

ARCHITECTURE-DRIVEN ASSURANCE

JENNIFER DAVIS, PH.D. TRUSTED SYSTEMS OCTOBER 2019



APPROACH: ARCHITECTURE-DRIVEN ASSURANCE

- 1. Architecture model is correct
 - Properties, structure, behavior, interaction of components, interfaces, contracts
 - Analyzable
- 2. Components are correct
 - Consistent/realizable contracts
 - Components verified to implement contracts
- 3. System does what the model says
 - No other information flows (memory safety, isolation)
 - OS executes model correctly (incl. timing)
- 4. System implementation corresponds to model
 - Automatic build from component and architecture models









SELECTED PROGRAMS LEVERAGING ARCHITECTURE-DRIVEN ASSURANCE

- DARPA High Assurance Cyber Military Systems (HACMS)
- DARPA Cyber Assured Systems Engineering (CASE)



HIGH ASSURANCE CYBER MILITARY SYSTEMS (HACMS) TECHNOLOGIES

Open source tools, languages, software

- 1. Architecture modeling and analysis tools (AADL)
 - Assume-Guarantee Reasoning Environment (AGREE)
 - Architecture-based assurance cases (Resolute)
- 2. Ivory/Tower embedded Domain Specific Languages
 - Memory safe component software
 - Code generation from high-level specification
- 3. seL4 formally verified OS kernel
 - Isabelle/HOL proof of correctness
 - Security properties proven to binary level
- 4. Automated build from models
 - Support for seL4, eChronos, VxWorks, Linux





galois







CYBER ASSURED SYSTEMS ENGINEERING (CASE)

- The goal of CASE is to develop the necessary design, analysis and verification tools to allow system engineers to design-in cyber resiliency and manage tradeoffs as they do the other non-functional properties when designing complex embedded computing systems
 - Cyber resiliency means that the system is tolerant to cyberattacks in the same way that safety critical systems are tolerant to random faults – they recover and continue to execute their mission function
 - Cyber security requirements are addressed today by penetration testing late in the development, resulting in expensive rework
 - Cyber requirements are often "shall not" statements about the system, and so are not testable (formal methods required)









APPROACH

- Start with initial design, new or legacy
 - Federated avionics system
- Generate new cyber requirements
 - Possibly based on modified system architecture
- Tool-assisted transformations of system architecture
 - Satisfy cyber requirements
 - Manage other design trade-offs
 - Insertion/synthesis of high-assurance components may be needed
- Verification of cyber resiliency
- Generate system from architecture model





[&]quot;After" (Integrated, Cyber-resilient)











Edit Navigate	Search Project Run (B S J C B I I I I I I I I I I I I I I I I I I	OSATE Analyse CASE AAHAA W Cyber Requirer Cyber Resilienc Component Ar System Build IS; IAS; IAS; Buses;	indow Help nents > Generate Requirements > GearCase by > Import Requirements DCRYPPS halysis > Formalize Requirement				
Import C	Cyber Security Re	equiremen <mark>t</mark> s		View/Edit Requiremen	at a second s		
Status	Туре	ID	Short Desciption	Status	Formalize V		
Formalize	well_formed	Req003_FlightPlanner	The FlightPlanner shall only accept well-formed messages fror	Generation Tool	DCRYPPS		
ToDo	monitored		The output of Third-party software shall be monitored for con The FlightPlanner shall only accept messages from a trusted G	Type	well_formed		
ToDo	trusted_source		The FlightPlanner shall only accept messages from a trusted G	ID	Reg003 FlightPlanner		
1000	Isolateu		Third-party software shall be isolated from critical component	Descrption	The FlightPlanner shall only accept well-formed messages from the GroundStation		
				Component	SW::SW.Impl.FPLN		
				Reason for omission	N/A		
<			>				





