



Use Cases For Model Execution

Žilvinas Strolia, 2016 May



- Introduction
- Automation and Debugging
- Behavior Simulation
- User Interface Mockups
- Engineering analysis
 - Automated Requirements Verification
 - Trade studies / trade-off analysis
 - Total mass/power/cost rollups
 - Model-based testing
- Integration of Analytics Models

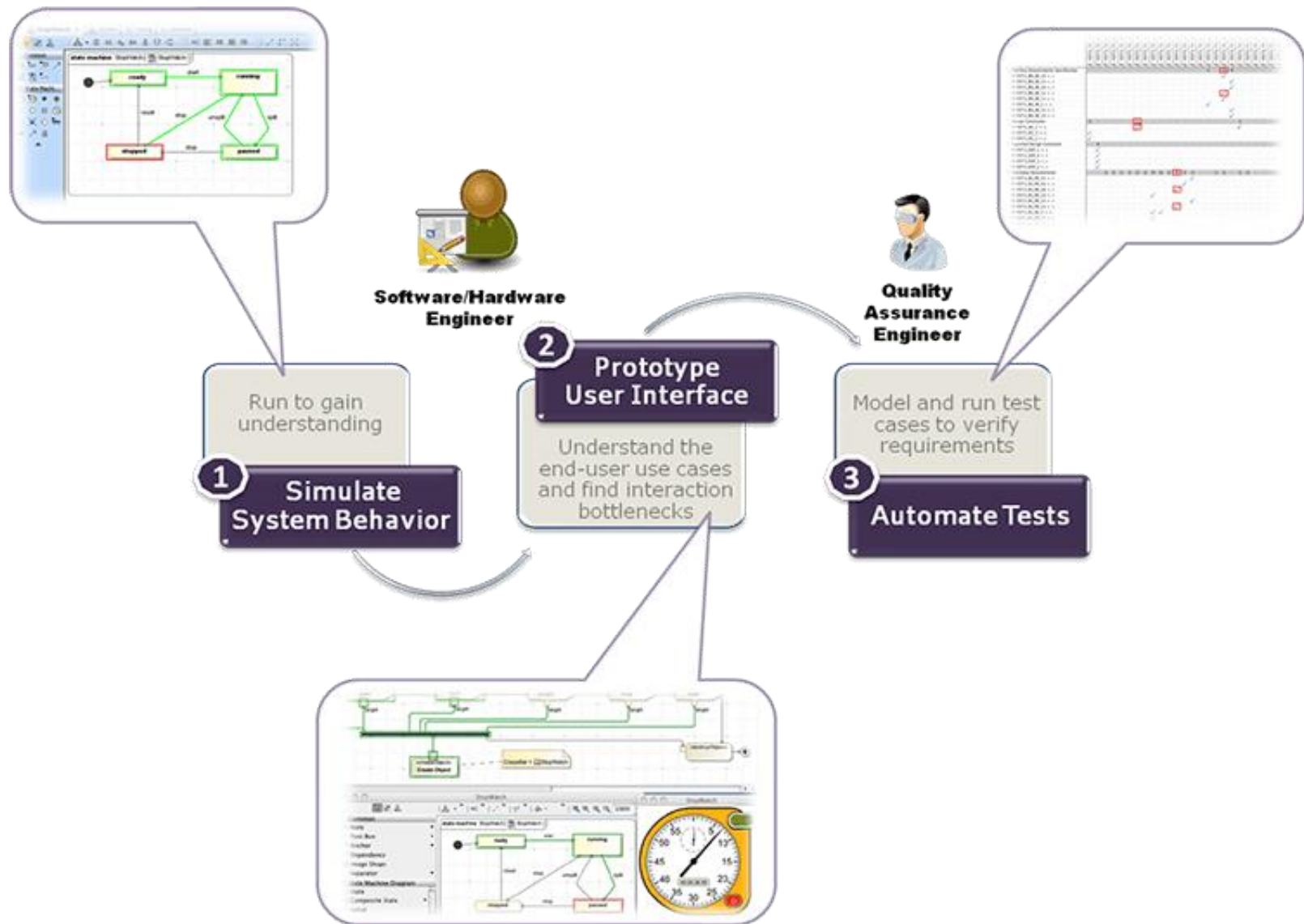


Introduction

Simulation



- The purpose of a simulation is to **gain system understanding without manipulating the real system**, either because it is not yet defined or available, or because it cannot be exercised directly due to cost, time, resources or risk constraints.
- Simulation is typically performed on a model of the system.



Cameo Simulation Toolkit (CST)



- Model execution framework and infrastructure:
 - Model debugging and animation environment
 - Pluggable engines, languages and evaluators
 - User Interface prototyping support
 - Model driven configurations and test cases
- The standard based model execution of:
 - Activities (OMG fUML standard)
 - Composite structures (OMG PSCS)
 - State Machines (W3C SCXML standard)
 - Actions/scripts (JSR223 standard)
 - Parametrics (OMG SysML standard)
 - Sequence diagrams (OMG UML Testing Profile)





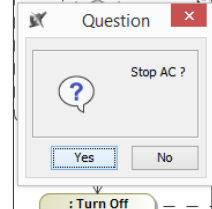
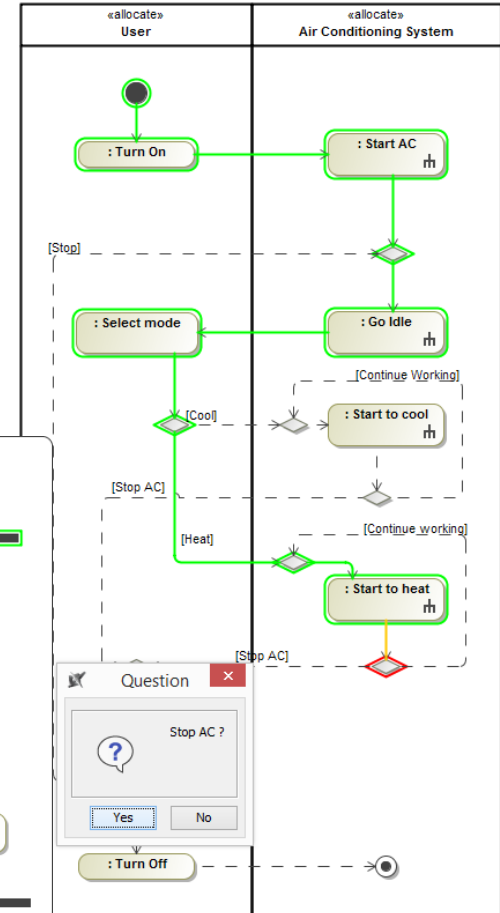
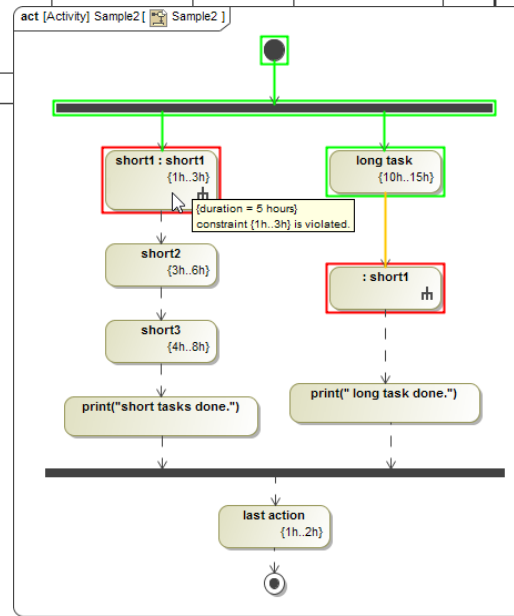
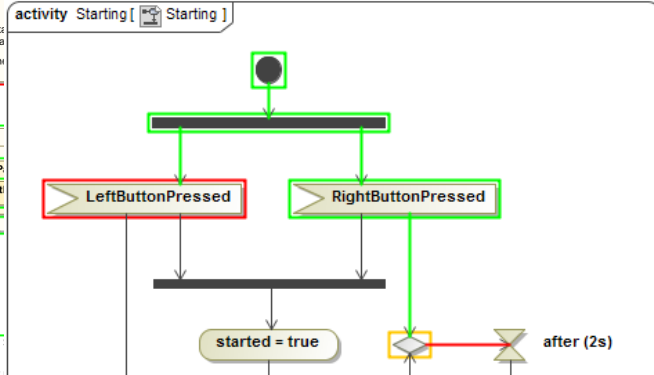
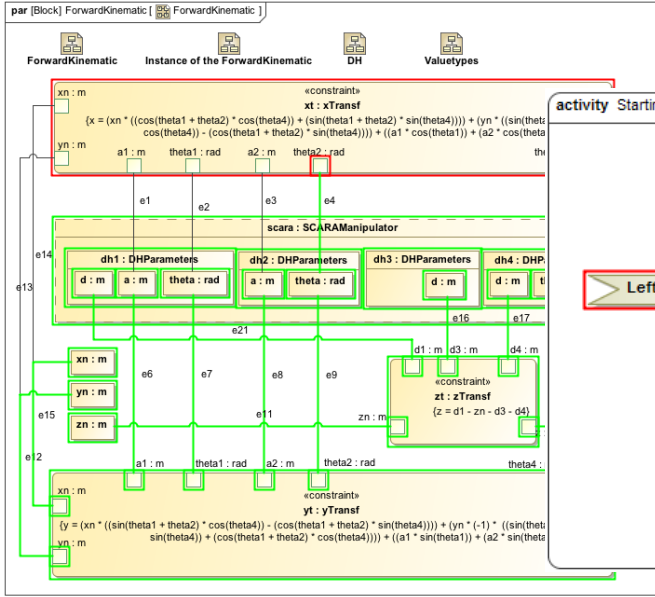
Animation and debugging

Enhance user understanding



- Animating various diagrams during the execution of a dynamic system model representing the system behavior can significantly enhance user understanding.
- A simple simulation can either rely on execution of pre-scripted scenarios, or it can react to specific user interaction (e.g., “toggle this input and see what happens”).

