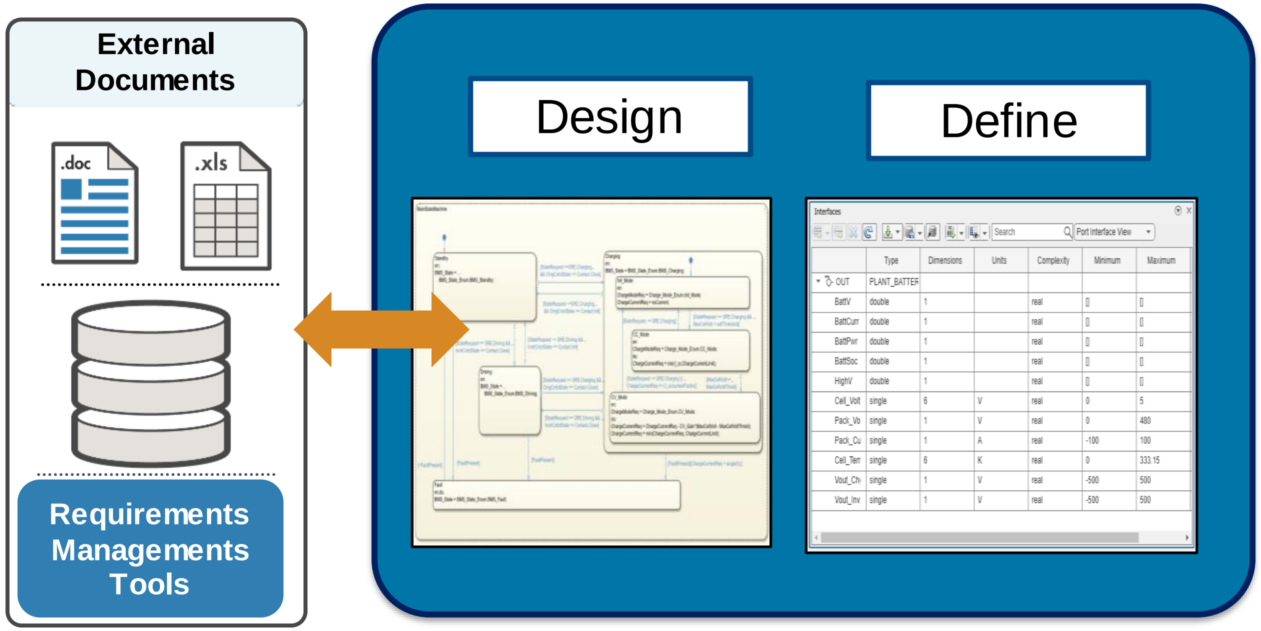
```
Charging
en:
BMS_State = ...
    BMS_State_Enum.BMS_Charging;
```

```
Fault
en,du:
BMS_State = BMS_State_Enum.BMS_Fault;
```

```
[StateRequest == SRE.Driving && ...
InvtrCntctState == Contact.Close]

[StateRequest ~= SRE.Driving || ...
InvtrCntctState ~= Contact.Close]

[StateRequest ==SRE.Charging...
&& ChrgCntctState == Contact.Close]

[StateRequest ~=SRE.Charging...
|| ChrgCntctState ~= Contact.Close]

[FaultPresent]

[FaultPresent]

[FaultPresent]
```

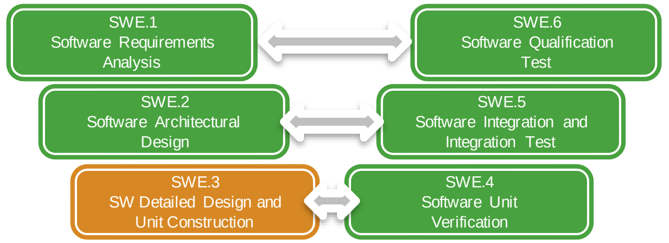

SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

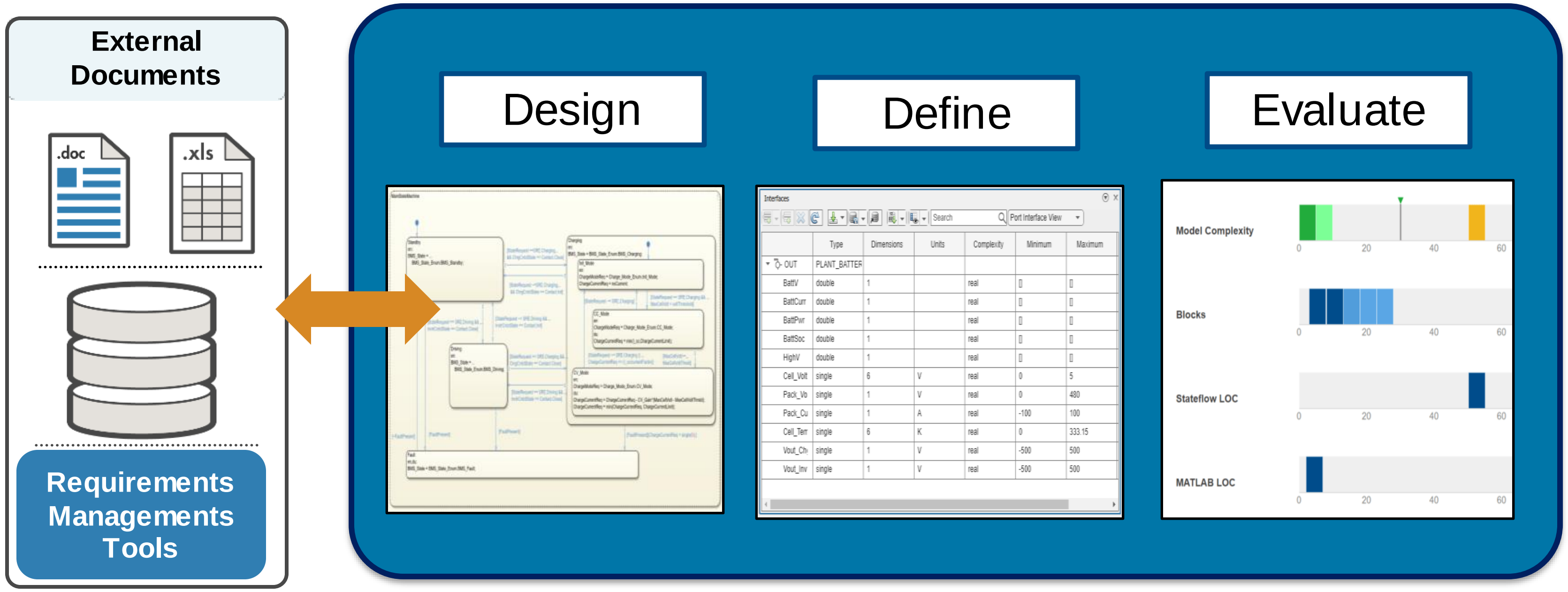
SWE.3 SW Detailed Design and Unit Construction



	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Ten	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