😚 runtime-osate2 - CASE_Si	mple_Example_\	V4/UAS.aadi - OSATE2
File Edit Navigate Sear	ch Project R	un OSATE Analyzik CASE AAHAA Window Help
📑 • 🔟 🖏 🐸 🗉 🖬	1 1 x 1 1	🔤 🔜 🔁 🗗 💆 Cyber Requirements 🔹 Generate Requirements 🔺 GearCase
💞 A 😂 🖉 A 😁 🗖	SW.aadl	Cyber Resiliency Import Requirements DCRYPPS UAS.aadl 2: Component Analysis Formalize Requirement System Build System Build Formalize Requirement
8 😫 🗵	1200	thread FlightPlanner
> CASE_Simple_Exa A	121	The FlightPlanner is an abstraction for UXAS.
CASE_Simple_E	122	It accepts a command message containing a map and flight pattern, and generates a mission.
🗸 🍉 diagrams	123	The FlightPlanner also has access to a No-Fly zone database, which it uses to generate the mission
MC_Missie	124	to avoid specified no-fly zones
SW_SW_In	125	features
UAV_UAV_	126	flight plan: out data port Mission;
UAS_UAS_	127	recy map: in event data port RF Msg.Impl;
Plusio Contr	128	request nofly zones: out event data port Map;
> Constances	129	nofly zones: in event data port MapArray;
> R FC.aadl	130	position status: in event data port Coordinate.Impl;
> 📑 GS.aadl	1310	annex agree {**
> 🔐 MC.aadl	1320	assume Req001 FlightPlanner "The Flight Planner shall receive a valid message from the Ground Station" : VALID MESSAGE(recv map);
> 🔐 SW.aadl	133	guarantee Reg002 FlightPlanner "The Flight Planner shall generate a valid mission" : good mission(flight plan):
> 💦 UAS_Buses.aa	134	assume Req003 FlightPlanner "The FlightPlanner shall only accept well-formed messages from the GroundStation" : WELL FORMED MESSAGE(recv map);
) > UAS.aadl	135	**
> II, UAV.aadl	136	end FlightPlanner;
> Case_lest	137	
Filter Test	1380	thread implementation FlightPlanner.Impl
> A Isolator Example	139	end FlightPlanner.Impl;







mple_Example_	4/UAS.aadi - OSATE2
ch Project F	un OSATE Analysis CASE AAHAA Window Help
1 De 1	E Cyber Requirements > Generate Requirements > Generat
th Project F SW.aadl 120@ 121 122 123 124 125 126 127 128 129 130 131@ 132@ 133 134 135 136 137 138@ 139	<pre>a OATI Ladar CASE AvAAA Wedee: Hep</pre>
	260 261 with CASE_Model_Transforma 262
	nple_Example_V h Project R SW.aad 120@ 121 122 123 124 125 126 127 128 129 130 131@ 132@ 133 134 135 136 137 138@ 139



268 **};





😵 runtime-osate2 - CASE_Sir	mple_Example_V4	/SW.aadl - OSATE2						
File Edit Navigate Search	h Project Rur	OSATE Analyses	CASE AAHAA Window	Help				
📑 • 🔛 🐚 👑 🖷 🛅	回 🔀 🛢 🖩		Cyber Requirements	>			🖴 骗 Export to JSON 🛛 Touch 🕻	8 9
			Cyber Resiliency	>	Model Annotations	1		
	D ON and 12	Dillord	Component Analysis	>	Model Transformations	>	Add Attestation Manager	
	SW.aadi 25	UAS-aadi	System Build	>	Proofs	>	Add Filter	
	203	uart_recv: uart_send:	out event data port CA	SE Mode	UART	Add Isolator		
> CASE_Simple_Exa A	205	wifi_recv:	cv: in event data port CASE_Model_Transformurcions::CASE_WIFI_				Add Monitor	
v 😕 diagrams	205	wifi_send: annex agree (**	out event data port CASE_Model_Transformations::CASE_WIFI Reg001_SW "The Mission Computer shall output a valid mission			WIFI	Add Simplex Architecture	
MC_Missi	208	guarantee R				issie	Transform for sel.4 build	1.0
SW_SW_In	209	***};						_
UAV_UAV_	211							
UAS_UAS_	2128 pr	ocess implementat	ion SW.Impl					





😵 runtime-osate2	- CASE_Simple_Example_V4/SW.aadl - OSATE	2							
File Edit Navig	ate Search Project Run OSATE Analy	ses CASE AAHAA Window Help							
📑 • 🔛 🗠 🖬	9 🖶 🖬 🗐 🖂 🖉 🗎 🛄 🔄 🖻 🖡	? 🖸 Cyber Requirements 🔹 🔛 🔛 🥔 😓 🤮 📟 📟 🦥 🖏 Export to JSON Touch 📽 💁							
		Cyber Resiliency > Model Annotations							
-0 A 11 11 A	T T D GW and 12 D LIAS and	Component Analysis Model Transformations Add Attestation Manager							
	de v 203 wart cer	System Build > Proofs > Add Filter							
> CASE_Simp > CASE_Sim > CASE_Sim > CASE_Sim > CASE_Sim MC B MC B SW B UA	Add Filter Enter Filter details. You may optional model.	imes Ily leave these fields empty and manually edit the AADL filter component once it is added to the							
UA:	Filter implementation name	FLT CakeML							
	Filter implementation language								
	Create log port	None O Event O Data O Event Data							
	Requirement	Req003_FlightPlanner ~							
	Preserve Guarantees from RadioDriver	Req001_RadioDriver "Only valid command messages shall be forwarded to message destination components"							
	Filter AGREE contract	guarantee Req001_Filter "The filter shall allow only well-formed messages to pass" : WELL_FORMED_MESSAGE(filter							
		OK Cancel							





