SoSECIE Webinar

Welcome to the 2019 System of Systems Engineering Collaborators Information Exchange (SoSECIE)



We will start at 11AM Eastern Time Skype Meeting +1 (703) 983-2020, 46013573# You can download today's presentation from the OUSD(R&E) Website: <u>https://www.acq.osd.mil/se/outreach/sosecollab.html</u> To add/remove yourself from the email list or suggest a future topic or

speaker, send an email to knharrington@mitre.org

NDIA System of Systems SE Committee

• Mission

- To provide a forum where government, industry, and academia can share lessons learned, promote best practices, address issues, and advocate systems engineering for Systems of Systems (SoS)
- To identify successful strategies for applying systems engineering principles to systems engineering of SoS

Operating Practices

- Face to face and virtual SoS Committee meetings are held in conjunction with NDIA
 SE Division meetings that occur in February, April, June, and August
- SoS Track at NDIA 22nd Annual Systems Engineering Conference, Grand Hilton Tampa Downtown, Tampa, FL, October 21-24, 2019
 - Conference Info: <u>http://www.ndia.org/events/2019/10/21/22nd-annual-systems-and-mission-engineering-conference</u>

NDIA SE Division SoS Committee Industry Chairs:

Mr. Rick Poel, Boeing

Ms. Jennie Horne, Raytheon

OSD Liaison:

Dr. Judith Dahmann, MITRE

Simple Rules of Engagement

- I have muted all participant lines for this introduction and the briefing.
- If you need to contact me during the briefing, send me an e-mail at <u>knharrington@mitre.org</u>.
- Download the presentation so you can follow along on your own
- We will hold all questions until the end:
 - I will start with questions submitted online via the CHAT window in Skype.
 - I will then take questions via telephone; State your name, organization, and question clearly.
- If a question requires more discussion, the speaker(s) contact info is in the brief.

Disclaimer

- MITRE, NDIA, and The Office of the Under Secretary of Defense for Research and Engineering makes no claims, promises or guarantees about the accuracy, completeness or adequacy of the contents of this presentation and expressly disclaims liability for errors and omissions in its contents.
- No warranty of any kind, implied, expressed or statutory, including but not limited to the warranties of non-infringement of third party rights, title, merchantability, fitness for a particular purpose and freedom from computer virus, is given with respect to the contents of this presentation or its hyperlinks to other Internet resources.
- Reference in any presentation to any specific commercial products, processes, or services, or the use of any trade, firm or corporation name is for the information and convenience of the participants and subscribers, and does not constitute endorsement, recommendation, or favoring by the Department of Defense or USD.

2019 System of Systems Engineering Collaborators Information Exchange Webinars Sponsored by OUSD(R&E) and NDIA SE Division

January 22, 2019

Systems of Systems Model Building and Acausal Simulation Environment Peter Menegay, SynaptiCAD

February 5, 2019

Development of New Standards for Systems of Systems Engineering Dr. Mike Yokell, Lockheed Martin Fellow and Deputy Director, Systems Engineering

February 19, 2019

Systems of Systems Engineering Managerial and Operational Affinity Dr. Mike Yokell, Lockheed Martin Fellow and Deputy Director, Systems Engineering

> March 12, 2019 Mission Engineering Competency Model Dr. Nicole A. Hutchison, Stevens Institute of Technology

> > March 26, 2019

Practical Modeling Concepts for Engineering Emergence in Systems of Systems Dr. Judith Dahmann, The MITRE Corporation Ms. Philomena Zimmerman, OUSD(R&E)

5

2019 System of Systems Engineering Collaborators Information Exchange Webinars Sponsored by OUSD(R&E) and NDIA SE Division

6

April 16, 2019 Mission Analysis and Operational Architectures Mr. Zane Scott, Vitech Corporation

April 30, 2019 Digital Engineering Transformation Mr. Thomas McDermott, Georgie Tech Research Institute, SERC

May 14, 2019 Toward Scaling Model-based Engineering for Systems of Systems Dr. Ryan B. Jacobs, The MITRE Corporation

> May 28, 2019 Modular Online Open SoS Education (MOOSE) Mr. Kyle Hastings, The MITRE Corporation

June 11, 2019 Mission Engineering and Prototype Warfare Mr. Matthew Horning, US ARMY FUTURES COMMAND

June 25, 2019 A Tool for Architecting Socio-Technical Problems: SoS Explorer Dr. Cihan Dagli

System of Systems Model Building and Acausal Simulation Environment

Peter Menegay – Director, Funded R&D, SynaptiCAD pete@syncad.com, 540-953-3390 Dale Burnham – Lead Systems Engineer, AFRL dale.burnham@us.af.mil

> SoSECIE Webinar Jan. 22, 2019





Technology Motivation

- DoD systems are increasingly complex and challenge human cognitive and organizational abilities. We are now combining those systems into System of Systems.
 - Engineering model flexibility & robustness.
 - Reuse of models.
 - SoS requirements expression and flowdown.
 - Discovering unforeseen behavior through trade studies.
 - Understanding complex results.
 - Accurate simulations, well before we commit.
 - Handling highly scaled simulation problems "digital twin".