SWE.3 SW Detailed Design and Unit Construction


SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

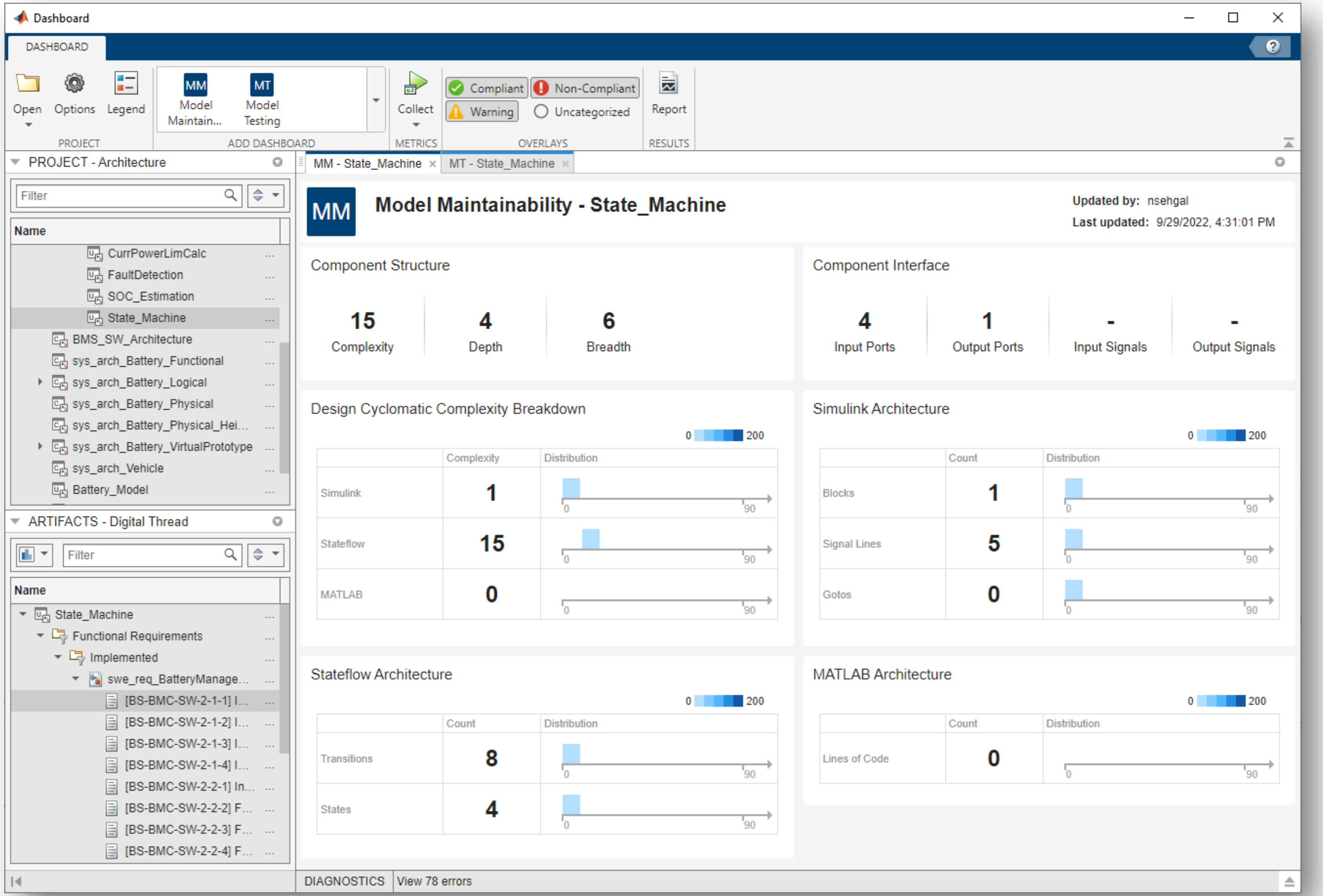


Interfaces

Search | Port Interface View

	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Terr	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

Evaluate Software Units



PROJECT - Architecture

Filter

Name
- CurrPowerLimCalc
- FaultDetection
- SOC_Estimation
- State_Machine
- BMS_SW_Architecture
- sys_arch_Battery_Functional
- sys_arch_Battery_Logical
- sys_arch_Battery_Physical
- sys_arch_Battery_Physical_Hei...
- sys_arch_Battery_VirtualPrototype
- sys_arch_Vehicle
- Battery_Model

ARTIFACTS - Digital Thread

Filter

Name
- State_Machine
  - Functional Requirements
    - Implemented
      - swe_req_BatteryManage...
        - [BS-BMC-SW-2-1-1] I...
        - [BS-BMC-SW-2-1-2] I...
        - [BS-BMC-SW-2-1-3] I...
        - [BS-BMC-SW-2-1-4] I...
        - [BS-BMC-SW-2-2-1] In...
        - [BS-BMC-SW-2-2-2] F...
        - [BS-BMC-SW-2-2-3] F...
        - [BS-BMC-SW-2-2-4] F...

MM - State_Machine | MT - State_Machine

MM Model Maintainability - State_Machine

Updated by: nsehgal
Last updated: 9/29/2022, 4:31:01 PM

Component Structure

15	4	6
Complexity	Depth	Breadth

Component Interface

4	1	-	-
Input Ports	Output Ports	Input Signals	Output Signals

Design Cyclomatic Complexity Breakdown (0 – 200)

	Complexity	Distribution
Simulink	1	0 – 90
Stateflow	15	0 – 90
MATLAB	0	0 – 90

Simulink Architecture (0 – 200)

	Count	Distribution
Blocks	1	0 – 90
Signal Lines	5	0 – 90
Gotos	0	0 – 90

Stateflow Architecture (0 – 200)

	Count	Distribution
Transitions	8	0 – 90
States	4	0 – 90

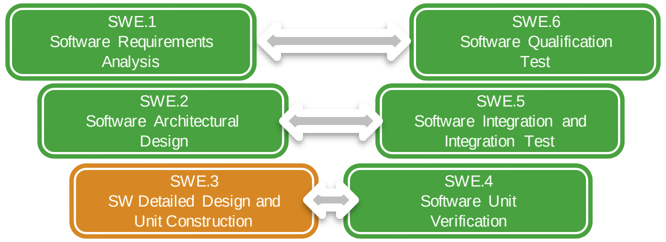
MATLAB Architecture (0 – 200)

