intercax

Integrating models and apps across the “Vee” for MBSE

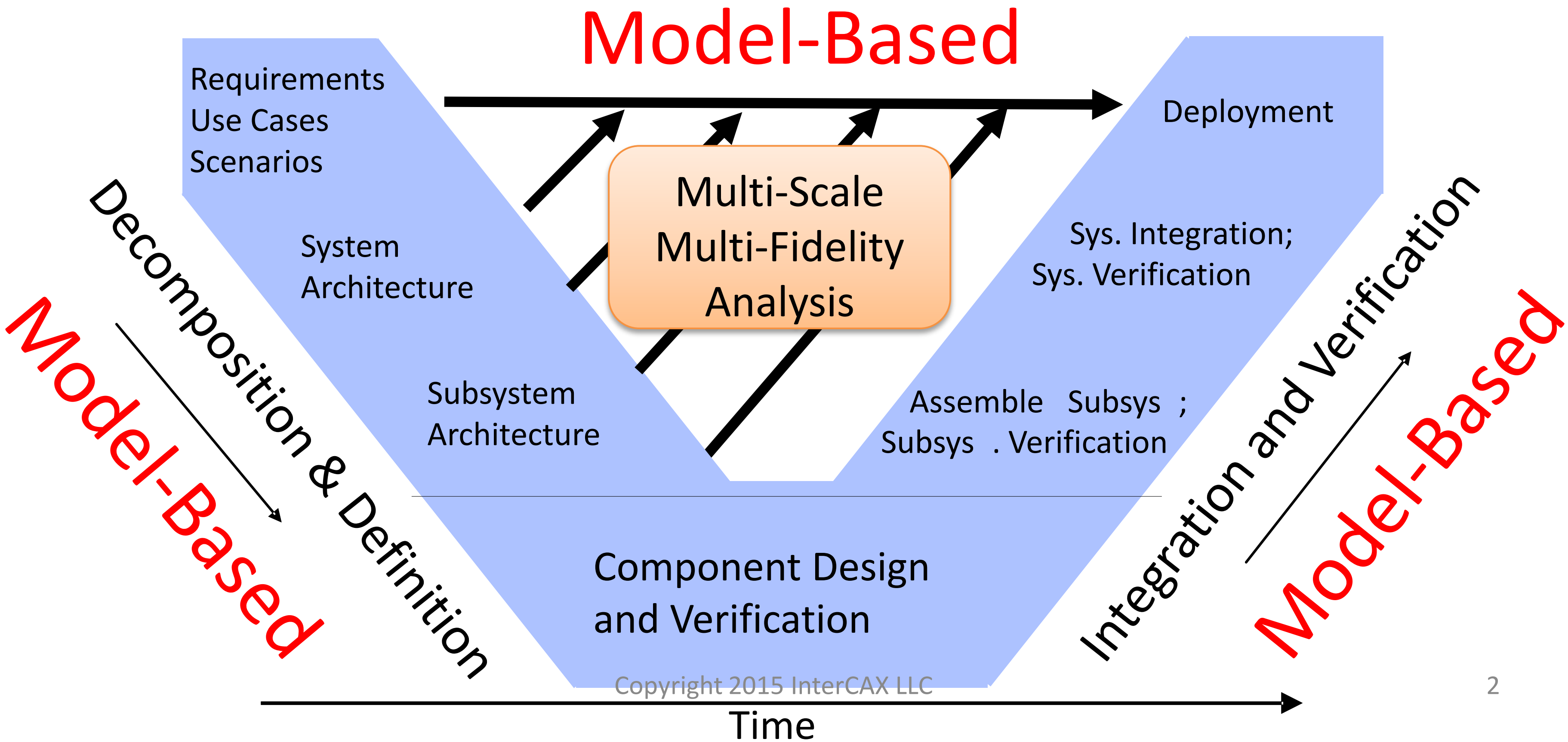
Aerospace Breakout Session
INCOSE IW, Torrance, CA
Jan 24-25, 2015

Manas Bajaj, PhD
Chief Systems Officer
manas@intercax.com
www.intercax.com

Model-Based Systems Engineering

Reinventing the traditional “Vee”

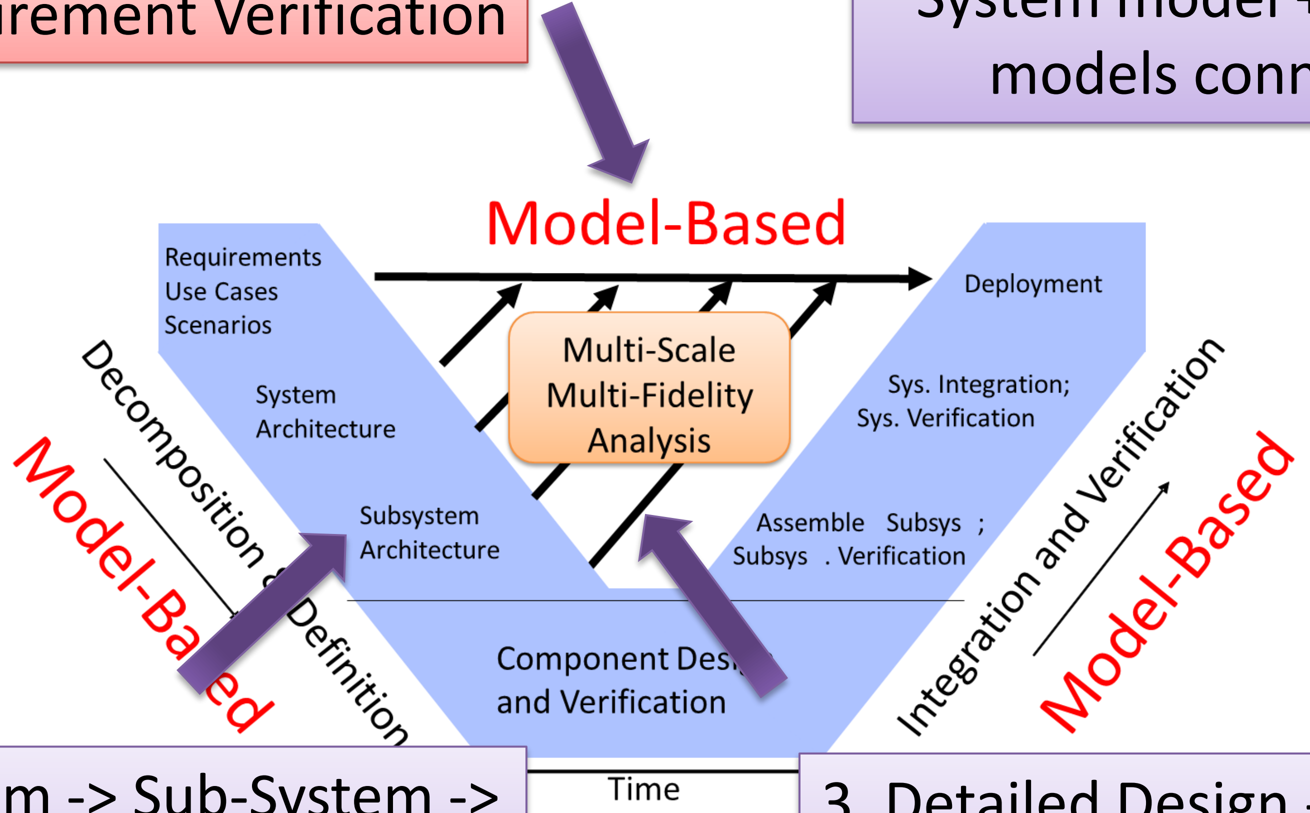
- Breaking big Vee into mini Vees – continuous verification
- Model-based communication within and across the Vee
- Traceability – single unified system model (federation)



How do we do this?

1. Design -> Analysis -> Requirement Verification

4. Total System Model – System model + domain models connected



2. System -> Sub-System -> Components (PLM/CAD, Software)

3. Detailed Design -> Analysis (Simulink/Modelica, FEA, CFD)

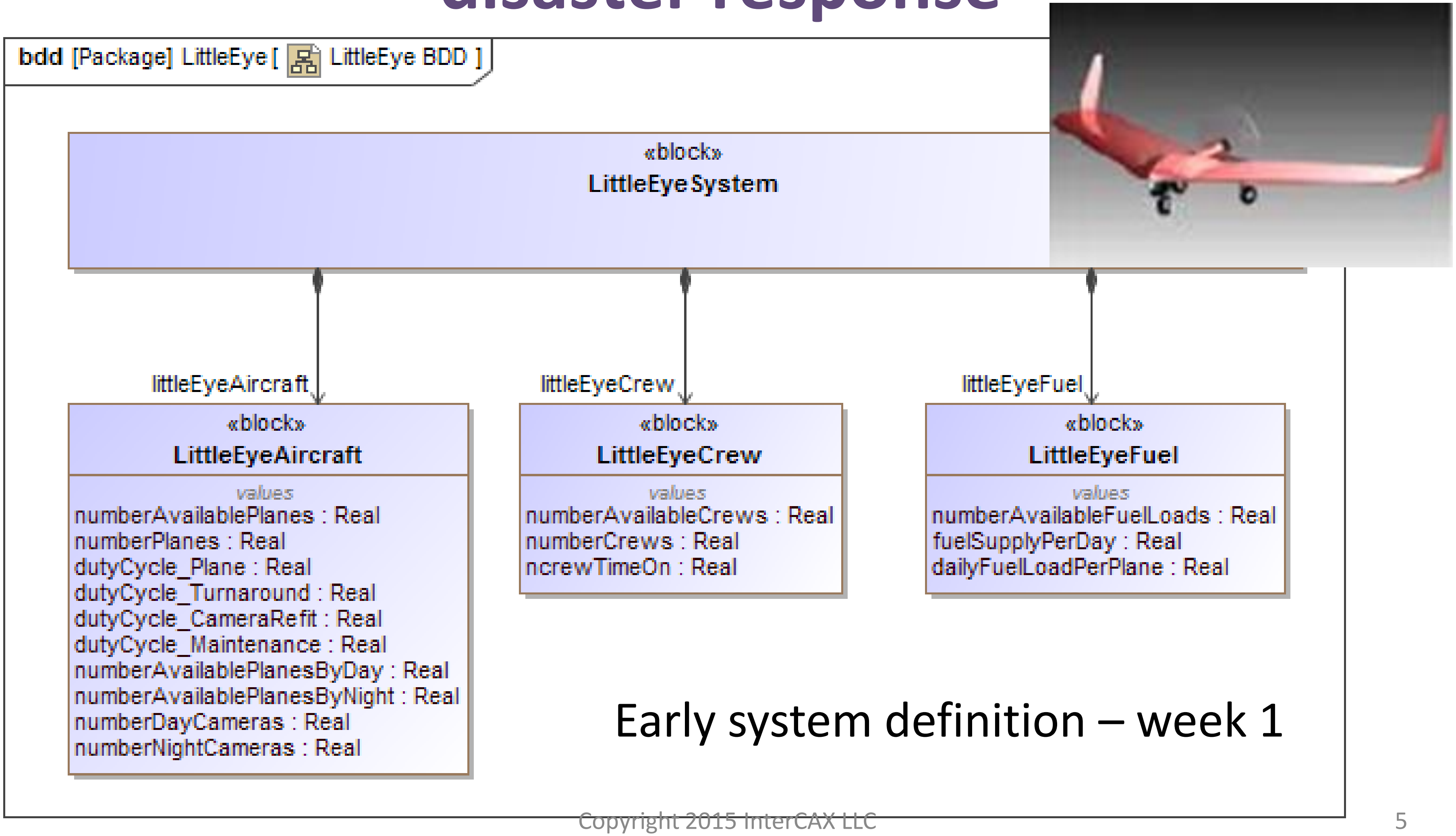
Design > Analysis > Verification

Jumping across the Vee from the start

- System architecture defined in SysML
- Block structure to represent the system
- Parametric models to compute system MoEs
- Relate requirements to MoEs
- Execute parametrics to compute MoEs and perform early trades (e.g. cost vs performance)
- Use results to verify requirements
- Alternatively, use activity / state machine models and execute them

