

IBM Analytics

Model-Based Interface Control Documents

Bruce Powel Douglass, Ph.D. Chief Evangelist, IBM IoT Bruce.Douglass@us.ibm.com www.bruce-douglass.com Twitter: @IronmanBruce





Agenda









Tables and matrices

Example





Agenda



Interfaces in SysML





Tables and matrices

Example



What's an ICD?

- An interface control document (ICD) in systems engineering and software engineering, provides a record of all interface information (such as drawings, diagrams, tables, and textual information) generated for a project. The underlying interface documents provide the details and describe the interface or interfaces between subsystems or to a system or subsystem. en.Wikipedia.org
- An ICD should only describe the detailed interface documentation itself, and not the characteristics of the systems which use it to connect.
 - The function and logic of those systems should be described in their own requirements and design documents as needed (there are DIDs for all of these).
 - For example, the ICD and associated interface documentation must include information about the size, format, and what is measured by the data, but not any internal design or implementation detail



Typical ICD Format

- System Overview
- Interface Overview
- Protocol Overview
- Security, Privacy, and Integrity Management
- Referenced Standards



Interface Control Document

Contents

1. Sys	stem Identification
1.1	<system 1="">1</system>
1.2	<system 2="">1</system>
2. Int	erface Description1
2.1	System Overview1
2.2	Interface Overview1
2.3	Functional Allocation1
2.4	Data Transfer
2.5	Transactions
2.6	Security and Integrity
3. De	tailed Interface Requirements
3.1	<interface 1=""> Requirements</interface>
3.1	.1 Interface Processing Time Requirements
3.1	.2 Message (or File) Requirements
3.1	.3 Communication Methods
3.1	.4 Security Requirements
3.1	.5 Physical Requirements
3.2	<interface 2=""> Requirements</interface>
4. Qu	alification Methods
Append	ix A: Interface Controls
Append	ix B: References 10
Append	ix C: Key Terms
Append	ix D: Information Exchange Matrix
Append	ix E: Context Flow Diagram describes the number data flows

ICD_XX

Page ii



Agenda



Interfaces in SysML





Tables and matrices

Example





Common ICD Forms

ICD Template (textual)

6. Detailed Interface Requirements

Instructions: This section specifies the requirements for one or more interfaces between two systems. This includes explicit definition of the content and format of every message or file that may pass between the two systems, and the conditions under which each message or file is to be sent. If an interface between the two systems is to be implemented incrementally, identify the implementation phase in which each message will be available. The structure in the "Requirements for <Given Interface>" section should be replicated for each defined interface between the two participating systems.

The template contained in the section named Requirements for <Given Interface> (including subsections) provides a generic approach to interface requirements definition. The specific interface definition should include only subsections relevant to the interface being defined, and liberty may be taken in the organization of subsections under the section named the section named Requirements for <Given Interface>. Where types of information not specified in the section named Requirements for <Given Interface> are required to clearly define the interface, additional subsections should be added. Other readily available documents (such as data dictionaries, standards for commercial protocols, and standards for user interfaces) may be referenced instead of stating the information here. It may be useful to include copies of such documentation as appendices to the ICD. Where possible, the use of tables and figures is encouraged to enhance the understandability of the interface definition. In defining interface requirements, clearly state which of the interfacing systems the requirement is being imposed upon.

6.1 Requirements for <Given Interface Name>

Instructions: Briefly summarize the interface for the name given above. Indicate what data protocol, communication methods, and processing priority are used by the interface. Data protocols may include messages and custom ASCII files. Communication methods may include electronic networks or magnetic media.

6.1.1 Assumptions

Instructions: Identify any assumptions that specify organizational responsibilities for specific activities or decisions, or that defines specific constraints. Assumptions might include:

- Data acceptance constraints
- Responsibility for establishing and managing the communication protocol
- Responsibility for providing and/or accepting file feeds for test and production processing
- Allowable file sizes
- · Responsibility for decisions on acceptance of test results

CMS XLC

6.1.2 General Processing Steps

Instructions: Describe the daily, weekly, monthly, etc., and threshold processing. Discuss the process to be used to confirm successful file transmission. Identify steps to be taken if all records in a file are received and the steps to be taken if all records are not received. Identify the reports to be used to document the results of daily, weekly, monthly, etc., processing. Describe any special processing that will be performed if a certain percentage (threshold) of the records is rejected.

Detailed Interface Requirements

6.1.3 Interface Processing Time Requirements

Instructions: If information is required to be formatted and communicated as the data is created, as a batch of data is created by operator action, or in accordance with some periodic schedule, indicate processing priority. Requirements for specific messages or files to be delivered to the communications medium within a set interval of time should be included in Subsection mande "Message Format (or Record Layout) and Required Protocols". State the priority that the interfacing entities must assign to the interface. Priority may be stated as performance or response time requirements defining how quickly incoming traffic or data requests must be processed by the interfacing system to meet the requirements of the interface. Considerable latitude should be given in defining performance requirements of the interface of the interfacing systems. Response time requirements, which are impacted by resources and beyond the control the interfacing systems (i.e., communication networks) are beyond the cosope of an ICD.

6.1.4 Message Format (or Record Layout) and Required Protocols

Instructions: Specify the explicit definitions of and the conditions under which each message is to be sent. Describe the definition, characteristics, and attributes of the command, also, document query and response descriptions.

6.1.4.1 File Layout

Instructions: This section should contain diagrams and short descriptions of both the header and detail layouts. This information may be included in an appendix to the document that is referenced here.