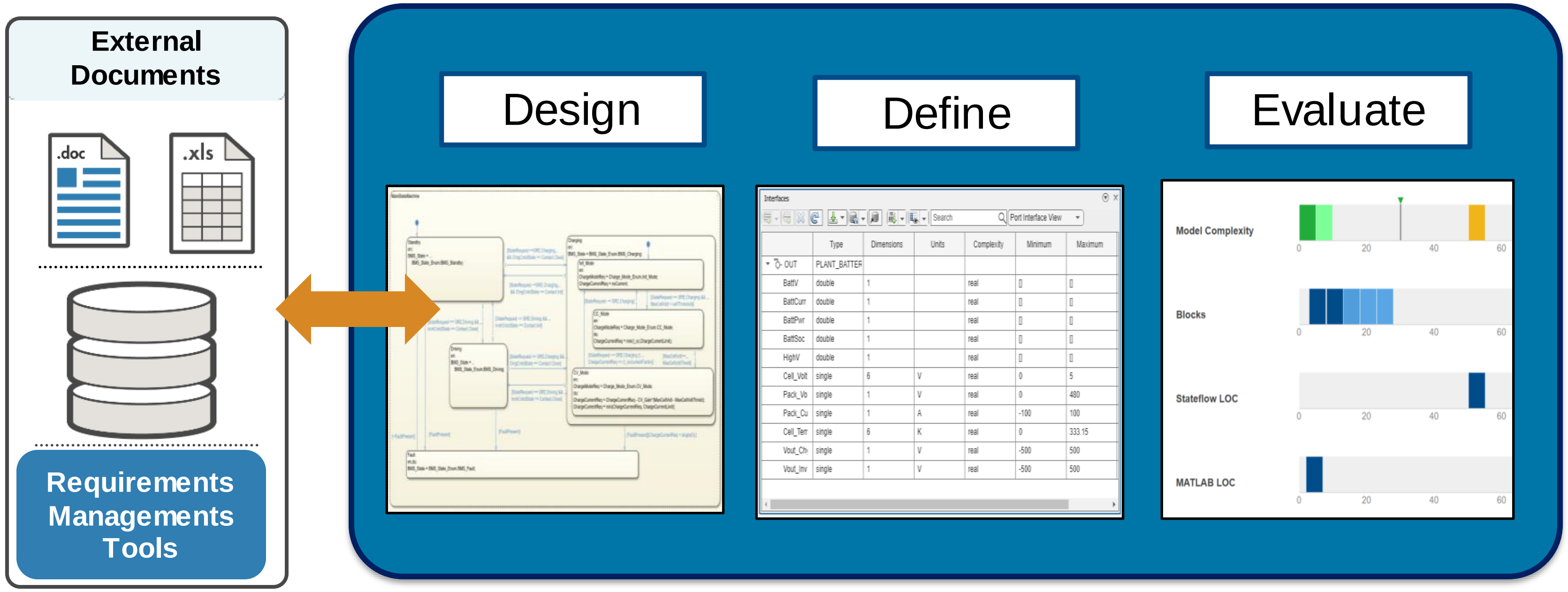
	Count	Distribution
Lines of Code	0	0 – 90

DIAGNOSTICS | View 78 errors

Monitor the Complexity of Your Design Using the Model Maintainability Dashboard

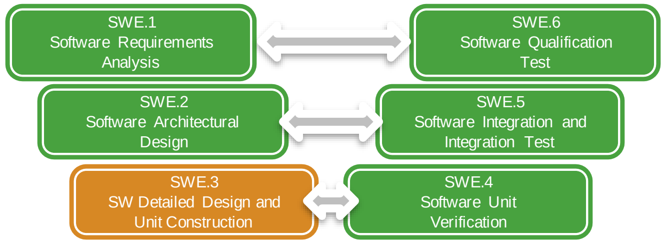
SWE.3 SW Detailed Design and Unit Construction

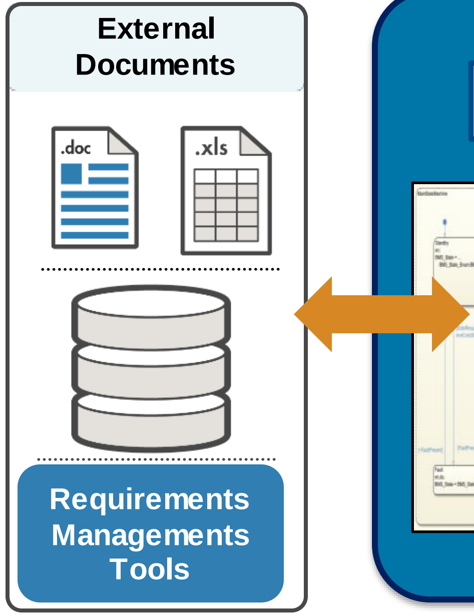

SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification



	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Tem	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