A sectors sector CAPP Clouds Form	- WARN							
runtime-osate2 - CASE_Simple_Exam	pie_V4/SW.aadi - USATE2							
File Edit Navigate Search Project	Run OSATE Analyses CAS	AAHAA Window H	telp			1.0.10		
🖬 • 🕅 👘 📾 🖬 🛄 🖂 >C		Cyber Requirements			Export to JSON Touch	0, 6		
		Cyber Resiliency	Model Annotations	1				
		Component Analysis	> Model Transformation	15 >	Add Attestation Manager			
VA 33 MA - D M SW.aa	di 13 III UAS aadi	System Build	> Proofs	>	Add Filter			
E 😵 🖤 203	uart_recv:	event data nort CASE	Model Transf _ innsuf	ASE HART	Add Isolator			
> CASE_Simple_Exa A 205	wifi recv: in e	went data port CASE	Model Transforwarcions::CA	SE WIFI I	Add Monitor			
V EP > CASE_Simple_E 206	wifi_send: out	event data port CASE	_Model_Transformations::C	ASE_WIFI	Add Simpley Architecture			
V Condiagrams 2070	annex agree {**	t ful "The Mission Co	enuter chall output a val	ad manual	Transform (as call & build			
MC_MISSP 200 209	**};	a_sw the mission co	abores, suarr onchor a sar	10 #1551	Transform for sec4 build			
1 HAV HAV 210	end SW;							
UAS UAS 211	process implementation	Sil Teol						
	process imprementation	SH. INPI						
282 With C	ASE_Model_Transforma	itions;						
283	1 + 1 **							
2049 annex	resolute (· data) ·
205 Ke	4005_FiightPlanner(C	The ElightDlage	ent, property_1d :	string, T	ilter : component,	connection_name	: string, message_type	: data) <=
200	[well_formed]	The FightPiann	er shall only arren	taglageta	wt filter correct	The droundstatio	a tuna)	
207	agree_prop_checked	(concexc, prope	rty_10) and add_111	ter (conte	xt, filter, connect	.ion_name, messag	e_cype)	
200 **1.								
209 53		210						
		319	MODEL TRANSFORM	ATIONS				
		320						
		321						
		322	Top-level claim	for prope	r insertion of a filte	er .		
		323	add_filter(context	: compone	nt, filter : component	, conn_name : stri	ng, msg_type : data) <=	
		324	filter evicts/	filter, co	ntext. conn name) and	filter not hypasse	d(filter, context, men tu	ne) and filter prop checked()
		326	TTTTEL_EXISTS(filler, co	incexc) com_name) and	TILCEL_HOL_Dypasse	(interi) concerci msg_cy	pe) and rifer_prop_checked()





