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

- What's an ICD?
- What's in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- Interfaces in SysML
- Tables and matrices
- Example

Agenda

- What's an ICD?
- What's in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- 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

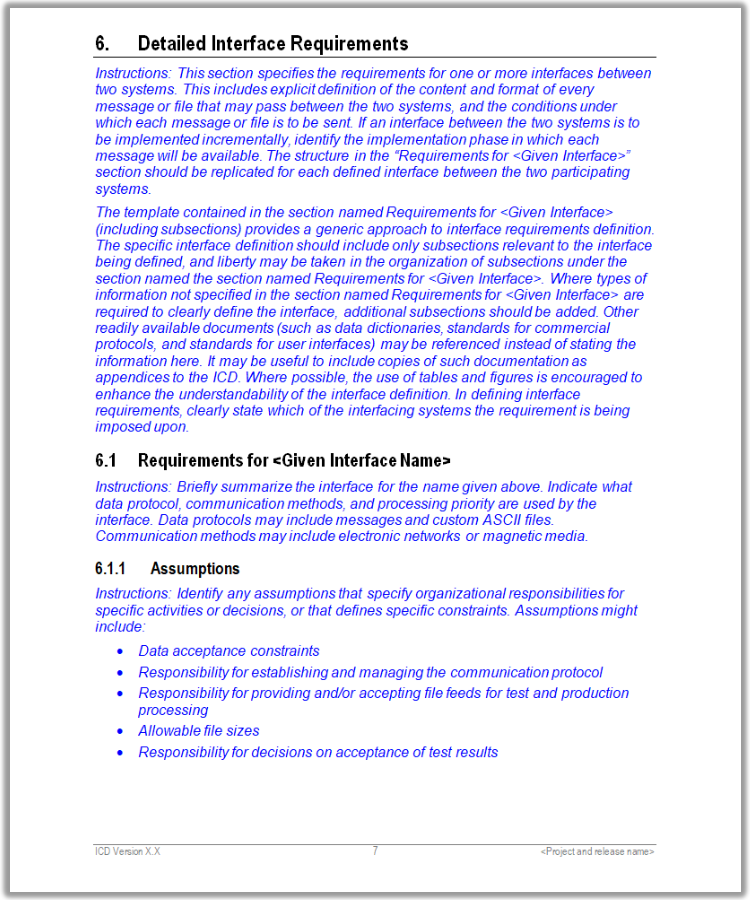
Contents

Agenda

- What's an ICD?
- What's in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- 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>

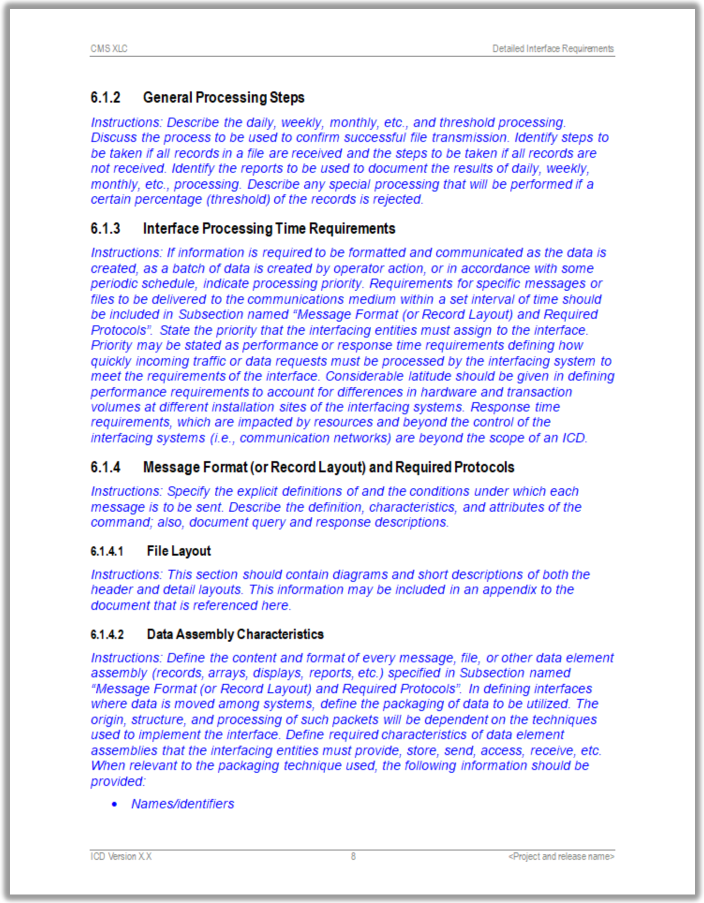
Instructions: Briefly summarize the interface for the name given above. Indicate what data protocol, communication methods, and processing priority are used by the interface. Data protocols may include messages and custom ASCII files. Communication methods may include electronic networks or magnetic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