6.1.4.2 Data Assembly Characteristics

Instructions: Define the content and format of every message, file, or other data element assembly (records, arrays, displays, reports, etc.) specified in Subsection named "Message Format (or Record Layout) and Required Protocols". In defining interfaces where data is moved among systems, define the packaging of data to be utilized. The origin, structure, and processing of such packets will be dependent on the techniques used to implement the interface. Define required characteristics of data element assemblies that the interfacing entities must provide, store, send, access, receive, etc. When relevant to the packaging technique used, the following information should be provided:

Names/identifiers

Issues:

- Laborious to construct and maintain
- Not directly connected to requirements and design



Common ICD Forms

- N² Charts
 - The N² chart, also referred to as N² diagram, N-squared diagram or N-squared chart, is a diagram in the shape of a matrix, representing functional or physical interfaces between system elements. It is used to systematically identify, define, tabulate, design, and analyze functional and physical interfaces. It applies to system interfaces and hardware and/or software interfaces.

	Provided Interfaces								
		Avionics Subsystem	Navigation Subsystem	Fire Control Subsystem	Attitude Control Subsystem	Thruster Man- agement Sub- system	Surveillance Subsystem	HUD Subsystem	Datalink Subsystem
	Avionics Subsystem		iConfigNav	iConfigFire Control	iConfigAttitude	iThruster		iConfigHUD	iConfig- Datalink
	Navigation Subsystem								
	Fire Control Subsystem								
	Attitude Control Subsystem								
	Thruster Management Subsys- tem								
	Surveillance Subsystem								
red Interfaces	HUD Subsystem	iDoors iAutopilot iVehicle- Man- agement	iNavData	iGun iMissile	iSetAttitude iAttitudeData		iOptical iRadar iFLIR		
Requi	Datalink Subsystem						iRegisterComm Object		iSendData iRegisterComm Object

Issues:

• Weak on interface properties and detail



Issues:

- · Lacking some detail
- Scalability to many interfaces
- Cross-referencing
- 9 Internet of Things



Agenda



Interfaces in SysML





Tables and matrices

Example





Type of Interface Specifications

- Logical interfaces provide logical characteristics of the interface, but omitting physical representation detail
 - System actor
 - Subsystem subsystem
 - Example
 - evHerzaRadarTrack(RadarTrack t)
 - RadarTrack
 - IFF (Identify Friend Foe)
 - Range 10m 100km
 - Accuracy 5m
 - Maximum lag 250ms
- Physical Interfaces specify physical properties including bit formats (where appropriate)
 - System actor
 - Subsystem subsystem
 - Example
 - 1553 Bus Message HerezaRadarTrack with payload RadarTrack
 - RadarTrack
 - Range: 32-bit integer having units of meters, Big Endian format, valid range 10 100,000
 - Accuracy 5m



Type of Interface Specifications

Interdisciplinary

- AKA a facet, such as Software Electronics or Electronics Mechanical
- Example
 - Software memory mapped interface to servo motor
 - Address: 0A100:1000
 - Size: 16 bits
 - Bit Mapping

Bit 0: 0=Off, 1=On Bit 1-4: Error codes 0 (no error) to 16

Bit 5-15: Unused

Bit 16-24: Commanded Motor Speed (Write only), units RPMx10 Range 0-250 (0 – 2500 RPM) Bit 25-31: Measured Motor Speed (Read only),), units RPMx10 Range 0-250 (0 – 2500 RPM)

- Electro-mechanical interface
 - Type: Rotary motor, shaft dimensions, 5cm length, 5mm diameter
 - Output:

Torque: 0 - 1.2 Newton-meter Speed: 0 - 2500 rpm





What's in an Interface?

- Services
- Flows
- Arguments and their Types
- Metadata
- Physical Infrastructure / protocols

Services in Interfaces

"A behavior that may be requested"

- Service Name
- Description
 - Input / output control/data transformation
- Inputs
 - *See Types
- Outputs
 - *See Type
- Metadata
 - *See Metadata

Note: interfaces may not implement behavior; they specify how behavior is invoked





Flows in Interfaces

"A quantity that moves; can be information, fluid, energy, materiel and may be continuous or discrete"

- Flow Name
- Description
 - Input / output control/data transformation
- Direction
- Type
 - *See Type
- Metadata
 - *See Metadata







Types

"A category of values"

- Type Name
- Dimensions
 - A measurable extent of some kind
- Units
 - A quantity chosen as a standard measure of a dimension
- Extent
 - A set of permissible values of a type
- Format
- Size
 - Example
 - Byte size
 - Bit size
 - Physical volume
 - Physical units
- 16 Internet of Things



© 2019 IBM Corporation



Metadata

"Information about or properties of information itself"

- Range
 - A continuous extent; may be of ordinal discrete or continuous values
- Precision
 - Degree of information; often expressed as number of valid digits of representation
- Fidelity
 - Precision of an input
- Accuracy
 - Precision of an output; degree of compliance to a quantifiable expectation
- Timeliness
 - Quality of being available or useful within a time frame, often expressed as a deadline or max response time
- Performance
 - Quantifiable property of action execution, such as response time, lag, bandwidth, capacity or throughput
- Synchronicity
 - Quantifiable properties around simultaneity of different events, such as synchronous, blocking, asynchronous event processing
- Flow control
 - Properties of the management of a flow (ex: ACK/NAK)
- Location / address
- Alignment
- Constraints
- 17 Internet of Things