Example 1 – LittleEye

UAV squadron for monitoring, surveillance & disaster response



bdd [Package] LittleEye [LittleEye BDD]

«block»
LittleEyeSystem

littleEyeAircraft

«block»
LittleEyeAircraft

values
numberAvailablePlanes : Real
numberPlanes : Real
dutyCycle_Plane : Real
dutyCycle_Turnaround : Real
dutyCycle_CameraRefit : Real
dutyCycle_Maintenance : Real
numberAvailablePlanesByDay : Real
numberAvailablePlanesByNight : Real
numberDayCameras : Real
numberNightCameras : Real

littleEyeCrew

«block»
LittleEyeCrew

values
numberAvailableCrews : Real
numberCrews : Real
ncrewTimeOn : Real

littleEyeFuel

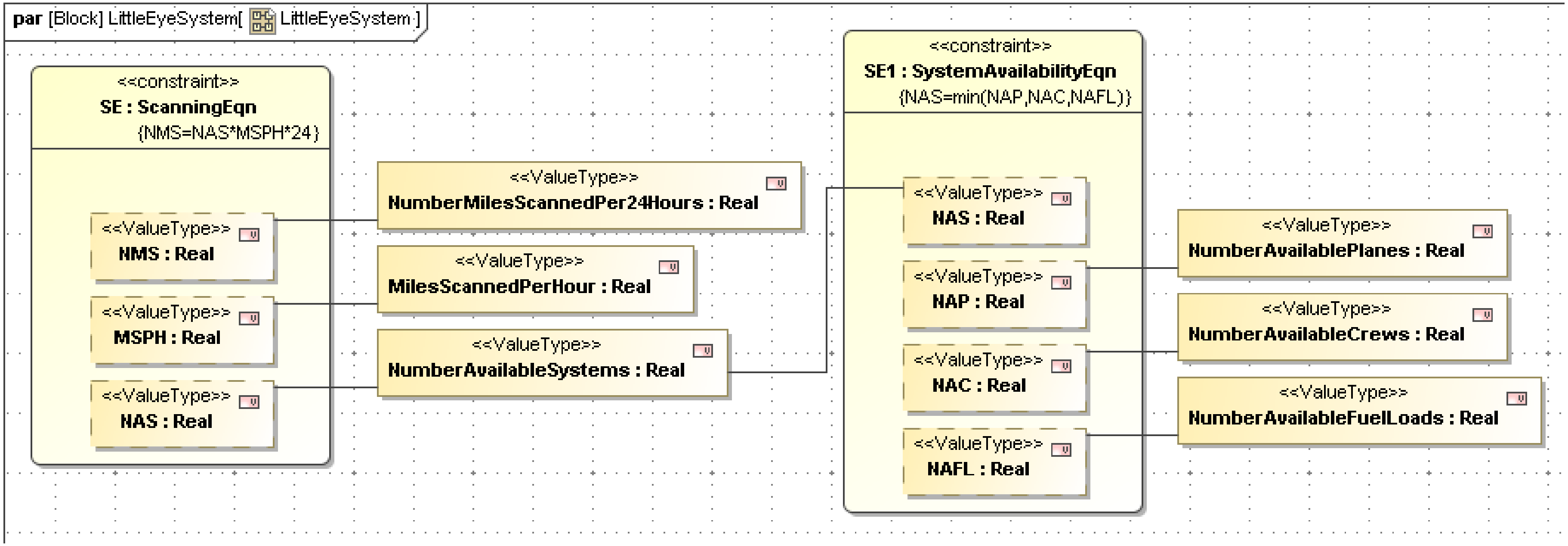
«block»
LittleEyeFuel

values
numberAvailableFuelLoads : Real
fuelSupplyPerDay : Real
dailyFuelLoadPerPlane : Real

Early system definition – week 1

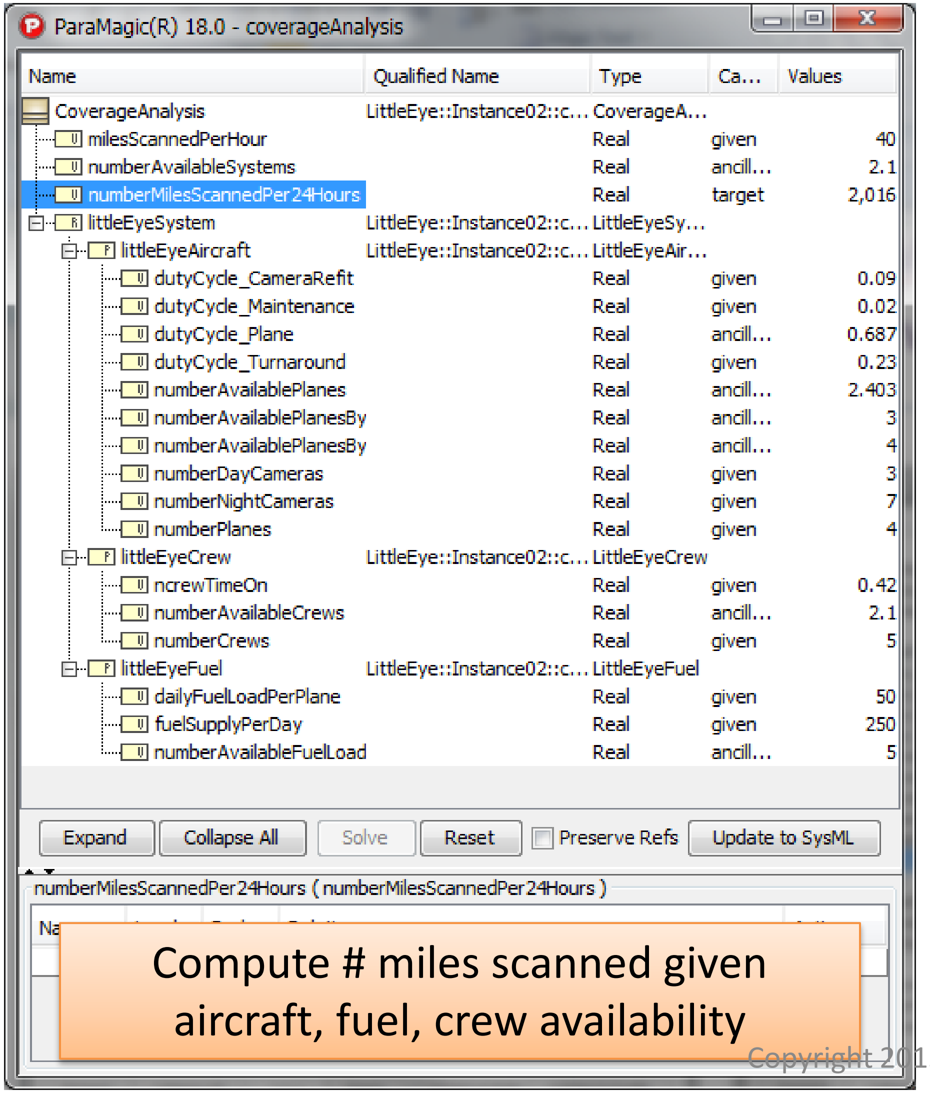
Example 1 – LittleEye (cont.)

Parametric models to relate MoEs to system definition, e.g. # miles scanned / 24 hrs and aircraft, crew, fuel availability



Example 1 – LittleEye (cont.)

Solve parametric models and verify requirements



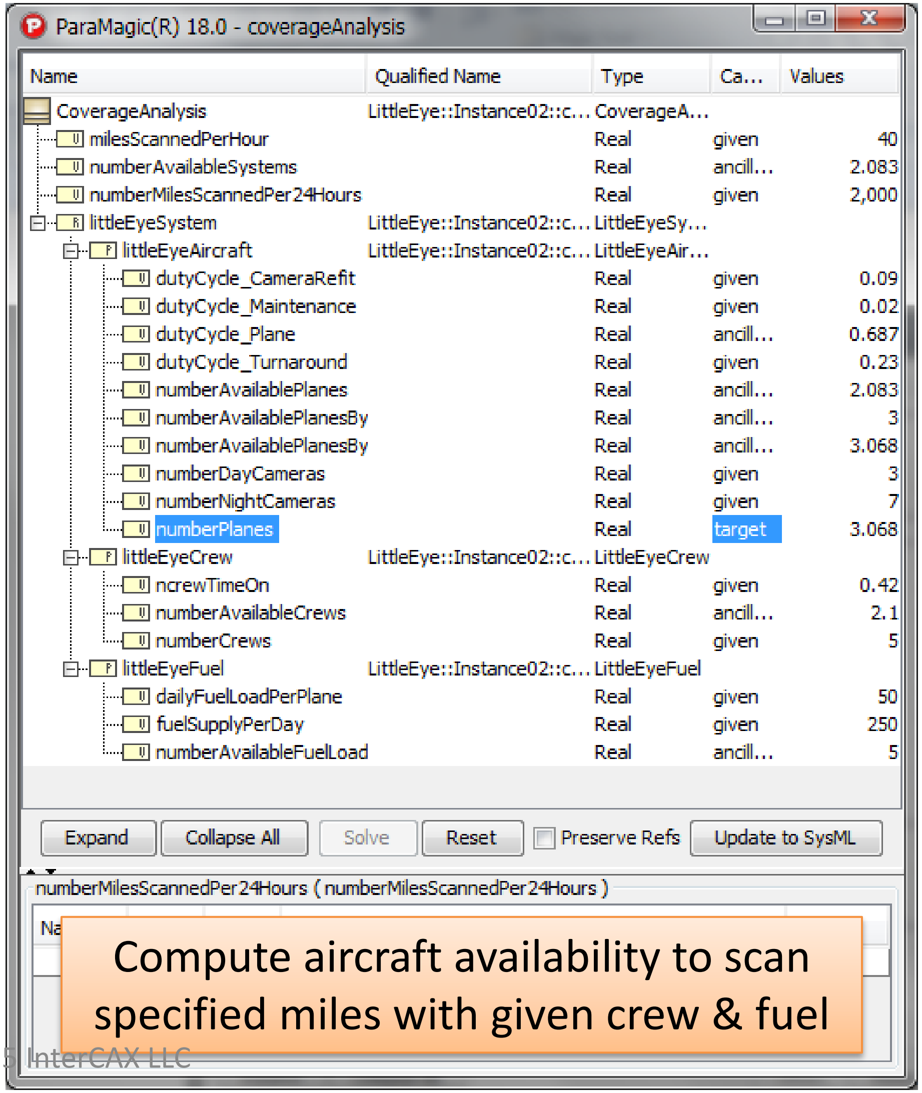
ParaMagic(R) 18.0 - coverageAnalysis

Name	Qualified Name	Type	Ca...	Values
CoverageAnalysis	LittleEye::Instance02::c...	CoverageA...		
milesScannedPerHour		Real	given	40
numberAvailableSystems		Real	ancil...	2.1
numberMilesScannedPer24Hours		Real	target	2,016
littleEyeSystem	LittleEye::Instance02::c...	LittleEyeSy...		
littleEyeAircraft	LittleEye::Instance02::c...	LittleEyeAir...		
dutyCycle_CameraRefit		Real	given	0.09
dutyCycle_Maintenance		Real	given	0.02
dutyCycle_Plane		Real	ancil...	0.687
dutyCycle_Turnaround		Real	given	0.23
numberAvailablePlanes		Real	ancil...	2.403
numberAvailablePlanesBy		Real	ancil...	3
numberAvailablePlanesBy		Real	ancil...	4
numberDayCameras		Real	given	3
numberNightCameras		Real	given	7
numberPlanes		Real	given	4
littleEyeCrew	LittleEye::Instance02::c...	LittleEyeCrew		
ncrewTimeOn		Real	given	0.42
numberAvailableCrews		Real	ancil...	2.1
numberCrews		Real	given	5
littleEyeFuel	LittleEye::Instance02::c...	LittleEyeFuel		
dailyFuelLoadPerPlane		Real	given	50
fuelSupplyPerDay		Real	given	250
numberAvailableFuelLoad		Real	ancil...	5