The Basics





System Computation Platform



Our approach is to fix weaknesses in the 3 key areas



3 Core Elements & Workflow

	ition					
e any system of simultaneous equation	ns					
Lick here to generate a scriptwrapper Lick here to generate a function for th ction return value? : e base units in generated function? Base Units () Guerfinput, Units	for this example is example all	$\label{eq:productions} \begin{split} & \mathbb{P}_{Quations} \text{Begin} \\ & p = zp + \text{Rearch} \\ & \text{e} = (a + p)/((a + p)) \\ & p = a^{1}(D - e) \\ & \text{vibar} = (mu/h)^{1}(D - e^{1}\text{LocalFunctions}. Sin(the \\ & \text{vibar} = (mu/h)^{1}(D - e^{1}\text{LocalFunctions}. C) \\ & \text{vithetabar}^{**2} = \text{vibar}^{**2} = viba$	etabae) oos(thetabar))	 Solution a = 16 wrtheta rop = 6i va = 2 h = 65 thetam vp = 9 wrbar = 	n1 of 3 9450 kilometer Islar - 59400963397 kilometer/second 4250914804 kilometer/second 4250914804 kilometer/second nax = 126860987646 degree 70013673615 kilometer/second = 342951623228 kilometer/second	
heres for filling out form:		#KnownSkejn mu-J86800 km*3/nee**2 PI-3.141526 kmensiolets Renth-63780 km zup-400 km e-L0.6 dimensionless	 #QuenesRegin T.3.06+-00.000+-00.1m T.3.26+-00.310++00.1m T.3.26+-00.310++00.1m T.0.26++00.30++00.1m+00 T.0.26++00.30++00.1m+00 T.0.26++00.00++00.1m+00 T.0.26++00.00++00 T.0.26++00.00++00 T.0.26++00.00++00 T.0.26++00.00++00 T.0.26++00.00++00 T.0.26++00.00++00 T.0.26++00.00++00 T.0.26++00 	E	def Cos(phi) import math return math.cos(phi) def Sin(phi): import math return math.com) return math.atml(phi) return math.atml(p) FLocalimEnd	× III ×
		#FunctionLinitsBegin LocaFunctions.ArcCox,radian LocaFunctions.Cox,dimensionless LocaFunctions.Sin,dimensionless LocaFunctions.ArcTan,radian	 MRandomGuessesBepin LiseRandomGuesses True NumRandomGuesses 10 		*EnCmBegin EnrCm=1.0e-7	

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



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



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



New Capability





Example Problem

PAR base_fuel_consumption_rate=0.045

GroundControlStation x Mission

OUT bombs dropped out=0.0

PAR init_distance_to_target=525.0 PAR target observation distance=6.0

INP SNRsar=Satellite x Comm.SNR INP SNRflir=Satellite x Comm.SNRflir gbu-Satellite_x_Comm.weight_gbu

INP None OUT fuel_weight=2000.0

PAR altitude_in=40000.0 PAR loiter_time_in=12.0 DAD cneed-175 @

UT altitude=40000.0 OUT loiter time=13.0 OUT distance_to_target=1000.0 OUT aircraft mode=0.0 OUT start loiter time=200000.0 OUT end loiter time=200000.0 OUT reached target=0.0 UT bombs_dropped=0.0 OUT Pdetect=0.0 OUT Phit=0.0 OUT Pkill-0.0 OUT Pshot=0.0

An MQ-9 Reaper flies to a target zone and collects sensor data, MULTIANK, MU It identifies a missile site, destroys it and returns to base. The aircraft is equipped with SAR, FLIR, and GBU as part of its system. The ground control station (GCS) controls mission parameters and determines probability of kill. A satellite handles communications between the two.

	#####_Aircraft1_x_SAR_#####	
PAR	Pt0=320.0	
PAR	Daz=44.5	
PAR	Dh=16.5	
PAR	rhor=1.0	
PAR	f=16.7	
PAR	fref=16.7	
PAR	aw=1.2	
PAR	Lsp=2.0	
PAR	Lradar=3.16	
PAR	Fn=1.33	
PAR	alpha=1.0077	
PAR	PI=3.14159265	
PAR	k=1.380648e-23	
PAR	c=299792458.0	
PAR	d=0.35	
PAR	T=290.0	
PAR	etap=0.5	
PAR	sigref=3.162e-3	
PAR	n=2	
PAR	one=1	
PAR	Daz0=44.5	
PAR	Dh0=16.5	
PAR	Le0=51.4	
PAR	We0=29.6	
PAR	He0=26.7	1
PAR	W0=83.0	1
PAR	C0=1500000.0	1
INP	aircraft_mode_in=Satellite_x_Comm.aircraft_mode	-
INP	h=Satellite_x_Comm.altitude	-
INP	dtt=Satellite_x_Comm.distance_to_target	-
INP	vx=Satellite_x_Comm.speed	4
OUT	aircraft_mode=0.0	
OUT	SNR=1.00	-
OUT	rs=0.0	
OUT	R=0.0	
OUT	V0=0.0	
OUT	V=0.0	
OUT	W=0.0	/
OUT	Pt=0.0	
OUT	C=0.0	
	M09AG5	

tbegin=0.0 tend=82800.0 deltat=3600.0

				OUT ShotDown=0.0	
				OUT KilledTarget=0.0	
				#####_Aircraft1_x_FLIR_#####	
				PAR lam1=2.0	
				PAR lam2=6.0	
			1	PAR T=500.0	\sim
				PAR PI=3.1415926	
1		#####_Satellite_x_Comm_#####	// ר	PAR eps=0.95	_
		PAR orbit_altitude_in=117406080.0] //	PAR sigma=5.67e-8	
		<pre>INP aircraft_mode_in=GroundControlStation_x_Mission.aircraft_mode</pre>	* //	PAR Ac=0.1	
		<pre>INP altitude_in=GroundControlStation_x_Mission.altitude</pre>	~ //	PAR Ae0=0.1	
		<pre>INP distance_to_target_in=GroundControlStation_x_Mission.distance_to_target</pre>	~/	PAR N=1.2e-7	
		INP speed_in=GroundControlStation_x_Mission.speed	-/	PAR AT=20.0	1
1		<pre>INP bombs_dropped_in=GroundControlStation_x_Mission.bombs_dropped</pre>	1	PAR vxref=200.0	\mathcal{I}
1	1	INP SNRsar_in=Aircraft1_x_SAR.SNR		PAR beta=6.7e-5	
	(INP SNRflir_in=Aircraft1_x_FLIR.SNR	-	PAR W0=100.0	_
1		INP weight_gbu_in=Aircraft1_x_GBU.weight_initial		PAR P0=280.0	
1	1	OUT orbit_altitude=117406080.0	ľΓ	PAR C0=1000200.0	
ŀ	≁	OUT aircraft_mode=0.0	\rightarrow	INP aircraft_mode_in=Satellite_x_Comm.aircraft_m	ode
7	-+	OUT altitude=40020.0	H	INP h=Satellite_x_Comm.altitude	
ŀ	•	OUT distance_to_target=6.0		INP dtt=Satellite_x_Comm.distance_to_target	
ł	•	OUT speed=175.0		INP vx=Satellite_x_Comm.speed	
		OUT bombs_dropped=0.0	ΗΛ	OUT aircraft_mode=0.0	
ŀ	/	OUT SNRsar=5.0	1	OUT C=0.0	
		OUT SNRflir=5.0	111	OUT F=0.0	
	- 1	OUT weight_gbu=1000.0	7Y //	OUT P=0.0	_
	- /		T MM	OUT R=0.0	
			100	OUT Rap=0.0	
]	/	(OUT S=0.0	
			`\##	OUT SNR=0.0	
			1000	OUT V0-0.0	
			NW)	OUT W=0.0	
			W	OUT tau=0.0	
			W		
			A	#####_Aircraft1_x_GBU_#####	
			N	PAR weight_initial=1000.0	
			N	INP aircraft_mode_in=Satellite_x_Comm.aircraft_m	ade
			N	INP h=Satellite_x_Comm.altitude	
			1/2	INP dtt=Satellite_x_Comm.distance_to_target	
			14	INP vx=Satellite_x_Comm.speed	
			7	INP bombs_dropped=Satellite_x_Comm.bombs_dropped	
				OIT woight-0.0	