IBM Analytics

Side Note: Metadata-ing with Stereotypes and Tags

- Stereotype: A "special kind of " a SysML model element. Stereotypes
 - are indicated with double guillemots, such as «qualified»
 - may be applied to all metaclasses or just selected metaclasses
 - Can carry additional metadata in specified "tags". These tags hold values that are added by the user to stereotyped elements

General Description Relations Tags P Name: bitmapped Applicable to: Argument, Attribute, CallOper New Term	ation, Class, Event, Row, ItemRow, Link, Object, Operation, Reception, TriggeredOperation, Type V	
Locate OK Apply		
SampleClass	*bitmapped* statusField:unsigned char	
«bitmapped» statusField:unsigned char «bytemapped» rangedPressure:int «bytemapped» measureStatus:char	This bit-mapped attribute contains HW status information Start Address: Hardware memory map bit 0: 0 if ok, 1 if bad bit 1-2: Active Channel 02 bit 3-7: Error Code 015	
	 Number_Of_Bits:RhpInteger=8 Start_Address:RhpString=0A00: IFFF Timing_Constraints:RhpString=Valid 1ms after a write to the measureStatus attribute Usage:RhpString=SW writes any value to measureStatus attribute, waits 1ms, then can read HW 	
	It	
	Content in the second s	

eneral	Description	Relations	Tags	Properties	
🗸 Use	default order				😭 🗖 🗙
- Lo	cal				
Nu	mber_Of_Bi	ts			
St	art_Address				
Tir	ning_Constra	aints			
Us	age				
bit	_0				
bit	_1				
bit	_10				
bit	_11				
bit	_12				
bit	_13				
bit	_14				
bit	_15				
bit	_2				
bit	_3				
bit	_4				
bit	_5				
bit	_6				
bit	_7				
bit	_8				
bit	_9				
Quick	Add				
Name		Val			ا داد ۸
Name		Valu	le.		Add



IBM Analytics

Metadata

 The Harmony aMBSE profile contains a «qualified» stereotype that provides some of these properties as tagged values.

-# X

>

Label..

🗟 🗘

 \sim

~

The «qualified» stereotype is usually applied to value properties but may be applied to

Value Property : position_set in Uc_ControlAirSurfaces

General Description Relations Tags Properties

position_set

Public

qualified in HarmonySE

CAS Surface Positions position set

OK

Name:

Visibility

Locate

Stereotype

- Attributes
- Arguments
- Types
- Value properties
- Flow properties

All input or output numeric values in the systems model should be qualified

Gen	eral Description Rela	ations Tags Properties	
\checkmark	Use default order		- 😭 🛙
-	HarmonySE		
-	qualified		
	Bit_Layout		
	Max_Latency	< 2ms	
	Prohibited_Value	s	
	Range_High	40	
	Range_Low	-40	
	Space_Complexit	у	
	accuracy	± 1.0	
	precision	± 0.5	
G	luick Add		
Na	ame:	Value:	1
Loc	ate OK /	Apply	







Agenda









Tables and matrices







Where do interfaces come from?

- Logical
 - Functional Analysis
 - Logical Data Schema
 - Architectural Design (Allocation Stage)
- Physical
 - Architectural Design (Realization Phase)
 - Deployment Architecture
 - Distribution Architecture



In the Harmony aMBSE process, most systems engineering work is done with logical interfaces and logical data schema; the physical schema is developed during the hand off to down stream engineering



Use Cases as a Source of Interfaces

Interfaces can be detailed with behavioral views



Architecture as a Source of Interfaces



Data Schema Define the Data on Interfaces





Example use of stereotypes in physical interface definition

SampleClass	«bitmapped» statusField:unsigned char	
 «bitmapped» statusField:unsigned char «bytemapped» rangedPressure:int «bytemapped» measureStatus:char 	This bit-mapped attribute contains HW status information Start Address: Hardware memory map bit 0: 0 if ok, 1 if bad bit 1-2: Active Channel 02 bit 3-7: Error Code 0 15	
	Number_Of_Bits:RhpInteger=8 Start_Address:RhpString=0A00:IFFF Timing_Constraints:RhpString=Valid 1ms after a write to the measures Usage:RhpString=SW writes any value to measures Usage:RhpString=Errors: 0 = No Error, 1 = Errors found bit_0:RhpString=Errors: 0 = No Error, 1 = Errors found bit_1:RhpString=low bit of Active Channel (value in range of 03) bit_3:RhpString=Error code (value in range 015) bit_4:RhpString=Error code (value in range 015) bit_5:RhpString=Error code (value in range 015) bit_6:RhpString=Error code (value in range 015) bit_5:RhpString=Error code (value in range 015) bit_6:RhpString=Error code (value in range 015) bit_7:RhpString=Error code (value in range 015) bit_7:RhpString=Error code (value in range 015)	restatus attribute aits 1ms, then can read HW status
Y	«bytemapped» measureStatus:char	
This is a write-only register to comman Endianism:RhpString=Big Format:RhpString Numer_Of_Bytes:RhpInteger=1 Start_Address:RhpString=0A00-01FE Starting_Byte_Number:RhpInteger Timing_Constraints:RhpString=A write of Units:RhpString Usage:RhpString=Write any value to get the		
«bytemap rangedPres	ved» sure:int	
This holds pressures in kilopas 100 with accuracy of 0.01 kP. I stored value is the integral part For example -32.98 kP is store	tals in the range of -100 to + is held as a scaled integer; of (actual pressure * 100). d as the integer value -3298.	
 Endianism:RhpString=Big Format:RhpString=32 bit scalled i Numer_Of_Bytes:RhpInteger=4 Start_Address:RhpString=0A00:1 Starting_Byte_Number:RhpIntege Timing_Constraints:RhpString=M Units:RhpString=Kilopascals 	nteger FF0 =0 axiimum lag on value is 500 ns	



Agenda



Interfaces in SysML





Tables and matrices







Interfaces in SysML

- Ports are named connection points.
- SysML has different kinds of ports
 - Standard Ports (directly from UML). These are typed by Interfaces and contain only services. Deprecated.
 - Flow ports are typed by flow properties. Deprecated.
 - Proxy ports are typed by Interface blocks and may contain both flow properties and services.
 - Full ports are internal elements exposed across a system boundary. Full ports are not recommended.