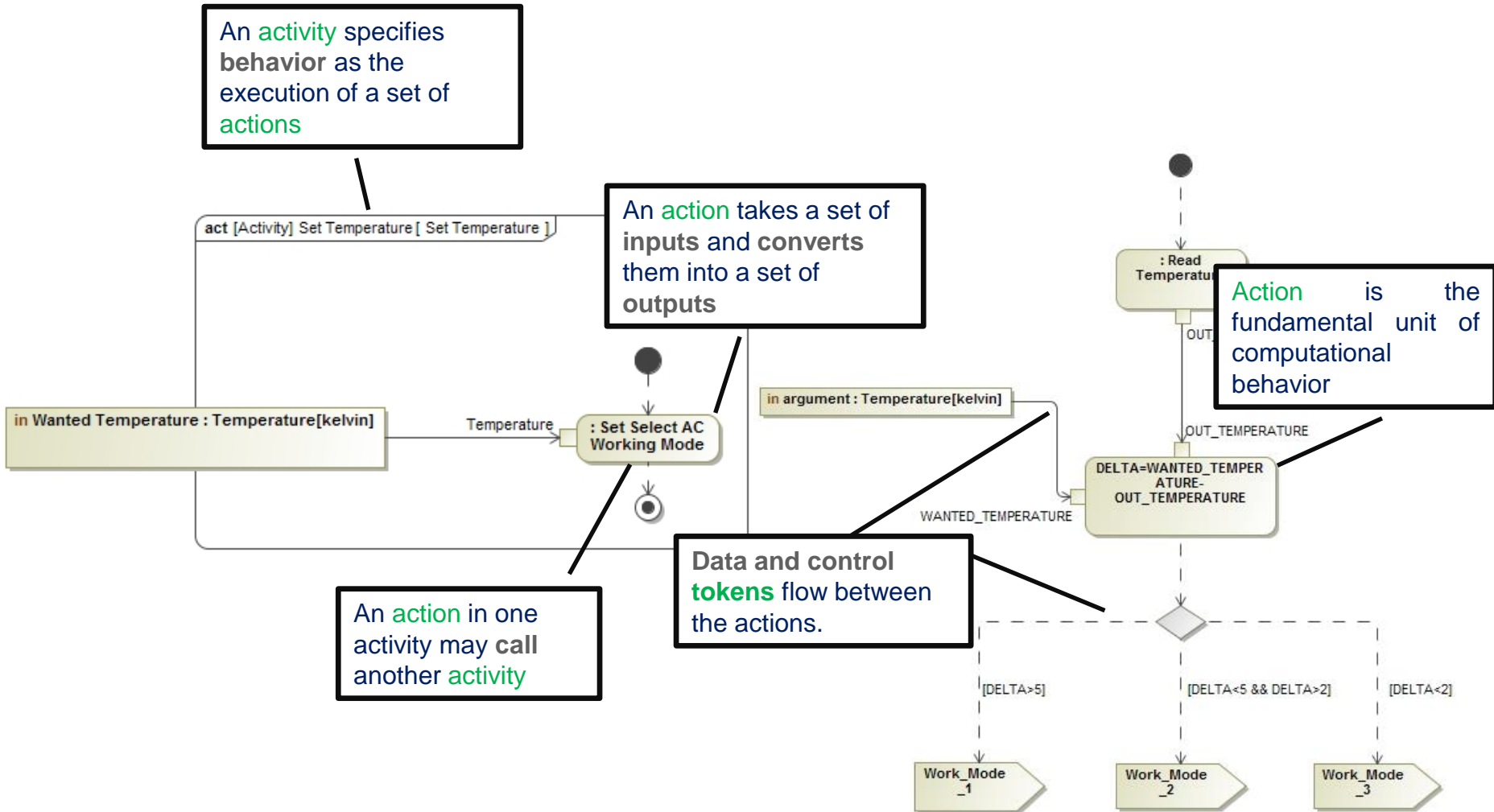
Animation



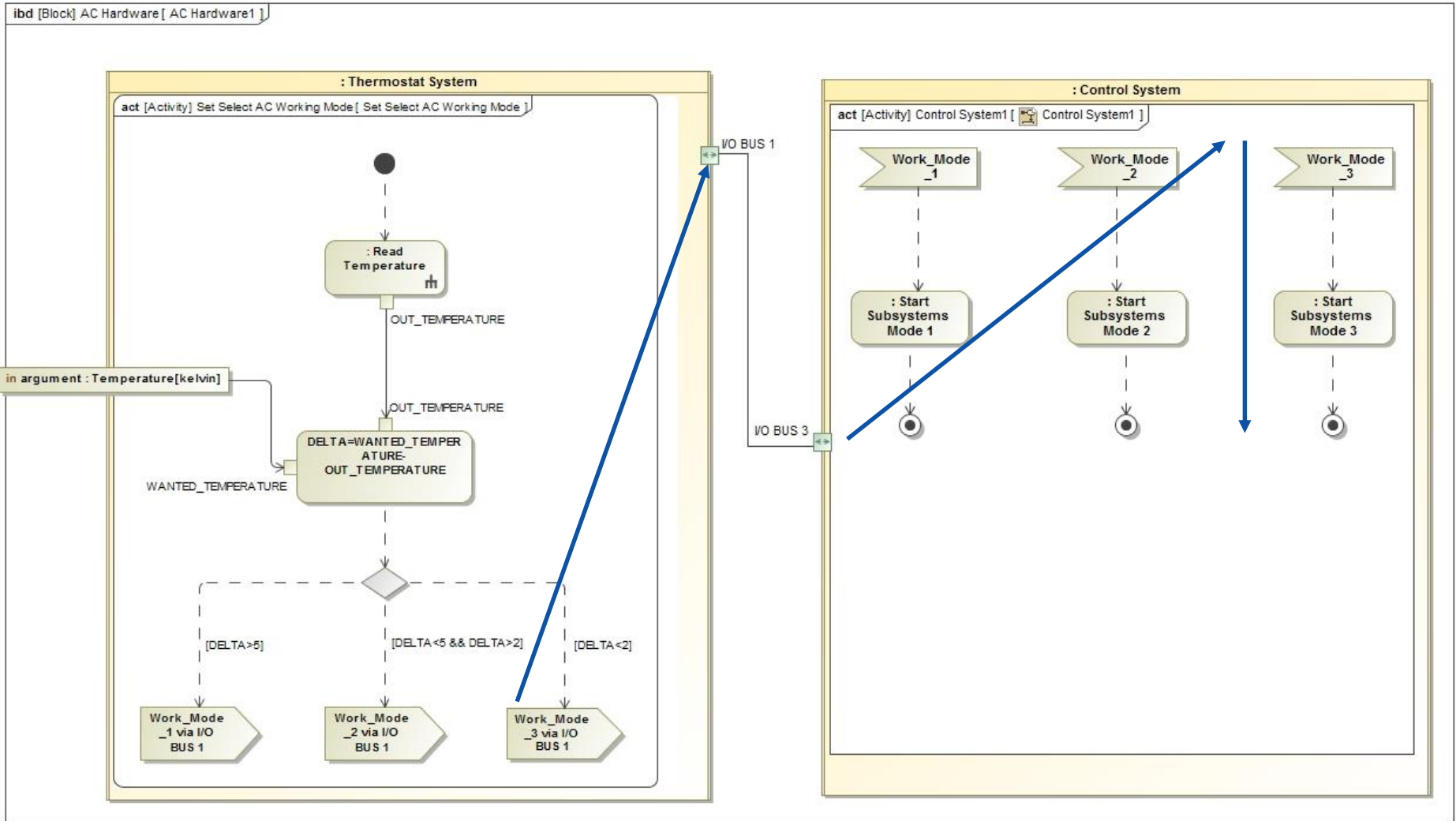


Behavior Simulation

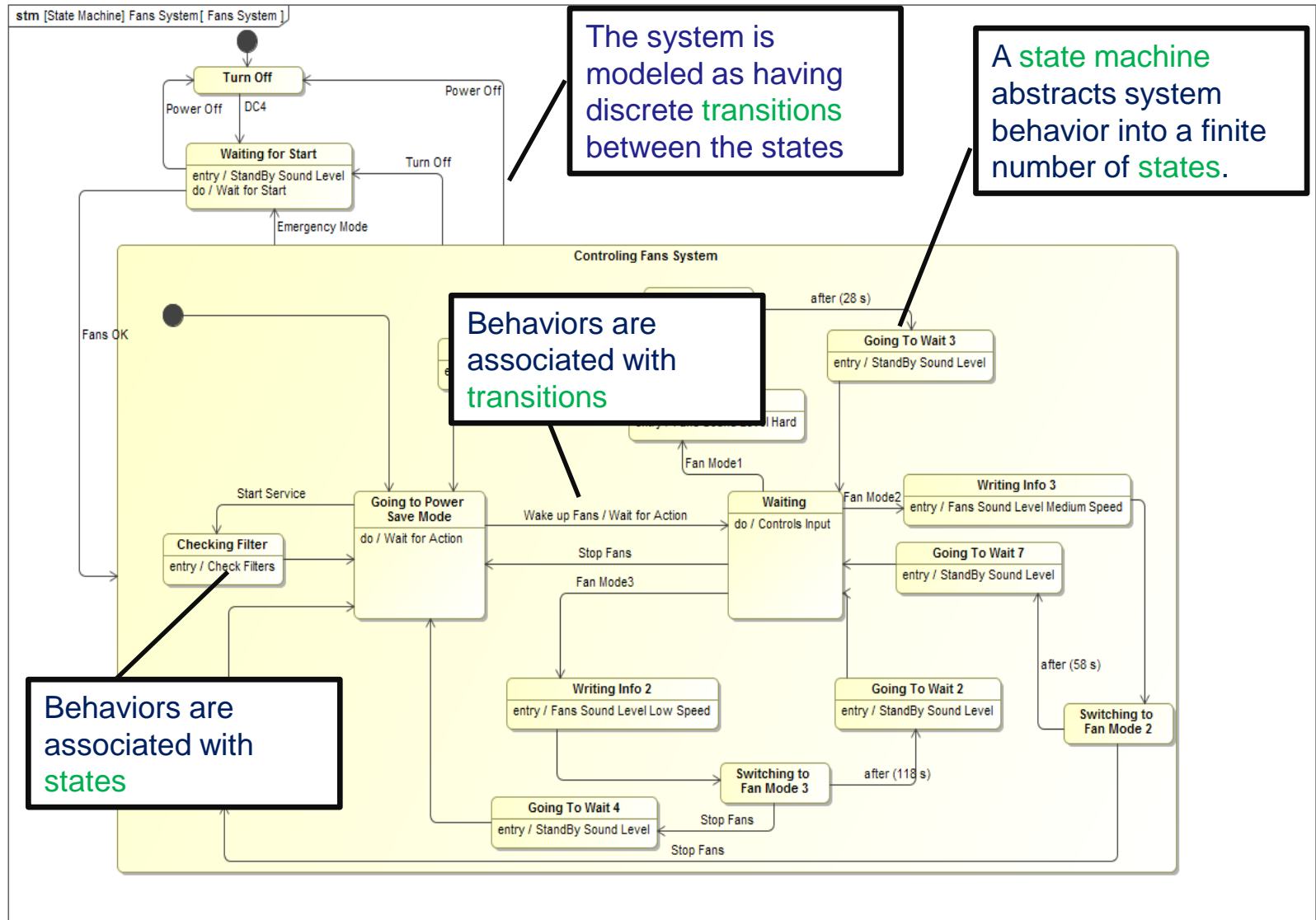
Executing Activity Diagrams



Sending Signals via Ports



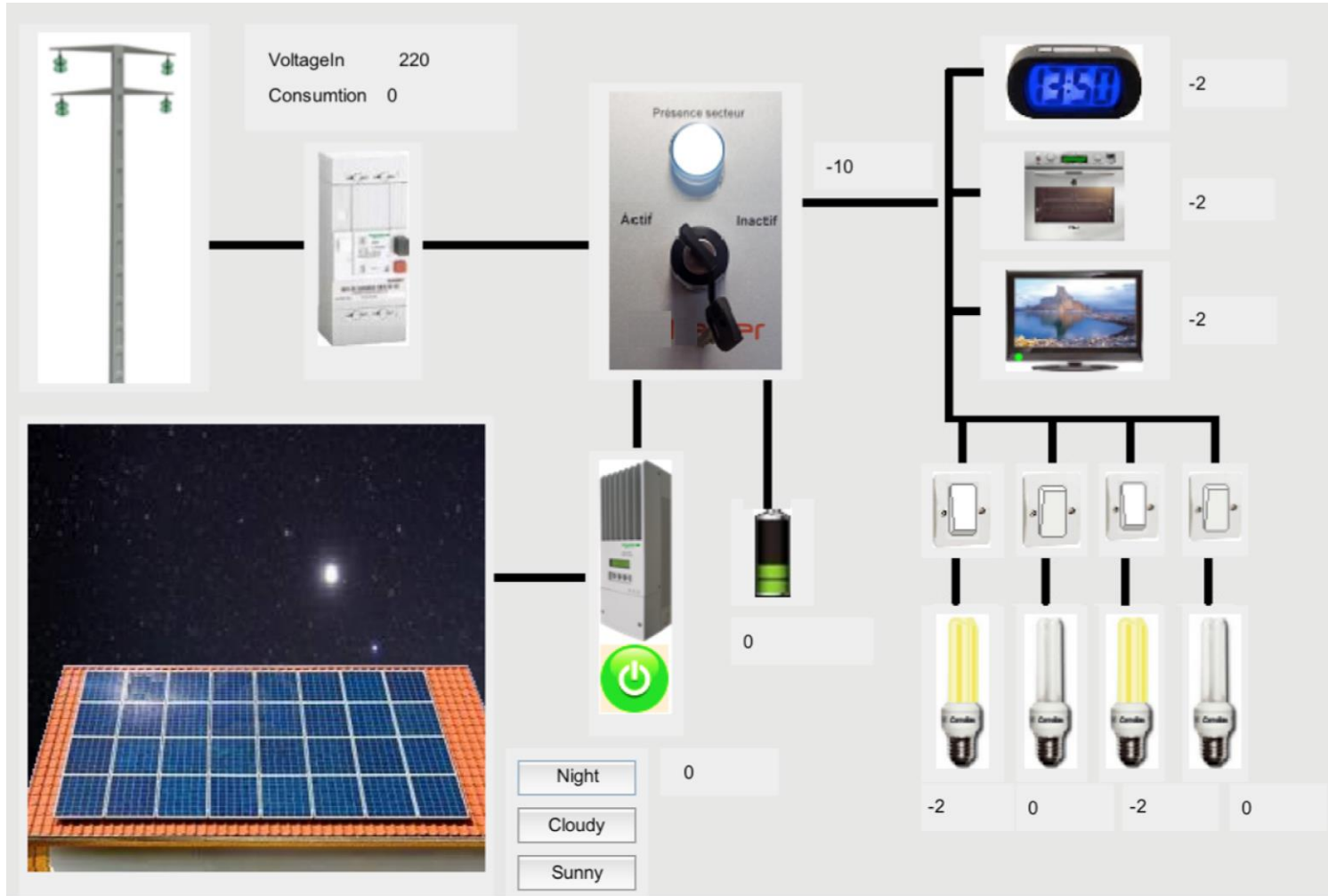
Executing State Machines





User Interface Mockups

GUI Examples



Train doors



EXE - Doors Open/Release

CBW (Left)

Cab Door A

OPEN
CLOSE

Key Switch

ON OFF

DOOR CLOSE

LEFT DOORS

RELEASE

CLOSE/INTERLOCK

TCMS HMI

Full Auto Manual Door Inhibit [OFF]
 Auto Close Auto Open Door Inhibit [ON]
 ATC Inhibit ATO Inhibit Door Close
 Clear Train

RIGHT DOORS

RELEASE

CLOSE/INTERLOCK

CBW (Right)

Cab Door B

OPEN
CLOSE

Key Switch

ON OFF

DOOR CLOSE

PASSENGER VEHICLE

EXTERNAL DOOR

EED

Leaf 1 Leaf 2

External Systems

Train Station Platform

01_Reading eastbound update
ETCS station train stopped on mark

01_Paddington PED OK PED NOK
CBTC station

01_Stratford PED Status true
AWS station Open? false

Train Speed (km/h) Train Direction

0 update eastbound update

Internal Variables

CBTC		
Active	false	Enable L false Enable R false
Full Auto	false	Open L false Open R false
Manual	false	Close L false Close R false
ADD	false	
ADC	false	Train Eastbound? true
0 Speed	true	Plat Eastbound? false
CTC		
Release L	0	Release R 0 CBTC CL L LEFT
Low Speed	true	CBTC CL R RIGHT

EDCU		
Release L	false	Release R false SDO EN true
Close HW L	false	Close HW R false SDO IN false
Close SW L	false	Close SW R false SDO BYP false
		SDO BYP IN true
		EED false
Door side	left	EAD false
Open/Rel	mode open	OBST false DRIVER IN false
Rel Inhibit	inhibit	OBST N. 3 PED IN false
Auto Close timer	-1	

Staff Controls

LOCK OOS UNLOCK ...

