Multi-Scale, Multi-Fidelity Systems Design and Simulation Environment

Peter Menegay – Director, Funded R&D pete@syncad.com, 540-953-3390, 540-557-7556 Dan Notestein – President SynaptiCAD

National Defense Industrial Association (NDIA) 19th Annual Systems Engineering Conference Oct. 26, 2016





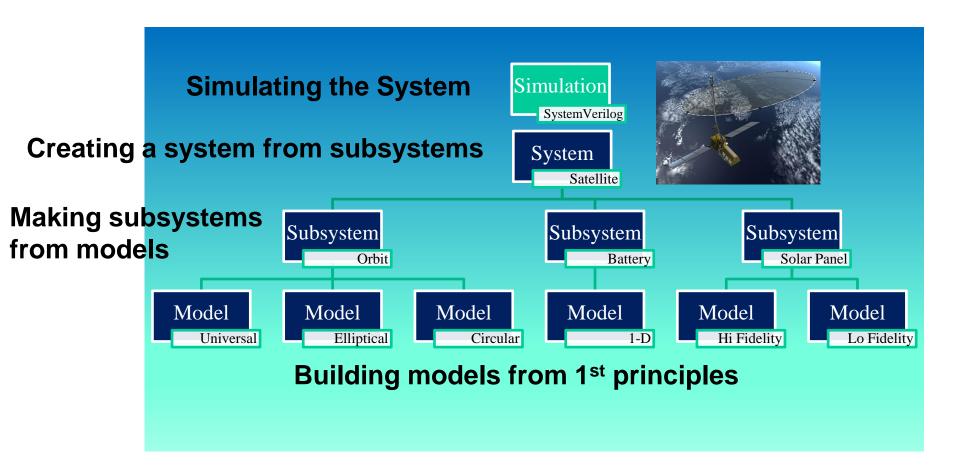
Promoting National Security Since 1919

Technology Motivation

- DoD systems are increasingly complex and challenge human cognitive, and organizational, abilities.
 - Model fidelity and connecting multi-fidelity models to coherent system views.
 - Using the lowest, most appropriate level of fidelity.
 - Engineering model robustness.
 - Reuse of models.
 - Handling highly scaled simulation problems "digital twin".
 - Discovering unforeseen behavior.
 - Understanding complex results.
 - Accurate simulations, well before we commit.
 - Simulations & models that live with us throughout the program's life.



The Basics

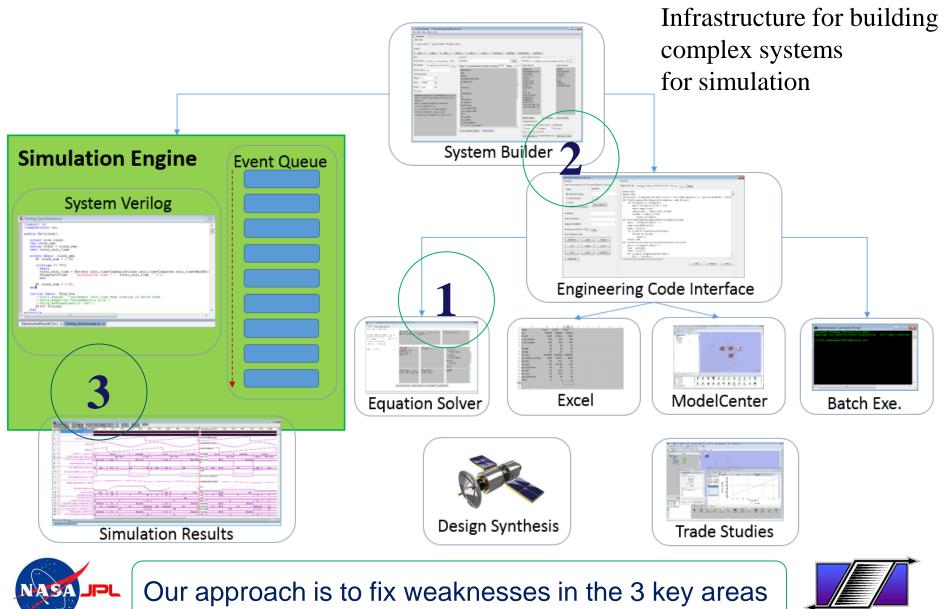




Some Observations

- Most simulation systems are not scalable. But those in the EDA space are. Why not use a paradigm like SystemVerilog to achieve scale?
- Most model-building tools are not generic, and if they are, are not robust enough to be used within simulation. We need to improve the areas where generic numerical methods are weak.
- Most system building tools have trouble handling fidelity transitions as models get more complex. Tools also tend to be domain specific. This leads to bifurcation of modeling efforts.
- System simulation needs to be able to *drive* other tools & models. It also needs to be *drivable* from other tools & models.
- When results from large scale simulations come in, they are often hard to interpret. How can we create views that make sense?