[ArchitecturalBlock	Sta	andard Ports			
FunctionPort	1	iCommunica SystemP	utions ort			
ConfigPort)C iOpen	1	iPower	Pro	xy Ports		
	д		<proxy* 1 prtAM5:a5U_AMS 2 pUc_StartUp:~iUc_StartUp_aSU_AMS</proxy* 			1 itsUc_StartUp:Uc_StartU
	iTest iDisplay ServicePort Port : ServicePort in ArchitecturalBlock		<pre></pre>	<pre>«proxy» paSU_AMS:iUc_S Up_aSU_Aircraft_Power</pre>	itartUp_aSU_AMS «proxy»	" Values
	General Contract Relations Tags Properties Provided Interfaces Provided Relations Tags Provided Add Relations Provided Relations Rel		*proxy» 1 prtAircraft_Hydraulics:aSU_Aircraft_Hydraulics 2 pUc_StartUp:	paSU_Aircraft_Power:iUc_StartUp_aS ~iUc_StartUp_aSU_Aircraft_Hydraulics	GU_Aircraft_Power «proxy»	Operations
	evTest		pat	SU_Aircraft_Hydraulics:iUc_StartUp_aSU_A	Vircraft_Hydraulics	
	Required Interfaces		<pre>«interfaceBlock» iUc_StartUp_aSU_Aircraft_Power</pre>	<pre>«interfaceBlock» iUc_StartUp_aSU_AMS</pre>	iUc_Start	<pre>«interfaceBlock» Up_aSU_Aircraft_Hydraulics</pre>
	Display Add		Operations reqd evSelect_Battery_As_Source() Flow Properties	Operations reqd evEnter_WARM_state() prov reqENABLE_Command() reqd evEnter_OPERATING_st reqd evReportError(err:Error reqd evEnter_EAILED_state()	Preqd ev	Operations LequestHydraulicStatus() Flow Properties
27	Locate OK Apply	1		Flow Properties		

Interface Blocks

An Interface Block has the following constraints:

- It cannot have behavior or Full Ports
- It may own:
 - Abstract Operations
 - Event Receptions
 - Flow Properties
 - Proxy / Flow Ports
 - Parametric Diagrams (to add some constraints to the inputs and outputs)





Agenda









Tables and matrices

Example





Tables and Matrices

- Tables and Matrices are primarily present to provide a textual summary view of model metadata
- Both provide flexible and powerful ways to view information in the model
- Both provide the ability to create model elements and relations within the context of the table that are added into the model
 - Personal preference: I prefer to create diagrammatically but view summary in tables and matrices
- Rhapsody uses the Layout View paradigm
 - A **Layout** is the definition of the contents and form of the table or matrix
 - The **View** is an instance of the layout applied in some context and scope
- Tables and matrices can be exported to files (CSV and HTML formats)

apsody 💌	Table/Matrix	
Req_9	Refresh Toggle empty row Add model elemer Switch Rows and Export to file	←
teq_9 ⊾/ InterfaceR		

Rhapsody Matrices

Matrices are primarily use to show specifically related model elements of different kinds

Layout

Matrix Layout : Use Case Trace Matrix Layout in Layouts 🛛 🛪 💌	Matrix Layout : Use Case Trace Matrix Layout in Layouts 🔹 💌 🗵	Matrix Layout : Use Case Trace Matrix Layout in Layouts 🔹 💌 🗵	Matrix Layout : Use Case Trace Matrix Layout in Layouts 💌 🗶
To Element Types Cell Element Types Relations Tags Properties General Description From Element Types	To Bernent Types Cell Bernent Types Relations Tags Properties General Description From Bernent Types	General Description From Bement Types To Bement Types Cell Bement Types Relations Tags Properties	General Description From Bement Types To Bement Types Cell Bement Types Relations Tags Properties
Name: Use Case Trace Matrix Layout Label Stereotype: V 🐘 🗞	Select Query Select All UseCase UseCaseDagram UseCaseDagram UseCaseDagram ValueProperty ValueProperty ValueType Verw 1 element type selected	Select Quey Select Benert Types Refinement Required Condition Requirements Dagram Re	Select Element Types Object Value Protects Realizes Realizes Selfaction Vitrace Trace to Requirement Value Binding (Deprecated) Verification I dement type selected
Locate OK Apply	Locate OK Apply	Locate OK Apply	Locate OK Apply