ICD Version X.X 7 <Project and release name>



CMS XLC Detailed Interface Requirements

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.

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 named "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 to account for differences in hardware and transaction volumes at different installation sites of the interfacing systems. Response time requirements, which are impacted by resources and beyond the control of the interfacing systems (i.e., communication networks) are beyond the scope 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*

ICD Version X.X 8 <Project and release name>

Issues:

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

Common ICD Forms

- N^2 Charts
  - The ***N*² chart**, also referred to as ***N*² 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.

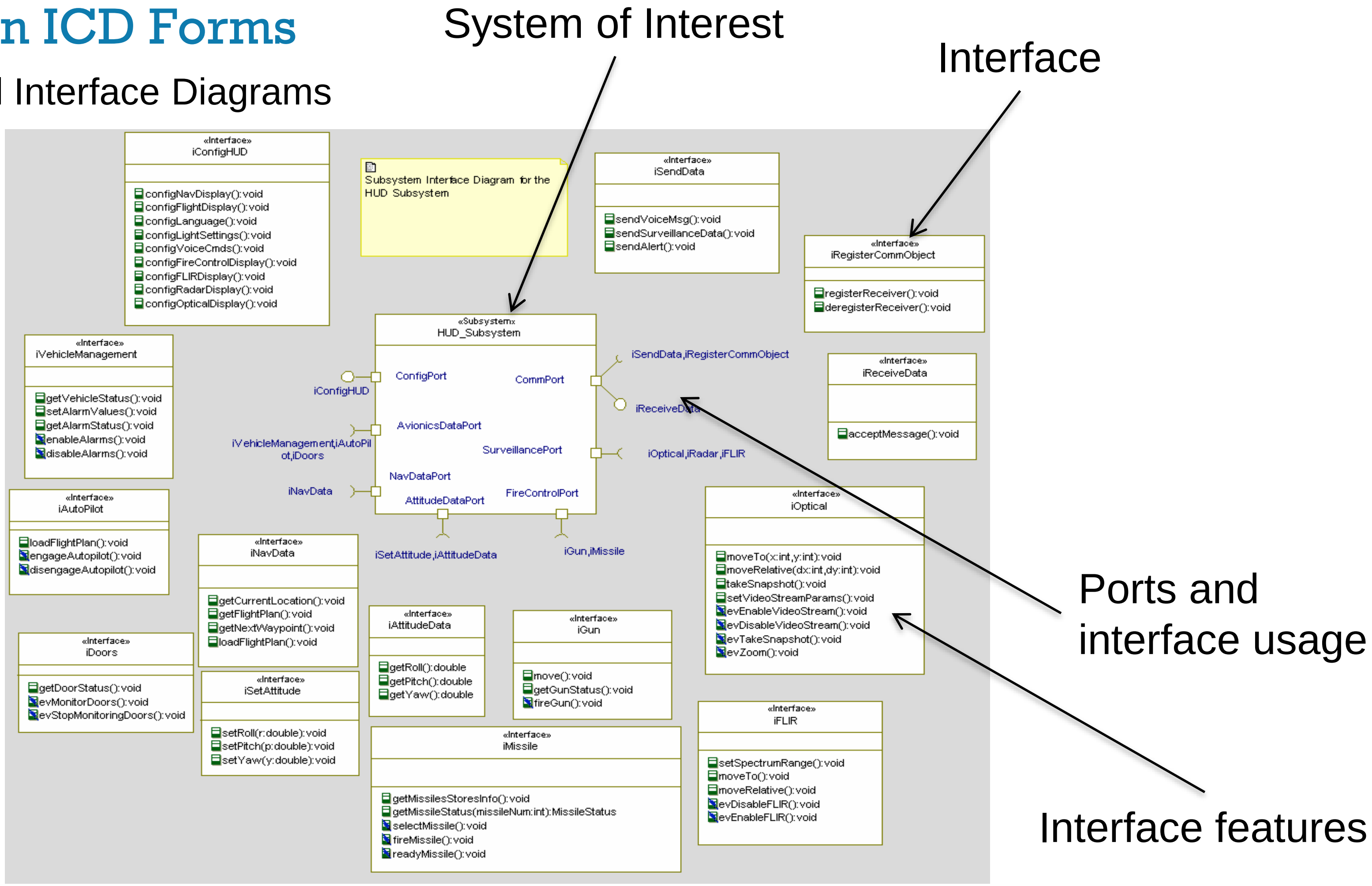
	Provided Interfaces								
		Avionics Subsystem	Navigation Subsystem	Fire Control Subsystem	Attitude Control Subsystem	Thruster Management Subsystem	Surveillance Subsystem	HUD Subsystem	Datalink Subsystem
Required Interfaces	Avionics Subsystem		iConfigNav	iConfigFire Control	iConfigAttitude	iThruster		iConfigHUD	iConfig-Datalink
	Navigation Subsystem								
	Fire Control Subsystem								
	Attitude Control Subsystem								
	Thruster Management Subsystem								
	Surveillance Subsystem								
	HUD Subsystem	iDoors iAutopilot iVehicle-Management	iNavData	iGun iMissile	iSetAttitude iAttitudeData		iOptical iRadar iFLIR		
	Datalink Subsystem						iRegisterComm Object		iSendData iRegisterComm Object