SWE.3 SW Detailed Design and Unit Construction


SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

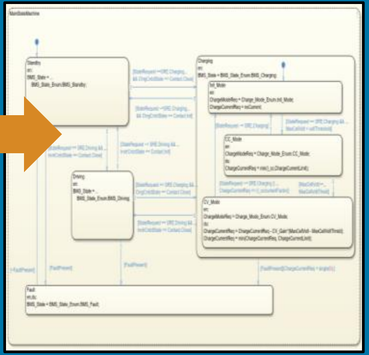


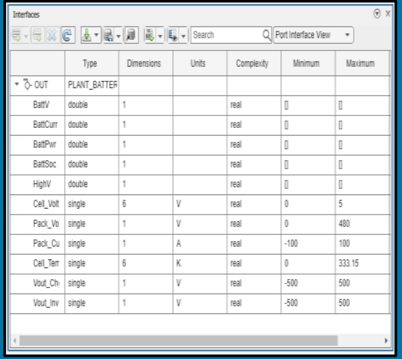
Design

Define

Evaluate

Develop

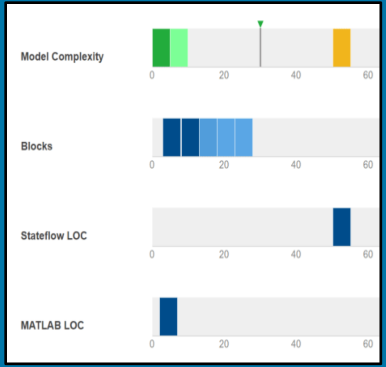


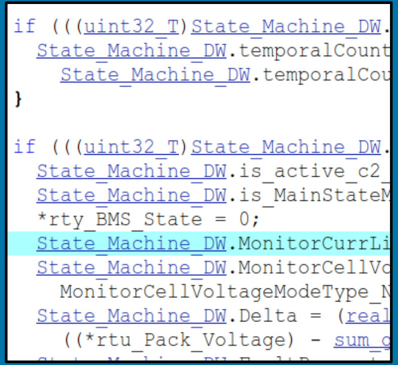


Interfaces

Search | Port Interface View

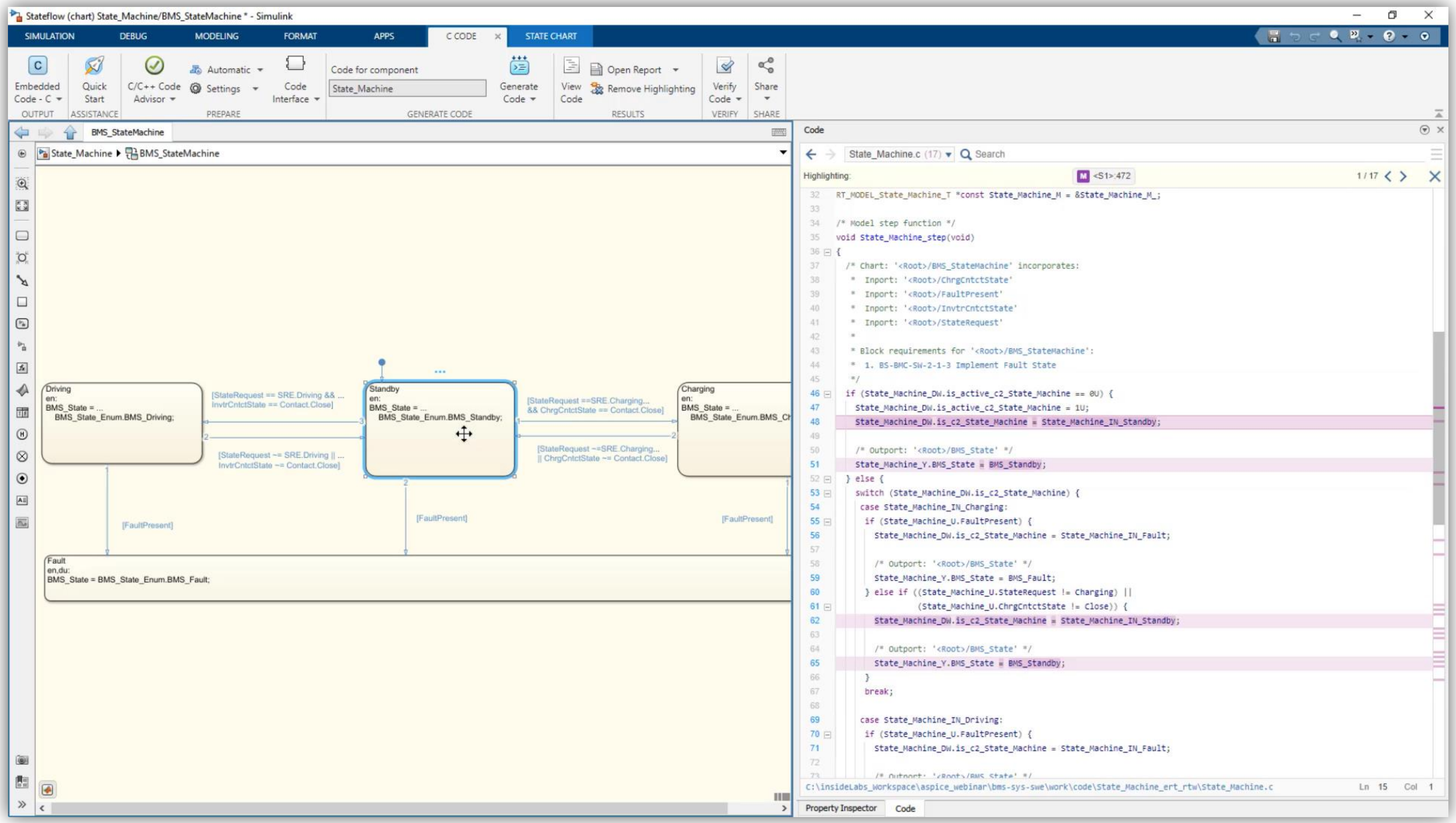
	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Ten	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500





```
if (((uint32_T)State_Machine_DW.
  State_Machine_DW.temporalCount
    State_Machine_DW.temporalCou
}

if (((uint32_T)State_Machine_DW.
  State_Machine_DW.is_active_c2_
  State_Machine_DW.is_MainStateM
  *rty_BMS_State = 0;
  State_Machine_DW.MonitorCurrLi
  State_Machine_DW.MonitorCellVo
    MonitorCellVoltageModeType_N
  State_Machine_DW.Delta = (real
    ((*rtu_Pack_Voltage) - sum_g
```

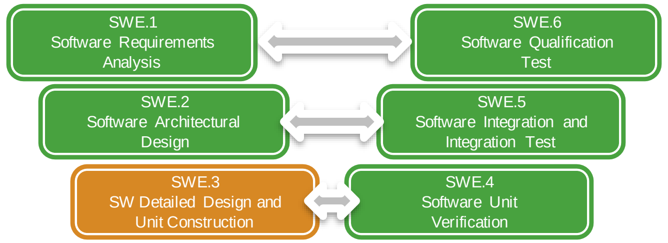
Develop Software Units – Code Generation



```
RT_MODEL_State_Machine_T *const State_Machine_M = &State_Machine_M_;

/* Model step function */
void State_Machine_step(void)
{
  /* Chart: '<Root>/BMS_StateMachine' incorporates:
   *  Inport: '<Root>/ChrgCntctState'
   *  Inport: '<Root>/FaultPresent'
   *  Inport: '<Root>/InvtrCntctState'
   *  Inport: '<Root>/StateRequest'
   *
   * Block requirements for '<Root>/BMS_StateMachine':
   *  1. BS-BMC-SW-2-1-3 Implement Fault State
   */
  if (State_Machine_DW.is_active_c2_State_Machine == 0U) {
    State_Machine_DW.is_active_c2_State_Machine = 1U;
    State_Machine_DW.is_c2_State_Machine = State_Machine_IN_Standby;

    /* Outport: '<Root>/BMS_State' */
    State_Machine_Y.BMS_State = BMS_Standby;
  } else {
    switch (State_Machine_DW.is_c2_State_Machine) {
     case State_Machine_IN_Charging:
      if (State_Machine_U.FaultPresent) {
        State_Machine_DW.is_c2_State_Machine = State_Machine_IN_Fault;

        /* Outport: '<Root>/BMS_State' */
        State_Machine_Y.BMS_State = BMS_Fault;
      } else if ((State_Machine_U.StateRequest != Charging) ||
                 (State_Machine_U.ChrgCntctState != Close)) {
        State_Machine_DW.is_c2_State_Machine = State_Machine_IN_Standby;

        /* Outport: '<Root>/BMS_State' */
        State_Machine_Y.BMS_State = BMS_Standby;
      }
      break;

     case State_Machine_IN_Driving:
      if (State_Machine_U.FaultPresent) {
        State_Machine_DW.is_c2_State_Machine = State_Machine_IN_Fault;

        /* Outport: '<Root>/BMS_State' */
```

```
C:\insideLabs_Workspace\aspice_webinar\bms-sys-swe\work\code\State_Machine_ert_rtw\State_Machine.c
```

SWE.3 SW Detailed Design and Unit Construction

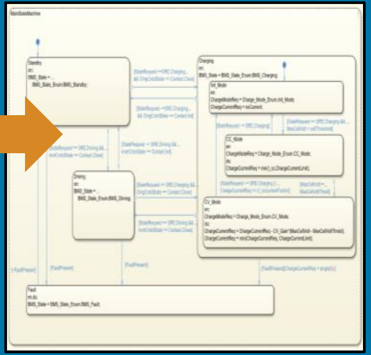

SWE.1
Software Requirements Analysis
SWE.6
Software Qualification Test
SWE.2
Software Architectural Design
SWE.5
Software Integration and Integration Test
SWE.3
SW Detailed Design and Unit Construction
SWE.4
Software Unit Verification