View



To: Requirement Scope: RequirementsPk	9										
n	[] InterfaceReq_0	[] InterfaceReq_1	[] InterfaceReq_2	[] InterfaceReq_3	[] InterfaceReq_4	[] InterfaceReq_5	[] InterfaceReq_6	[] InterfaceReq_7	[] InterfaceReq_8	D InterfaceReg_9	[] F
Shut Down											
Manage Power											
🖉 👝 Manage Data											
Configure System											
Control Air Surfaces	Number InterfaceReq_0	_ InterfaceReq_1		Number InterfaceReg_3							
Update Status											
Start Up											
 Rotate Control Surface 											
Ai											
Configure Movement											
Perform Self Test											
Check Hydraulics											
Check Power											
Extend Control Surface	-										
Rotate Trim Tab											
Coordinate Start Up											
Coordinate Surface Movement	N InterfaceReg 0	N InterfaceReg 1									
Manage System State	-	-	Number InterfaceReg 2	N InterfaceReg 3	InterfaceReg 4	Number InterfaceReg 5	InterfaceReg 6	N InterfaceReg 7	interfaceReg 8	N InterfaceReg 9	N. In
Handle Control Errors											
Manage System Communicatons											
Move Control Surface											
Rotate Control Surface											
Retract Control Surface											
Rotate Trim Tab											
Coordinate Movement											

Rhapsody Tables

- Tables are primarily used to show metadata around elements of a common kind
- Layout

Boloti	ione	Tag			Propertiee	
General	Descriptio	iags	- Flement Type	es & Criteria	Columns	
	2 ccc.iptio					
Name: B	lock Feature Tal	ble Layout			Label	
Stereotype:				\sim	9h 14	
Relation Ta	sble					
Locate	OK Apr	olv.				
ble Layout : Blo	ock Feature Tab	ole Layout in l	ayouts.			-#
Relati	ions	Taos			Properties	
General	Descriptio	in .uga	Element Type	es & Criteria	Columns	
Select Flemen	nt Types					
	,pos					
Association	onEnd			^	Select All	
Attack Fi	ow Diagram				Clear All	
Attribute	e				Invert	
Authentic Basic Fai	ation				arron.	
BindingCo	onnector					
Block				¥		
3 element type	s selected					
Enable C	interia:					
Locate	OK Apr	ply			,	
		1				
le Layout : Bloc	ck Feature Table	e Layout in La	youts			-#
Conorrol Donorie	ntian Element	Turnon & Critoria	Columns	Polationa	Tage Proper	tico
serierar Descrip	Juon Demeni	Types a criterie	Coldinino	neiduuris	Tays Troper	uco
Adv	vanced Options.				X × +	Ť
Туре	Property	, C	olumn nam	e Col	umn width	
General Attrib	oute Name	N	ame			
General Attrib	oute Element	type El	ement type			
General Attrib	oute Descript	ion D	escription			
General Attrib	ute Owner	0	wner			

Name 🚽	Element type	Description 🔹	Owner
adjustPosition	Operation		ACES_ControlSurface
adjustPosition	Operation		Uc_RotateControlSurfac
augmentErrorCount	Operation		ACES_Management
Check_Hydraulic_Pressure	Operation		ACES_Management
Check_Power_Status	Operation		ACES_Management
Check_SW_Integrity	Operation		ACES_Management
checkCmdRange	Operation		Uc_RotateControlSurfac
checkCmdRange	Operation		ACES_ControlSurface
cmdPos	ValueProperty		Uc_RotateControlSurfac
Command_To_Minimum_Position	Operation		ACES_Management
Determine_Time_since_last_restart	Operation		ACES_Management
Determine_Time_since_last_restart	Operation		ACES_Power
determine Position Error	Operation		Uc_RotateControlSurfac
determinePositionError	Operation		ACES_ControlSurface
elapsedTime	ValueProperty		Uc_RotateControlSurfac
Enter_FAILED_state	Operation		ACES_Management
Enter_OPERATING_state	Operation		ACES_Management
Enter_WARM_state	Operation		ACES_Management
errorCount	Attribute		ACES_Management
errorFound	Attribute		ACES_Management
faultCount	Attribute		ACES_Management
getMeasuredPosition	Operation		ACES_ControlSurface
getMeasuredPosition	Operation		Uc_RotateControlSurfac
getPosition	Operation		ACES_ControlSurface
highPositionLimit	ValueProperty		Uc_RotateControlSurfac
hydraulicsOK	Attribute		ACES_Hydraulics
hydraulicsOK	Attribute		ACES_Management
initializeSurfaces	Operation		ACES_Management
isFlyable	Operation		ACES_Management
lowPositionLimit	ValueProperty		Uc_RotateControlSurfac
measuredPos	ValueProperty		Uc_RotateControlSurfac
movementOK	Operation		Uc_RotateControlSurfac
movementOK	Operation		ACES_ControlSurface
moveSurfaces	Operation		ACES_Management
moveTo	Operation		ACES_ControlSurface
moveTo	Operation		Uc_RotateControlSurfac
-	A		

View



Rhapsody Tables and Context Patterns

- Context patterns add a new level of flexibility locating and viewing model metadata
 - A context pattern is a pattern of metatypes (core metatypes or stereotypes)
 - The context pattern replaces the Element Types & Criteria Tab on normal tables
 - Any metatype can be used in a context pattern.
 - Only elements that fully match the entire pattern are returned (put in table)
 - A selector field label can be added to be used in the context of a property. This is encased in curly brackets { }
 - * or + can be added for recursion or iteration
 - A check box allows for expandable lists in first column ("Collapse 1st column")