PULL EAD

Door P/B (3s)

Passenger Controls

Door P/B

Door P/B (2.5s)

PULL EED

CLOSE DOOR MANUALLY

PASS THROUGH DOOR

OBSTACLE

NO OBSTACLE



Engineering Analysis



- Automated Requirements Verification
- Trade studies / trade-off analysis
- Total Mass/cost/power rollup
- Model-based testing



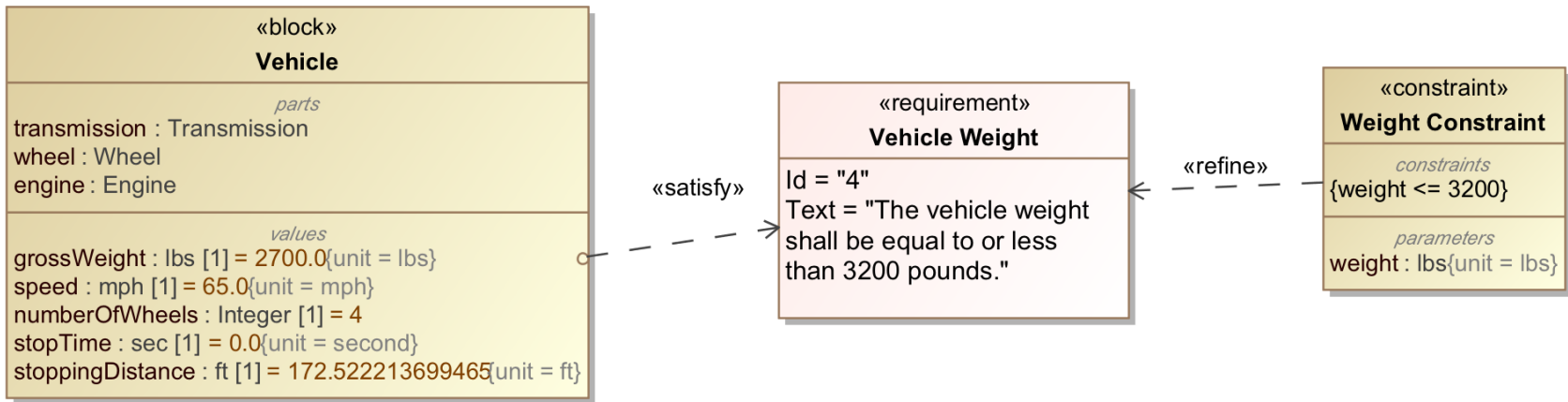
Automated Requirements Verification

Formalize requirements

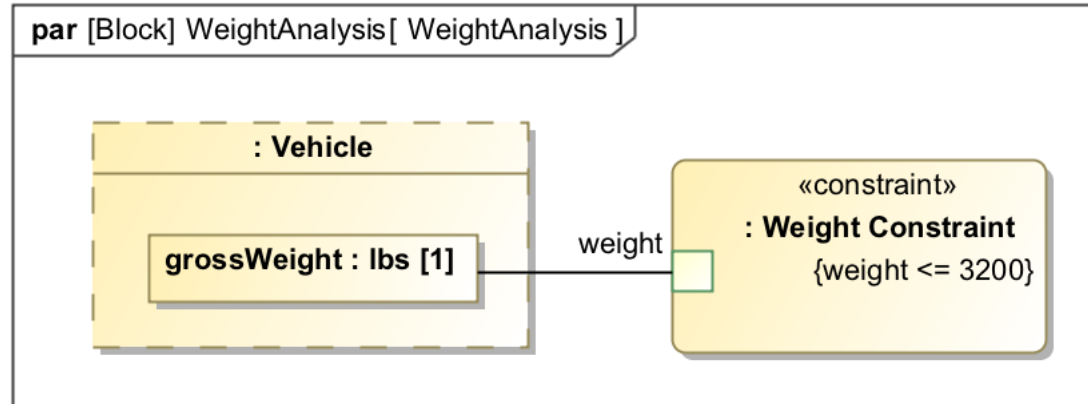
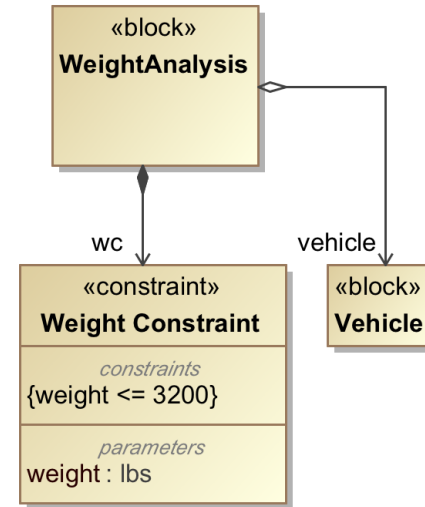
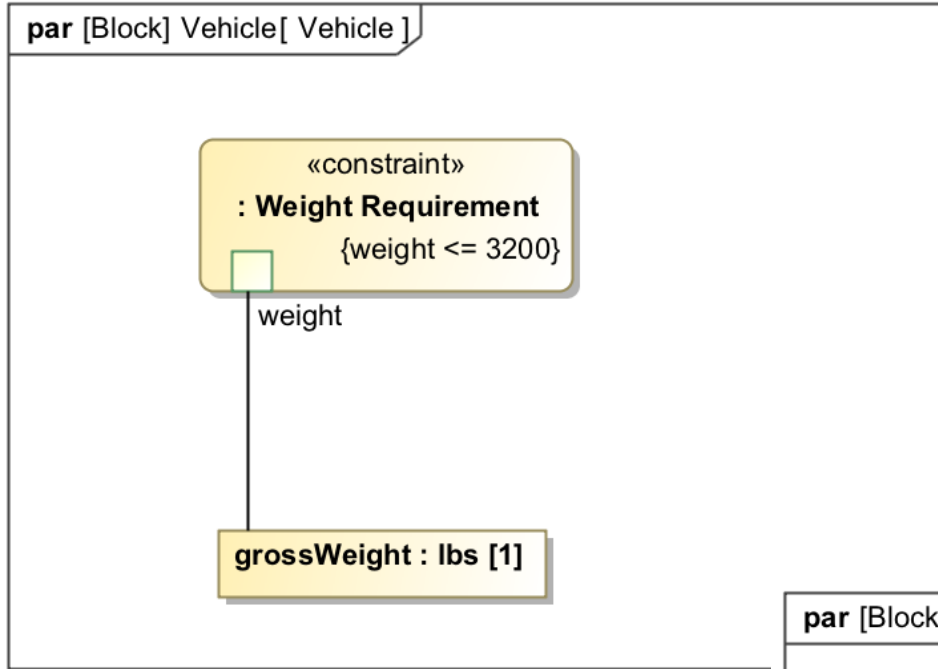


- Represent requirement in BDD
- Identify system parameter which should satisfy the requirement
- Refine requirement into more formal/computable description
- Use constraint block in analysis context
- Bind required system parameters

Step 1. Refine (formalize)



Step 2. Binding



Constraints Verification



Variables x Breakpoints x Console x Sessions x

Name	Value
WeightAnalysis	WeightAnalysis@73b24bee
Vehicle	Vehicle@71ec5be8
grossWeight : lbs [1]	2700.0000
numberOfWheels : Integer [1]	4
speed : mph [1]	60.0000
stoppingDistance : ft [1]	172.5222
stopTime : sec [1]	0.0000
engine : Engine	: Engine@1c578899
transmission : Transmission	: Transmission@8846d59
wheel : Wheel	: Wheel@339ccce9

Requirement 4 - "The vehicle weight shall be equal to or less than 3200 pounds." is satisfied.

Variables x Breakpoints x Console x Sessions x

Name	Value
WeightAnalysis	WeightAnalysis@73b24bee
Vehicle	Vehicle@71ec5be8
grossWeight : lbs [1]	3300.0000
numberOfWheels : Integer [1]	4
speed : mph [1]	60.0000
stoppingDistance : ft [1]	172.5222
stopTime : sec [1]	0.0000
engine : Engine	: Engine@1c578899
transmission : Transmission	: Transmission@8846d59
wheel : Wheel	: Wheel@339ccce9

Requirement 4 - "The vehicle weight shall be equal to or less than 3200 pounds." is not satisfied.



Trade Studies/Trade-Off Analysis

What Is A Trade Study?