System Computation Platform

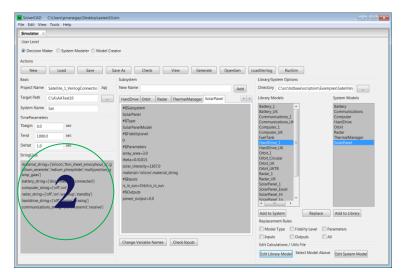


SYNAPTI**CAD**

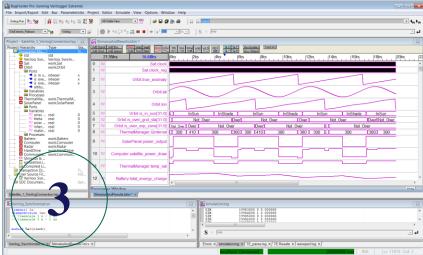
3 Core Elements & Workflow

Equation Solver Code Generation Simulation							
Solve any system of simultaneous equations							
Click here to generate a scriptwrapper for this example Click here to generate a function for this example Function return value? Use tase units in generated function? Base_Units Userfrout_Units	$\label{eq:productions} \begin{split} & = \mathcal{C}(uationsBegin \\ rp = zp + Rearth \\ e^-(r_0 + r_0)/(a + rp) \\ rp = a^*(1, 0 - e) \\ vrbar = (m_u/h)^*(1, 0 - e^*localFunctionsC) \\ vrbetaleas the structure defined and the structure defined and$		 Solution 1 of 3 Sight Solution to 1 Sight Solution terret virtetabar - 1.580005197 kilometer/second virtetabar - 2.58005197 kilometer/second h = 5374,250796 kilometer/second the sight Sight Solution terret/second the sight Sight Solution terret/second virtetamax - 12.856987466 degree virtetamax - 12.856987466 degr				
Set herts for filling out form:	#GrownBegin mu-398000 km ⁺¹ /sec ⁺¹ 2 PH-1141530 km Reambin 3720 km e-0.6 dimensionless	 KluenesBegin 1.3.06+00.006+00.10e 1.3.26+0.11e 0.12e+0.12e+0.01e 1.3.26+0.10e+0.01e 1.4.20e+0.20e+0.10e 1.4.20e+0.20e+0.10e+0.11e 1.4.20e+0.20e+0.11e 1.4.20e+0.20e+0.11e 1.4.20e+0.20e+0.11e 1.4.20e+0.20e+0.11e 1.4.20e+0.20e+0.11e 1.4.20e+0.20e+0.20e+0.11e 	 Idef Cociphi): mean meta-cociphi) def Sociphi: def Sociphi: mean meta-cociphi) def Sociphi: mean meta-cociphi def ArcTan(r); def ArcTan(r	• III •			
	#FunctionUnitsBegin LocalFunctions.ArcCostadian LocalFunctions.CostImensionless LocalFunctions.Sin,dimensionless LocalFunctions.ArcTanzadian	ARandomGuessesRegin LiveRandomGuesses10	 #ErrCrittExpin ErrCritt=10e-7 				

Eqn. Based Model Creator -- Simultaneous equation solver for creating new engineering models for subsystems.



System Builder -- User interface for creating new system models.





System Simulator -- SystemVerilog simulation engine for system model execution and results.