Issues:

- Weak on interface properties and detail

Common ICD Forms

- Graphical Interface Diagrams



Issues:
- Lacking some detail
- Scalability to many interfaces
- Cross-referencing

Agenda

- What’s an ICD?
- What’s in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- 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

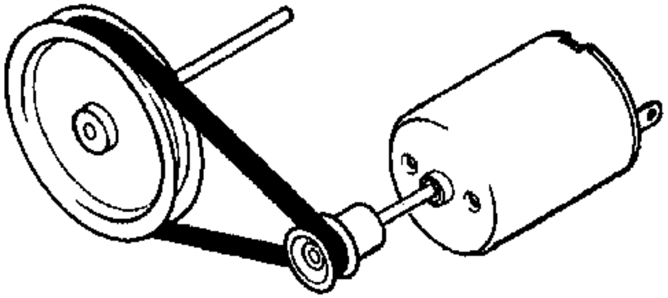
        Bit 5-15: Unused

        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

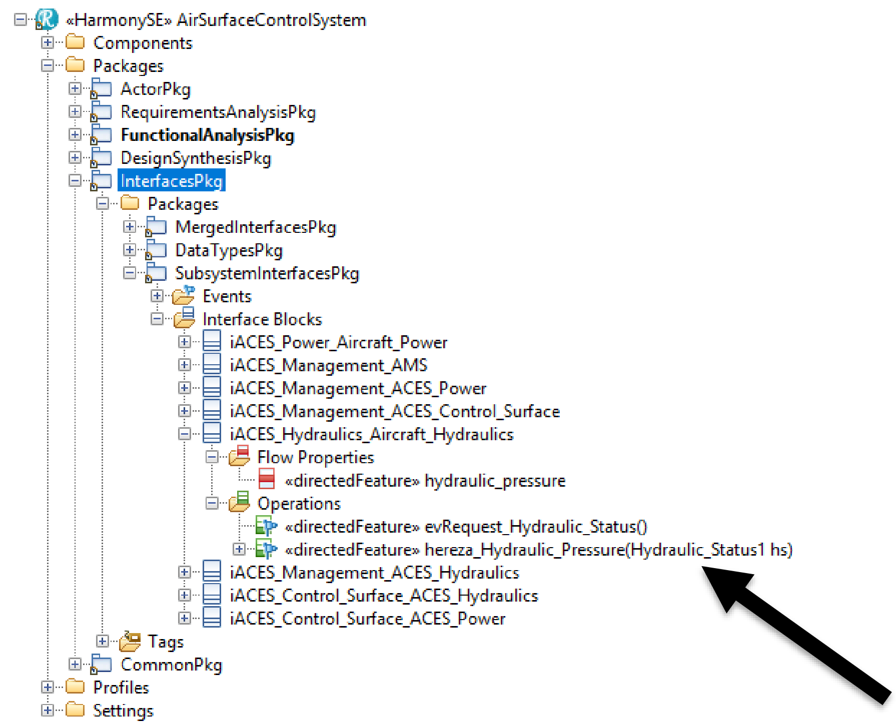


What's in an Interface?

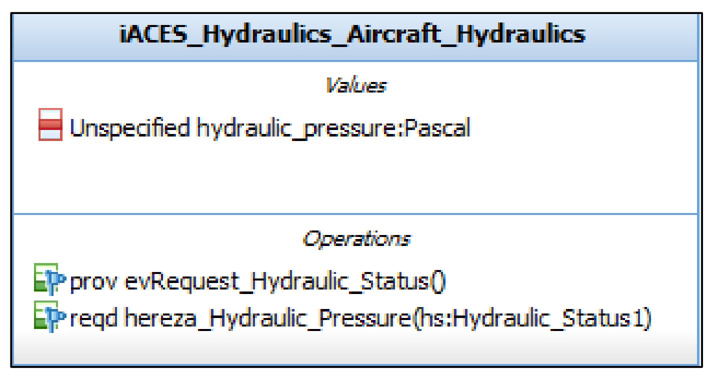
- Services
- Flows
- Arguments and their Types
- Metadata
- Physical Infrastructure / protocols