PAR base_weight=4000.0 INP fuel_weight=Aircraftl_x_fuelTank.fuel_weight INP fuel_weight=Aircraftl_x_SAR.N INP flin_weight=Aircraftl_x_fLIR.N INP gbu_weight=Aircraftl_x_GBU.weight OUT total weight=4900.0		#####_Aircraft1_x_Structure_#####
<pre>INP fuel_weight=Aircrafti_x_FuelTank.fuel_weight INP sam_weight=Aircrafti_x_FAR.W INP flim_weight=Aircrafti_x_FLIR.W INP gbu_weight=Aircrafti_x_GBU.weight OUT total weight=AB00.0</pre>	PAR	base_weight=4900.0
INP sar_weight=Aircraft1_x_SAR.W INP flir_weight=Aircraft1_x_FLIR.W INP gbu_weight=Aircraft1_x_GBU.weight OUT total weight=4980.0	INP	<pre>fuel_weight=Aircraft1_x_FuelTank.fuel_weight</pre>
INP flir_weight=Aircraft1_x_FLIR.W INP gbu_weight=Aircraft1_x_GBU.weight DUT total weight=4920.0	INP	sar_weight=Aircraft1_x_SAR.W
INP gbu_weight=Aircraft1_x_GBU.weight DUT total weight=4980.0	INP	flir_weight=Aircraft1_x_FLIR.W
OUT total weight=4980.0	INP	gbu_weight=Aircraft1_x_GBU.weight
	DUT	total_weight=4900.0



SoS Integration - 1

• First build independent systems: aircraft, ground control station, and satellite. Aircraft is composed of multiple subsystems (e.g. SAR) which can be modeled in equation solver and connected to overall system:

SolverCAD C:\A\PythonWork\solvercad\engine\ExampleP	roblems\SAR.scin			×
] 🕛 🛹 👌 fx 🔇 🎧			
New Open Save Save Guess Errors Solve Sort Adjust Che	ck Solve Graph Generated Local Time Remove	Show Window Tab Lock Help About		
As	Function Functions Functions Time	Variables -		
Documentation Equations	Nisc		Solution	-
1 SNR=Pt*d* 2 rs=(dtt** 3 R=(0az*0h	etap**2*(Daz*Dh)**2*rhor*(sigref/fref**n)*f 2+h**2)**0.5 /(Daz0*Dh0))**0.5 h00\%Lad%Lao	**(n+one)/(8.0*PI*aw*c*k*T*vx*Lsp*Lr	1 Solution 1 of 1 2 C = 1500000.0 USD 3 Pt = 320.0 watt 4 R = 1.0 dimensionless	
Settinge B V=V0*R**3			5 SNR = 3.8833034251815737 dimensionless	
Generate function	*LocalFunctions.GetK(R)		6 V = 53662.7280000001 centimeter**3	
Scatting attended 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	.562		8 W = 83.00000000000014 pound	
Punction return value: 9			9 rs = 9.6639589634190965 mile	-
Units for generated function Knowns	E	Guesses	Local Functions	8
Base_Units UserInput_Units 1 vx=175.0	mile/hour	1 SNR,1.0,20.0,1,dimensionless	1 import math	
Graphics Editor	foot	2rs,5.0,15.0,1,mile 3R 0 1 10 0 1 dimensionless	2 def mylog(x): 3 return math log10(x)	
h SNR ^ 4 Pt0=320.0	watt	4 V0.50000.0.55000.0.1.centimeter*	*3 4 def mvln(x):	
30000.0 8.9606452188404955 5 Daz=44.5	Add Calculations to Utils File			
40000.0 3.8833034251815/3/ = 0.01=10.3 C	e Options	Utils File		
\$616.7 gi + 9 fref=16.7 10 au=1.2 di	Insert Calculation from External Model or Library Type Location	Open Utils File C:\Users\pmenegay\Desktop\M	Q9_Sensor\1 Save	Work\AirForce_DUE_Feb8_2017\
Graph Settings II daw=1.2 dil 11 Lsp=2.0 d	Function Library	n=passed_in_object.n		System Models
Graph 1 12 Lradar=3.	1 O ModelCenter	one=passed_in_object.one		Aircraft1 x Structure
Title	© Excel	Dh0=passed_in_object.Dh0		Aircraft1_x_FuelTank
X Label x axis 1 LocalFunc	Copen Model	Le0=passed_in_object.Le0		Aircraft1_x_SAR
Y Label y axis		We0=passed_in_object.We0 He0=passed_in_object.He0		Aircraft1_X_FLIK
Plot 1	Function	W0=passed_in_object.W0		GroundControlStation
Add Graph Add Plot elet 3D	Input Variables	C0=passed_in_object.C0	ovided inputs	
Multigraph Settings	Output Veriables	r = FunctionLibrary.SAR_3413(vx,dt	t,h,Pt0,Daz,Dh,rhor,f,fref,aw,Lsp,Lradar,Fn,alpha,PI,k,c,	
	Output variables	#get the outputs in a convenient n	umerical form	
	Generate and Insert Code Go	rs = r['rs'].magnitude		
	Insert Python Code	R = r['R'].magnitude		
	def (fn) class import	V0 = r['V0'].magnitude		
		W = r['W'].magnitude		
	for while print	Pt = r['Pt'].magnitude		
	lict dist road file	return [SNR,rs,R,V0,V,W,Pt,C]	=	
	list dict lead life		*	4
	write file	•	•	place Add to Library
			OK Cancel Help	
				delity Level 📃 Parameters
				utputs 🔲 V/V Level
				File
	Edit Requirements	Generate System Of Systems	Generate Acausal System Edit Library M	odel Select Model Above Edit System Model
	Trade Studies	Generate	Generate Re-generate	



SoS Integration - 2

• Next, SoS is built by specifying which systems to unite. A flat hierarchy of all contained subsystems results. Variable links must be created.