A trade study or trade-off study is the activity of a multidisciplinary team to identify the most balanced technical solutions among a set of proposed viable solutions

(System Engineering Manual,
Federal Aviation Administration, 2006)

Parametric Trade Study



- Examining various design alternatives by comparison

#	Name	Manufacturer	Capacity	Voltage	Refrigerant	Power	Frequency	Current	Warranty	Price	Application
1	4JKW3	Danfoss	32000.0	230.0	R-22	1.677825	60.0	16.7	1.0	937.0	Home
2	4JKW4	Danfoss	34000.0	230.0	R-22	1.86425	60.0	16.7	1.0	936.0	Home
3	4JKW5	Danfoss	38000.0	230.0	R-22	2.050675	60.0	19.5	1.0	895.0	Home
4	4JKW6	Danfoss	38000.0	230.0	R-22						
5	5AGX7	Bristol	18000.0	230.0	R-22						
6	5AGX9	Bristol	22000.0	230.0	R-22						
7	6AGX9	Panasonic	6926.0	115.0	R-410A						
8	6AGY1	Panasonic	8190.0	115.0	R-410A						
9	6AGY2	Panasonic	8155.0	230.0	R-410A						

#	Name	Manufacturer	Power	Speed Number	RPM	Voltage	Current	Frequency	Price	Shaft Diameter
1	2PRA8	Genteq	0.246081	1	825.0	230.0	1.8	60.0	193.0	0.5
2	2PRD4	Genteq	0.559275	1	1075.0	460.0	2.0	60.0	258.0	0.5
3	3LU99	Dayton	0.559275	1	1075.0	460.0	2.0	60.0	218.0	0.5
4	3M265	Dayton	0.246081	1	1625.0	230.0	3.1	60.0	213.0	0.5
5	3RCT2	Century	0.6524875	1	1075.0	230.0	4.6	60.0	417.0	0.5
6	3RCT6	Century	0.7457	1	1140.0	230.0	5.6	60.0	623.0	0.625
7	4M261	Dayton	0.1237862	1	1075.0	230.0	1.3	60.0	106.0	0.5
8	4M263	Dayton	0.37285	1	825.0	230.0	3.1	60.0	216.0	0.5
9	10K094	Genteq	0.37285	2	1625.0	230.0	2.5	60.0	277.0	0.5
10	20RK82	Century	1.11855	1	1140.0	230.0	5.3	60.0	550.0	0.625
11	32NA62	Century	0.186425	1	825.0	230.0	1.2	60.0	175.0	0.5

#	Name	Total Power	EER	COP	Compressor	Compressor.Capacity	Condenser Fan Motor	Condenser Fan Motor.Power
1	29AT83_45CE98_	1.677825	10.430170011771192	3.056039813448959	29AT83 : Compressor	17500.0	45CE98 : Condenser Fan Motor	0.559275
2	29AT83_44YU91_	12.30405	1.4222959106960715	0.41673270183394895	29AT83 : Compressor	17500.0	44YU91 : Condenser Fan Motor	11.1855
3	29AT83_40PK42_	1.5659699999999999	11.175182155469136	3.2743283715524565	29AT83 : Compressor	17500.0	40PK42 : Condenser Fan Motor	0.44742
4	29AT83_32NA71_	1.4914	11.73394126324259	3.438044790130079	29AT83 : Compressor	17500.0	32NA71 : Condenser Fan Motor	0.37285
5	29AT83_32NA68_	1.677825	10.430170011771192	3.056039813448959	29AT83 : Compressor	17500.0	32NA68 : Condenser Fan Motor	0.559275
6	29AT83_32NA67_	1.4914	11.73394126324259	3.438044790130079	29AT83 : Compressor	17500.0	32NA67 : Condenser Fan Motor	0.37285
7	29AT83_32NA66_	1.364631	12.823979522669498	3.757426000142163	29AT83 : Compressor	17500.0	32NA66 : Condenser Fan Motor	0.246081
8	29AT83_32NA64_	1.4914	11.73394126324259	3.438044790130079	29AT83 : Compressor	17500.0	32NA64 : Condenser Fan Motor	0.37285



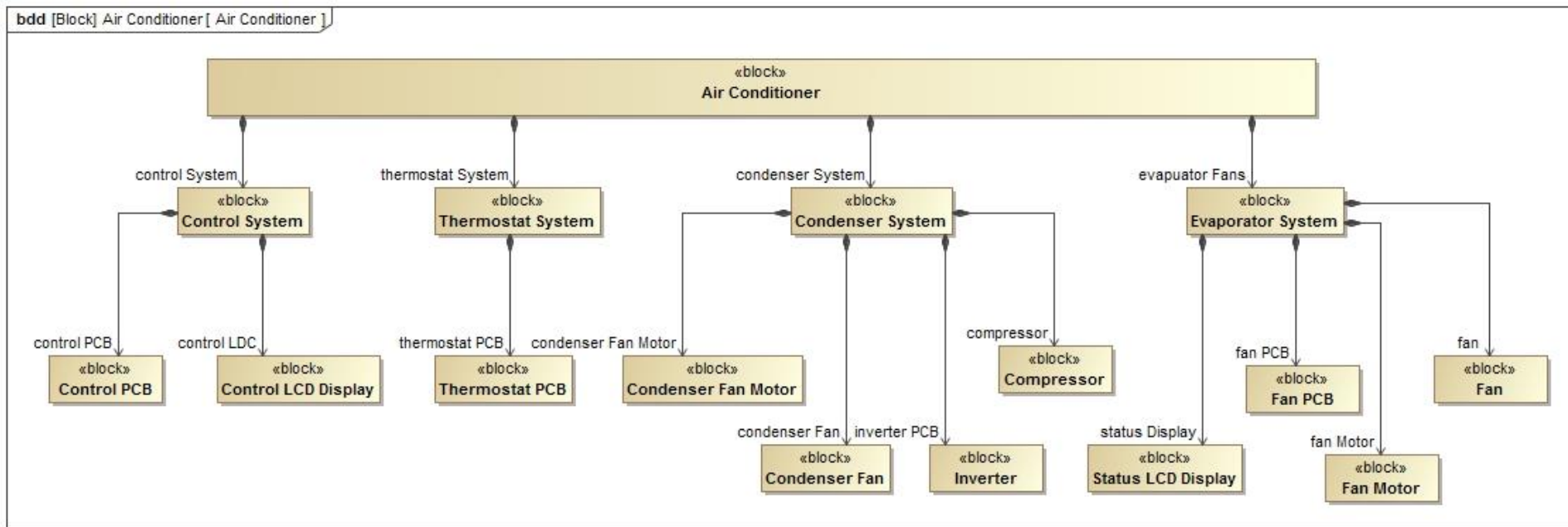
Mass, power, cost rollups

The problem



Requirement:

The total power of air conditioner shall not exceed 1.0 kW.



A typical way

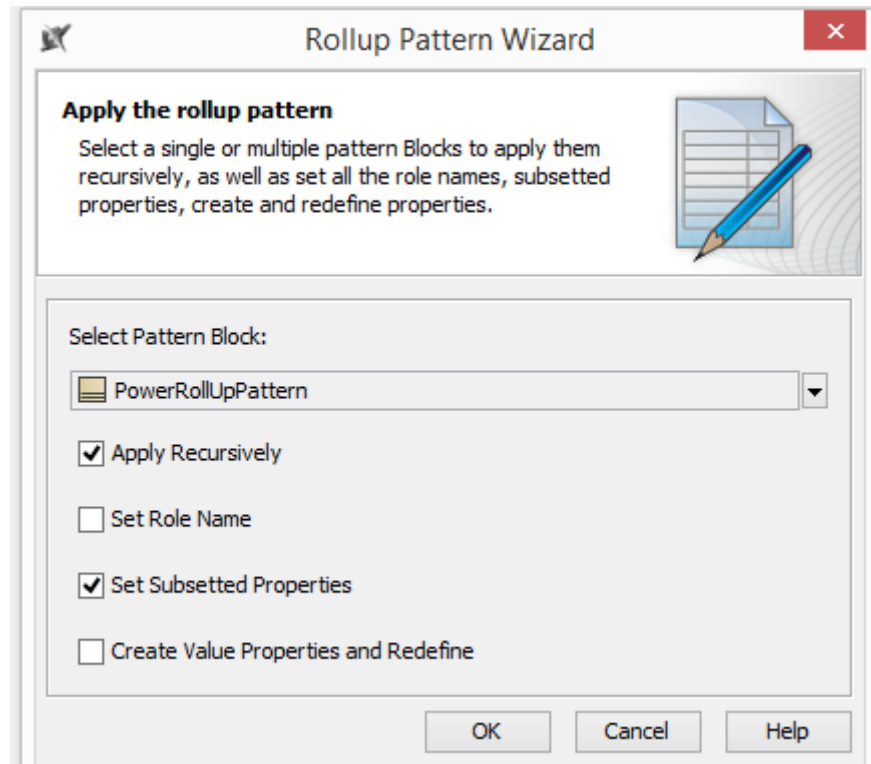
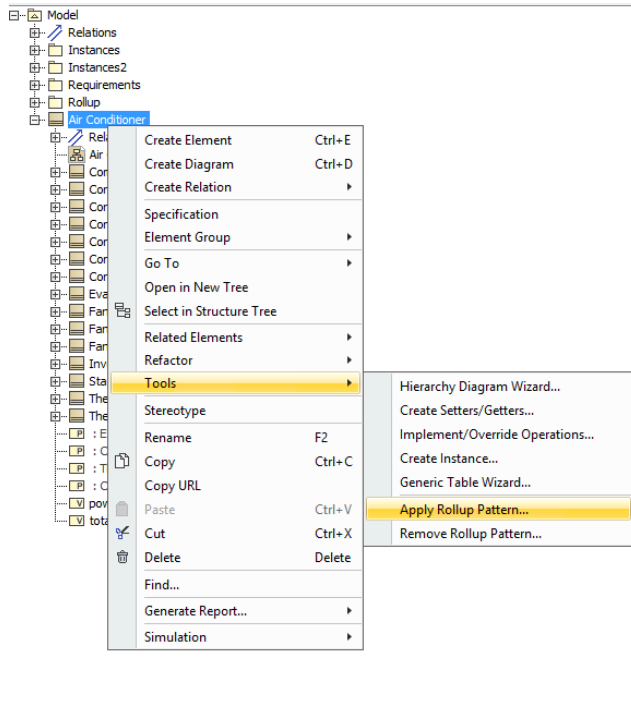