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

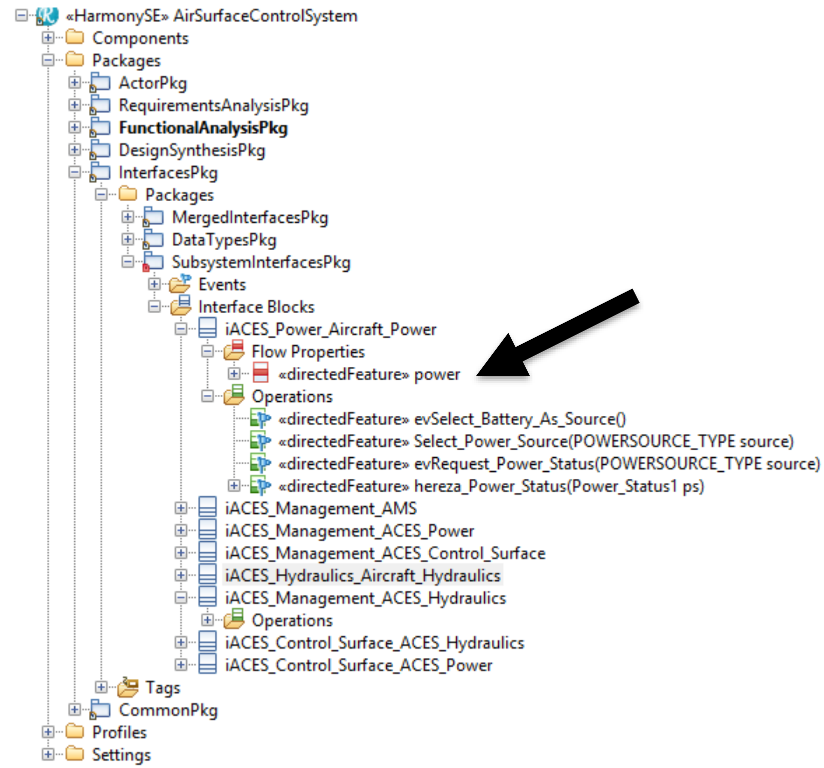


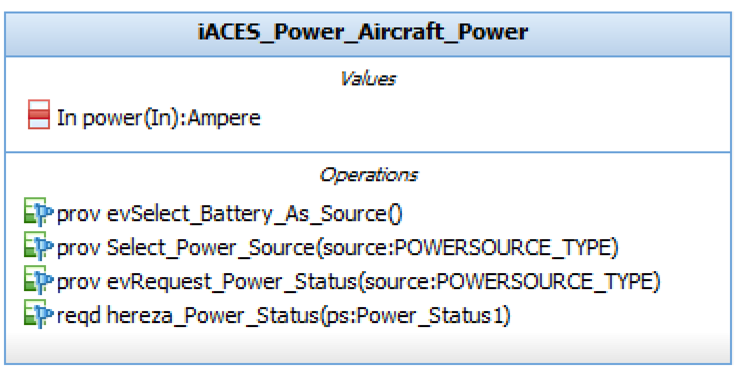
Note: interfaces may not implement behavior; they specify how behavior is invoked