🔩 SimModelBuilder		
File Help		
Simulator Equation Solver		
Project Files Simulation	User Level	
	🗇 Decision Maker 💿 System Modeler 💿 Model Creator	
Basic	Subsystems	Library/System Options
Project Name MQ9AGS_VerilogConnection.hpj .hpj	Aircraft1_x_SAR Aircraft1_x_FLIR Aircraft1_x_GBU GroundControlStation_x_Mission Satellite_x_Comm	Directory C:\A\PythonWork\AirForce_DUE_Feb8_2017\Work\Code\!
Target Path C:\Users\pmenegay\Desktop\MQ9_AGS	#\$Subsystem	Library Models System Models
System Name MQ9AGS	#\$Type	Aircraft1_x_SAR Aircraft1_x_Structure Battery UK Aircraft1 x FuelTank
Time Parameters	MissionModel	Communications_UK Aircraft1_x_SAR
Thegin 0.0 sec	#\$FidelityLevel	Compressor Aircraft1_x_FLIR Computer LIK Aircraft1_x_GPL
	1	FuelTank GroundControlStation_x_Mission
Tend 82800.0 sec	#\$VVLevel	HardDrive_UK Satellite_x_Comm
Deltat 3600.0 sec	# #\$Deltatl.ocal	Orbit_UK
StringLists	3600.0	Orbit_UKTB
hattan: string - l'connected' 'disconnected'	#\$Parameters	Plant Radar UK
communications string=['transmit' 'receive' 'off']	altitude_in=40000.0 😑	SolarPanel_Excel
computer_string=['on','off']	loiter_time_in=12.0	SolarPanel_Hi
harddrive_string=['erasing','saving','off']	speed=175.0	SolarPanel_UK
material_string=['silicon','thin_sheet_amorphous_si','gallium_a	init_distance_to_target=525.0	ThermalManager_MC
rsenide','multijunction_gainp_gaas','indium_phosphide']	target_observation_distance=6.0	ThermalManager_UK
radar_string=['standby','on','off','warmup']	* \$Inpuls SNRcar-Satellite v Comm SNRcarEvent	
	SNRfir=Satellite x Comm SNRfir/Fvent	
	weight_gbu=Satellite_x_Comm.weight_gbu:Event	
	#\$Outputs	
	altitude=40000.0:Event	
	loiter_time=13.0:Event	
	distance_to_target=1000.0:Event	
	aircraft_mode=0.0:Event	
	start_loiter_time=200000.0:Event	Add to System Replace Add to Library
	end_loiter_time=200000.0:Event	Replacement Rules
Descriments	bombs dropped=0.0:Fvent	Model Type Fidelity Level Parameters
Kequirements		Inputs Outputs V/V Level
Edit Requirements	Add Subsystem Change Variable Names Check Inputs	- Edit Calculations / Litils File
Trade Studies	Generate	
Edit Trade Studies	System Of Systems Acausal System Regenerate Acausal Scin File	Edit Library Model Edit System Model

Simulation Results

• The SoS is simulated using the System Verilog engine and results are shown in a timing diagram:

<u>الم</u>	Stimu	lusAndResults.btim *																				×
Add Add	Signal (Clock (Add Bus Delay Setup S Add Spacer Hold Text 1	Sample HIGH LOW TRI	/AL INVa →=	N MI	LO HEX	Q + Q F Q - Q R	ViewVa Class M	riables Pro ethods	operties												
		0.000ms	0.000ms		0ks	5ks	10ks	15ks	20ks	25ks	30ks	35ks	40ks	45ks	50ks	55ks	60ks	65ks	70ks	75ks	80ks	8
0	л		MQ9AGS.	clock																	<u> </u>	
1	л	Aircraft1	1_x_Structure.total_w	/eight	7900	9847 968	85 9523	9361	9199 90	37 8875	6 <mark>)(8713)</mark> (8	551 838	8227	8065 79	03 774	1 (7579)	7417 72	55 7093	5931 57	769 5607	5445	Ξ
2	л	Aircraft	t1_x_FuelTank.fuel_w	/eight	3764	<u>3602 (34</u>	40 3278	(3116)	2954 27	92 2630	2468 2	306 214	4) <mark>(1982)</mark> (1820 (16	658 <mark>(149</mark>	6 (1334 (1	1172 (10 ⁻	10) 848)	686 5	24 (362	200	_
3	лı		Aircraft1_x_SAR	.SNR		0)	(3.8	8330342	518						0		_
4	л		Aircraft1_x_SA	AR.W									83									_
5	лı		Aircraft1_x_S/	AR.Pt									320									_
6	л		Aircraft1_x_S	AR.C									1500000									_
7	л		Aircraft1_x_FLIR	.SNR		0)	(2.0	7371672	473				_(0		_
8	Л		Aircraft1_x_FL	IR.W									100									_
9	лı		Aircraft1_x_FL	R.Ae		0.1										_						
10	л		Aircraft1_x_FI	IR.C									1000000)								_
11	лı		Aircraft1_x_F	LIR.P									280									_
12	лı		Aircraft1_x_GBU.w	reight							1000						_(0		_	
13	лı	GroundControl	IStation_x_Mission.al	titude							40000								_			
14	ហ	ndControlStation_x_N	lission.distance_to_t	arget	(1000)	350 (17	<u>′5)</u>				6 (181) 356) 531)						706 8	81 (1056	1231	_		
15	л	GroundContro	olStation_x_Mission.s	peed									175									_
16	лл	oundControlStation_x	_Mission.bombs_dro	pped							0					(1			_
17	л	GroundControl	Station_x_Mission.Po	letect			0			_(0.8406	0378970	3					_
18	л	GroundCon	trolStation_x_Mission	n.Phit			0			_(0.9643	2600665	3					_
19	л	GroundCon	trolStation_x_Mission	ı.Pkill		0				_(0.8106	1609570	1					_
20	л	GroundContro	olStation_x_Mission.	Pshot		0				_(3.333708	357252e-0	002					_
21	лı	GroundControlSta	tion_x_Mission.Shot	Down									0									_
22	л	GroundControlStati	ion_x_Mission.Killed	arget			0									1						
•			111	•	•				111													•
Para	amet	er Window																			Hide	,
Sti	mulus	AndResults.btim* ×																				



Requirements Modeling

• A template is used to create requirements sentences in English which are translated to Python for analysis. Requirements can also be written directly in Python.