Rhapsody Tables and Context Patterns

Advanced Table Options X	Table Layout : N	ame the Ports Layout in Comr	nonPkg		¥ ×
Appearance: Collapse 1st column When pushing "Enter" move selection to: Down Align column beader menu to:	General Des	cription Element Types & Criteria Advanced Options	Columns Relation	ns Tags Properties	Â
Context table: Context pattern: {Pkg)Package*, {Blk}Block, {Prt}Port,	Type General Att General Att General Att	Property C ribute Name N ribute Name N ribute Name N	Column name Co Jame in Pkg Pkg Jame in blk blk Jame in Prt Prt	ntext Column width	
Column name format: \$(Property) in \$(Context)					~
OK Cancel	Locate	OK Apply			

	Found 181 elements				
	Name in Pkg	Name in Blk	Name in Prt		
	ACESDecompositionPkg				
	ACES_ControlSurfacePkg				
		ACES_ControlSurface	PACES_Hydraulics		
		ACES_ControlSurface	PACES_Management		
		ACES_ControlSurface	pACES_Power		
	ACES_ControlSurfaceRetractingPkg	ACES_ControlSurfaceRetracting			
	ACES_ControlSurfaceWithTrimPkg	ACES_ControlSurfaceWithTrim			
	ACES_HydraulicsPkg				
		ACES_Hydraulics	pACES_ControlSurface		
		ACES_Hydraulics	PACES_Management		
		ACES_Hydraulics	pAircraft_Hydraulics		
	Activities				
Tratar	📩 ActorPkg				
34 Interi	ictori mingo				



Agenda



Common ICD Forms





come from?

Interfaces in SysML





Tables and matrices in Rhapsody

Example



35 InternetofThings

© 2019 IBM Corporation



The Air Control Surfaces Enactment System (ACES)





ACES Example: Logical Interfaces

- In this example, the ACES system has a defined set of logical interfaces derived from
 - Use case functional analysis
 - Definition of the logical data schema
 - Creation of a subsystem architecture including internal connections and interfaces

Table La	yout : Interface Features Table Layout in CommonPkg	Ŧ
Table Layout : Interface Features Table Layout in CommonPkg 🛛 🕇 🔀	al Description Columns Relations Tags Properties	
General Description Columns Relations Tags Properties	Advanced Options	Ŷ
Name: Interface Features Table Layout Label Stereotype: ✓ ♥↓	e Property Column name Context Column width heral Attri Name Name in Pkg Pkg heral Attri Name Name in Inter Inter heral Attri Name Name in feature feature heral Attri Name Name in arg arg heral Attri Classifier Classifier in arg arg heral Attri	.h
Locate OK Apply	Advanced Table Options Appearance: Collapse 1st column	×
	When pushing "Enter" move selection to: Down ~ Align column header menu to: Right ~	
	Context table: Context pattern: {Pkg}Package*, {Inter}Block, {feature}AttributelOperationIReceptionIFlowProperty, {ar	
Pkg}Package*, {Inter}Block,	Column name format: \$(Property) in \$(Context)]
arg}Argument	OK Cancel	
37 Internet of Things	© 2019 IBM Co	orpor

Resulting Logical Interfaces Table

ound 42 elements				
lame in Pkg	Name in Inter	Name in feature	Name in arg	Classifier in arg
SubsystemInterfaces	Pkg			
	iACES_Control_Surface_ACES_Hydraulics	hydraulic_pressure		
	ACES_Control_Surface_ACES_Power	power		
	iACES_Hydraulics_Aircraft_Hydraulics	evRequest_Hydraulic_Status		
	iACES_Hydraulics_Aircraft_Hydraulics	hereza_Hydraulic_Pressure	a ₀hs	Hydraulic_Status1
	iACES_Hydraulics_Aircraft_Hydraulics	Hydraulic_pressure		
	iACES_Management_ACES_Control_Surface	Command_To_Position	🗐 id	SurfaceID
	iACES_Management_ACES_Control_Surface	Command_To_Position	pos	🔷 int
	iACES_Management_ACES_Control_Surface	Req_config_parameter		
	iACES_Management_ACES_Control_Surface	Request_SW_Integrity_Check		
	iACES_Management_ACES_Control_Surface	www.Status	🗐 id	SurfaceID
	iACES_Management_ACES_Control_Surface	🕞 SW_Status	sw_status_msg	W_Status
	iACES_Management_ACES_Control_Surface	Update_Position	🗐 id	SurfaceID
	iACES_Management_ACES_Control_Surface	Update_Position	movement_duration	Second
	iACES_Management_ACES_Control_Surface	Update_Position	pos	🔷 int
	iACES_Management_ACES_Control_Surface	herezaConfiguration	Config	Surface_Configura
	iACES_Management_ACES_Control_Surface	herezaConfiguration	🗐 id	SurfaceID
	iACES_Management_ACES_Hydraulics	Deck_Hydraullics		
	iACES_Management_ACES_Hydraulics	Request_SW_Integrity_Check		
	iACES_Management_ACES_Hydraulics	www.Status	🗐 id	SurfaceID
	iACES_Management_ACES_Hydraulics	www.status	w_status_msg	SW_Status
	iACES_Management_ACES_Hydraulics	Dpdate_Hydraulic_Status	h_status	Hydraulic_Status1
	iACES_Management_ACES_Power	Request_SW_Integrity_Check		
	iACES_Management_ACES_Power	www.Status	🖬 id	SurfaceID
	iACES_Management_ACES_Power	www.Status	w_status_msg	SW_Status
	iACES_Management_ACES_Power	Select_Battery_As_Source		
	iACES_Management_ACES_Power	Update_Power_Status	p_status	Power_Status1
	iACES_Management_ACES_Power	power		
	iACES_Management_AMS	ev Disable		
	iACES_Management_AMS	evEnter_FAILED_State		
	iACES_Management_AMS	evEnter_Operating_State		
	iACES_Management_AMS	evEnter_Operational_State		
	iACES_Management_AMS	evEnter_WARM_State		
	iACES_Management_AMS	evReport_Error	err	•
	iACES_Management_AMS	evUpdate_Positions	sp	Surface_Positions
	iACES_Management_AMS	regENABLE_Command		
	ACES_Power_Aircraft_Power	Operation_1	a n x	♦ int
	ACES_Power_Aircraft_Power	Operation_1	a ny	♦ double
	iACES_Power_Aircraft_Power	Select_Power_Source	source	POWERSOURCE.
	iACES_Power_Aircraft_Power	evRequest_Power_Status	source	POWERSOURCE
	ACES_Power_Aircraft_Power	evSelect_Battery_As_Source		
	ACES_Power_Aircraft_Power	hereza_Power_Status	a nps	Power_Status1
	ACES_Power_Aircraft_Power	power	SF.	