External Documents

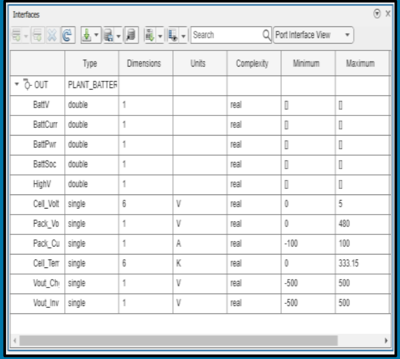
.doc .xls

Requirements Managements Tools

Design



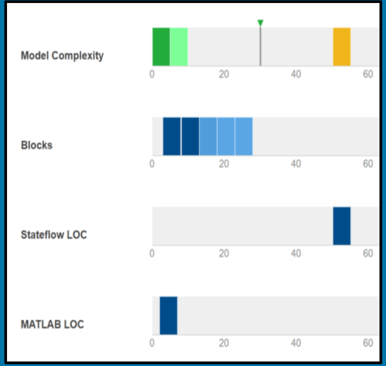
Define



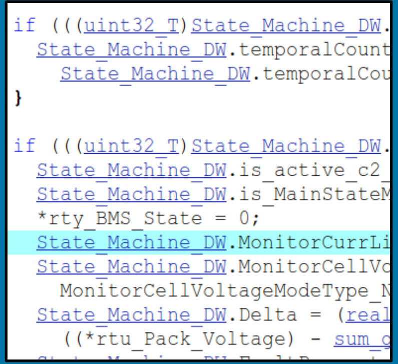
Interfaces

	Type	Dimensions	Units	Complexity	Minimum	Maximum
OUT	PLANT_BATTER					
BattV	double	1		real	[]	[]
BattCurr	double	1		real	[]	[]
BattPwr	double	1		real	[]	[]
BattSoc	double	1		real	[]	[]
HighV	double	1		real	[]	[]
Cell_Volt	single	6	V	real	0	5
Pack_Vo	single	1	V	real	0	480
Pack_Cu	single	1	A	real	-100	100
Cell_Terr	single	6	K	real	0	333.15
Vout_Ch	single	1	V	real	-500	500
Vout_Inv	single	1	V	real	-500	500

Evaluate



Deploy



```
if (((uint32_T)State_Machine_DW.
  State_Machine_DW.temporalCount
    State_Machine_DW.temporalCou
}

if (((uint32_T)State_Machine_DW.
  State_Machine_DW.is_active_c2_
  State_Machine_DW.is_MainStateM
  *rty_BMS_State = 0;
  State_Machine_DW.MonitorCurrLi
  State_Machine_DW.MonitorCellVo
    MonitorCellVoltageModeType_N
  State_Machine_DW.Delta = (real
    ((*rtu_Pack_Voltage) - sum_g
```

Output Work Products

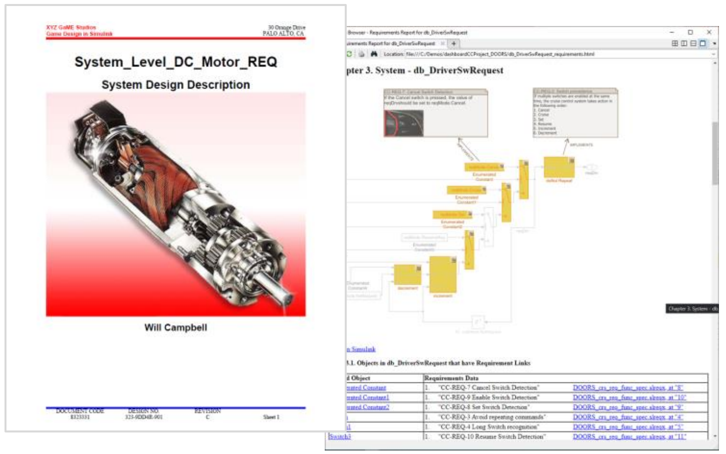
Detailed Design

Software Unit

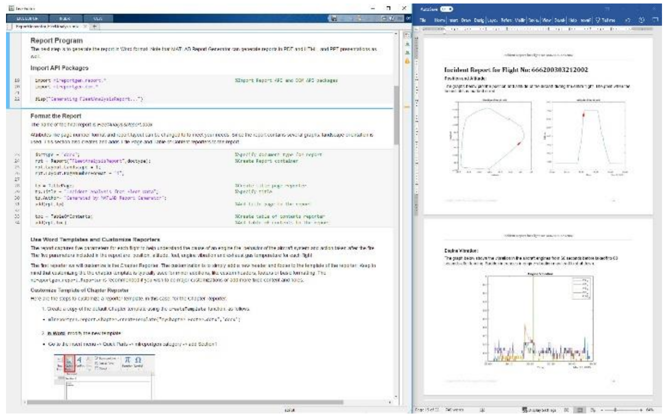
Interfaces

Traceability

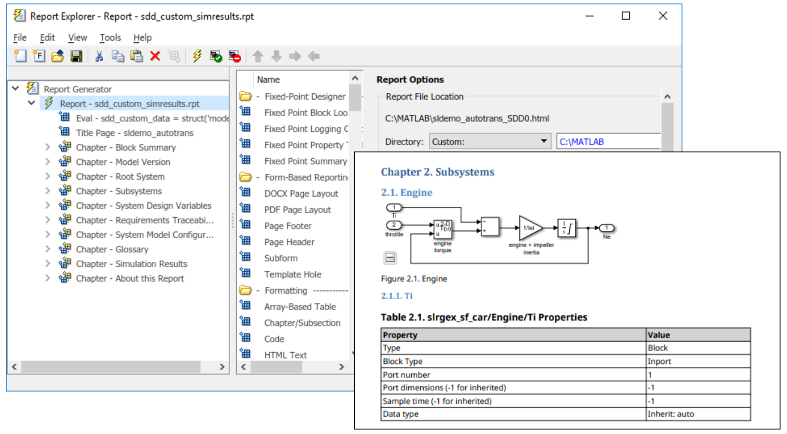
Output Work Products using MATLAB/Simulink Report Generator