Expand | Collapse All | Solve | Reset | □ Preserve Refs | Update to SysML

numberMilesScannedPer24Hours (numberMilesScannedPer24Hours)

Compute # miles scanned given aircraft, fuel, crew availability



ParaMagic(R) 18.0 - coverageAnalysis

Name	Qualified Name	Type	Ca...	Values
CoverageAnalysis	LittleEye::Instance02::c...	CoverageA...		
milesScannedPerHour		Real	given	40
numberAvailableSystems		Real	ancil...	2.083
numberMilesScannedPer24Hours		Real	given	2,000
littleEyeSystem	LittleEye::Instance02::c...	LittleEyeSy...		
littleEyeAircraft	LittleEye::Instance02::c...	LittleEyeAir...		
dutyCycle_CameraRefit		Real	given	0.09
dutyCycle_Maintenance		Real	given	0.02
dutyCycle_Plane		Real	ancil...	0.687
dutyCycle_Turnaround		Real	given	0.23
numberAvailablePlanes		Real	ancil...	2.083
numberAvailablePlanesBy		Real	ancil...	3
numberAvailablePlanesBy		Real	ancil...	3.068
numberDayCameras		Real	given	3
numberNightCameras		Real	given	7
numberPlanes		Real	target	3.068
littleEyeCrew	LittleEye::Instance02::c...	LittleEyeCrew		
ncrewTimeOn		Real	given	0.42
numberAvailableCrews		Real	ancil...	2.1
numberCrews		Real	given	5
littleEyeFuel	LittleEye::Instance02::c...	LittleEyeFuel		
dailyFuelLoadPerPlane		Real	given	50
fuelSupplyPerDay		Real	given	250
numberAvailableFuelLoad		Real	ancil...	5

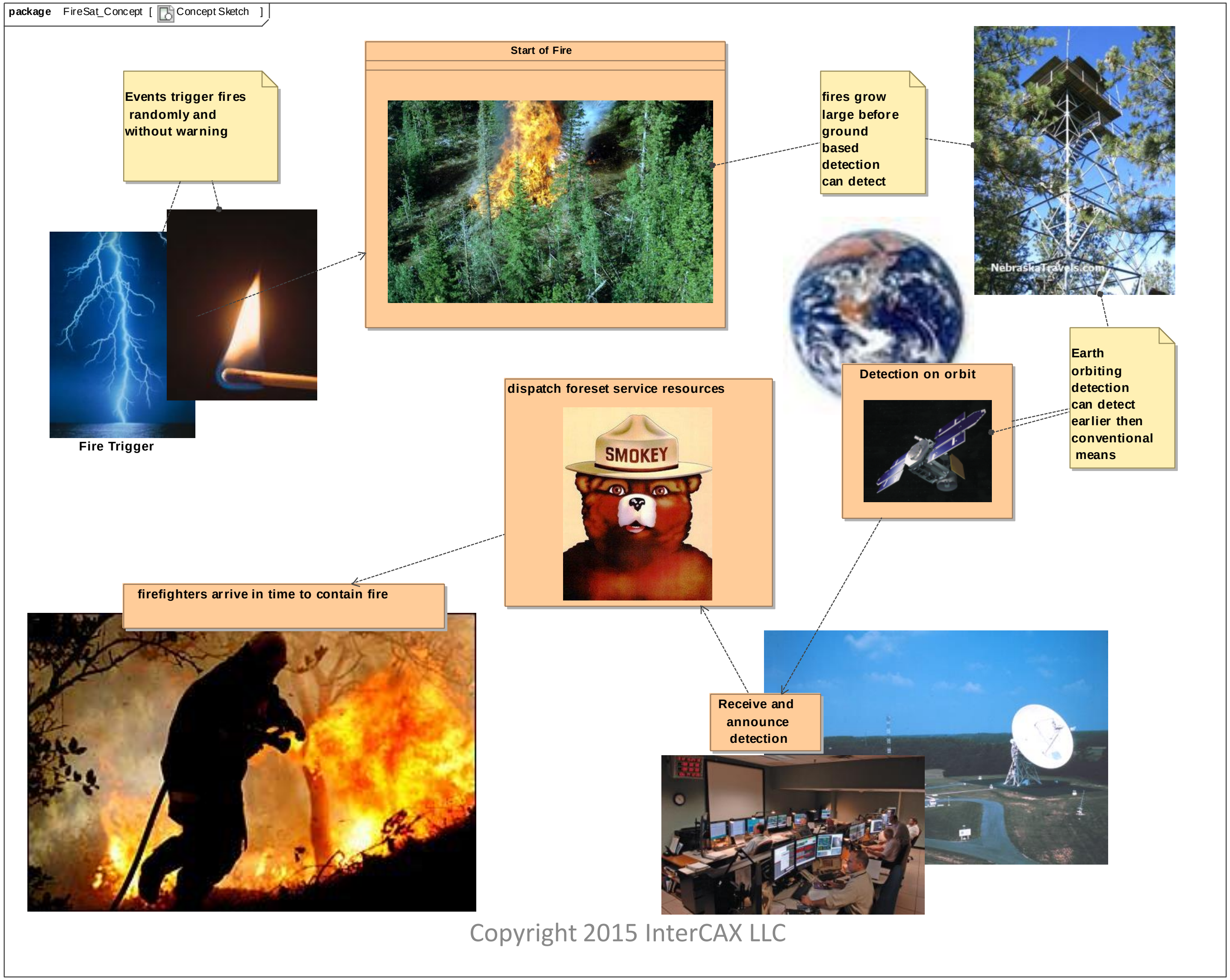
Expand | Collapse All | Solve | Reset | □ Preserve Refs | Update to SysML

numberMilesScannedPer24Hours (numberMilesScannedPer24Hours)

Compute aircraft availability to scan specified miles with given crew & fuel

Example 2 – FireSat (INCOSE SSWG)

Satellite to detect forest fires



package FireSat_Concept [Concept Sketch]

Events trigger fires randomly and without warning

Fire Trigger

Start of Fire

fires grow large before ground based detection can detect

Earth orbiting detection can detect earlier then conventional means

Detection on orbit

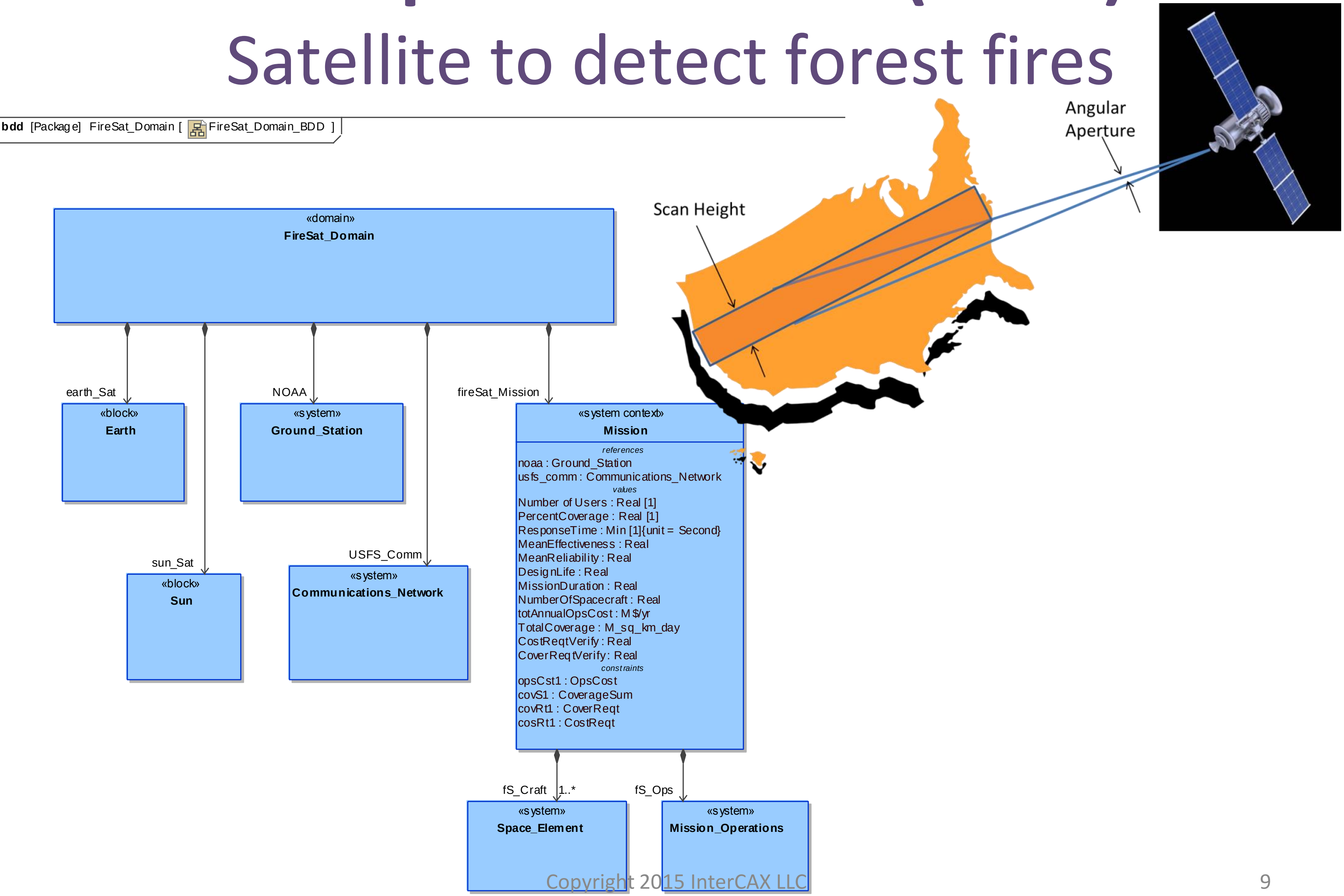
dispatch foreset service resources

firefighters arrive in time to contain fire

Receive and announce detection

Example 2 – FireSat (cont.)

Satellite to detect forest fires



bdd [Package] FireSat_Domain [FireSat_Domain_BDD]

«domain»
FireSat_Domain

earth_Sat
«block»
Earth

NOAA
«system»
Ground_Station

sun_Sat
«block»
Sun

USFS_Comm
«system»
Communications_Network

fireSat_Mission
«system context»
Mission

references
noaa : Ground_Station
usfs_comm : Communications_Network

values
Number of Users : Real [1]
PercentCoverage : Real [1]
ResponseTime : Min [1]{unit = Second}
MeanEffectiveness : Real
MeanReliability : Real
DesignLife : Real
MissionDuration : Real
NumberOfSpacecraft : Real
totAnnualOpsCost : M$/yr
TotalCoverage : M_sq_km_day
CostReqtVerify : Real
CoverReqtVerify : Real

constraints
opsCst1 : OpsCost
covS1 : CoverageSum
covRt1 : CoverReqt
cosRt1 : CostReqt

fS_Craft 1..*
«system»
Space_Element

fS_Ops
«system»
Mission_Operations

Example 2 – FireSat (cont.)