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

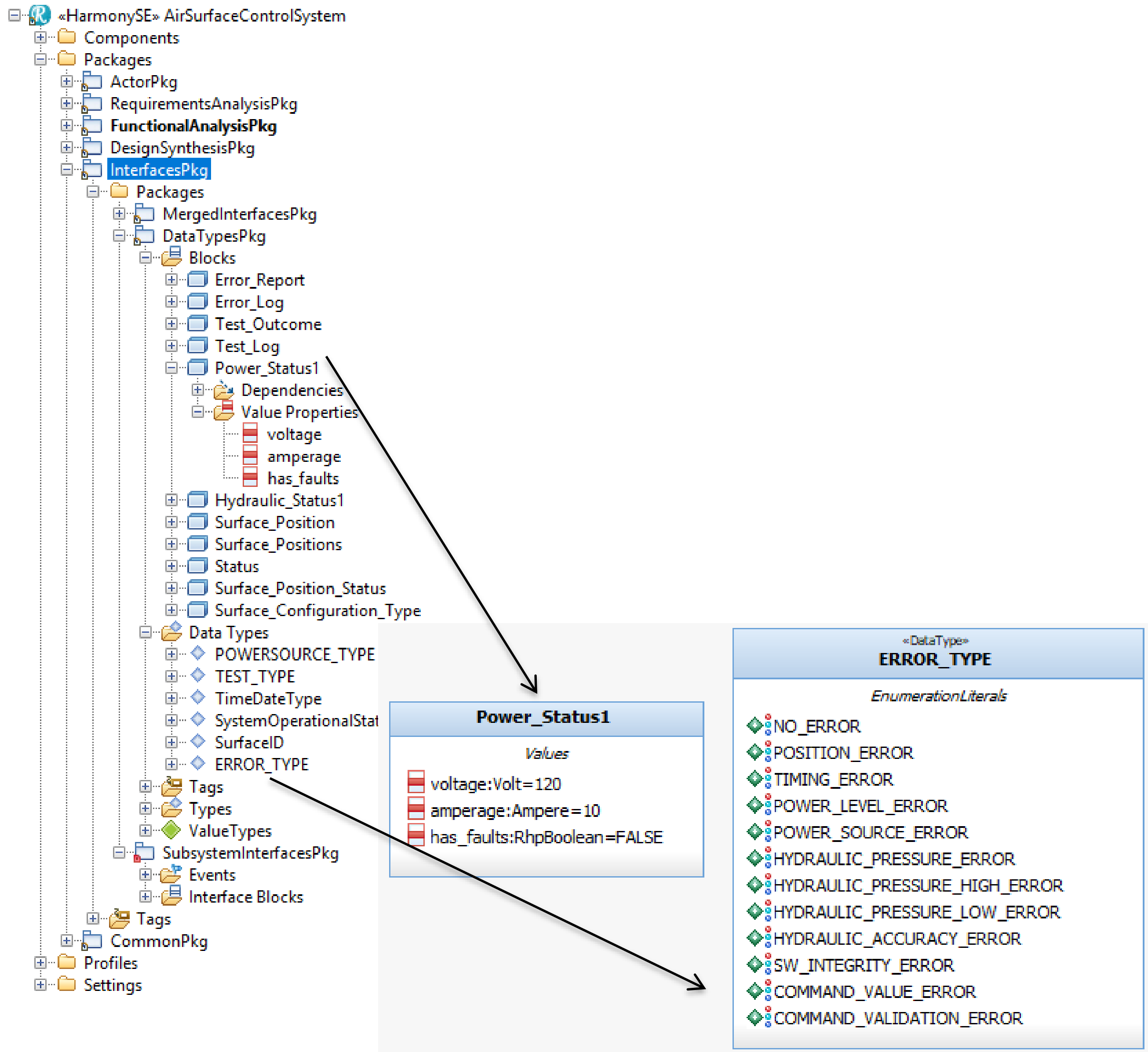




Types

"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