Predefined Standard Reports



Report and DOM APIs



Chapter 2. Subsystems

2.1. Engine

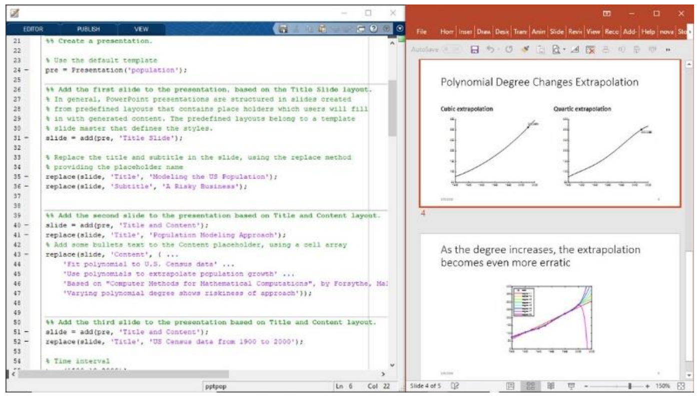
Figure 2.1. Engine

2.1.1. Ti

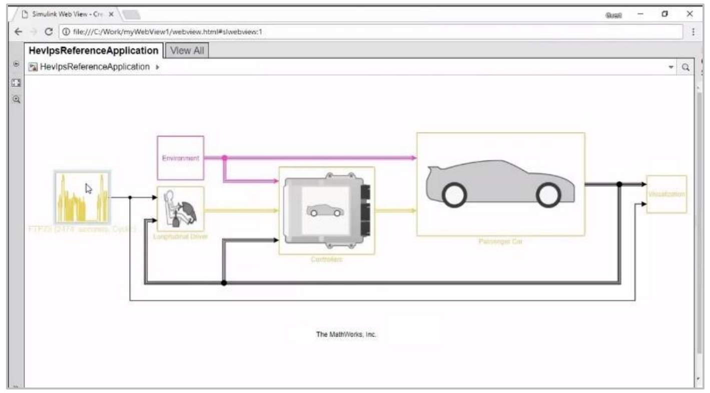
Table 2.1. slrgex_sf_car/Engine/Ti Properties

Property	Value
Type	Block
Block Type	Inport
Port number	1
Port dimensions (-1 for inherited)	-1
Sample time (-1 for inherited)	-1
Data type	Inherit: auto

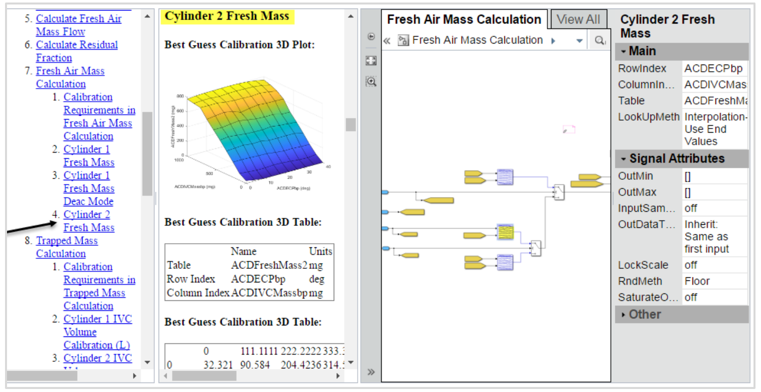
Report Explorer



PowerPoint® API



Web View



Embedded Web View

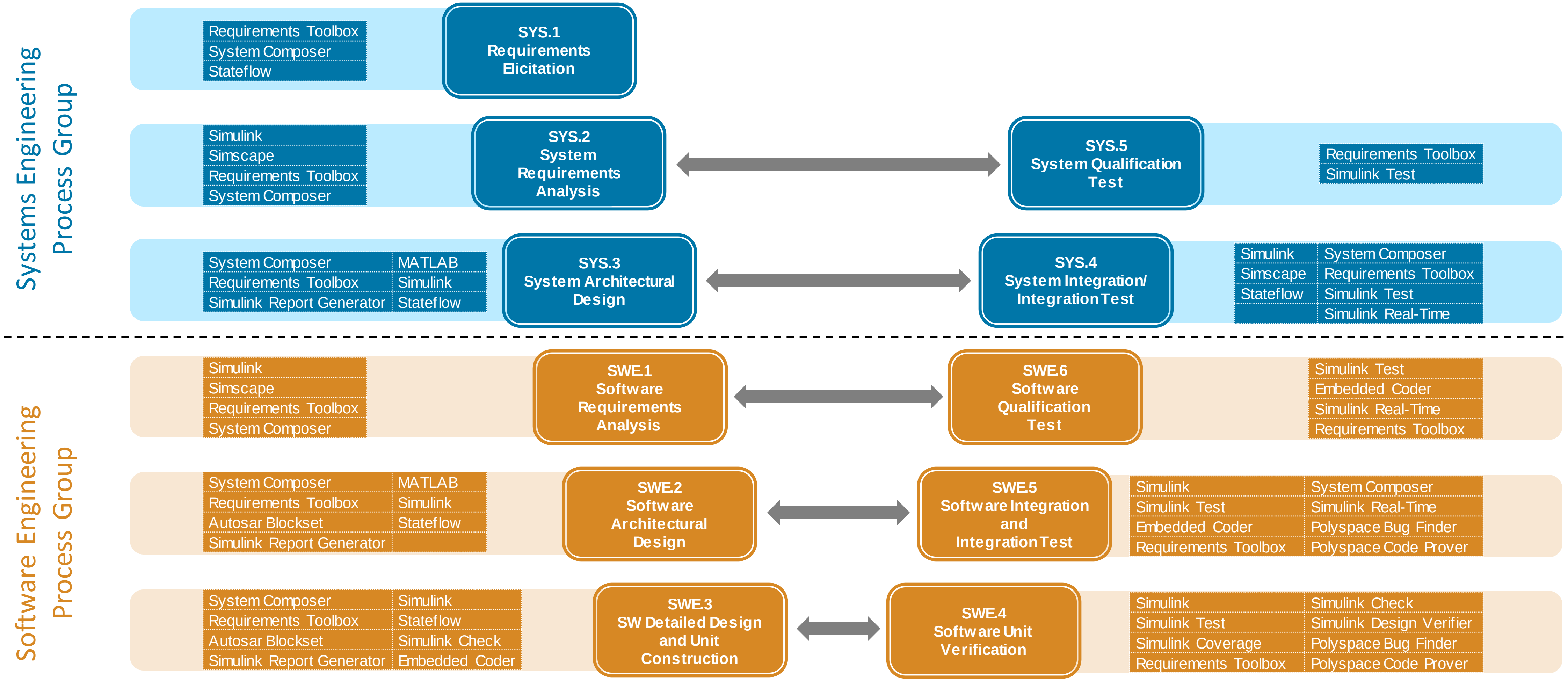
Key Takeaways

Model-Based Design and Model-Based Systems Engineering enable:

1. **Fast development and realization** of **system and software** architecture and design
2. **Early testing** to detect errors in designs and their realization
3. **Fast and efficient iterations**