Parametrics for space element – *coverage*, *resolution*, and *cost*

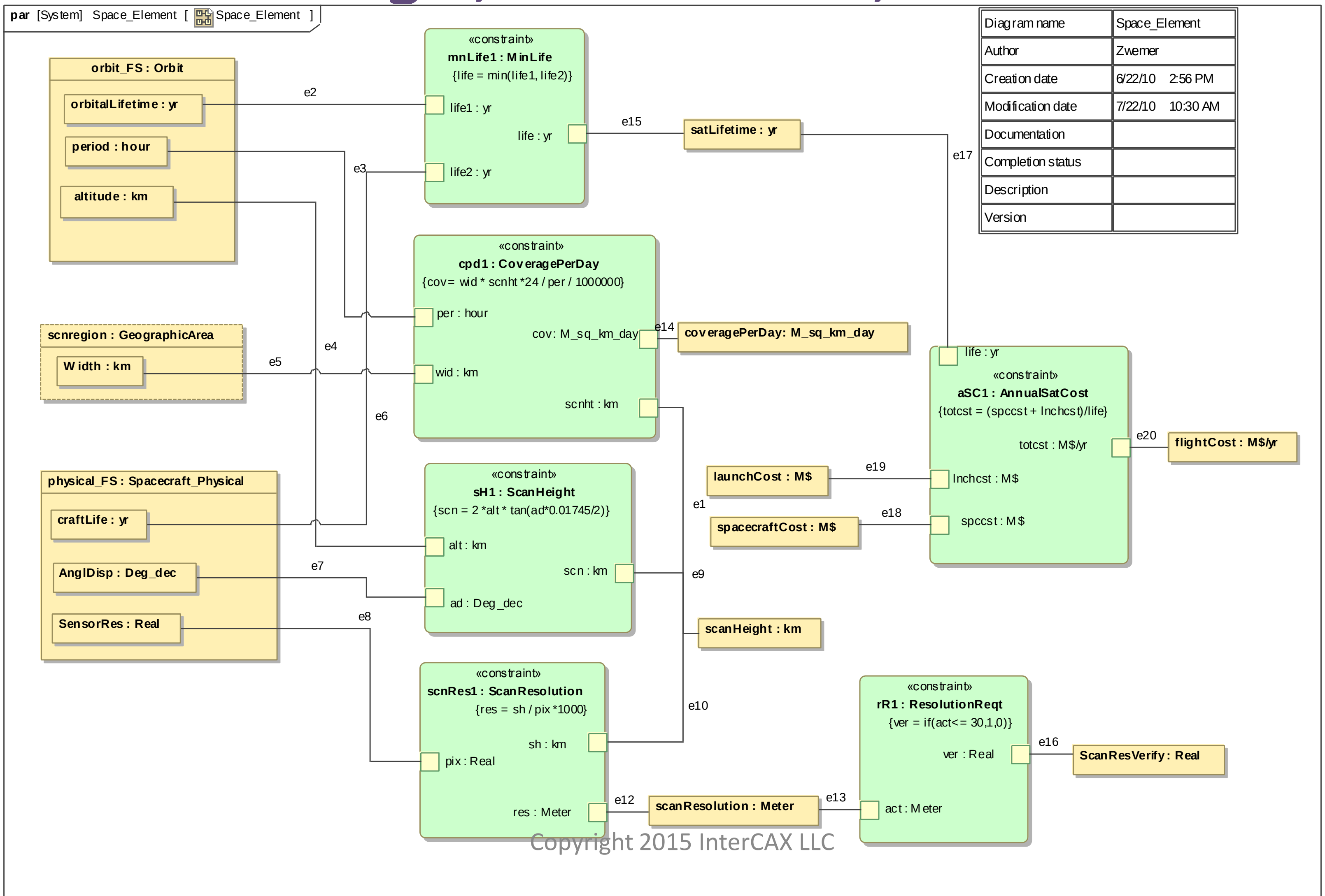


Diagram name	Space_Element
Author	Zwemer
Creation date	6/22/10 2:56 PM
Modification date	7/22/10 10:30 AM
Documentation	
Completion status	
Description	
Version	

Example 2 – FireSat (cont.)

Trade study performed using ParaMagic®

Altitude	Angular Aperture	Annual Cost		Daily Coverage		Target Resolution	
km	deg	M$/yr		M sq km/day		meters	
300	3	38.28	0	1.25	0	15.71	1
325	3	26.29	0	1.35	0	17.02	1
350	3	19.11	1	1.44	0	18.33	1
375	3	14.60	1	1.54	0	19.64	1
400	3	11.66	1	1.63	0	20.94	1
425	3	11.33	1	1.72	0	22.25	1
450	3	11.33	1	1.82	0	23.56	1
475	3	11.33	1	1.91	0	24.87	1
500	3	11.33	1	2.00	0	26.18	1
525	3	11.33	1	2.08	0	27.49	1
550	3	11.33	1	2.17	0	28.80	1
575	3	11.33	1	2.26	0	30.11	0
600	3	11.33	1	2.34	0	31.42	0

Single satellite option does not satisfy all of cost, coverage, and resolution requirements at any operational altitude

Example 2 – FireSat (cont.)

Trade study performed using ParaMagic®

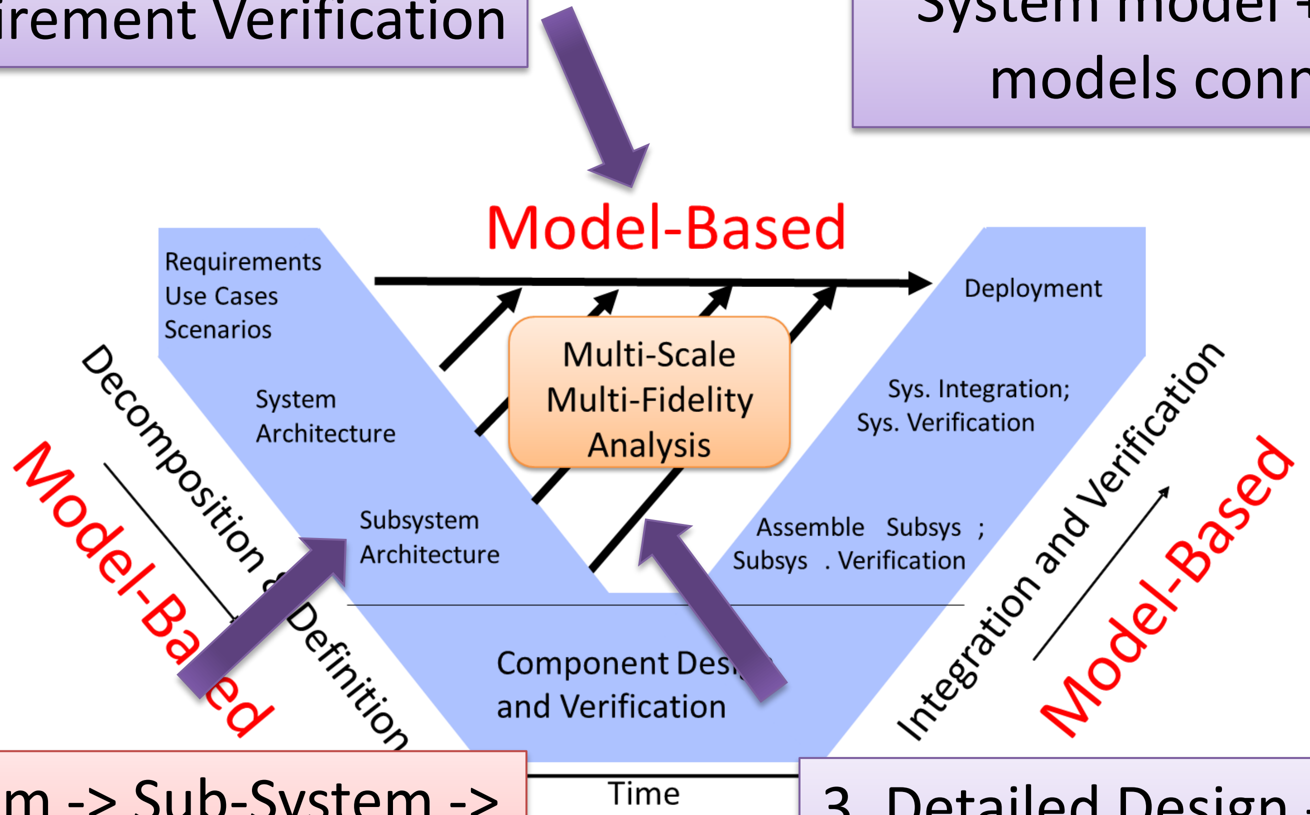
Satellite 1		Satelite 2									
	Angular		Angular			Daily		Target		Target	
Altitude	Aperture	Altitude	Aperture	Annual Cost		Coverage		Resolution		Resolution	
km	deg	km	deg	M$/yr		M sq km/day		meters		meters	
300	3	300	3	77.23	0	2.50	0	15.71	1	15.71	1
325	3	325	3	51.61	0	2.70	0	17.02	1	17.02	1
350	3	350	3	36.28	0	2.89	0	18.33	1	18.33	1
375	3	375	3	26.65	0	3.08	1	19.64	1	19.64	1
400	3	400	3	20.36	0	3.26	1	20.94	1	20.94	1
425	3	425	3	19.67	1	3.45	1	22.25	1	22.25	1
450	3	450	3	19.67	1	3.63	1	23.56	1	23.56	1
475	3	475	3	19.67	1	3.81	1	24.87	1	24.87	1
500	3	500	3	19.67	1	3.99	1	26.18	1	26.18	1
525	3	525	3	19.67	1	4.17	1	27.49	1	27.49	1
550	3	550	3	19.67	1	4.34	1	28.80	1	28.80	1
575	3	575	3	19.67	1	4.52	1	30.11	0	30.11	0
600	3	600	3	19.67	1	4.69	1	31.42	0	31.42	0

Two satellite options has operational altitude range where cost, coverage, and resolution requirements are satisfied

How do we do this?