1. Eqn. Based Model Creator

- A general purpose model-creation environment for engineering analysis. Under development.
- Solves any system of nonlinear simultaneous equations.
- Manages the core numerical library to achieve robustness.
- Generates Python functions to use in simulation code.
- Uses a library concept for storing functions for later use.

ile Help guation Solver					
Solve any system of simultaneous equations					
Click here to generate a scriptwrapper for this example Click here to generate a function for this example Function return value? p2 Use base units in generated function? Base_Units UserInput_Units	#EquationsBegin v=(mu/(Rearth+z))**0.5 T=((20*P)(u*0.5))*(Rearth+z)**1.5 h=(Rearth+z)*v energy=-10*mu/(20*(Rearth+z)) t=(h*3/mu*2)*Theta LocalFunctions.Cos(Ninety-ThetaShade2) ThetaShade1=ThreeSixty-ThetaShade2 <	=Rearth/(Rearth+z)	T = 58 Theta ThetaS ThetaS energy fakeva	on 1 of 1 01.0665553 second = 0.0108310862147 radian hade1 = 5.13011155582 radian hade2 = 1.1303775136 radian r= -28651184.1327 meter*2/second*2 r = 0.0 dimensionless 739279452.8 meter**2/second	
Get hints for filling out form:	#KnownsBegin mu=396614 mt*/sect*2 Pl=3.1415926 dimensionless Rearth=6378000.0 meter t=10.0 sec Ninety=1.5707963267949 rad Ninety=1.5707963267949 rad Ispos=1.0 dimensionless	#GuessesBegin v10000.10000.1,m/sec TA800.04800.01.sec h.0.610.264E11.1m*2/sec energy.5.0.1430.1m*2/sec*2 Theta.00175.637.1.nd ThetaShade2.00175.1571,rad ThetaShade1.471.628.1.nd is_in_sun.0.1.0.1.dimensionless fakevar,0.0.0.1.dimensionless		if(thetash2 > 0.0): return 1.0 else: return 0.0 def isinSun(theta,thetash1,thetash2): if(theta < 0.05 and theta >= 0.0): return 0 else: return 1 #LocalFnsEnd	
marker_list=ro-,bo- title=the big one legend_list=red,blue xiabel=x axis ylabel=y axis	LocalFunctions.Cos,dimensionless LocalFunctions.InShade,dimensionless	UseRandomGuesses True NumRandomGuesses 10	• •	< ErrCrit=1.0E-7	+ =
y2label=z axis index,2nd,yaxis=6 < x x column hdr x axis title: y column hdr y axis title: title here y column hdr y axis title: title here	#ConstraintsBegin	 #OptimizationBegin #OptimizationEnd 	, ,	#DOEBegin #DOEEnd	
z z column hdr z axis title: title here Graph Title Your Title Plot Graph	< Þ	* 4	t v	<	w. F
Check System Get Guesses Solve and Generate Code	See Scriptwrapper See Generated Fur	nction Sort Guesses Adjust Guesse	es :		



2. System Builder

A GUI that helps you build, connect, and modify sub-system models for simulation.

- Create system models from a library of prebuilt subsystems.
- Create subsystem models from equation solver and external tools.
- Publish subsystems to library.
- Easily replace subsystems with higher/lower fidelity ones.

SimModelBuilder C:\Users\pmenegay\Desktop\base	2.sim	
File Edit View Tools Help		
Simulator ×		
User Level		
O Decision Maker O System Modeler Model Creat	pr	
Actions		
New Load Save Sav	e As Check View Generate OpenGen Lo	adVerilog RunSim
Basic	Subsystem	Library/System Options
Project Name Satellite_1_VerilogConnectio .hpj	New Name Add	Directory C:\src\hdbase\scriptsim\Examples\Satellite\
Target Path C:\A\AATest32	HardDrive Orbit Radar SolarPanel ThermalManager	Library Models System Models
System Name Sat	#\$Subsystem Orbit #\$Syubsystem Orbit #\$Syupe OrbitalMechanicsModel #\$FidelityLevel 0 #\$DeltatLocal 1.0 #\$DeltatLocal 1.0 #\$Parameters mu = 39860.0 Rearth = 6378.1 2_sat_perigee = 500.0 z_sat_aporgee = 510.0 i=40.0 Omega = 30.0 Omega = 20.0 JD_base = 245000.0	Battery_UK Battery Communications_UK Computer HardDrive_UK Computer HardDrive_UK HardDrive Orbit_UKTB Orbit Radar_UK Radar SolarPanel_UK SolarPanel ThermalManager_UK ThermalManager Add to System Replace Add to Library
	JD_relative_to_base = 7287.9 lat_gnd = 34.2 lon_gnd = 241.81 Change Variable Names Check Inputs	Replacement Rules Replacement Rules Model Type Fidelity Level Parameters Inputs Outputs All Edit Calculations / Utils File Edit Library Model Select Model Above Edit Library Model



2. System Builder, cont.

Includes a feature to help you edit calculations or link to them from external tools.