Select Template		Requirement #1: ""The GroundControlStation_x_Mission shall have a Pkill that is greater than 0.8 while the time is greater than to 43200.0"""
Select Sentence Template: The <subsystem> shall have a <parameter> that is <equality> <number> The <subsystem> shall have a <parameter> that is <equality> <number> while the <parameter The <subsystem> shall have its <parameter> set to <string> while the <parameter> is <equality> The <subsystem> shall have its <parameter> set to <string> if the <parameter> is <equality> The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter> set to <string> when the time is equal to <number The <subsystem> shall have its <parameter <subsystem=""> shall have its <parameter< pre=""></parameter<></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></number </string></parameter></subsystem></equality></parameter></string></parameter></subsystem></equality></parameter></string></parameter></subsystem></parameter </number></equality></parameter></subsystem></number></equality></parameter></subsystem>	ter> is <equality> <number> lity> <number> > <number> ber> ~</number></number></number></equality>	<pre>if GroundControlStation_x_Nission_time > 43200.0: if GroundControlStation_x_Nission_Pkill > 0.8:</pre>
	OK Cancel	Time #1 0.0 N/A 3600.0 N/A 7200.0 N/A 18900.0 N/A 14900.0 N/A
Requirements Requirement #1 The <subsystem> shall have a <parameter> that is <equality> <number< td=""><td>>> while the <parameter> is <equality> <number></number></equality></parameter></td><td>18090.0 N/A 25200.0 N/A 25200.0 N/A 28800.0 N/A 32400.0 N/A 32400.0 N/A 35000.0 N/A 35000.0 N/A 35000.0 N/A 35000.0 N/A 35000.0 N/A 43200.0 N/A</td></number<></equality></parameter></subsystem>	>> while the <parameter> is <equality> <number></number></equality></parameter>	18090.0 N/A 25200.0 N/A 25200.0 N/A 28800.0 N/A 32400.0 N/A 32400.0 N/A 35000.0 N/A 35000.0 N/A 35000.0 N/A 35000.0 N/A 35000.0 N/A 43200.0 N/A
Select a subsystem Select a parameter FLR FLR FUR GBU GBU SaR Stactarte SaR Stactarte SaR Stactarte SaR Stactarte Sartellite Comm Sartellite Comm Satellite Statellite Satellite Comm Satellite OK Cancell Water Satellite Satellite Satellite OK Cancell Mater OK Cancell OK Add Requirement Delete Requirement Add Requirement Delete Requirement	Equality Select One: equal to less than not equal to not equal to OK Cancel	Subsystem Sde00.0 True GroundControlStation_XMission_Pkill,GroundControlStation_XMission_Pshot Sde00.0 True GroundControlStation_XMission_altitude_in=15000.0to50000.0,levels=8 Sde00.0 True GroundControlStation_XMission_altitude_in=15000.0to50000.0,levels=8 Sde00.0 True GroundControlStation_XMission_altitude_in=15000.0to50000.0,levels=8 Sde00.0 True GroundControlStation_XMission_Pkill,GroundControlStation_XMission_Pshot Samples: 3 Smoles: 3 Glaubiton True Gated.0 True Formadiant station_XMission_Pshot Formadiant Results over time by combining trade studies for each requirement Tsoudo: 0 True Formadiant study point is combined with the others using AND. WANITIK6: If a point is N/A and another is True, the combined result is N/A. Your requirem Your requirem Your requirem Time #1_res 100.0% of the Ide0.0 14400.0 N/A 22600.0 N/A 22600.0 N/A 23000.0 N/A 24000.0 N/A 25000.0 N/A 2600.0 False 54000.0 False 54000.0
Requirements Requirement #1 The GroundControlStation_x_Mission shall have a Pkill that is great	ter than 0.8 while the time is greater than to 4326	66409.0 False 72009.0 False 75609.0 False 72020.0 False 72020.0 False Requirement 1 across trade studies held true 0.0% of the time Requirement 1 across trade studies held true 0.0% of the time when applicable Requirement 1 across trade studies was N/A 56.5217391304% of the time Subsystem

SoS Trade Studies

• Operate over the entire SoS and perform full simulations for each point. DOE, Monte-Carlo, and single variable optimization is available.

18



SYNAPTI**CAD**

AADL Integration

• Simulation models are architecturally analogous to AADL language.





Acausal Solver/Driver

• Acausal simulation means that what we consider an input or output is not fixed. All models are solved as a system of equations using two methods:





Acausal Method 1

• Each system is rendered as an equation to be solved at every timestep. The simulator is the driver and the equation solver runs behind the scenes:



0.17022134363

SYNAPTI**CAD**

48

Aircraft1 x FLIR GroundControlStation x Mission.Ae

Acausal Method 2

• The entire SoS is viewed as an equation to be solved over the entire span of time of the simulation. The equation solver is the driver and the simulator runs behind the scenes.