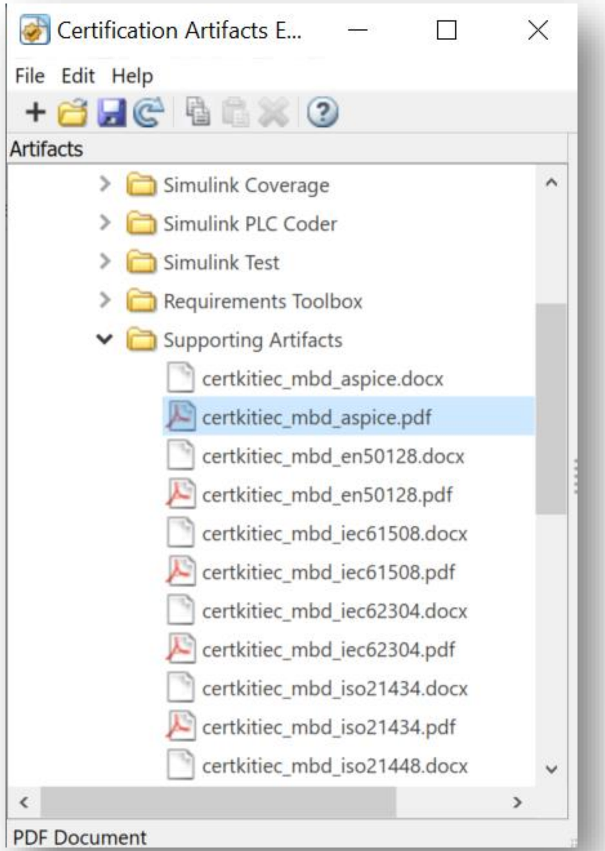
→ Develop **high quality products** following an **efficient Automotive SPICE® compliant process**

Mapping of A-SPICE® Processes to MathWorks Solution



Referencde MBD Process for A-SPICE®

IEC Certification Kit


Certification Artifacts E...

File Edit Help

Artifacts

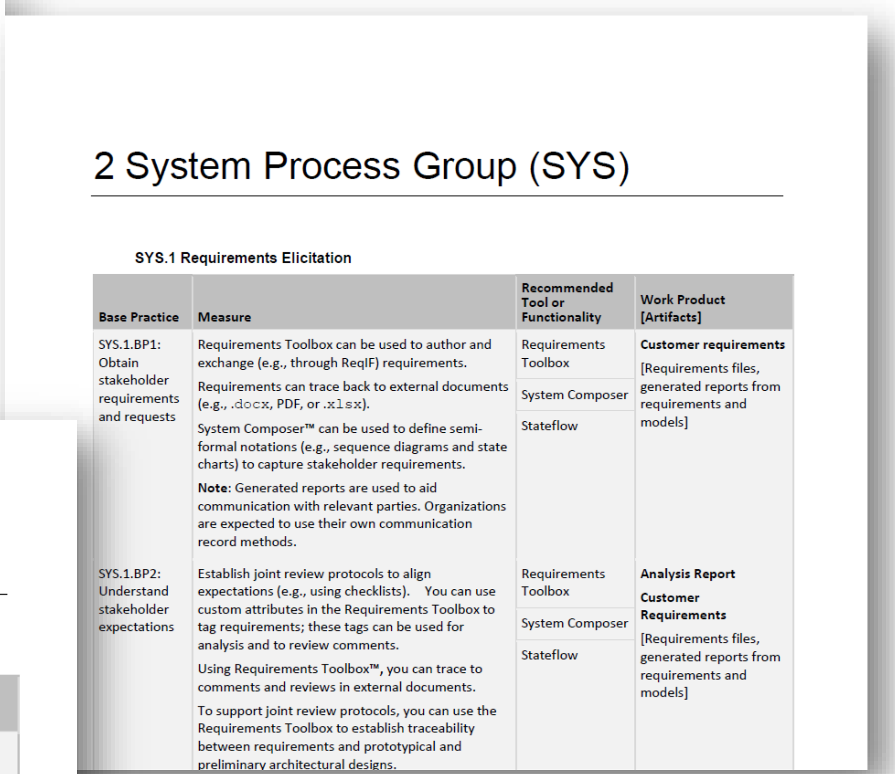
- Simulink Coverage
- Simulink PLC Coder
- Simulink Test
- Requirements Toolbox
- Supporting Artifacts
  - certkitiec_mbd_aspice.docx
  - certkitiec_mbd_aspice.pdf
  - certkitiec_mbd_en50128.docx
  - certkitiec_mbd_en50128.pdf
  - certkitiec_mbd_iec61508.docx
  - certkitiec_mbd_iec61508.pdf
  - certkitiec_mbd_iec62304.docx
  - certkitiec_mbd_iec62304.pdf
  - certkitiec_mbd_iso21434.docx
  - certkitiec_mbd_iso21434.pdf
  - certkitiec_mbd_iso21448.docx

PDF Document

IEC Certification Kit

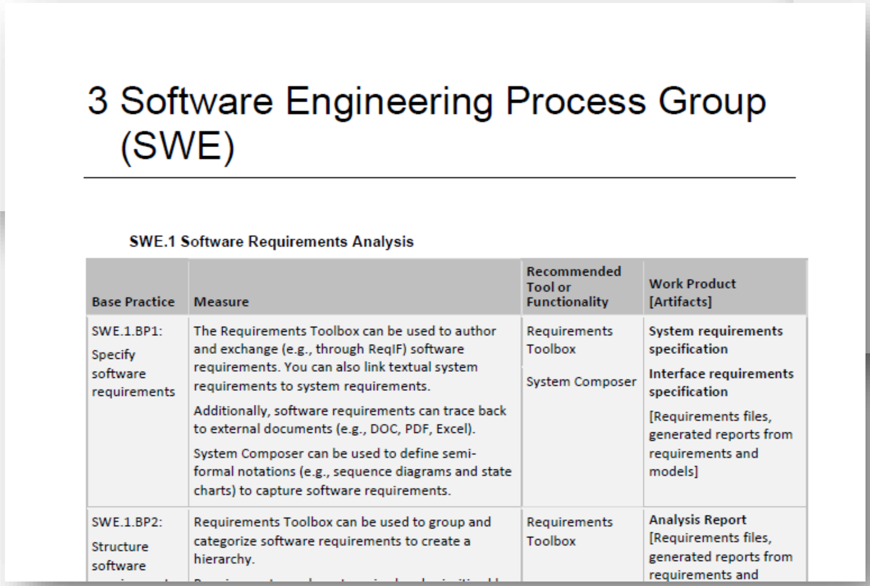
Reference Model-Based Design Process for Automotive SPICE®