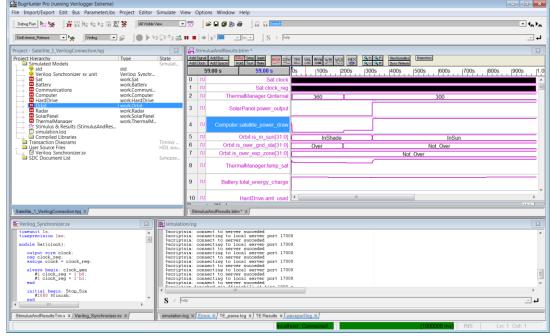
	Add Calculations to Utils File	2		×
	Options		Utils File	
	Insert Calculation from Exte		Open Utils File C:\src\hdbase\scriptsim\Examples\Satellite\ Save	
	Туре	Location	import math	
	Function Library		import time	
	ModelCenter		<pre>def ConvertToVerilogValue(val, stringlist_name=''): val = str(val)</pre>	=
	C Excel	Open Model	try:	
	O ASCII I/O Executable	Open Model	<pre>val_type = type(eval(val)) if(val_type is float):</pre>	
Link to pre-built models in	Function		return eval(val) elif(val_type is int):	
-	Input Variables		return eval(val)	
SolverCAD, ModelCenter,	Cutavit Variables		<pre>elif(val_type is bool): if(val == 'False'):</pre>	
Excel, or your own	Output Variables		return 0	
Exect, or your own	Generate and Insert Code	Go	else: return 1	
program.	Insert Python Code		except NameError:	
	def (fn.) class for while list dict write file	print	<pre>#means it is a string stringlist = FindStringListInStringList(stringlist, stringlist_name) int_val = ConvertStringInStringListToInt(stringlist,val) return int_val def SetDelayAndVerilogOutputs(pasted_in_object): delta_wait = passed_in_object Tcurrent - float(passed_in_object.Sim.time()) passed_in_object.Qinternal_vvc.delay = (str(delta_wait), "Transport") nassed in object.temn_sat_vvo.delav = (str(delta_wait). "Transport") OK Cancel Hel </pre>	P
Buttons for quickly creating common Python constructs.			Edit the calculation file for any subsystem in Python.	
			SYNAP	TICAD

3. System Simulator

The system simulator combines a high-performance compiled-code SystemVerilog simulator with a Python interpreter to enable engineering level modeling of real world systems.

- Timing diagram with simulation results
- Generated SystemVerilog code.
- Hierarchical view of subsystems and components.
- Full IDE including single step debugging, breakpoints, etc.
- Design browsing & navigation.
- Various output formats

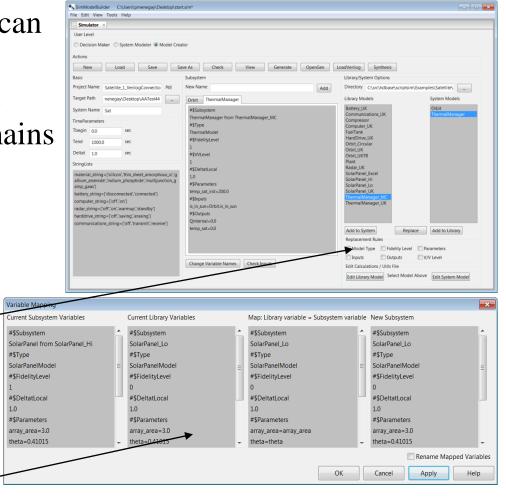






Multifidelity Modeling

- Models of different fidelity can be switched on the fly.
- As the project advances, the simulation environment remains in place, and maintains connectiveity with previous models.



Replacement rules for switching model fidelity

Variable mapping to ensure continuity between models



Model Libraries

• Subsystem model library.

Builder.