- Add property to every block in the structure
- Create Parametric diagrams for every block
- Create “sum(cost1, cost2, ...) constraint blocks
- Results
 - many hours of work
 - Polluted model
 - Hundreds of parametric diagrams
 - Need to remodel when add any new part

How could we improve that?



- Use inheritance
 - Define everything in some abstract “CostOwner”
 - Inherit parametrics
- Use subsets - say what roles are children
- Automate the process



Run Analysis



Criteria
 Classifier: PowerRollUpPattern Scope (optional): Instances2 Filter: Q

#	Name	Power : Real	Total Power : Real
1	Instances2		
2	air Conditioner	0.0	847.0
3	air Conditioner.evaporator System	0.0	122.0
4	air Conditioner.evaporator System.fan	0.0	0.0
5	air Conditioner.evaporator System.fan Motor	100.0	100.0
6	air Conditioner.evaporator System.fan PCB	10.0	10.0
7	air Conditioner.evaporator System.status LCD Display	12.0	12.0
8	air Conditioner.condenser System	0.0	650.0
9	air Conditioner.condenser System.compressor	500.0	500.0
10	air Conditioner.condenser System.inverter	50.0	50.0
11	air Conditioner.condenser System.condenser Fan	0.0	0.0

Simulation

Variables Breakpoints

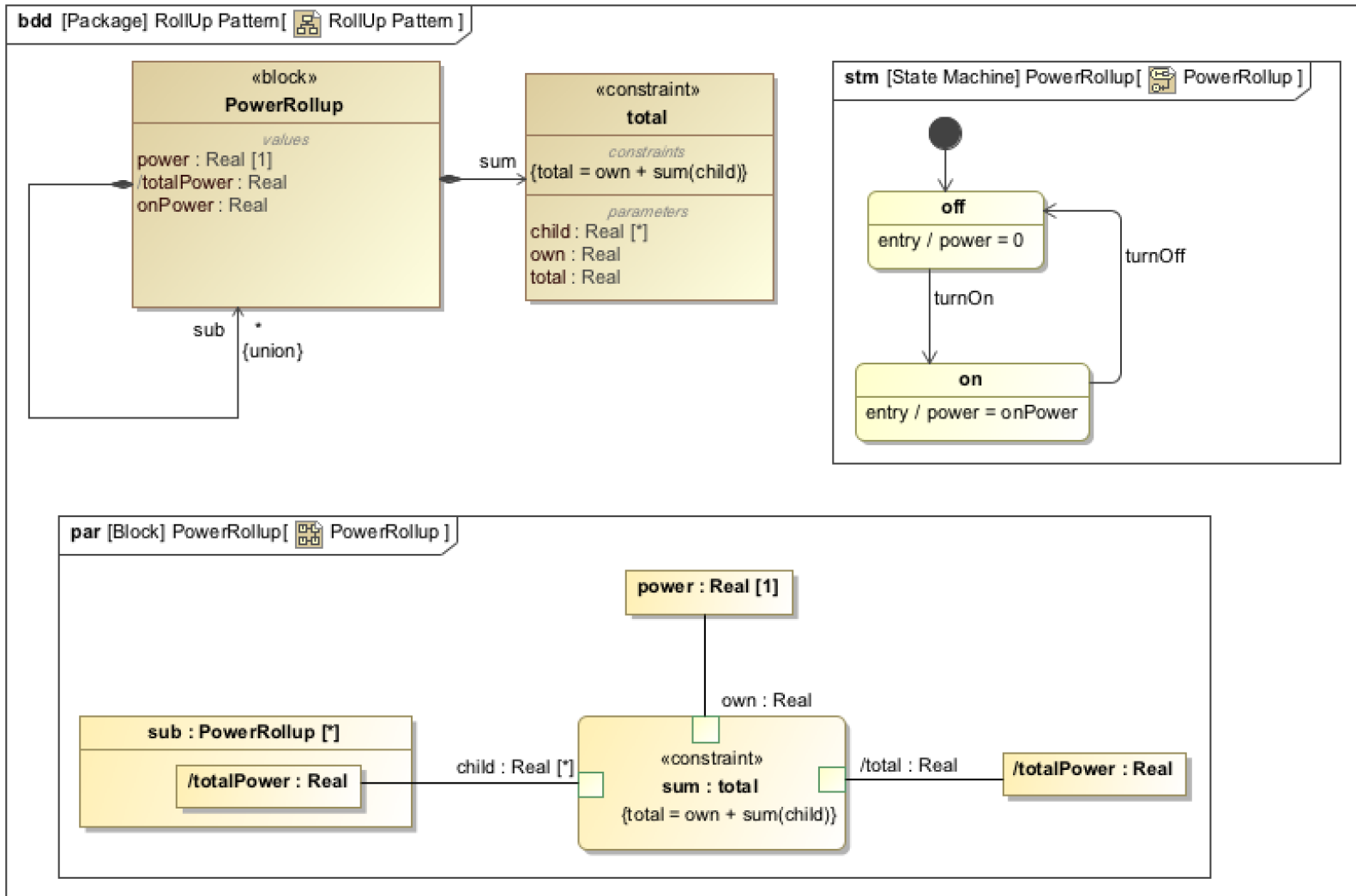
Name	Value
Air Conditioner {totalPower <= 1000.0}	air Conditioner : Air Conditioner@4626a343
power : Real	0,0000
totalPower : Real	847,0000
Evaporator System {subsets subPower}	air Conditioner.evaporator System : Evaporator System.
power : Real	0,0000
totalPower : Real	122,0000
Fan {subsets subPower}	air Conditioner.evaporator System.fan : Fan@4942b145
power : Real	0,0000
totalPower : Real	0,0000
sum : total {total = parent + sum(child)}	total@2bd06335
Fan Motor {subsets subPower}	air Conditioner.evaporator System.fan Motor : Fan Moto
power : Real	100,0000
totalPower : Real	100,0000
sum : total {total = parent + sum(child)}	total@5d93afac
Fan PCB {subsets subPower}	air Conditioner.evaporator System.fan PCB : Fan PCB@
power : Real	10,0000
totalPower : Real	10,0000
sum : total {total = parent + sum(child)}	total@47bbc3f

Simulation

Variables Breakpoints

Name	Value
Air Conditioner {totalPower <= 1000.0}	air Conditioner : Air Conditioner@4626a343
power : Real	0,0000
totalPower : Real	1147,0000
Evaporator System {subsets subPower}	air Conditioner.evaporator System : Evaporator System.
power : Real	0,0000
totalPower : Real	122,0000
Fan {subsets subPower}	air Conditioner.evaporator System.fan : Fan@4942b145
power : Real	0,0000
totalPower : Real	0,0000
sum : total {total = parent + sum(child)}	total@2bd06335
Fan Motor {subsets subPower}	air Conditioner.evaporator System.fan Motor : Fan Moto
power : Real	100,0000
totalPower : Real	100,0000
sum : total {total = parent + sum(child)}	total@5d93afac
Fan PCB {subsets subPower}	air Conditioner.evaporator System.fan PCB : Fan PCB@
power : Real	10,0000
totalPower : Real	10,0000
sum : total {total = parent + sum(child)}	total@47bbc3f
Status LCD Display {subsets subPower}	air Conditioner.evaporator System.status LCD Display :
power : Real	12,0000
totalPower : Real	12,0000
sum : total {total = parent + sum(child)}	total@69686fd8
sum : total {total = parent + sum(child)}	total@69d7ed2e