1. Design -> Analysis -> Requirement Verification

4. Total System Model – System model + domain models connected



2. System -> Sub-System -> Components (PLM/CAD, Software)

3. Detailed Design -> Analysis (Simulink/Modelica, FEA, CFD)

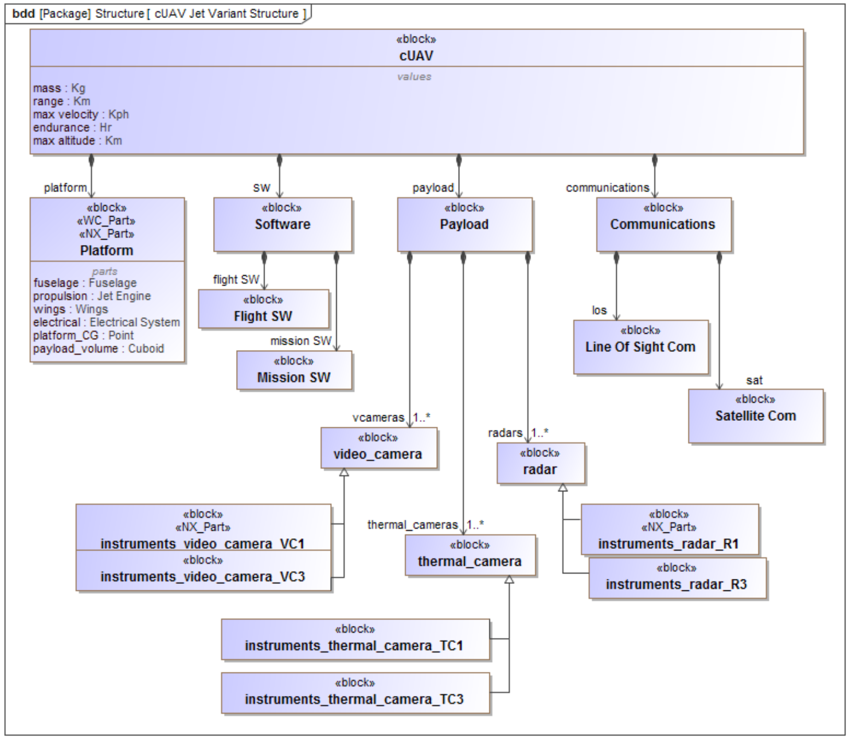
System > Sub-System > Components

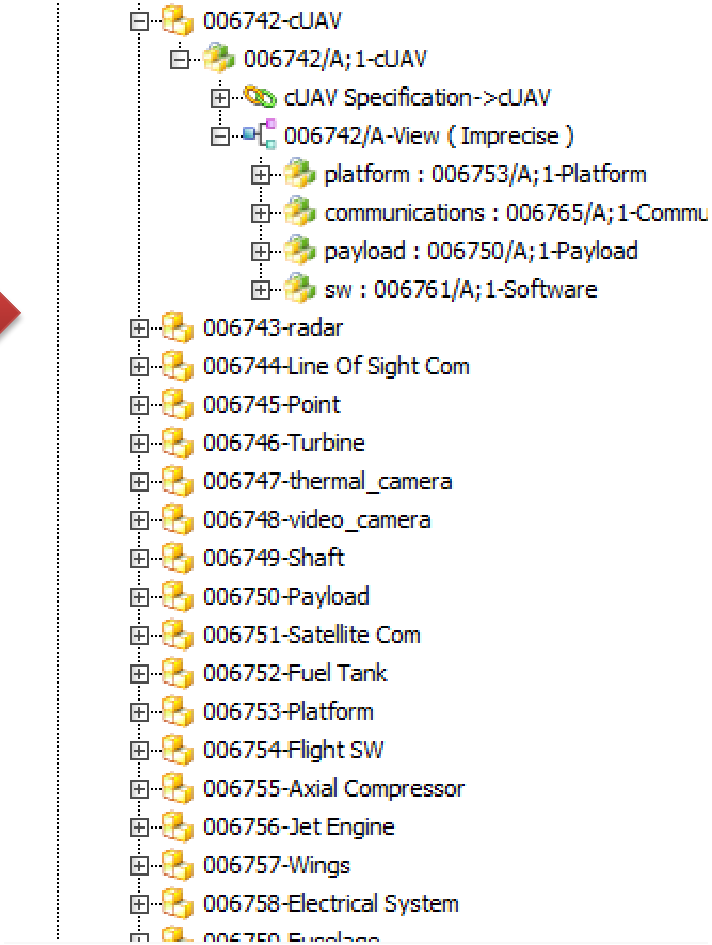
Maturing the system definition with CAD, PLM, Databases, and SW

- Different teams / organizations responsible for system, sub-system, component development
- Different modeling environments (SysML, CAD, PLM, ALM, Databases,…)
- Use cases
  - System definition (hardware) -> PLM part structure
  - System definition (software) -> ALM system
  - System/sub-system **requirements -> geometry** (CAD)
  - Reuse PLM / Databases / CAD -> system architecture

System definition > PLM

Using SLIM to seed part structure in PLM systems (Teamcenter / Windchill) from SysML



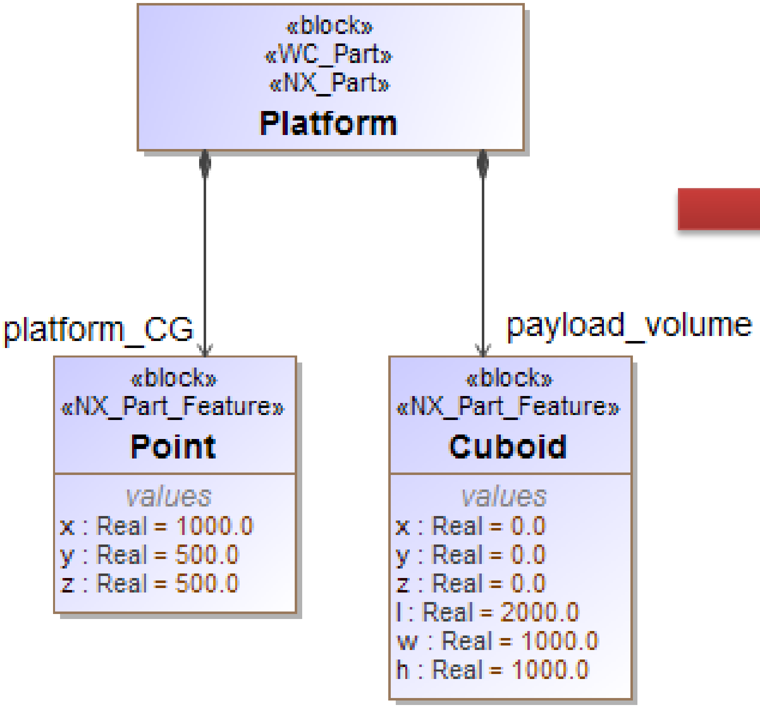


- 006742-cUAV
  - 006742/A;1-cUAV
    - cUAV Specification->cUAV
    - 006742/A-View (Imprecise)
      - platform : 006753/A;1-Platform
      - communications : 006765/A;1-Commu
      - payload : 006750/A;1-Payload
      - sw : 006761/A;1-Software
- 006743-radar
- 006744-Line Of Sight Com
- 006745-Point
- 006746-Turbine
- 006747-thermal_camera
- 006748-video_camera
- 006749-Shaft
- 006750-Payload
- 006751-Satellite Com
- 006752-Fuel Tank
- 006753-Platform
- 006754-Flight SW
- 006755-Axial Compressor
- 006756-Jet Engine
- 006757-Wings
- 006758-Electrical System
- 006759-Fuselage

Seed part structure in PLM from SysML

System definition > PLM

Using SLIM to seed system / sub-system requirements from SysML as geometry in CAD tools (e.g. NX)



«block»
«WC_Part»
«NX_Part»
Platform

platform_CG

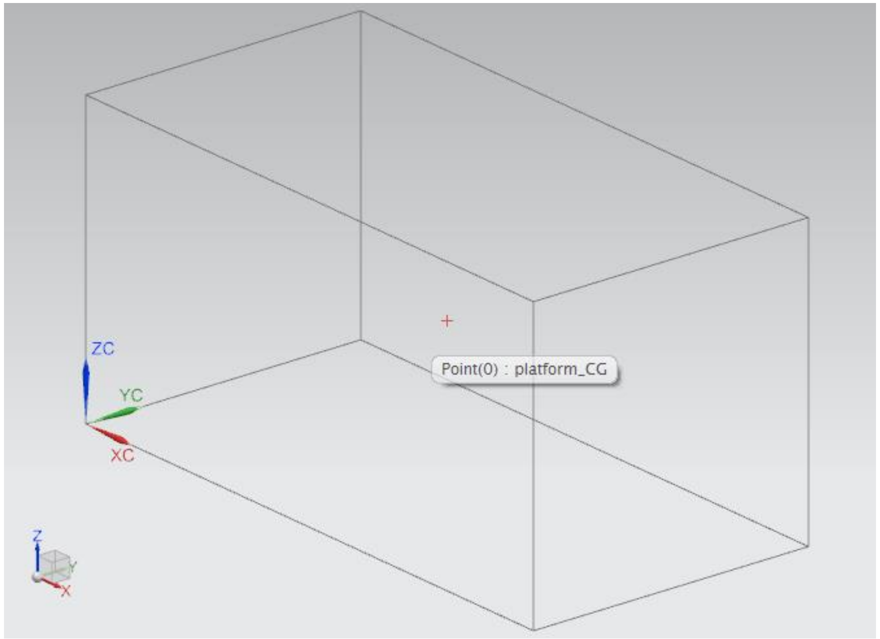
payload_volume

«block»
«NX_Part_Feature»
Point

values
x : Real = 1000.0
y : Real = 500.0
z : Real = 500.0

«block»
«NX_Part_Feature»
Cuboid

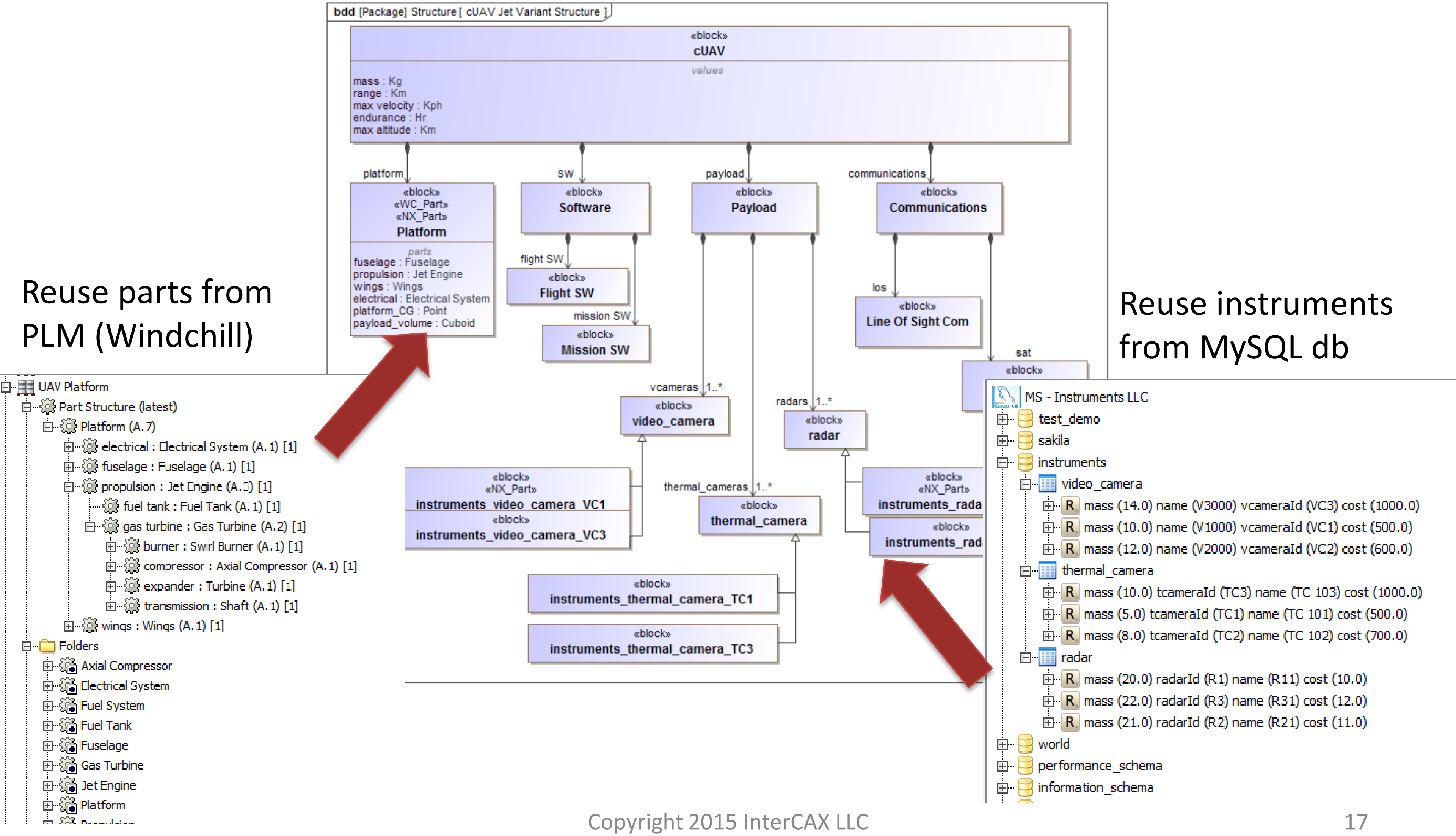
values
x : Real = 0.0
y : Real = 0.0
z : Real = 0.0
l : Real = 2000.0
w : Real = 1000.0
h : Real = 1000.0