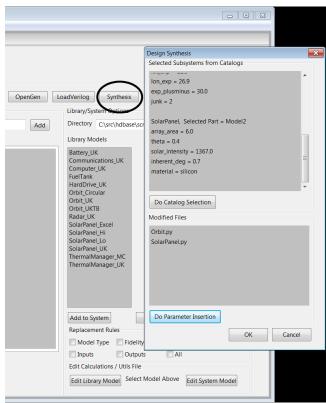
- Orbit calculations, solar panel, battery, etc. are publishable and retrievable from library.
- Function library for equation solver.
 - Generated functions can be accessed by System

Organize 🔹 🛛 🥞 Open	Include in library Share with	» 🎚 • 🗍	● 000000000000000000000000000000000000
Name	Date modified	Type Si:	
🗼 Plant	12/1/15 4:23 PM	File folder	1 #\$Subsystem 2 Orbit UK
L Compressor	12/1/15 4:22 PM	File folder	3 #\$Type
L Orbit_UK	11/28/15 10:28 AM	File folder	4 OrbitalMechanicsModel
L Orbit_Circular	11/17/15 4:34 PM	File folder	5 #≸FidelityLevel 6 Ø
🗼 SolarPanel_Hi	11/16/15 3:10 PM	File folder	7 #≸DeltatLocal
SolarPanel_Lo	11/16/15 3:10 PM	File folder	8 1.0
L ThermalManager_MC	11/16/15 2:35 PM	File folder	9 #\$Parameters
SolarPanel_Excel	11/16/15 2:20 PM	File folder	10 mu = 398600.0
👃 FuelTank	11/16/15 1:52 PM	File folder	11 Rearth = 6378.1 12 z_sat perigee = 500.0
1 ThermalManager_UK	11/13/15 2:40 PM	File folder	12 z_sat_perigee = 500.0
🗼 SolarPanel_UK	11/13/15 2:40 PM	File folder	14 i=40.0
📙 Radar_UK	11/13/15 2-39 PM	File folder	15 Omega = 30.0
Orbit_UKTB	11/13/15 2:39 PM	File folder	16 omega = 20.0 17 JD base = 2450000.0
HardDrive_UK	11/13/15 2:39 PM	File folder	18 JD relative to base = 7287.9
Computer_UK	11/13/15 2:38 PM	File folder	19 lat_gnd = 34.2
Communications_UK	11/13/15 2:38 PM	File folder	20 lon_gnd = 241.81
🗼 Battery_UK	11/13/15 2:38 PM	File folder	21 gnd_plusminus = 30.0 22 lat exp = 22.9
			22 lat_exp = 22.9 23 lon exp = 26.9
			24 exp_plusminus = 30.0
	III		25 theta = 0.0
Orbit_UK Date	e modified: 11/28/15 10:28 AM		26 #\$Inputs 27 None
File folder			27 None 28 #\$Outputs
4-4-2			29 true_anomaly = 0.0
			30 lat = 0.0
🔍 🗢 📕 « ModelLi	Orbit_UK v 4 Search Orbit		31 lon = 0.0 32 is in sun=False
Modeld	siden orbig		33 is_over_gnd_sta=False
Organize 🔹 🛛 🌄 Open	Burn New folder	··· • • •	34 is_over_exp_zone=False
Name	Date modified Type	Size	35 altitude = 505.0
Orbit UK.definition		1 KB	lengtl Ln : 15 Col : 15 Sel : 0 0 Dos\Windows UTF-8
	11/28/15 10:28 AM DEFINITION File		



Synthesis

- Compares simulation subsystems with a catalog of parts.
 - User has presumably optimized the subsystem and now wants to select hardware.
 - Software will choose the closest part from catalog and resimulate.