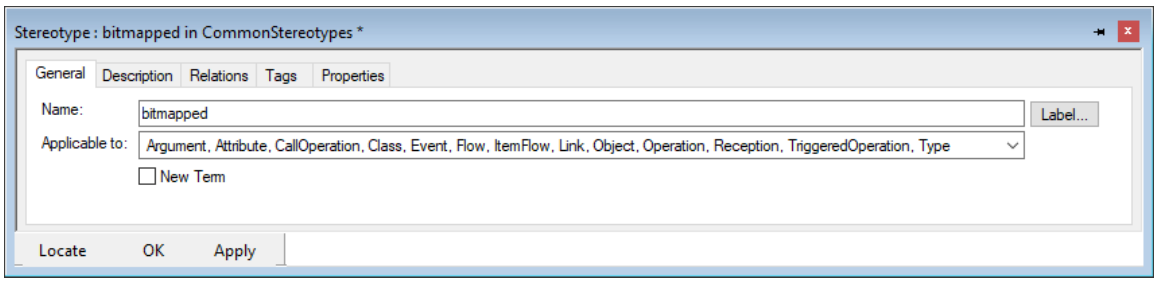
Metadata

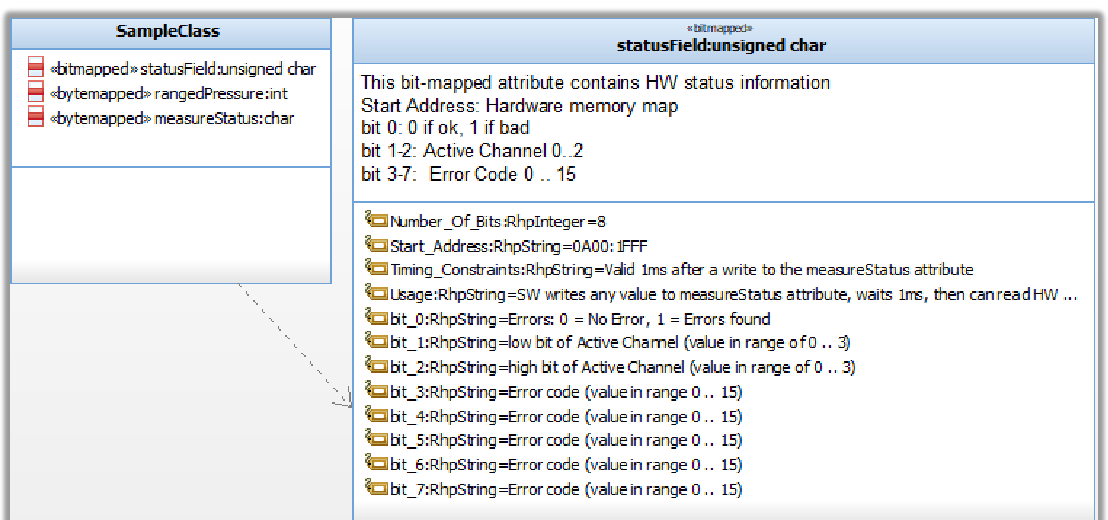
"Information about or properties of information itself"