Seed spatial requirements, such as CG, bounding boxes, keep-out zones defined in SysML seeded as geometry in CAD tool (NX shown here)

PLM/CAD/Database -> System architecture

Using SLIM to reuse existing CAD/PLM/Database artifacts to compose system architecture



Reuse parts from PLM (Windchill)

- UAV Platform
  - Part Structure (latest)
    - Platform (A.7)
      - electrical : Electrical System (A.1) [1]
      - fuselage : Fuselage (A.1) [1]
      - propulsion : Jet Engine (A.3) [1]
        - fuel tank : Fuel Tank (A.1) [1]
        - gas turbine : Gas Turbine (A.2) [1]
          - burner : Swirl Burner (A.1) [1]
          - compressor : Axial Compressor (A.1) [1]
          - expander : Turbine (A.1) [1]
          - transmission : Shaft (A.1) [1]
      - wings : Wings (A.1) [1]
  - Folders
    - Axial Compressor
    - Electrical System
    - Fuel System
    - Fuel Tank
    - Fuselage
    - Gas Turbine
    - Jet Engine
    - Platform
    - Propulsion

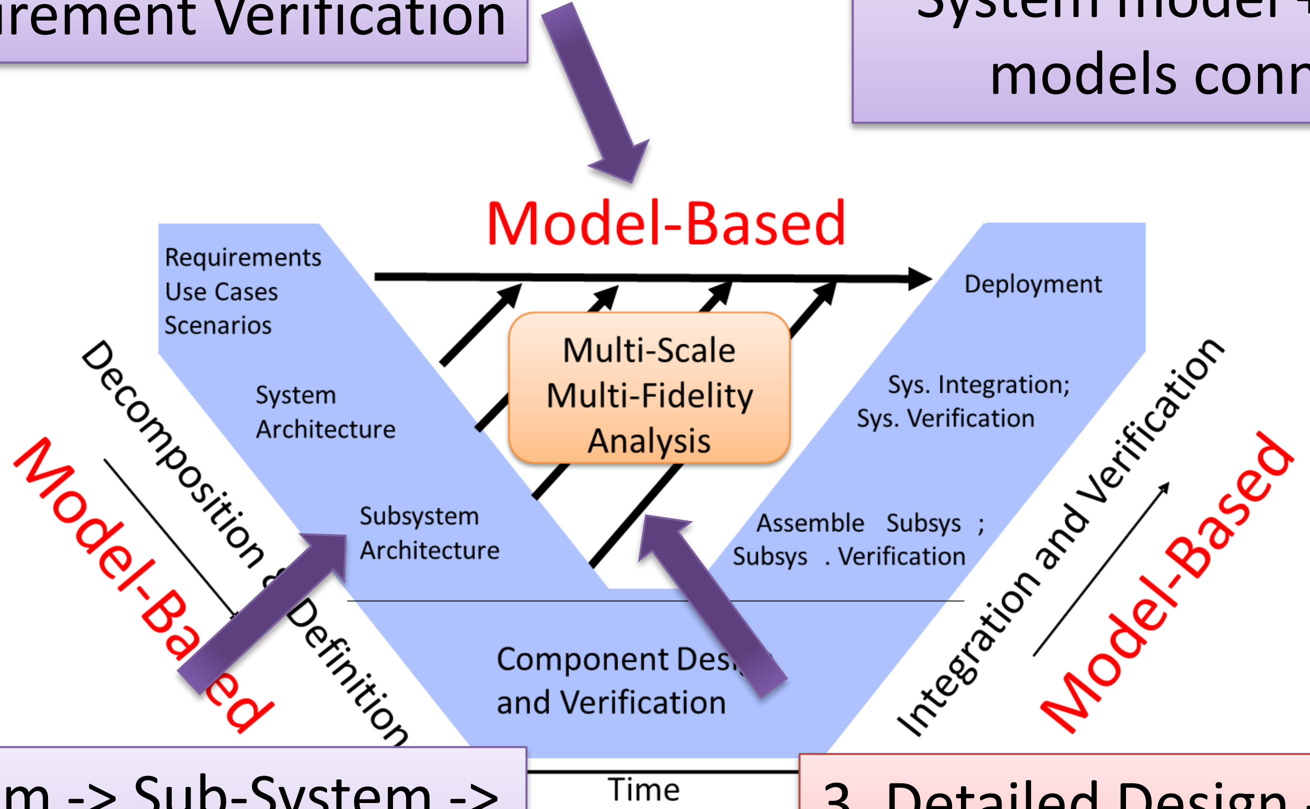
Reuse instruments from MySQL db

- MS - Instruments LLC
  - test_demo
  - sakila
  - instruments
    - video_camera
      - R mass (14.0) name (V3000) vcameraId (VC3) cost (1000.0)
      - R mass (10.0) name (V1000) vcameraId (VC1) cost (500.0)
      - R mass (12.0) name (V2000) vcameraId (VC2) cost (600.0)
    - thermal_camera
      - R mass (10.0) tcameraId (TC3) name (TC 103) cost (1000.0)
      - R mass (5.0) tcameraId (TC1) name (TC 101) cost (500.0)
      - R mass (8.0) tcameraId (TC2) name (TC 102) cost (700.0)
    - radar
      - R mass (20.0) radarId (R1) name (R11) cost (10.0)
      - R mass (22.0) radarId (R3) name (R31) cost (12.0)
      - R mass (21.0) radarId (R2) name (R21) cost (11.0)
  - world
  - performance_schema
  - information_schema

How do we do this?

1. Design -> Analysis -> Requirement Verification

4. Total System Model – System model + domain models connected



2. System -> Sub-System -> Components (PLM/CAD, Software)

3. Detailed Design -> Analysis (Simulink/Modelica, FEA, CFD)

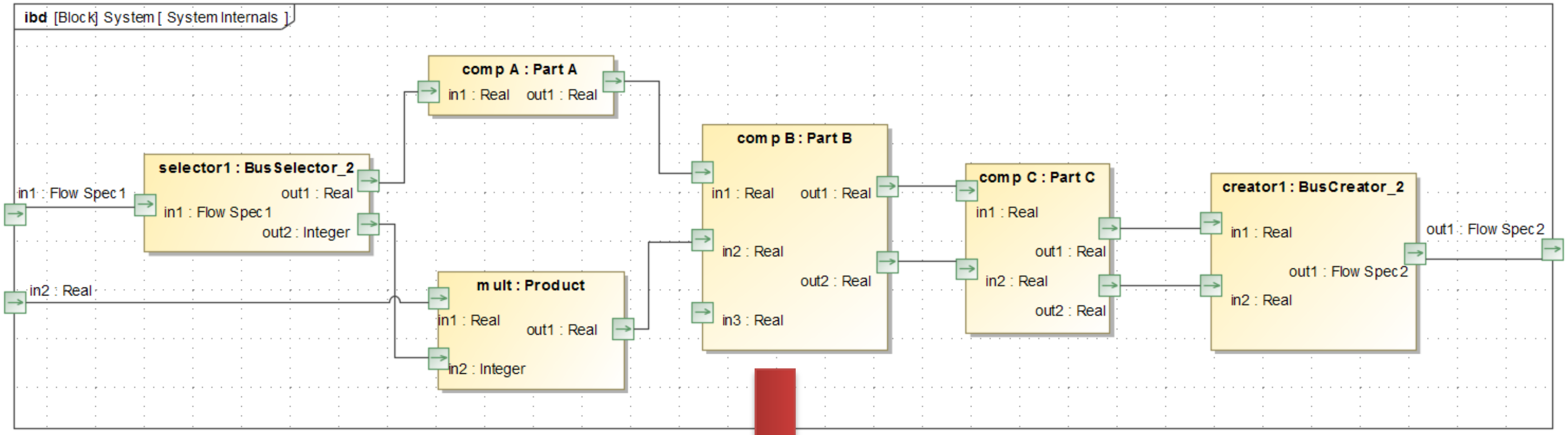
Detailed Design > Analysis

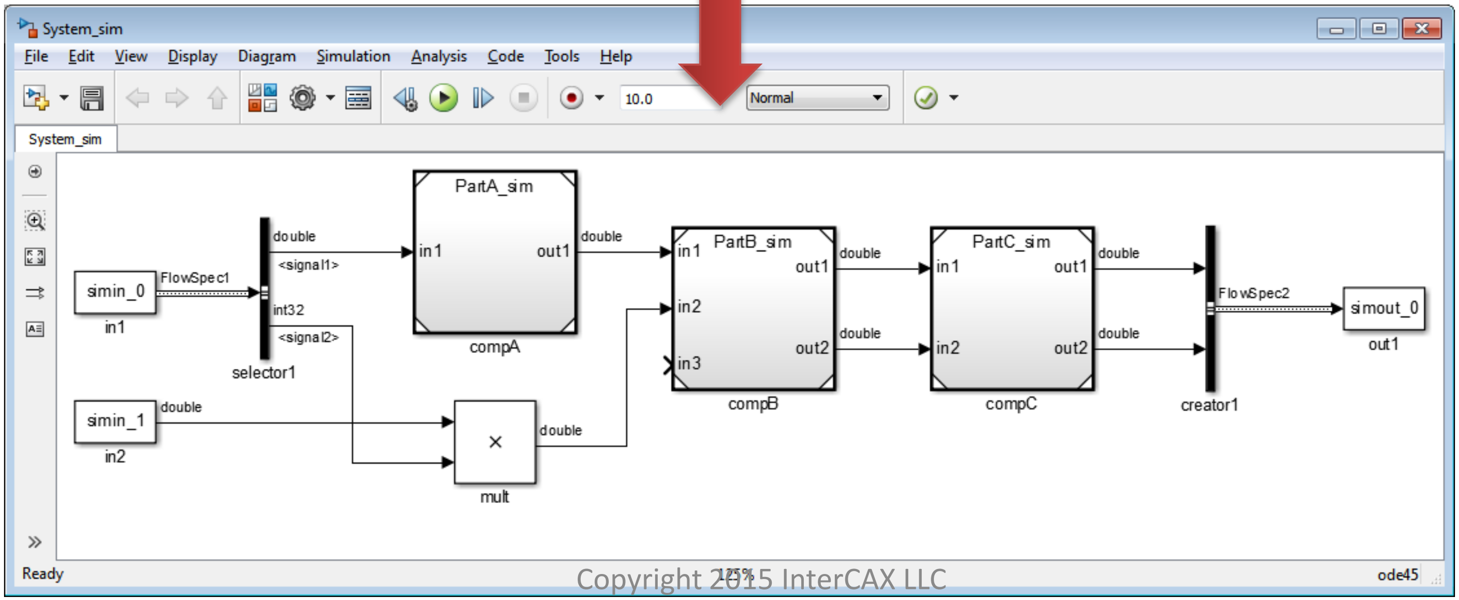
Seeding analysis models from design models

- Model-based communication between designers and analysts
  - Saves time to create analysis models
  - Communicate design intent to analysts
- Use cases of design > analysis model transform
  - SysML (IBD / Activity) > Simulink or Modelica
  - SysML > FEA/CFD
  - CAD > FEA/CFD

SysML (IBD / Activity) > Simulink / Modelica

Using SLIM to Seed Simulink / Modelica models from SysML internal block structure or activity models