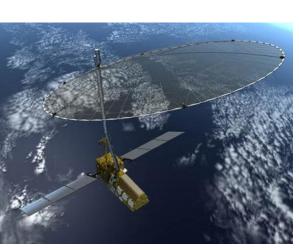
Orbit ##### PAR mu=398600.0 PAR Rearth=6378.1 PAR z sat perigee=500.0 PAR z sat apogee=510.0 PAR i=40.0 PAR Omega=30.0 PAR omega=20.0 PAR JD base=2450000.0 PAR JD_relative_to_base=7287.9 PAR lat gnd=34.2 PAR lon gnd=241.81 PAR gnd plusminus=30.0 PAR lat exp=22.9 PAR lon exp=26.9 PAR exp_plusminus=30.0 PAR theta=0.0 INP None OUT true anomaly=0.0 OUT lat=0.0 OUT lon=0.0 OUT is_in_sun=False OUT is over gnd sta=False OUT is over exp zone=False OUT altitude=505.0

Satellite Model

Sat

Satellite circles the earth in a standard elliptical orbit. It's mission is to collect earth data over an experimental zone and download it to a ground station at another location. It charges a battery in the sun and depletes the battery in the shade. The simulation objective is to understand if the subsystems are sized properly.

Sat
tbegin=0.0
tend=1000.0
deltat=100.0



SolarPanel

PAR material='silicon',material_string

PAR array area=3.0

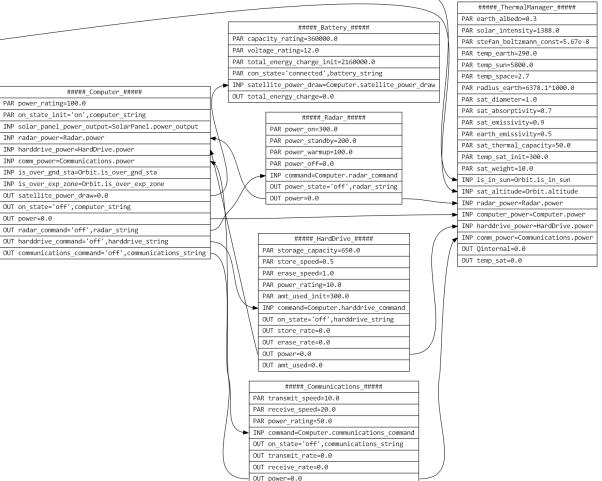
PAR theta=0.41015

PAR inherent deg=0.77

OUT power output=0.0

PAR solar intensity=1367.0

INP is in sun=Orbit.is in sun



NASA / JPL

Results

Add C	lock A	Add Bus Delay Setup Sample HIGH LON Add Spacer Hold Text Marker	-	VAL INVal			View Variable Class Method	5														
	11.5	52ks 0.000ms	0ks	1ks	2ks	3ks	4ks	5ks 6	ks 7	ks 8ks	9ks	10k	s 11ks	11519201	ms 13ks	14k	s 15ks	s <mark>16ks</mark>	17ks	18ks	19ks	20k
)	v	Sat.clock												0								<u> </u>
1	v	Sat.clock_reg												0					_			
2	л	Orbit.true_anomaly												0.1691188	4765501	7						
3	л	Orbit.lat	-											18.680678	1407817							
4	л	Orbit.lon			1					7				220.53248	934	1						
5	лı	Orbit.is_in_sun[31:0]	6.1		InSun	Υ	InS	hade)		InSun		γ h	nShade	InShade	InS	Sun	- r	InShad	e Y	Ir	nSun	~
6	лı	Orbit.is over gnd sta[31:0]	التنتقا		Not	Over		Over	γ		Not Over		Y	Over 1			Not Over	mondu			Not Ove	er 📃
7	л	Orbit.is_over_exp_zone[31:0]		ot Over			Not	Over	<u></u>	Over			Not Over	Not_Over		(Ov.)			Not Over	0101 1	1101_010	<u> </u>
8	л	SolarPanel.power_output			<u>, , , , , , , , , , , , , , , , , , , </u>			[,]		0		101.1			[_
9	л	ThermalManager.Qinternal	ñ	300	X 410 X	30	0	1 360	300	X 410		300	γ	360) 🚶	300	410)		300		360	300	_
10	л	Computer.satellite_power_draw												-360				000				_
11	л	ThermalManager.temp_sat	F			L						<u>٦</u>		244.41267	5100314							_
			-							_				1503634.8	9507634							
12	10	Battery.total_energy_charge	1																			
13	л	HardDrive.amt_used						$\overline{}$						212		_						
14	лı	Radar.power		200	300		2	200		300			200	200		(300)			200			
15	лı	HardDrive.power	O	0) 10)	C)) 10) 0	(10		0	χ	100	0	(10)		0	X	10)	0	
16	лı	Computer.power	Ć									100		100								
17	JU	Communications.power	0C		()) 50			0		X	500)			0		X	50)	0	
18	л (Computer.radar_command[31:0]		Standby) On (Sta	indby) On			Standby	Standby)(On)(Standby			
19	ллош	puter.harddrive_command[31:0]	DC	Off	Saving (0	ff	Erasir	g Off	Saving		Off		Erasing	Off	Sa.)		Off)	Erasing)	Off	
20	лл _{г.с}	communications_command[31:0]	DC		0	ff		Transm	it)		Off		т)(Transmit			Off)	Transmit)	Off	
21	лı	Radar.power_state[31:0]		Standby	(On)		Sta	indby) On			/	Standby		On)			Standby			
22	лı	Communications.transmit_rate	OC		_,, ()) 10	χ		0		χ	100			0)	10)	0	
			•																			•
ara	meter	r Window																				Hide