SolverCAD C:/A/PythonWork/solvercad	l/engine/ExampleProblems/	MQ9AGS b.scin						
File Help		<u></u>						
Cimulator Equation Column								
Simulator Equation Solver								
	🔷 🕅 🖸 !	🛹 🗟 fx 🔇 🍞	(32)		i			
New Open Save Save Guess Errors Solve	Sort Adjust Check Solve	Graph Generated Local Time Remov	ve Show	Window Tab Lock Help	About			
File Guesse	Solver	Mise	valiables	Lavout H	lelp			
Desumentation	Solver	IWISC		Layout		Solution		
	1 Ainenef+1 x Stru	tuna total unight LocalFunctions	CottimDocu	ults/Aincoaft1 v Stow		1 Colution 1 of 1		
	2 Aircraft1_x_Fuel 3 Aircraft1_x_SAR_ 4 Aircraft1_x_SAR_ 5 Aircraft1_x_SAR_	<pre>lank_full_weight=LocalFunctions.Ge aircraft_mode=LocalFunctions.GetSis SNR=LocalFunctions.GetSimResults(A rs=LocalFunctions.GetSimResults(A)</pre>	tSimResults(/ inResults(/ ircraft1_x	ts(Aircraft1_x_Structure_ Aircraft1_x_Structure_ x_Structurebase_weig _Structurebase_weig	eb ight sht,	2 Aircraft1_x_FLIR_Ae = 0.1702 3 Aircraft1_x_FLIR_C = 1161202 4 Aircraft1_x_FLIR_F = 0.27860 5 Aircraft1_x_FLIR_F = 0.27860	1165469305916 5784799999 di 741803900003 d 47828099995 di	dimensionless mensionless imensionless mensionless
P	5 Aircraft1_X_SAK_	_R=LocalFunctions.GetSimResults(Ain	cratti_x_3	Structurebase_weigh	IT,A	$6 \text{ Alrcraftl}_X \text{-} \text{FLIR}_R = 15552.6$	343/3999999 dl	dimensionless
Settings	8 Aircraft1 x SAR	V=LocalFunctions GetSimResults(Air	$craft1 \times $	_structure_base_weigh	ht Δ	$8 \text{Aircraft1} \times \text{FLIR} S = 4.23564$	190609700001e-0	7 dimensionless
Generate function	9 Aircraft1_x_SAR_		craft1_x_9	Structurebase_weight	nt,A	9 Aircraft1_x_FLIRSNR = 3.529	70755081 dimen	sionless
Function return value:	10 Aircraft1_x_SAR_ 11 Aircraft1_x_SAR_	_Pt=LocalFunctions.GetSimResults(Ai _C=LocalFunctions.GetSimResults(Air	rcraft1_x_ craft1_x_9	_Structurebase_weig Structurebase_weight	ght, : nt,A :	10 Aircraft1_x_FLIRV0 = 0.0237 11 Aircraft1_x_FLIRW = 222.066	88321751599999 95652900001 di	dimensionless mensionless
Units for generated function	12 Aircraft1_x_FLIR	aircraft_mode=LocalFunctions.Get	imResults	(Aircraft1_x_Structure	re	12 Aircraft1_x_FLIRaircraft_mo	de = 1.0 dimen	sionless
Base Units OUserInput Units	13 Aircraft1_x_FLIR	C=LocalFunctions.GetSimResults(A)	rcraft1_x	_Structurebase_weigh	ght,	13 Aircraft1_x_FLIR_tau = 0.352	73913024300002	dimensionless
Costiniput_onits	14 Aircraft1_x_FLIR		rcraft1_x	_Structurebase_weigh Structure_base_weigh	gnt, .	14 Aircratti_x_Fuellanktuel_we	light = 1820.0	aimensionless
				buse_weig		4		4
	Knowns		8	Guesses		8	Local Functions]
Graphics Editor	44 Aircraft1_x_FLIR 45 Aircraft1_x_FLIR 46 Aircraft1_x_GBU 47 GroundControlSta 48 GroundControlSta 49 GroundControlSta	_P0=280.0 dimensionless _C0=1000000.0 dimensionless weight_initial=1000.0 dimensionlesf tion_x_Mission_altitude_in=40000.0 tion_x_Mission_loiter_time_in=12.0 tion_x_Mission_speed=175.0 dimension	s dimens dimens onless	38 GroundControlStat 39 Satellite_x_Comm_ 40 Satellite_x_Comm_ 41 Satellite_x_Comm_ 42 Satellite_x_Comm_ 43 Satellite_x_Comm_	tion_x_M orbit_ aircra altitu distan speed,	<pre>Mission_KilledTarget,0.9,1.1 ^ altitude,105665472.0,1291466 aft_mode,0.9,1.1,1,dimensionl ude,36000.0,44000.0,1,dimensi nce_to_target,5.4,6.6,1,dimen ,157.5,192.5,1,dimensionless</pre>	1 import time 2 import Util 3 import Glob 4 schome, user 5 6 import sys	s als home,exhome,pubhome = Globals.{
	50 GroundControlSta	tion_x_Missioninit_distance_to_ta	distanc	44 Satellite_x_comm_	DOMDS_ SNRsar	_aropped,-5.0,5.0,1,aimension	8 sys nath an	pend('C:/A/PythonWork/AirForce_ pend('C:/A/PythonWork/solverca
Graph Settings	52 Satellite_x_Comm	_orbit_altitude_in=117406080.0 dir	ensionl	46 Satellite_x_Comm_		ir,2.7,3.3,1,dimensionless	9 import Trad	eStudy
Title:	53 t=43200.0 dimens	ionless	II	47 Satellite_x_Comm_	weight	t_gbu,900.0,1100.0,1,dimensic	10 res_dir = '	C:/Users/pmenegay/Desktop/MQ9_/
x: x axis title:	GroundControlSta	ion_x_MissionPkill=0.9 dimension	less	8 Aircraft1_x_FLIR_	Ae,0.1	1,0.2,1,dimensionless	11	
	•	111	• I	•			12 def SaveAns	wer_GetSimResults(arguments,ca.
y: v axis title:	Function Units			8	Random G	Guess Settings		Error Criteria Settings t
z: z axis title:	1 LocalFunctions.G	SimResults, dimensionless, 0			1 UseR	RandomGuesses True		1 ErrCrit=1.0E-7
	2 LocalFunctions.G	tSimResults,dimensionless,1		=	2 NumR	RandomGuesses 1		2 SolveMethod=Numerical
	4 LocalFunctions.G	etSimResults.dimensionless.3			4 Rand	domGuessMethod LatinHvpercube		4 SolutionToUseForAdjustment=1
	5 LocalFunctions.G	tSimResults, dimensionless, 4			5			5 PercentSpreadAroundSolution=5
	6 LocalFunctions.G	tSimResults, dimensionless, 5						6 InputFileToModify=y_junk.scin
	7 LocalFunctions.G	SimResults, dimensionless, 6						7
	8 LocalFunctions.G	tSimKesults, dimensionless, 7						
	10 Local Functions	stSimResults dimensionless 0						
	11 LocalFunctions.G	etSimResults, dimensionless. 10						
Simple Graph Settings Multigraph Setting	12 LocalFunctions.G	tSimResults, dimensionless, 11		-				
	1	tet e la la la la	(lotuput of	ion Urobiom				

Some Observations

- Scale up.
 - Human scalability: our ability to see, understand, and manipulate large SoS models.
 - Computational scalability: Multiple nested engineering models, solved iteratively by equation solver, are possible. Trade studies make this worse.
- "Explosion" of data caused by trade studies.
- English requirements are backed up by requirements expression through a computer language (i.e. Python).
- Requirements modeling can be enforced through equation solver.
- The acausal capability allows for many scenarios without having to rebuild the model. Inputs/outputs can be switched.
- This tool is designed to be linked from 3rd party systems engineering tools such as AADL or SysML.