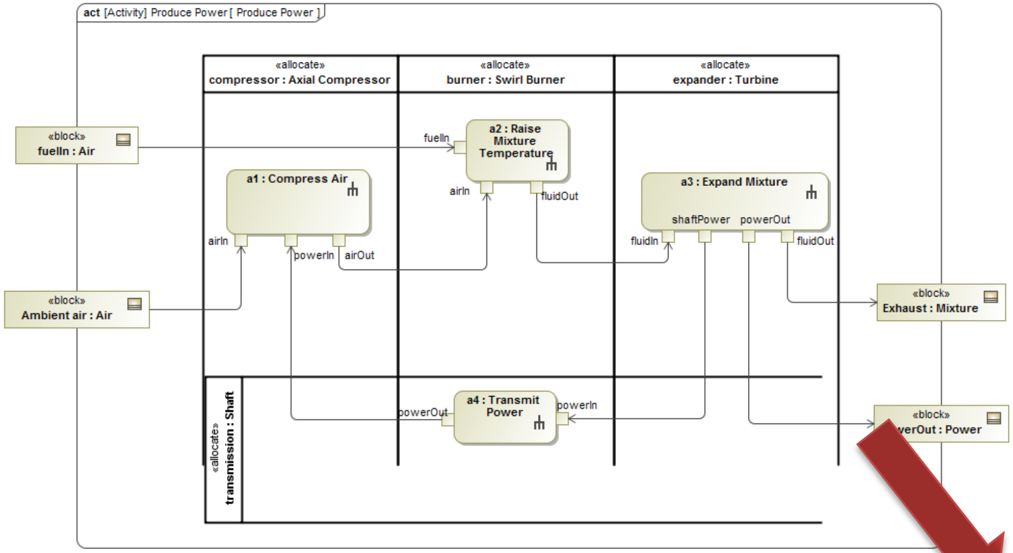




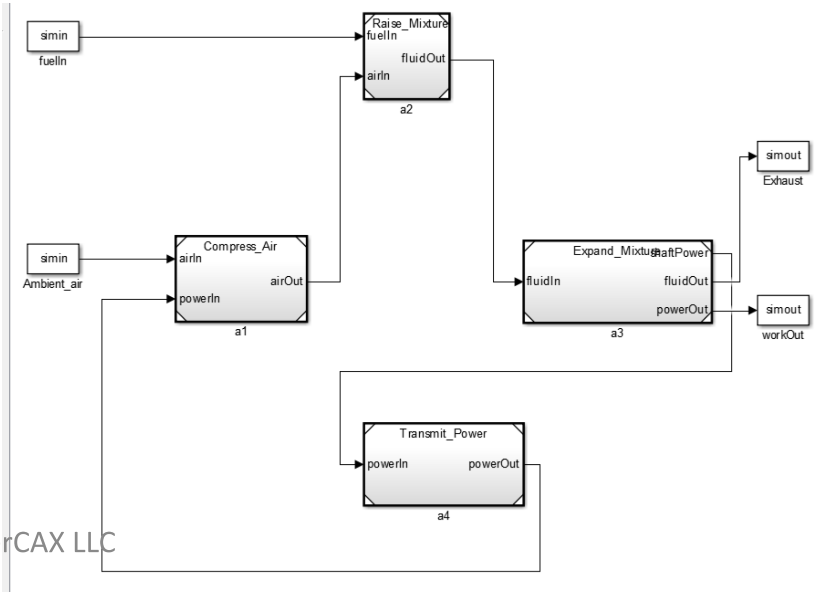
Copyright 2015 InterCAX LLC

SysML (IBD / Activity) > Simulink / Modelica

Using SLIM to Seed Simulink / Modelica models from SysML internal block structure or activity models



SysML activity model for produce power behavior of Gas Turbine

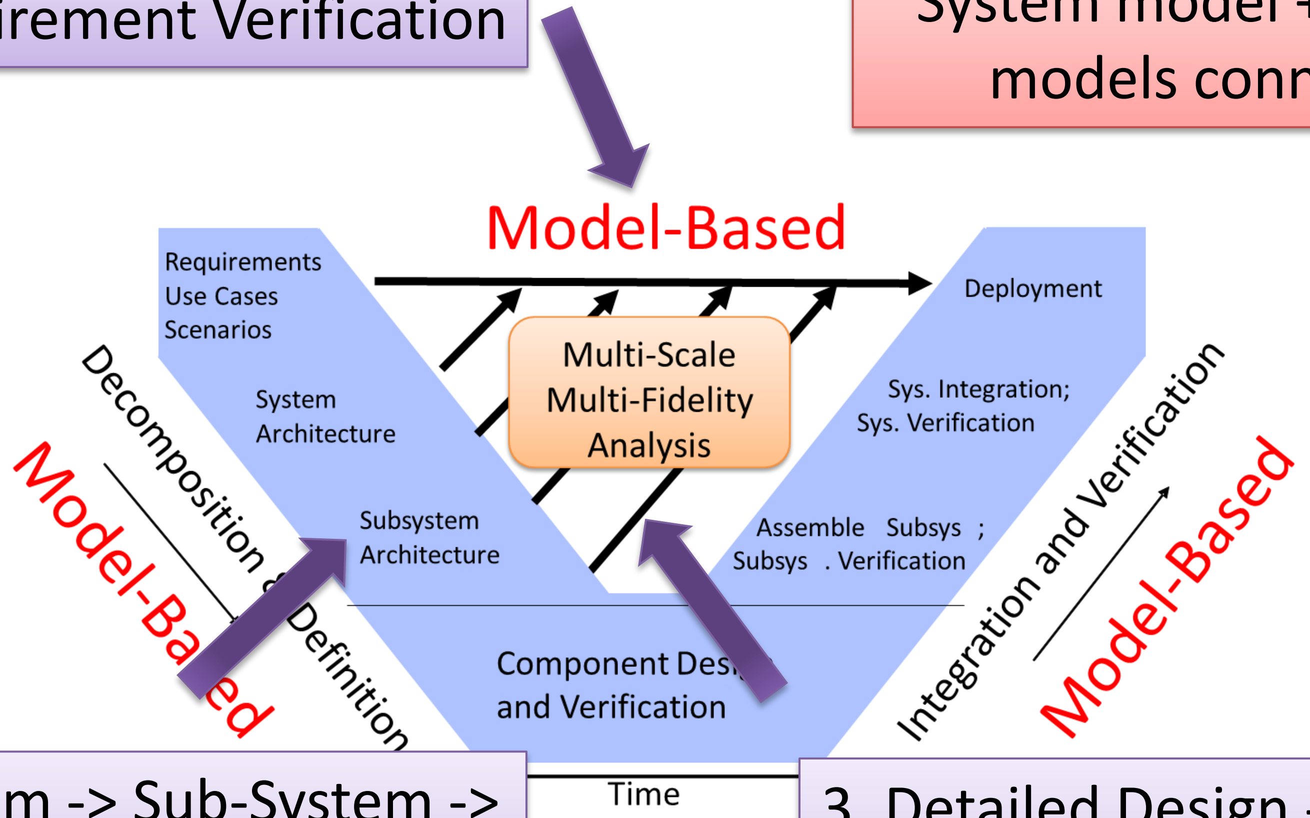


Simulink model seeded from the SysML activity model

How do we do this?

1. Design -> Analysis -> Requirement Verification

4. Total System Model – System model + domain models connected

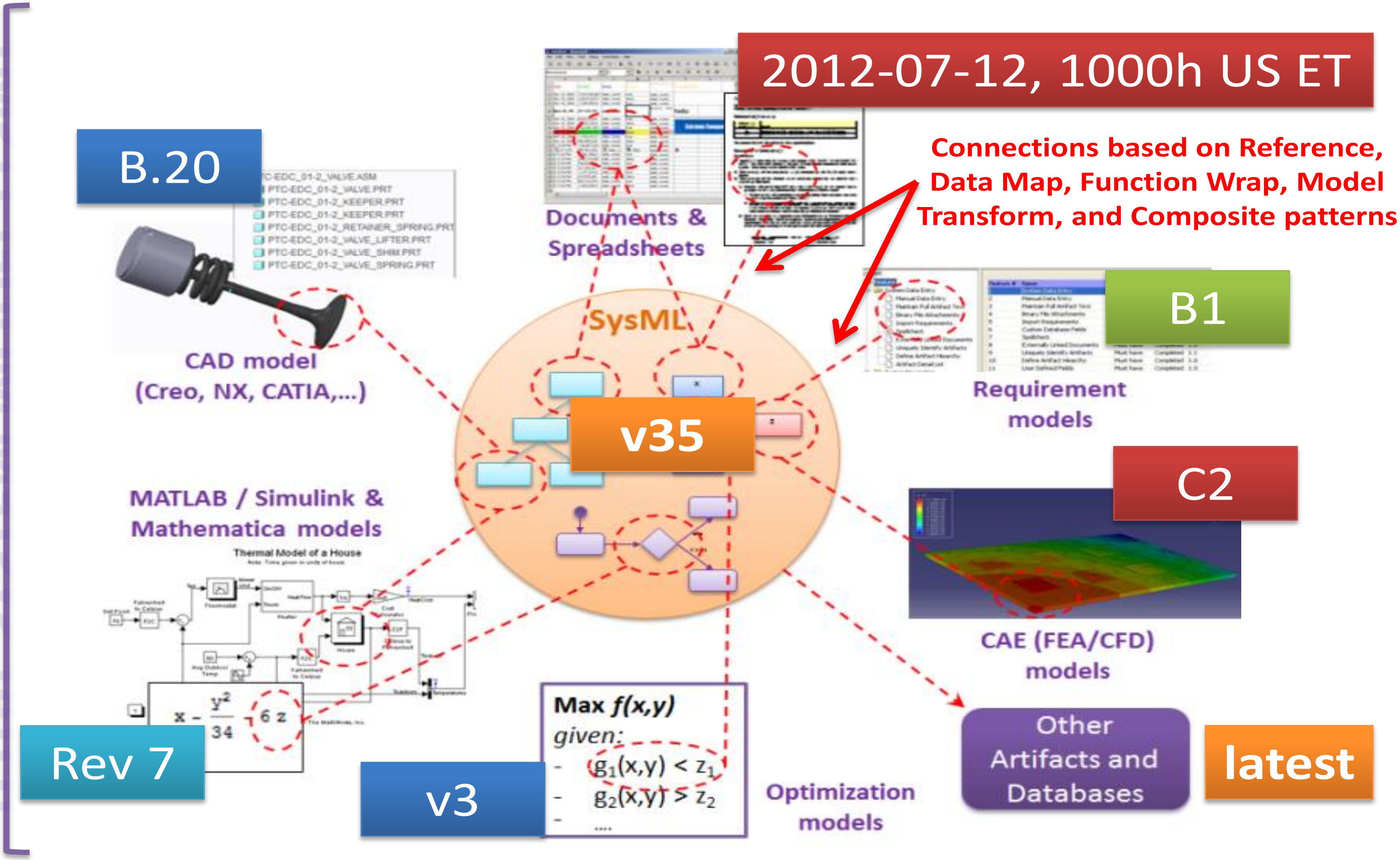


2. System -> Sub-System -> Components (PLM/CAD, Software)

3. Detailed Design -> Analysis (Simulink/Modelica, FEA, CFD)

Total System Model
Federation of models across the "Vee"

TOTAL SYSTEM MODEL (TSM)



2012-07-12, 1000h US ET

Connections based on Reference, Data Map, Function Wrap, Model Transform, and Composite patterns

B.20

CAD model
(Creo, NX, CATIA,…)

Documents & Spreadsheets

SysML

v35

B1

Requirement models

C2

CAE (FEA/CFD) models

MATLAB / Simulink & Mathematica models

Thermal Model of a House

Rev 7

v3

Max ***f(x,y)***
given:
- $g_1(x,y) < z_1$
- $g_2(x,y) > z_2$
- ....