Logical Data Schema (Graphical View)



© 2019 IBM Corporation

Logical Data Schema (Tabular)

Name in okg	Name in type	Name in feature	Classifier in feature
PataTupesPkg	manie in type	Ivane il leature	Ciassiler in reduie
Data Typesrkg			
	Error_Log	- data Naca	A True Data True
		date_time	 ImeDateType EDBOD_TYPE
	Error_Report	error	<pre> ERHUR_ITPE A DOWEDOOLUDOE TVDE </pre>
	Error_Report	source	POWERSOURCE_TYPE
	Error_Report	surfaceID	SurfaceID
	Hydraulic_Status1	has_faults	RhpBoolean
	Hydraulic_Status1	e pressure	Pascal
	MAX_MOVE_TIME		
	MEASUREMENT_INTERVAL		
	NORMAL_RESTART_INTERVAL		
	NUMBER_OF_SURFACES		
	POWERSOURCE_TYPE		
	Power_Status1	amperage	Ampere
	Power_Status1	has_faults	RhpBoolean
	Power_Status1	voltage	🕀 Volt
	STATIONKEEPING INTERVAL		-
	SURFACE POSITION JITTER TOLERA		
	SUBFACE POSITION TO FRANCE		
	Status	status	SystemOperationalState
	SufaceID		Option oppiration alotato
	Surface Configuration Type	high limit	▲ int
	Suface_Configuration_Type		V III.
	Surface_Conliguration_Type		VIIL
	Surface_Position	commanded_position	✓ ITL
	Surface_Position	has_fault	HhpBoolean
	Surface_Position	high_range	♦ int
	Surface_Position	low_range	♦ int
	Surface_Position	measured_position	♦ int
	Surface_Position	surfaceID	SurfaceID
	Surface_Position	time_to_achieve_position	Second
	Surface_Position_Status	commanded_position	🔷 int
	Surface_Position_Status	is_functional	RhpBoolean
	Surface_Position_Status	e measured_position	♦ int
	Surface_Position_Status	surfaceID	SurfaceID
	Surface_Position_Status	time_of_failure	TimeDateType
	Surface Position Status	time of measurement	TimeDateType
	Surface_Position Status	time_to_move	♦ int
	Surface Positions	a getFault Status	
	Surface Positions	getSurface Position	
	Surface Positions		
	Surface Positions	estFault Statue	
	Surface_Positions		
	Surface_Positions	selveasured_Position	
	Surface_Positions		
	U Surface_Positions	setSurface_Position	
	USurface_Positions	setSurface_Range	
	Usurface_Positions	set fime_To_Achieve_Position	
	Surface_Positions	setTime_to_achieve_position	
	SystemOperationalState		
	TEST_TYPE		
	Test_Log		
	Test_Outcome	date_time	TimeDateType
	Test_Outcome	pass	RhpBoolean
	Test_Outcome	surfaceID	SurfaceID
	Test Outcome	test ID	♦ TEST TYPE



Physical Data Schema (using CBP Protocol) – Graphical



41 Internetof

Physical Schema – Message Types



Physical – Logical Trace Table

Ξ

Name in pkg	Name in PhysicalRepresentation	To element in LogicalRepresenta
는 Message Types Pkg		
	CBPMessage	check Power
	CBPMessage	evRequestHydraulicStatus
	CBPMessage	reqENABLE_Command
	CBPMessage	request_Hydraulic_Status
	CBP_ControlSet	herezaPositionSet
	CBP_Error	evError
	CBP_HydraulicStatus	herezaHydraulic_Pressure
	CBP_HydraulicStatus	herezaHydraulic_Pressure
	CBP_Move	Command_To_Position
	CBP_Move	📪 ev Move To
	CBP_MoveDone	Updated_Position
	CBP_MoveDone	evMovementDone
	CBP_MoveDone	herezaPosition
	CBP_PowerSource	Select_Battery_As_Source
	CBP_PowerStatus	Update_Power_Status
	CBP_PowerStatus	updatePowerStatus
	CBP_ReportError	Peport Error
	CBP_RequestConfiguration	Req_config_parameter
	CBP_RequestConfiguration	Req_config_parameters
	CBP_RequestSWStatus	request_SW_Integrity_Check
	CBP_RequestSWStatus	request_SW_Integrity_Check
	CBP_SWStatus	🕞 SW_Status
	CBP_SWStatus	📪 SW_Status
	CBP_State	evEnter_FAILED_state
	CBP_State	evEnter_OPERATING_state
	CBP_State	evEnter_WARM_state
	CBP_State	📭 evGoto Failed State
	CBP_SurfaceConfiguration	herezaConfiguration