Backup Slides



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.

e Help							
quation Solver							
olve any system of simultaneous equations							
Click here to generate a scriptwrapper for this example Click here to generate a function for this example unction return value? P2 Use base units in generated function? Sease_Units UserInput_Units	$\label{eq:second} \begin{split} & \texttt{FEquationsBegin} \\ & \texttt{v}_{(mu/(Rearth+z)^{**0.5})} \\ & \texttt{r}_{((2,0^{*}))(mu^{*0.5})^{*}(Rearth+z)^{**1.5} \\ & \texttt{h}_{=}(Rearth+z)^{*} \\ & \texttt{nerrgy} = -10^{*mu}(20^{*}(Rearth+z)) \\ & \texttt{t}_{(h^{*3})}(mu^{*2})^{*} \\ & \texttt{heta} \\ & \texttt{LocalFunctions Cos(Ninety-ThetaShade2} \\ & \texttt{HetaShade1} = ThreeSixty-ThetaShade2} \end{split}$	92)=Rea	arth/(Rearth+z)	Sc T Th Th Th Th Th Th fa h	elutio = 580 netas netas nergy kevan = 523	n 1 of 1 106655553 second = 0.0108310862147 radian hadel = 5.13011155582 radian hade2 = 1.1530731516 radian = -28561184.1327 meter**2/second**; = 0.0 dimensionless 739279452.8 meter**2/second	2
et hints for filling out form:	#KnownsBegin mu=3986E14 m**3/sec**2 PI=3.141956 dimensionless Rearth=6378000.0 m z=600002.0 meter t=10.0 sec Ninety=1.5707963267949 rad Ninety=1.5707963267949 rad InreeSity=6.28318530717959 rad ispos=1.0 dimensionless	*	#GuessesBegin v,1000.01000.01.m/sec TA800.28400.01.sec h,10E10.264E11.1.m*12/sec*12 Thets0.75.30.1430.1.m*12/sec*12 Thets0.715.37.1.rad Thets0.715.73.1.rad Thets0.715.71.rad Thets0.716.28.1.rad is_in_sun_0.0.10.1.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	Ē
StartPlot						#LOCAIFIISEIIU	
xdataydata_list=x,y;x,z	4	т. Р	4		-	4	
marker_list=ro-Do- title=the big one legend_list=red,blue klabel=x axis Jabel=y axis	LocalFunctions.Cos, dimensionless LocalFunctions.InShade, dimensionless	*	UseRandomGuesses True NumRandomGuesses 10		•	ErrCrit=1.0E-7	* III
y2label=z axis	4	•	4		_	4	•
<	#ConstraintsBegin	*	#OptimizationBegin #OptimizationEnd		^	#DOEBegin #DOEEnd	Â
imple Graphics Settings							
x column hdr x axis title: title here							
y column hdr y axis title: title here							
z column hdr z axis title: title here			4		-	4	
raph Title Your Title Plot Graph		r	X	,		×	P



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	e2.sim	- • ×
File Edit View Tools Help		
Simulator ×		
User Level		
Obecision Maker System Modeler Model Creation	tor	
Actions		
New Load Save Sav	ve As Check View Generate OpenGen LoadVerilog RunSim	
Basic	Subsystem Library/System Options	
Project Name Satellite_1_VerilogConnectio .hpj	New Name Add Directory C\src\hdbase\scriptsim\Example	es\Satellite\
Target Path C:\A\AATest32	HardDrive Orbit Radar SolarPanel ThermalManager	System Models
System Name Sat	#\$Subsystem Battery_UK	Battery
TimeParameters	Orbit Computer_UK	Computer
Tbegin 0.0 sec	#\$Type HardDrive_UK	HardDrive
Tread and the	#\$Fidelityl evel Radar_UK	Radar
Tend 5000.0 sec	0 SolarPanel_UK	SolarPanel
Deltat 1.0 sec	#\$DeltatLocal	Inermaimanager
StringLists	1.0	
material string=l'silicon' thin sheet amorphous si' 'g	#\$Parameters	
allium_arsenide','indium_phosphide','multijunction_g	mu = 398600.0	
ainp_gaas']	Rearch = 0578.1	
battery_string=['disconnected','connected']	z_sat_pengee = 5000	
computer_string=['off','on']	i=40.0	
radar_string=['off','on','warmup','standby']	Omega = 30.0	
narddrive_string=[off, saving , erasing]	omega = 20.0	
communications_sumg=[on, transmit, receive]	JD_base = 2450000.0 Add to System Replace	Add to Library
	JD_relative_to_base = 7287.9	Add to Elorary
	lat_gnd = 34.2	
	Ion_gnd = 241.81	ameters
	Change Variable Names Check Inputs All	
	Edit Calculations / Utils File	
	Edit Library Model Select Model Above	Edit System 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								
	Options		Utils File						
	Insert Calculation from Exte	ernal Model or Library	Open Utils File C:\src\hdbase\scriptsim\Examples\Satellite\ Save						
	Туре	Location	import math						
	Function Library		import time		[]				
	ModelCenter		<pre>def ConvertToVerilogValue(val, str. </pre>	inglist_name=''):	E	E .			
	C Excel		val = str(val) 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)						
Link to pre-built models in	Tanetion		elif(val_type is int): return eval(val)						
SolverCAD. ModelCenter.	Input Variables		elif(val_type is bool):						
	Output Variables		<pre>if(val == 'False'): return 0</pre>						
Excel, or your own	Generate and Insert Code		else:						
n no 210 0 2		00	return 1						
program.	Insert Python Code		except Namecrror: #means it is a string						
	def (fn.) class for while list dict write file	import print read file	<pre>stringlist = FindStringLis int_val = ConvertStringInS return int_val def SetDelayAndVerilogOutputs(pas delta_wait = passed_in_object. passed_in_object.Qinternal_vvc nassed in object.temn sat vvo.</pre>	<pre>tInStringLists(stringlists, tringListToInt(stringlist,va ed_in_object): Tcurrent - float(passed_in_c .delay = (str(delta_wait),"T delav = (str(delta_wait)."Tr</pre>	<pre>stringlist_name) al) object.Sim.time()) Transport") ransport") OK Cancel Help</pre>	-			
Buttons for quickly creating common Python constructs.			Edit the calculation for any subsystem	ation file tem in Pythor	ı.				
					SYNAPTI	CAD			