Optimization models

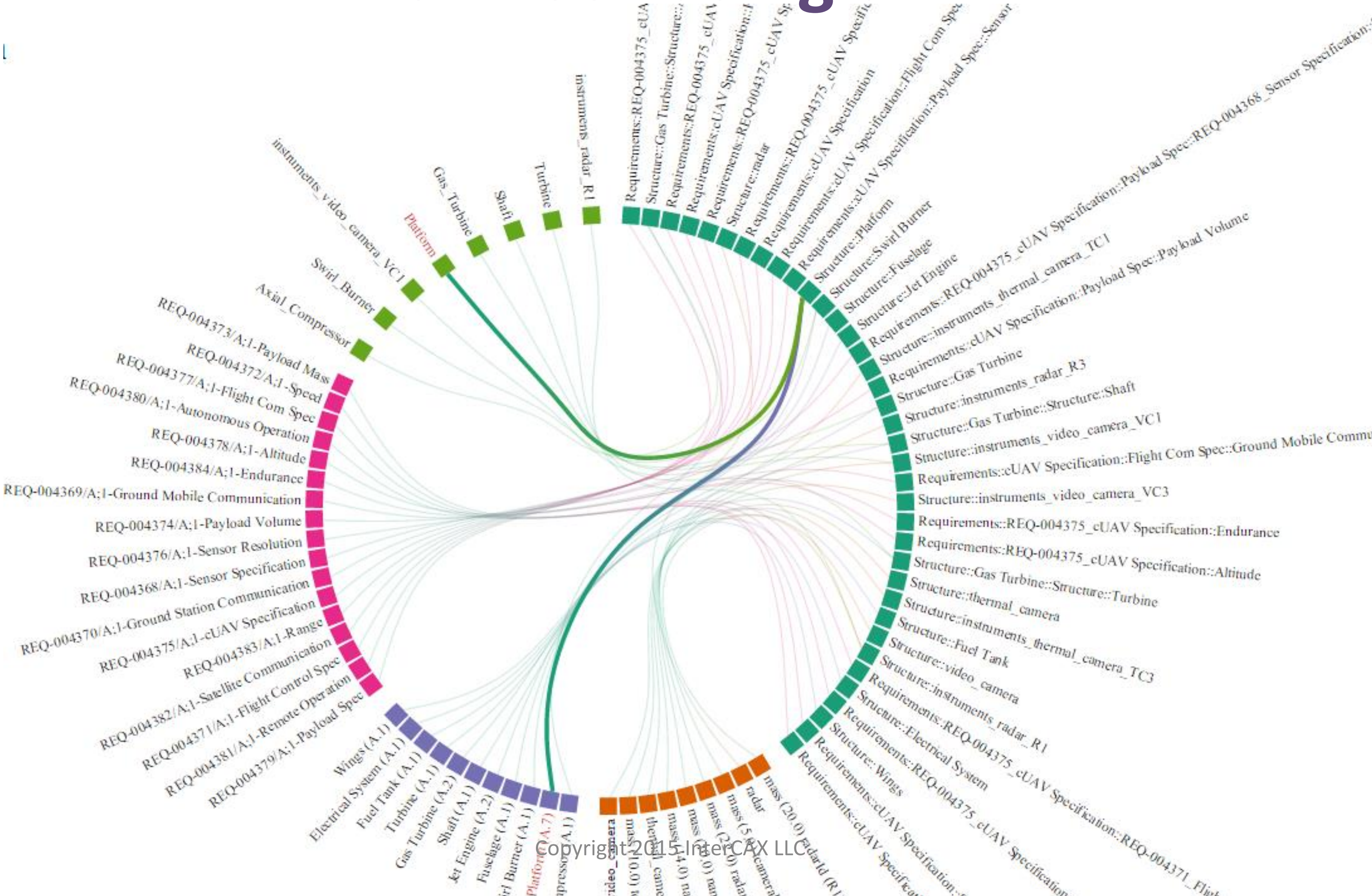
Other Artifacts and Databases

latest

Patterns of model connections

- Reference
  - Compare versions of connected models
- Data map
  - Compare and sync attributes bi-directionally
- Function wrap
  - Execute target model from the source model
- Model transform
  - Compare full model structure
  - Sync model structure bi-directionally

Visualizing models connections across the “Vee” using SLIM


instruments_radar_R1
Turbine
Shaft
Gas_Turbine
Platform
instruments_video_camera_VC1
Swirl_Burner
Axial_Compressor
REQ-004373/A;1-Payload Mass
REQ-004372/A;1-Speed
REQ-004377/A;1-Flight Com Spec
REQ-004380/A;1-Autonomous Operation
REQ-004378/A;1-Altitude
REQ-004384/A;1-Endurance
REQ-004369/A;1-Ground Mobile Communication
REQ-004374/A;1-Payload Volume
REQ-004376/A;1-Sensor Resolution
REQ-004368/A;1-Sensor Specification
REQ-004370/A;1-Ground Station Communication
REQ-004375/A;1-cUAV Specification
REQ-004383/A;1-Range
REQ-004382/A;1-Satellite Communication
REQ-004371/A;1-Flight Control Spec
REQ-004381/A;1-Remote Operation
REQ-004379/A;1-Payload Spec
Wings (A.1)
Electrical System (A.1)
Fuel Tank (A.1)
Turbine (A.1)
Gas Turbine (A.2)
Shaft (A.1)
Jet Engine (A.1)
Fuselage (A.2)
irl Burner (A.1)
Platform (A.7)
pressor (A.1)
video_camera
mass (0.0) n
thermal_came
mass (4.0) na
mass (3.0) nam
mass (2.0) radar
mass (5.0) camera
radar
mass (20.0) radarId (R1
Requirements::cUAV Specificat
Requirements::cUAV Specification::
Structure::Wings
Requirements::REQ-004375_cUAV Specification
Structure::Electrical System
Requirements::REQ-004375_cUAV Specification::REQ-004371_Fligh
Structure::instruments_radar_R1
Structure::video_camera
Structure::Fuel Tank
Structure::instruments_thermal_camera_TC3
Structure::thermal_camera
Structure::Gas Turbine::Structure::Turbine
Requirements::REQ-004375_cUAV Specification::Altitude
Requirements::REQ-004375_cUAV Specification::Endurance
Structure::instruments_video_camera_VC3
Requirements::cUAV Specification::Flight Com Spec::Ground Mobile Commu
Structure::instruments_video_camera_VC1
Structure::Gas Turbine::Structure::Shaft
Structure::instruments_radar_R3
Structure::Gas Turbine
Requirements::cUAV Specification::Payload Spec::Payload Volume
Structure::instruments_thermal_camera_TC1
Requirements::REQ-004375_cUAV Specification::Payload Spec::REQ-004368_Sensor Specification::
Structure::Jet Engine
Structure::Fuselage
Structure::Swirl Burner
Structure::Platform
Requirements::cUAV Specification::Payload Spec::Sensor
Requirements::cUAV Specification::Flight Com Spec
Requirements::cUAV Specification
Requirements::REQ-004375_cUAV Specific
Structure::radar
Requirements::REQ-004375_cUAV Sp
Requirements::cUAV Specification::
Requirements::REQ-004375_cUAV
Structure::Gas Turbine::Structure::
Requirements::REQ-004375_cUA

Questions / Comments

Manas Bajaj, PhD
Chief Systems Officer
manas@intercax.com

Dirk Zwemer, PhD
President
dirk@intercax.com

web: www.intercax.com

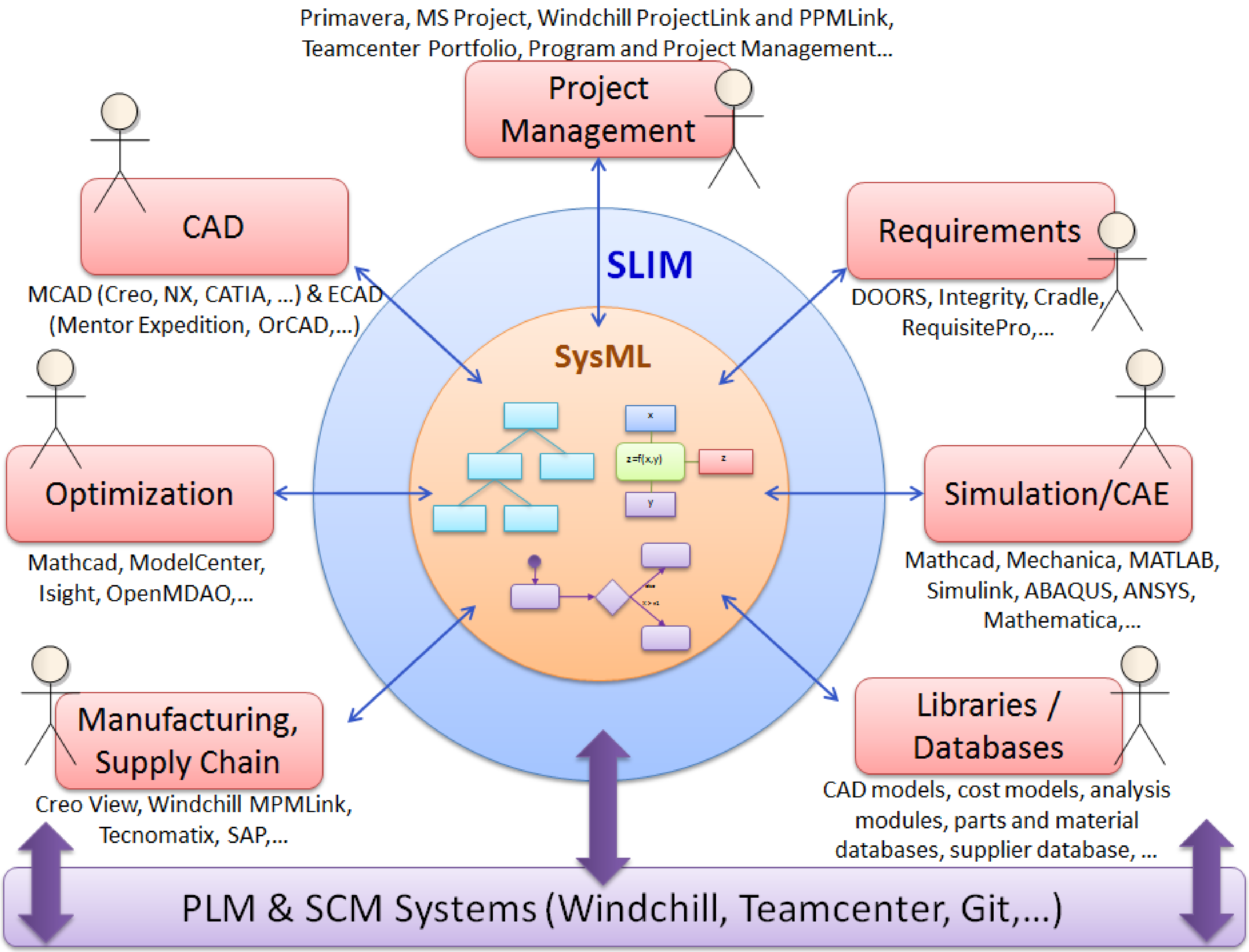
email: slim@intercax.com

Twitter @intercax | LinkedIn intercax-llc

BACKUP SLIDES

System LIfecycle Management (SLIM)

Enabling Model-Based Systems Engineering



Primavera, MS Project, Windchill ProjectLink and PPMLink,
Teamcenter Portfolio, Program and Project Management...

Project Management

CAD

MCAD (Creo, NX, CATIA, ...) & ECAD
(Mentor Expedition, OrCAD,...)

Requirements

DOORS, Integrity, Cradle,
RequisitePro,...

SLIM

SysML

Optimization

Mathcad, ModelCenter,
Isight, OpenMDAO,...

Simulation/CAE

Mathcad, Mechanica, MATLAB,
Simulink, ABAQUS, ANSYS,
Mathematica,...

Manufacturing, Supply Chain

Creo View, Windchill MPMLink,
Tecnomatix, SAP,...

Libraries / Databases

CAD models, cost models, analysis
modules, parts and material
databases, supplier database, ...

PLM & SCM Systems (Windchill, Teamcenter, Git,...)