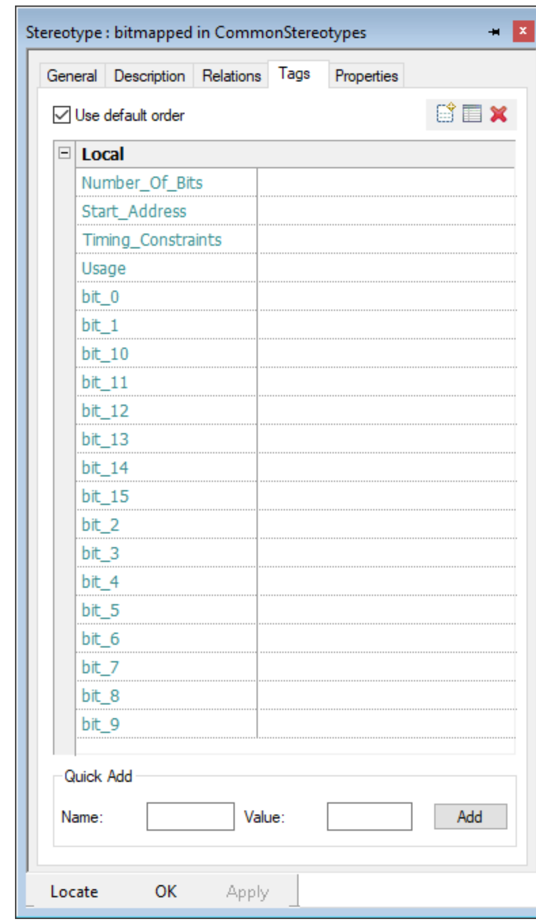
- Range
  - A continuous extent; may be of ordinal discrete or continuous values
- Precision
  - Degree of information; often expressed as number of valid digits of representation
- Fidelity
  - Precision of an input
- Accuracy
  - Precision of an output; degree of compliance to a quantifiable expectation
- 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