Dynamic rollup



Dynamic rollup



Simulation

Trigger: turnOff

Variables Breakpoints

Name	Value
Air Conditioner [ON] {totalPower <= 1000.0}	air Conditioner : Air Conditioner @1e633db8
onPower : Real	0,0000
power : Real	0,0000
standByPower : Real	0,0000
totalPower : Real	916,0000
Evaporator System [ON] {subsets subPower}	air Conditioner.evaporator System : Evaporator System...
Condenser System [ON] {subsets subPower}	air Conditioner.condenser System : Condenser System@...
Thermostat System [ON] {subsets subPower}	air Conditioner.thermostat System : Thermostat System...
Control System [ON] {subsets subPower}	air Conditioner.control System : Control System@6eafb8a
sum : total {total = parent + sum(child)}	total@49679965

Simulation

Trigger: turnOff

Variables Breakpoints

Name	Value
Air Conditioner [StandBy] {totalPower <= 1000.0}	air Conditioner : Air Conditioner @124fc593
onPower : Real	0,0000
power : Real	0,0000
standByPower : Real	0,0000
totalPower : Real	24,0000
Evaporator System [StandBy] {subsets sub...}	air Conditioner.evaporator System : Evaporator System...
Condenser System [StandBy] {subsets sub...}	air Conditioner.condenser System : Condenser System@...
Thermostat System [StandBy] {subsets su...}	air Conditioner.thermostat System : Thermostat System...
Control System [StandBy] {subsets subPo...}	air Conditioner.control System : Control System@9add509
sum : total {total = parent + sum(child)}	total@236e290

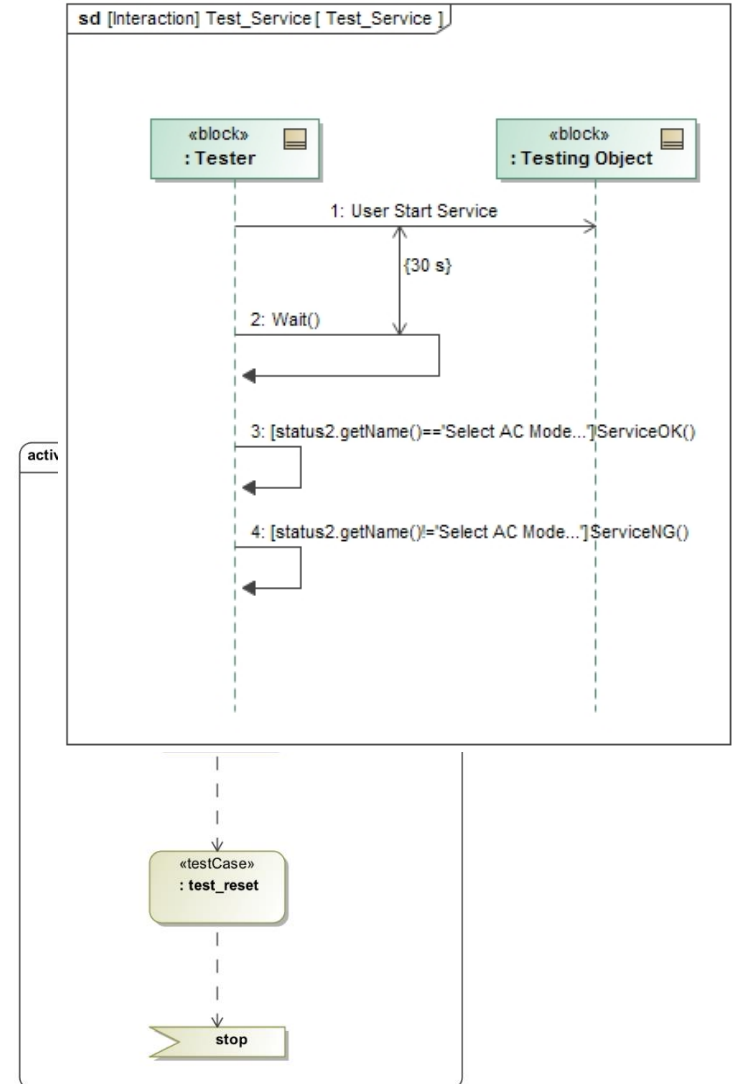
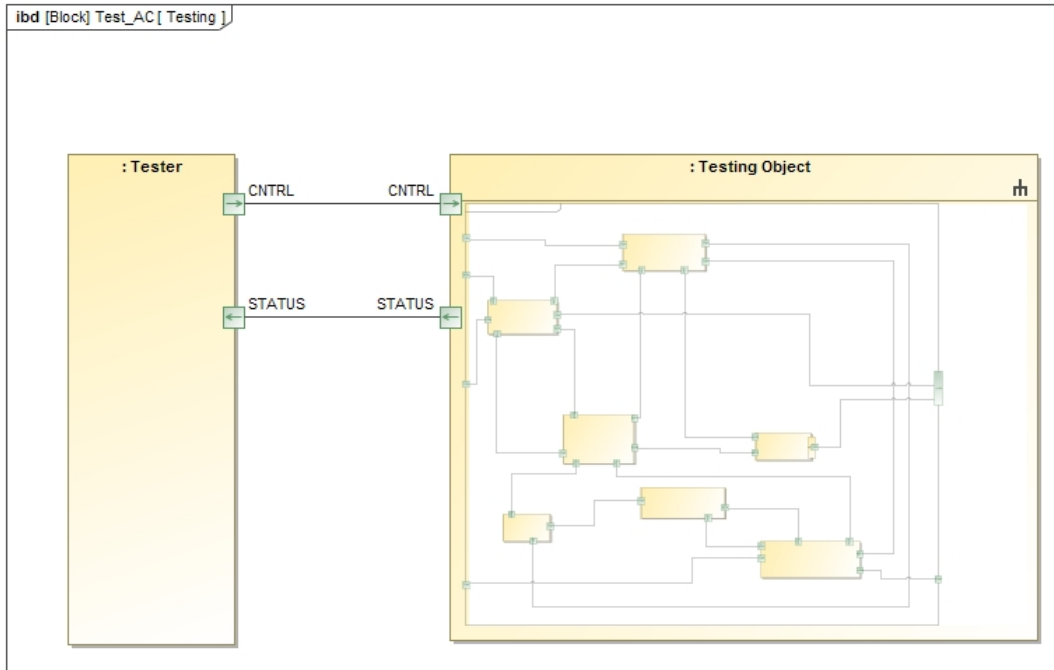
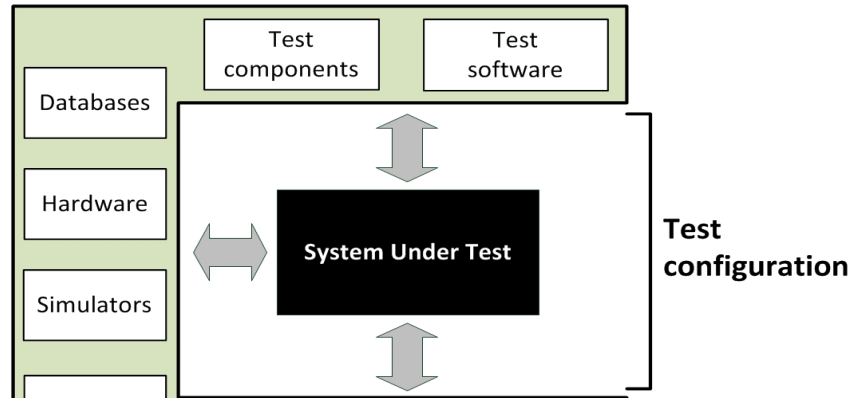


Model based testing

Model-based testing in SysML



Test environment



Testing report (Instance Table + Excel)



Run Test

Instance Table

Add New Add Existing Delete Export

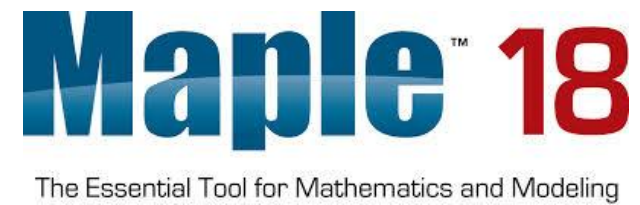
#	Name	Testcase1 verdict	Testcase2 verdict
1	Test at 2014.05.02 19.36	pass	pass
2	Test at 2014.05.02 19.49	pass	pass
3	Test at 2014.05.03 09.28	fail	fail
4	Test at 2014.05.03 09.33	pass	fail

	A	B	C
1			
2	Test	testcase1 verdict	testcase2 verdict
3	Test at 2014.05.02 19.49	pass	pass
4	Test at 2014.05.02 19.36	pass	pass
5	Test at 2014.05.03 09.28	fail	fail
6	Test at 2014.05.03 09.33	pass	fail