File Edit Navigate Search Project Run OSATE Analyses CASE Analyses CASE Analyses CASE Analyses CASE Analyses CASE Coper Requirements Image: Component Analysis Image: Component Analysis Image: Component Analysis Image: Component Analysis Add Attestation Manager Add Image: Component Analysis Image: Component Analysis Image: Component Analysis Add Image: Component Analysis Add Image: Component Analysis Add Image: Component Analysis Image:	
Cyber Requirements Cyber Req	
Cyber Resiliency Model Annotations Model Transformations Add Attestation Manager System Build Proofs Add Islator V CASE_Simple_E 203 uart_recvi Add Islator V CASE_Simple_E 203 uart_recvi Add Islator 204 uart_send: Ut event data port CASE_Model_Transformations:/CASE_UNRT Add Islator 205 uiff_send: uiff_send: uiff_send: uiff_send: 2079 annex agree (** guarantee Reg001_SN "The Mission Coeputer shall output a valid missi: Add Monitor MC_Missi SW_SW_In 209 end SN; Transformations; iCASE_WIFF 204 with CASE_Model_Transformations; Impl Transform for setA build * : 203 with CASE_Model_Transformations; Error Log Console 203 with CASE_Model_Transformations; Add Romation # Add Stolator 204 with CASE_Model_Transformations; Impl # Add Romation SN. Impl # Add Romation 205 with CASE_Model_Transformations; Impl # Add Romation SN. Impl # Add Romation 204 Problems <	
SW add W add W add W add W add W add W add W add W add W add W add W add W add W add W add W add W add W add W add W add W add	
With CASE_Model_Transformations; With	
With CASE_Model_Transformations; Wit	
CASE_Simple_E 205 wiff_recv: in event data port CASE_Model_Transformations::CASE_NIFL 206 wiff_recv: in event data port CASE_Model_Transformations::CASE_NIFL 207 annex agret (** 208 guarantee Reg001_SN "The Mission Computer shall output a valid mission 208 with CASE_Model_Transformation SN.Impl 209 end SN; 200 end SN; 200 end SN; 200 process implementation SN.Impl 282 with CASE_Model_Transformations; 283 2840 283 2840 283 Problems III Properties Assurance Case IIII Properties Assurance Case IIII Properties Assurance Case IIII Properties IIII Properties Assurance Case IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	
Add simplex Architecture guarantee Reg001_SN "The Mission Computer shall output a valid mission SW_SW_JN UAV_UAV_2 UAS_UAS_U UAS_UAS_U With CASE_Model_Transformations; 282 283 284 283 284 285 286 286 286 286 286 286 286 286 286 286	
Wit_Missip 209 **); Wit_Wissip 209 **); UAV_UAV_UAV_UAV_UAV_UAV_UAV_UAV_UAV_UAV_	
UW, UW, UW, UW, UN, UN, UN, UN, UN, UN, UN, UN, UN, UN	
UAS_UAS_ 2120 process implementation SW.Impl 282 with CASE_Model_Transformations; 283 283 2840 Problems 2850 Problems 286 Req003_FlightPlanner(SW_Impl_Instance : SW::SW.Impl, "Req003_FlightPlanner", SW_Impl_Instance : SW::SW.Impl, RF_	
<pre>with CASE_Model_Transformations; with CASE_Model_Transformations; Problems Properties Assurance Case X CARE Results Provide Console Req003_FlightPlanner(SW_Impl_Instance : SW::SW.Impl, "Req003_FlightPlanner", SW_Impl_Instance : SW::SW.Impl, SW_Impl_Instance : SW::SW.Impl, RF_ Req003_FlightPlanner(SW_Impl_Instance : SW::SW.Impl, "Req003_FlightPlanner", SW_Impl_Instance : SW::SW.Impl, RF_ Req003_FlightPlanner(SW_Impl_Instance : SW::SW.Impl_Impl_Impl_Impl_Impl_Impl_Impl_Impl_</pre>	
 Problems Properties Assurance Case S GAGREE Results Provide Console Req003_FlightPlanner(SW_Impl_Instance : SW::SW.Impl, "Req003_FlightPlanner", SW_Impl_Instance : SW::SW.Impl, SW_Impl_Instance : SW::SW.Impl, RF_ 	
2850 2860 2860 2860 2860 2860 2860 2860 286	
287	(lgml)
207 v of [well formed] The FlightPlanner shall only accent well-formed messages from the GroundStation	-
288 wein_formed i the higher lainer shan only accept wein-formed messages norm the Groundstation	
289 V AGREE properties passed	
Filter FLT : SW::CASE_Filter.Impl is properly added to component FPLN : SW::FlightPlanner.Impl	
A CASE Filter FLT : SW::CASE Filter.Impl is connected to component FPLN : SW::FlightPlanner.Impl by connection SW::SW.Impl.c4	
Filter ELT : SW/#CASE Eilter Impl cannot be hunassed	
Filter FET : SW.CASE_Filter.impr cannot be bypassed	
 Filter property implemented 	
FLT : SW::CASE_Filter.Impl implementation is appropriate for OS	
Filter proof checked	



CYBER RESILIENT ARCHITECTURE PATTERNS

- Library of general, tool-assisted *architecture model transformations* that mitigate vulnerabilities or address cyber requirements
- Automatic insertion and verification of transform properties as assumeguarantee contracts and assurance case claims
- Examples
 - Filter 🗹
 - Attestation
 - Isolation
 - Monitor/Simplex
 - Distributed Action (e.g., Zeroize)
 - seL4 implementation



CASE TARGETS



• Experimental platform: AFRL UxAS



• Demonstration platform: CH-47 CAAS



FOR MORE INFORMATION...

- Send me an email at jen.davis@collins.com
- Releases of CASE tool suite (includes Resolute and AGREE) <u>https://github.com/loonwerks/formal-methods-workbench/releases</u>
- HAMR (for System Build): <u>https://github.com/sireum/hamr-plugin-update-site</u>
- More information on CASE project at http://loonwerks.com/projects/case.html
- Tool descriptions and papers
 - Resolute: <u>http://loonwerks.com/tools/resolute.html</u>
 - AGREE: <u>http://loonwerks.com/tools/agree.html</u>
- The seL4 Microkernel: <u>https://sel4.systems/</u>

The tools are open source and free to use!