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

rganize 🔹 🛛 🥞 Open	Include in library Share with	» 📕 •		2 8 8 2
ame	Date modified	Туре	Size	
Plant	12/1/15 4:23 PM	File folder	1 #\$Subsystem	
Compressor	12/1/15 4:22 PM	File folder	3 #\$Type	
Orbit_UK	11/28/15 10:28 AM	File folder	4 OrbitalMechanicsModel	
Orbit_Circular	11/17/15 4:34 PM	File folder	5 #\$FidelityLevel	
SolarPanel_Hi	11/16/15 3:10 PM	File folder	6 Ø 7 #≸Deltatiocal	
SolarPanel_Lo	11/16/15 3:10 PM	File folder	8 1.0	
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 = 63/8.1	
ThermalManager_UK	11/13/15 2:40 PM	File folder	12 z sat apogee = 510.0	
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	
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 23 lon_exp = 26.9	
Orbit UK Date	III		24 exp_plusminus = 30.0 25 theta = 0.0 26 #\$Inputs	
File folder	noonco: xa/xoyxo xoxo ren		27 None 28 #\$Outputs 29 true anomaly = 0.0	
			30 lat = 0.0 31 lon = 0.0	
ModelLi	• Orbit_uk • • • Search Orbit	UK	32 15_1n_sun=False 33 is over and sta=False	
rganize 🔹 📑 Open 🦄	Burn New folder	81 · 🔟 🤇	34 is_over_exp_zone=False	
ame	Date modified Type	Size	35 altitude = 505.0	



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.







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.



Orbit

PAR JD_relative_to_base=7287.9

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 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 theta=0.0

INP None

OUT lat=0.0

OUT lon=0.0

PAR exp_plusminus=30.0

OUT true anomaly=0.0

OUT is in sun=False

OUT altitude=505.0

OUT is over gnd sta=False

OUT is over exp zone=False



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 intensitv=1367.0

INP is in sun=Orbit.is in sun



32

Results

	11.52ks 0.000ms	0ks 1	(s 2ks	3ks 4ks	5ks	6ks	7ks 8ks	9ks 1)ks ∣11ks	11519201 r	ns 13ks 1	4ks 15k	ks 16ks	17ks	18ks	19ks	201
л	Sat.clock									n							
1 л	Sat.clock_reg									0							
2 л	Orbit.true_anomaly									0.16911884	7655017						-
3 л	r Orbit.lat									18.6806781	407817						-
4 л	ſ Orbit.lon		1							220.532489)34						
5 л	Orbit.is_in_sun[31:0]	(In:.)	InSun	X	InShade	χ	InSun	X	InShade	InShade	InSun	X	InSha	de)	InSur	1 🗲	-
6 л	Orbit.is_over_gnd_sta[31:0]	c)	Not_(Over) Ov	er (N	ot_Over)(Over _		Not_Over	r	0	iver (No	ot_Over	1
7 л	Orbit.is_over_exp_zone[31:0]	Not_Over	Over (Not_Over		(Over)		Not_Over	Not_Over	(Ov.)			Not_Over			
8 л	SolarPanel.power_output									0							-
9 л	ThermalManager.Qinternal	(.) 300	(410)	300	(36	0 300	0 (410)	300	χ	360)	300 410		300) 3	360	300	1
10 л	Computer.satellite_power_draw									-360							-
11 л	ThermalManager.temp_sat									244.412675	5100314	ι					-
12 л	Battery.total_energy_charge									1503634.89	9507634						-
13 л	f HardDrive.amt_used	<u> </u>								212							
14 л	Radar.power	200	(300)		200		(300)		200	200	300)			200			1
15 л	r HardDrive.power	0)	(10)	0) 1	0 (0)(10)(0	χ	100 (0 (10)		0)(10	0	1
16 л	r Computer.power							100		100							1
17 л	Communications.power	0	()) 5) (0	X	50 <u>0</u> (0)()	50)	0	1
18 л	Computer.radar_command[31:0]	Standby	<u>(On)</u>		Standby) On)		Standby	Standby)On)			Standby			1
19 л	omputer.harddrive_command[31:0]	i.) Off	Saving (Off)Eras	ing) Of	f (Saving)	Off) ((Erasing	Off Sa.)		Off)Era	asing)	Off	1
20 л	r.communications_command[31:0]	1)	0	ff	(Tran	smit)		Off)(T	Transmit		Off		(Tra	insmit)	Off	1
21 л	Radar.power_state[31:0]	Standby	<u>(On)</u>		Standby		(On)		Standby	Standby)On)			Standby			1
22 л	Communications.transmit_rate	0	())(1) (0)	100		0		(10	0	Ŧ
		4														•	

-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.

Constrain_Batt_SolarPanel.sv	23
<pre>program Constrain_Batt_SolarPanel;</pre>	
class Constrainer; rand integer cr.to_aar; rand integer cr. rand integer aar;	
<pre>constraint c1 { cr_to_aar < 20000;} constraint c2 { cr_to_aar > 10000;}</pre>	
<pre>constraint c3 { cr >= 55000;} constraint c4 { cr <= 200000;}</pre>	
<pre>constraint c5 { aar >= 4;} constraint c6 { aar <= 10;}</pre>	
<pre>constraint c0 { cr_to_aar == cr / aar;} endclass</pre>	
<pre>initial begin Constrainer obj = new(); int wfile; fwfine vfile; fwfine vfile; fwfile;</pre>	<pre>cr_to_aar = %0d*,orj.cr,obj.aar,obj.cr_to_aar);</pre>





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.