-Battery slowly drains to 0

Solar Panel does not recharge it when exposed to sun Ie, the Solar Panel is undersized. Battery is oversized.



Results, cont.

- One way to vary the solar panel / battery size is to use constrained randomization.
- Solution was to increase the solar panel area from 3.0 to 4.0 m**2 and decrease the battery capacity from 360,000 to 60,000 amp-sec.

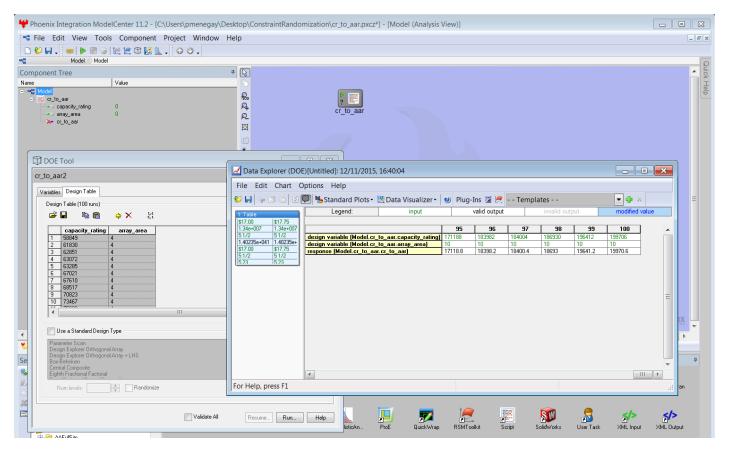
	olarPanel.sv			Σ
program Constr	ain_Batt_SolarPanel;			
class Constrat rand integer of rand integer of rand integer a	r_to_aar; r;			
$\begin{array}{c} \text{constraint c1} \\ \text{constraint c2} \end{array}$	{ cr_to_aar < 20000;} { cr_to_aar > 10000;}			
	{ cr >= 55000;} { cr <= 200000;}			
constraint c5 constraint c6				
constraint c0 endclass	{ cr_to_aar == cr / aar;}			
<pre>\$fwrite(wfile; for(int i=0; : if(obj.random: begin \$fwrite(wfil \$display(" F end else \$display("Reference: \$d</pre>	<pre>("c:/users/pmenegay/deskt "capacity_rating, array_a (100; i++) ze()) e,"%0d , %0d \n",objcr.o andomization successful : indomization failed"); .cr_to_ear);</pre>	rea \n"); bj.aar);	cr_to_aar = %0d*,obj.	cr,obj.aar,obj.cr_to_aar);





Results, cont.

• This could also have been achieved by driving the simulator from a ModelCenter DOE.





Overall Results

- Once engineering models were made, system integration was fast, 1-2 days for this case.
 - Model libraries were key.
- Provision for multi-fidelity model switching allowed project to remain within a single environment throughout its life.
- Scalability tests on a simple vehicle object lends credence to the SystemVerilog approach.
 - SystemVerilog can simulate up to memory limits of computer. 18 million vehicles for 32-bit and 40 million for 64 bit.
 - SimPy by contrast could simulate 900,000 such objects.
- Runs could be made faster by using event-driven simulation. A 10 fold speed up was achieved this way.
 - Important for long run times over the life of the system.

We thank the NASA SBIR program for sponsoring this work. Contract NNX15CP26P.