Physical Message Properties Table

Name in cls 🚽	Name in Attr	Classifier in Attr 🔹	Name in tags 🔹 💌	Value in tags
CBP_Error	duration	Interval_In_MS	Context Inter_Of_Bytes	C 2
CBP_Error	e measuredPos	♦ double	🔄 Usage	Construction After movement completes
CBP_Error	e measuredPos	♦ double	Context Inter_Of_Bytes	4
CBP_Error	e measuredPos	♦ double	🔁 Format	🔄 4-byte IEEE floating point format
CBP_Error	e measuredPos	♦ double	🔄 Starting_Byte_Number	* 7
CBP_Error	when	TimeDate_Type		
CBP_Error	emdPos	♦ double	🔄 Starting_Byte_Number	C 3
CBP_Error	emdPos	♦ double	🔄 Usage	Commanded position
CBP_Error	emdPos	♦ double	🔁 Format	🔄 4-byte IEEE floating point format
CBP_Error	emdPos	♦ double	Context Inter_Of_Bytes	k 4
CBP_Error	surfaceID	SurfaceIDType	Cart_Address	š 0
CBP_Error	surfaceID	SurfaceIDType	Context Inter_Of_Bytes	ka 1
CBP_Error	errorType	ERROR_TYPE	Context Inter_Of_Bytes	ka 2
CBP_Error	errorType	ERROR_TYPE	Cart_Address	ka 1
CBP_Error	duration	Interval_In_MS	🔁 Starting_Byte_Number	ka 11
CBP_HydraulicStatus	💻 status	Hydraulic Status		
CBP_Move	position	♦ double	Context Inter_Of_Bytes	k 4
CBP_Move	position	♦ double	🔄 Format	🔄 4-byte IEEE floating point format
CBP_Move	position	♦ double	🔁 Usage	Commanded position
CBP_Move	surfaceID	SurfaceIDType		
CBP_MoveDone	surfaceID	SurfaceIDType	🔁 Starting_Byte_Number	š 0
CBP_MoveDone	posAchieved	♦ double	🔁 Format	🔄 4-byte IEEE floating point format
CBP_MoveDone	posAchieved	♦ double	🔁 Endianism	🔁 Big
CBP_MoveDone	surfaceID	SurfaceIDType	Context Inter_Of_Bytes	ka 1
CBP_MoveDone	surfaceID	SurfaceIDType	🔁 Usage	🔄 ID of the referenced control surface
CBP_MoveDone	posAchieved	♦ double	Context Inter_Of_Bytes	∛ □ 4
CBP_MoveDone	posAchieved	♦ double	🔄 Usage	🔄 The measured position achieved in movement
CBP_MoveDone	posAchieved	♦ double	🔄 Starting_Byte_Number	ka 1
CBP_MoveDone	timeUsed	Interval_In_MS	Context Inter_Of_Bytes	k 4
CBP_MoveDone	timeUsed	Interval_In_MS	🔄 Endianism	🔁 Big
CBP_MoveDone	surfaceID	SurfaceIDType	🔁 Endianism	🔁 Big
CBP_MoveDone	timeUsed	Interval_In_MS	🔁 Starting_Byte_Number	ka 5
CBP_MoveDone	timeUsed	Interval_In_MS	🔁 Usage	Contraction of movement time in ms
CBP_PowerSource	powerSource	POWERSOURCE_TYPE		
CBP_PowerStatus	💻 status	PowerStatus		
CBP_ReportError	when	TimeDate_Type		
CBP_ReportError	errorType	ERROR_TYPE		
CBP_ReportError	surfaceID	SurfaceIDType		
CBP_RequestConfiguration	surfaceID	SurfaceIDType		
CBP RequestSWStatus	surfaceID	SurfaceIDType		



Report Generation

 A good approach is to create a report template for your modeling tool. Recommended format is:

Overview	Types and Classifiers
Reference model	Blocks and Classes
Scope	Name
Date Created	Value and Flow Properties
Interfaces & Interface blocks, each with	Name
Name	Description
Value and Flow Properties	Туре
Name	Tags
Value description	Services
Туре	Name
Tags	Service Description
Services	Parameters
Name	Parameter type
Service Description	Return type
Parameters	Tags
Parameter type	Types, Value Types, and Data Types
Return type	Name
Tags	Description
Description	Based Type
Flow ports	If Enumeration, then list
Name	enumerations
Owning Classifier	
Туре	
Tags	





Summary





Real-Time Agile Systems and Software Development

Welcome to www.bruce-douglass.com



You've found yourself on **www.brucedouglass.com**, my web site on all things real-time and embedded.

On this site you will find papers, presentations, models, forums for questions / discussions, and links (lots of links) to areas of interest, such as

- Developing Embedded Software
- Model-Driven Development for Real-Time Systems
- Model-Based Systems Engineering
- Safety Analysis and Design
- Agile Methods for Embedded Software
- Agile Methods for Systems Engineering
- The Harmony agile Model-Based Systems Engineering process
- The Harmony agile Embedded Software Development process
- Models and profiles I've developed and authored
- List and links to many of my books.

The menu at the top of each page either takes you to the relevant page or to a list of relevant pages.

There is even a members only site for those who want to access to even more stuff. I teach and consult on all these topics - see my <u>About</u> page.

Policy on Using These Materials

All materials on this web site are free for reuse and distribution, provided their source (me or this web site) is appropriately attributed. I retain sole copyright.