R2022a


2 System Process Group (SYS)

SYS.1 Requirements Elicitation

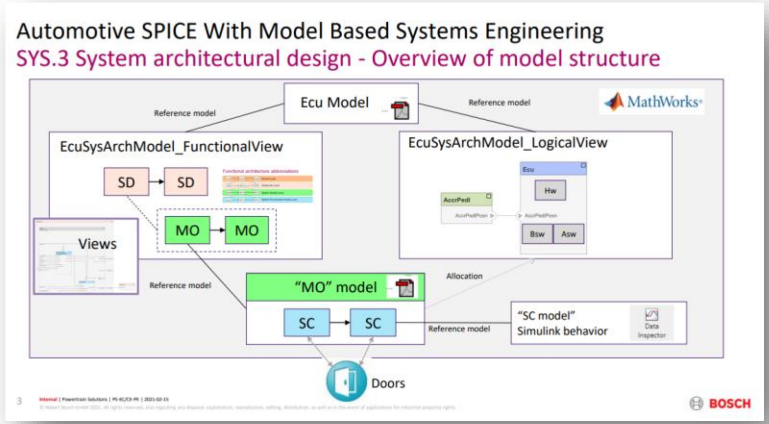
Base Practice	Measure	Recommended Tool or Functionality	Work Product [Artifacts]
SYS.1.BP1: Obtain stakeholder requirements and requests	Requirements Toolbox can be used to author and exchange (e.g., through ReqIF) requirements. Requirements can trace back to external documents (e.g., `.docx`, PDF, or `.xlsx`). System Composer™ can be used to define semi-formal notations (e.g., sequence diagrams and state charts) to capture stakeholder requirements. **Note**: Generated reports are used to aid communication with relevant parties. Organizations are expected to use their own communication record methods.	Requirements Toolbox System Composer Stateflow	**Customer requirements** [Requirements files, generated reports from requirements and models]
SYS.1.BP2: Understand stakeholder expectations	Establish joint review protocols to align expectations (e.g., using checklists). You can use custom attributes in the Requirements Toolbox to tag requirements; these tags can be used for analysis and to review comments. Using Requirements Toolbox™, you can trace to comments and reviews in external documents. To support joint review protocols, you can use the Requirements Toolbox to establish traceability between requirements and prototypical and preliminary architectural designs.	Requirements Toolbox System Composer Stateflow	**Analysis Report** **Customer Requirements** [Requirements files, generated reports from requirements and models]


3 Software Engineering Process Group (SWE)

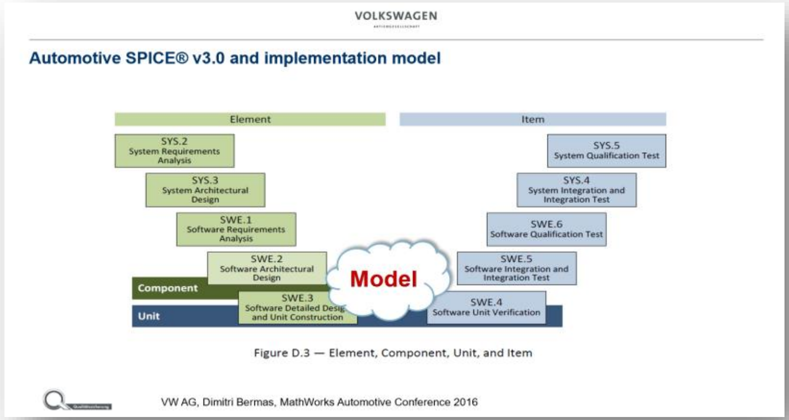
SWE.1 Software Requirements Analysis

Base Practice	Measure	Recommended Tool or Functionality	Work Product [Artifacts]
SWE.1.BP1: Specify software requirements	The Requirements Toolbox can be used to author and exchange (e.g., through ReqIF) software requirements. You can also link textual system requirements to system requirements. Additionally, software requirements can trace back to external documents (e.g., DOC, PDF, Excel). System Composer can be used to define semi-formal notations (e.g., sequence diagrams and state charts) to capture software requirements.	Requirements Toolbox System Composer	**System requirements specification** **Interface requirements specification** [Requirements files, generated reports from requirements and models]
SWE.1.BP2: Structure software	Requirements Toolbox can be used to group and categorize software requirements to create a hierarchy.	Requirements Toolbox	**Analysis Report** [Requirements files, generated reports from requirements and

Video Resources: A-SPICE® with Model-Based Design


Automotive SPICE With Model Based Systems Engineering
SYS.3 System architectural design - Overview of model structure

[Robert Bosch:
System Architectural Design According to Automotive SPICE Using the MathWorks toolchain]

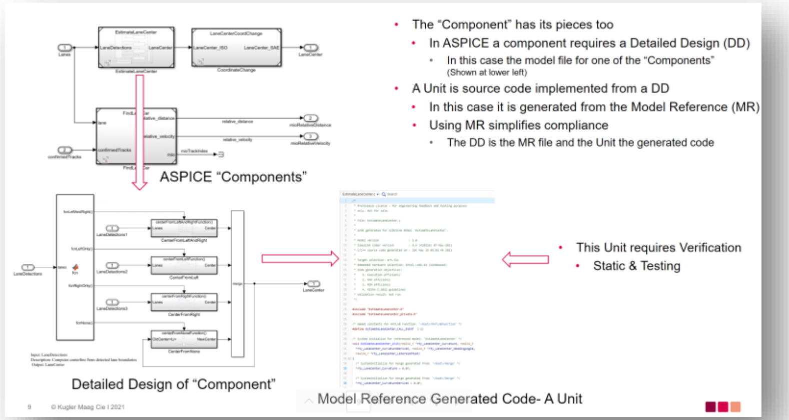

Automotive SPICE® v3.0 and implementation model

VOLKSWAGEN
Element
Item
SYS.2 System Requirements Analysis
SYS.3 System Architectural Design
SWE.1 Software Requirements Analysis
SWE.2 Software Architectural Design
SWE.3 Software Detailed Design and Unit Construction
SYS.5 System Qualification Test
SYS.4 System Integration and Integration Test
SWE.6 Software Qualification Test
SWE.5 Software Integration and Integration Test
SWE.4 Software Unit Verification
Component
Unit
Model

Figure D.3 — Element, Component, Unit, and Item

VW AG, Dimitri Bermas, MathWorks Automotive Conference 2016

[Volkswagen:
Software Detailed Design for Model-Based Development]


- The "Component" has its pieces too
  - In ASPICE a component requires a Detailed Design (DD)
    - In this case the model file for one of the "Components" (Shown at lower left)
- A Unit is source code implemented from a DD
  - In this case it is generated from the Model Reference (MR)
  - Using MR simplifies compliance
    - The DD is the MR file and the Unit the generated code

ASPICE "Components"

- This Unit requires Verification
  - Static & Testing

Detailed Design of "Component"

Model Reference Generated Code- A Unit

[Kugler Maag:
Effective Model-Based Development Strategies for ASPICE and Safety Compliance]

MathWorks

Accelerate A-SPICE® Compliance with Model-Based Design – Part 1

System Engineering Processes

Mohammad Abu-Alqumsan, *IEC Certification Kit, MathWorks, mabualqu@mathworks.com*
Giovanni Vagnoni, *Consulting Services, MathWorks, giovannv@mathworks.com*
Marc Segelken, *Application Engineering, MathWorks, msegelke@mathworks.com*

© 2022 The MathWorks, Inc.

[Accelerate A-SPICE compliance with Model-Based Design - Part 1/2 (Systems Engineering Processes)]

MathWorks

Accelerate A-SPICE Compliance with Model-Based Design – Part 2

Software Engineering Process Group

Mohammad Abu-Alqumsan, *IEC Certification Kit, MathWorks, mabualqu@mathworks.com*
Alex Shin, *Application Engineering, MathWorks, ashin@mathworks.com*
Nukul Sehgal, *Application Engineering, MathWorks, nsehgal@mathworks.com*

© 2022 The MathWorks, Inc.

[Accelerate A-SPICE compliance with Model Based Design - Part 2/2 (Software Engineering Processes)]

MathWorks

AUTOMOTIVE CONFERENCE 2023

Korea

Thank you

MathWorks®