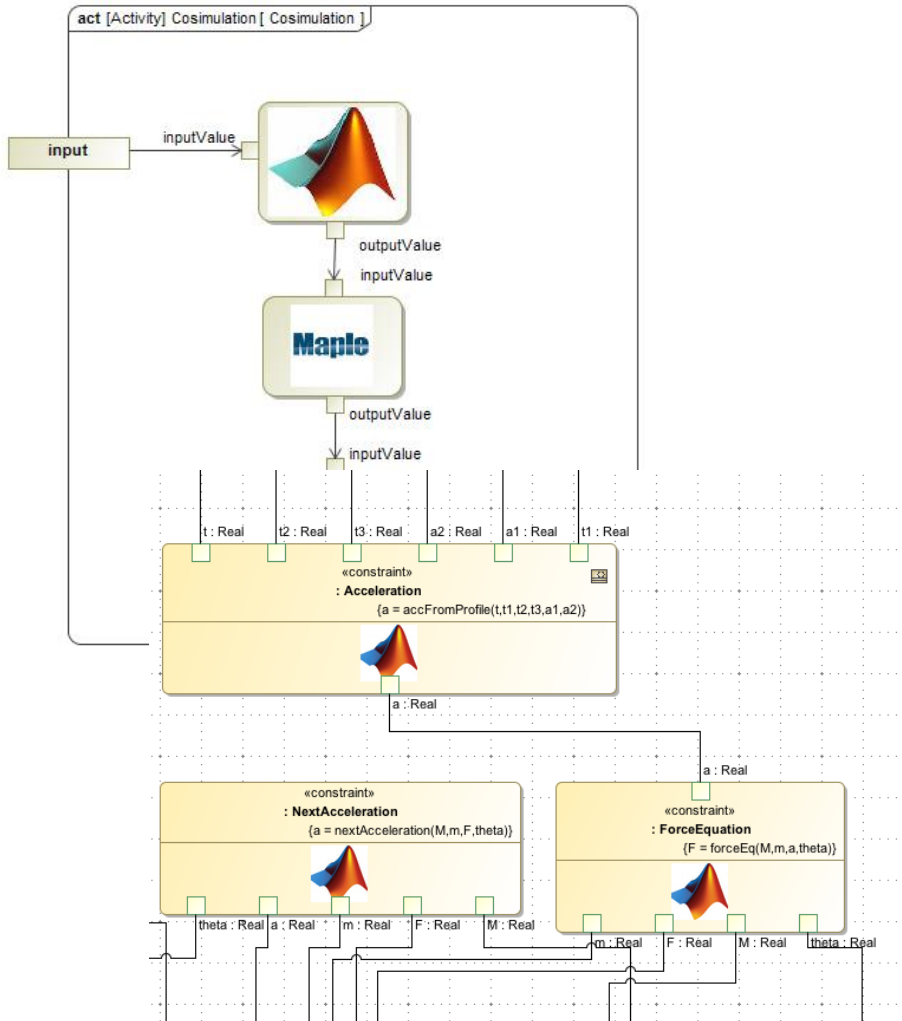


Integration of Analytics Model

Integration of Analytics Models

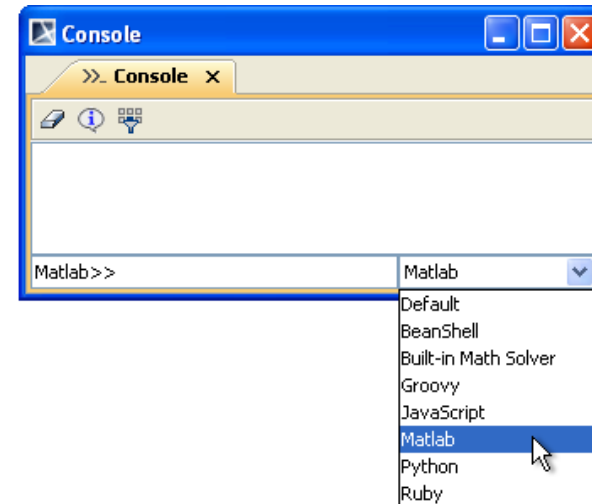


Cameo Simulation Toolkit



Math engines

- Matlab/Simulink
- Mathematica
- Maple
- Open Modelica
- Scripting
 - Javascript
 - Python
 - Groovy
 - Ruby



Co-simulation: Invoking Simulink Model



act [Activity] Call Sim [Call Sim]

```
h = sim('simple', [1 2])
```

h =
1.0000
1.0000
1.0002
1.0012
1.0062
1.0262
1.0462
1.0662
1.0862
1.1062
1.1262

simple

File Edit View Display Diagram Simulation Analysis Code Tools

simple

Step Transfer Fcn Scope

Ready 100% ode45

Scripting, external APIs



Opaque Behavior - setOutputState

Specification of Opaque Behavior properties
Specify properties of the selected Opaque Behavior in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

History: `setOutputState(port : Integer=0, power : Integer=0, mode : Int...`

setOutputState

Properties: All Customize

Name	setOutputState
Qualified Name	NXT_Model_Library::opaqueBehavior::coreBehavior::s...
Owner	coreBehavior [NXT_Model_Library::opaqueBehavior]
Applied Stereotype	importClass(Packages.com.nomagic.magicdraw.legomindstorms.NXTManager);
Body	NXTManager.startRunMoter(port,power,mode,regulationMode,turnRatio,runState,tachoLimit);
Language	

Body
Specifies the behavior in one or more languages.

Q- Type here to filter properties

Close Back Forward Help

package User Interface [ShooterBot Interface]

NXT Controller

Status

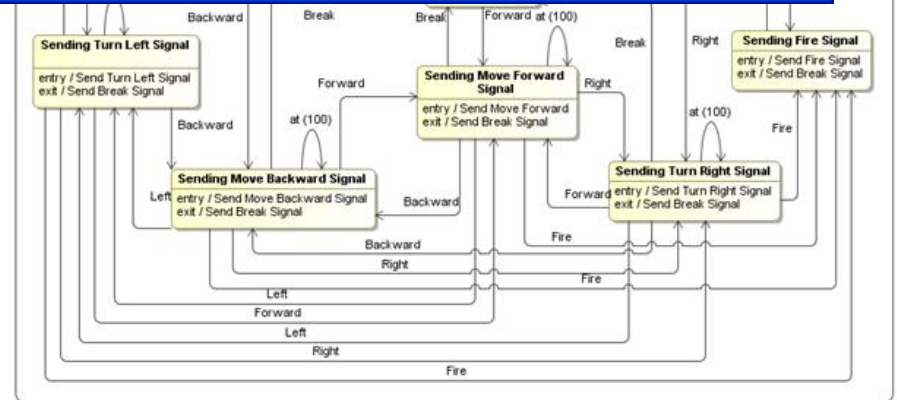
Sensor
Nearest Object : NearestIObjct

Forward

Left Break Right Fire

Backward

Connect (USB) Connect (Bluetooth) Disconnect

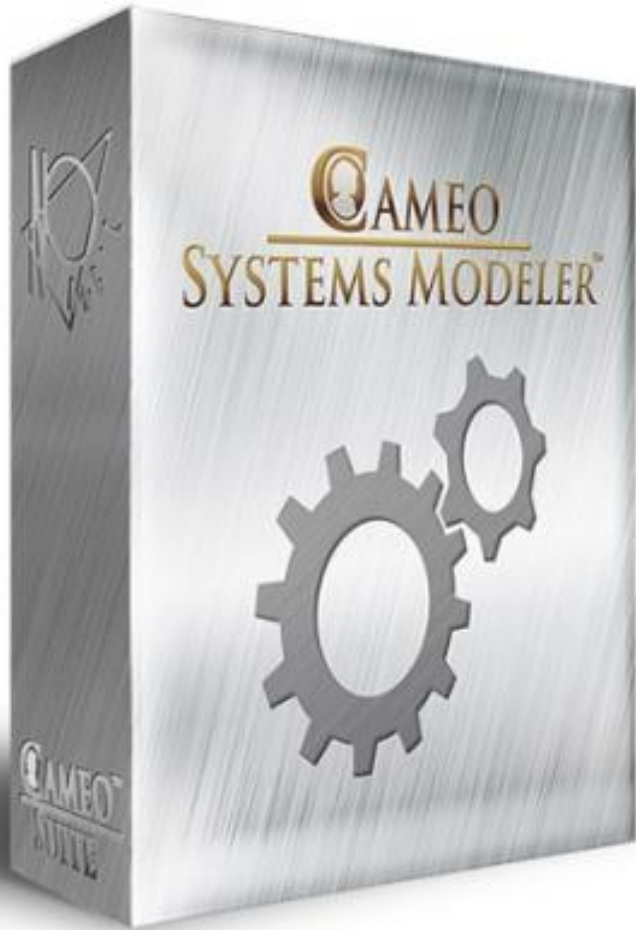




With simulation you can:

- Enhance user understanding by using animation and debugging
- Define your system behavior by using activity and states diagrams
- Create user interface mockup
- Do engineering analysis by using rollups, trade studies, system testing
- Integrate with other analytic tools

Thank You !



+