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

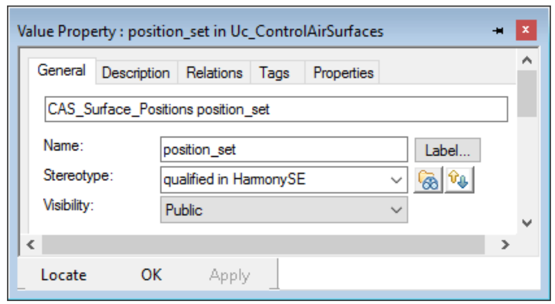


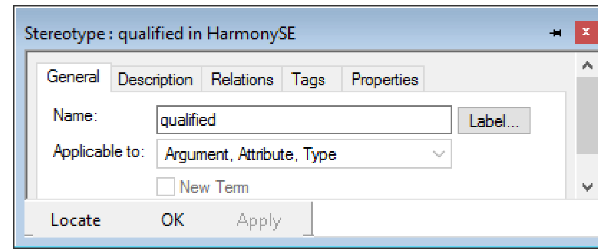


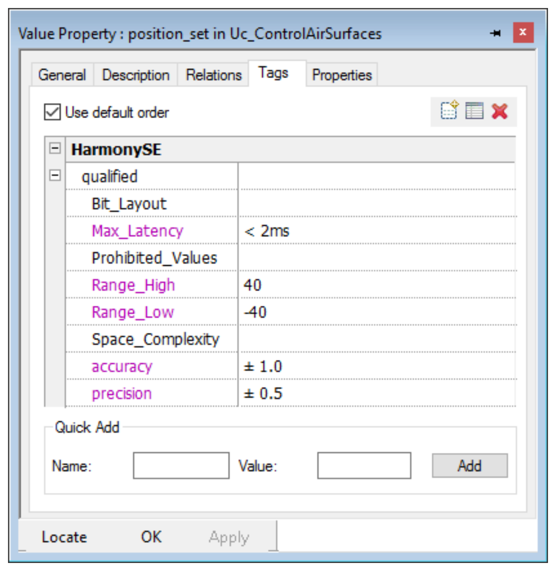


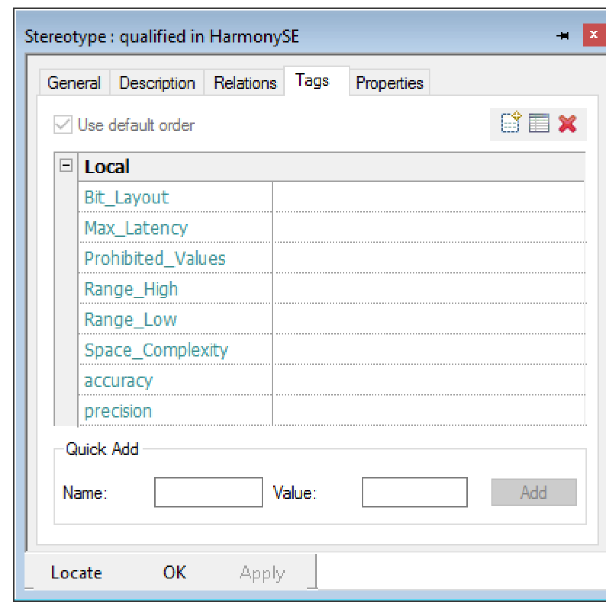
Metadata

- The Harmony aMBSE profile contains a «qualified» stereotype that provides some of these properties as tagged values.
- The «qualified» stereotype is usually applied to value properties but may be applied to
  - Attributes
  - Arguments
  - Types
  - Value properties
  - Flow properties









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

Agenda

- What's an ICD?
- What's in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- Interfaces in SysML
- Tables and matrices
- Example

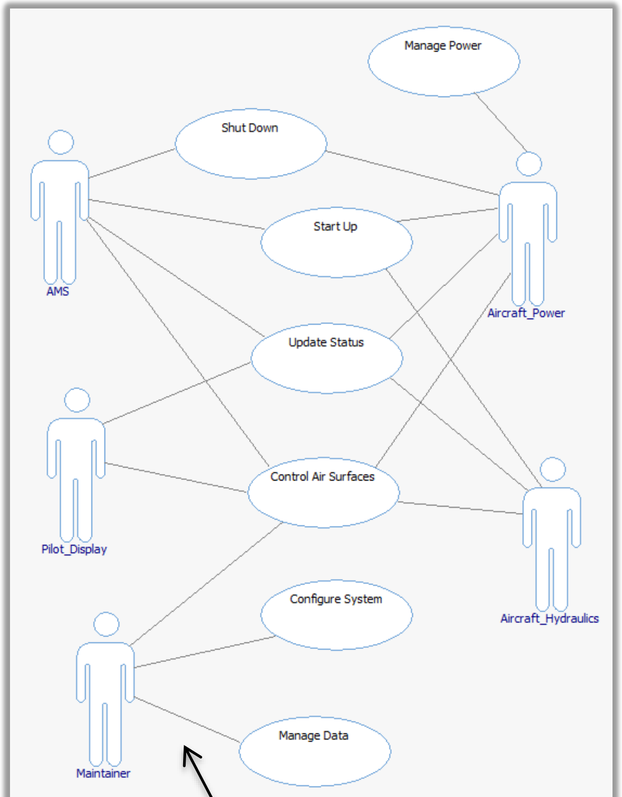
Where do interfaces come from?

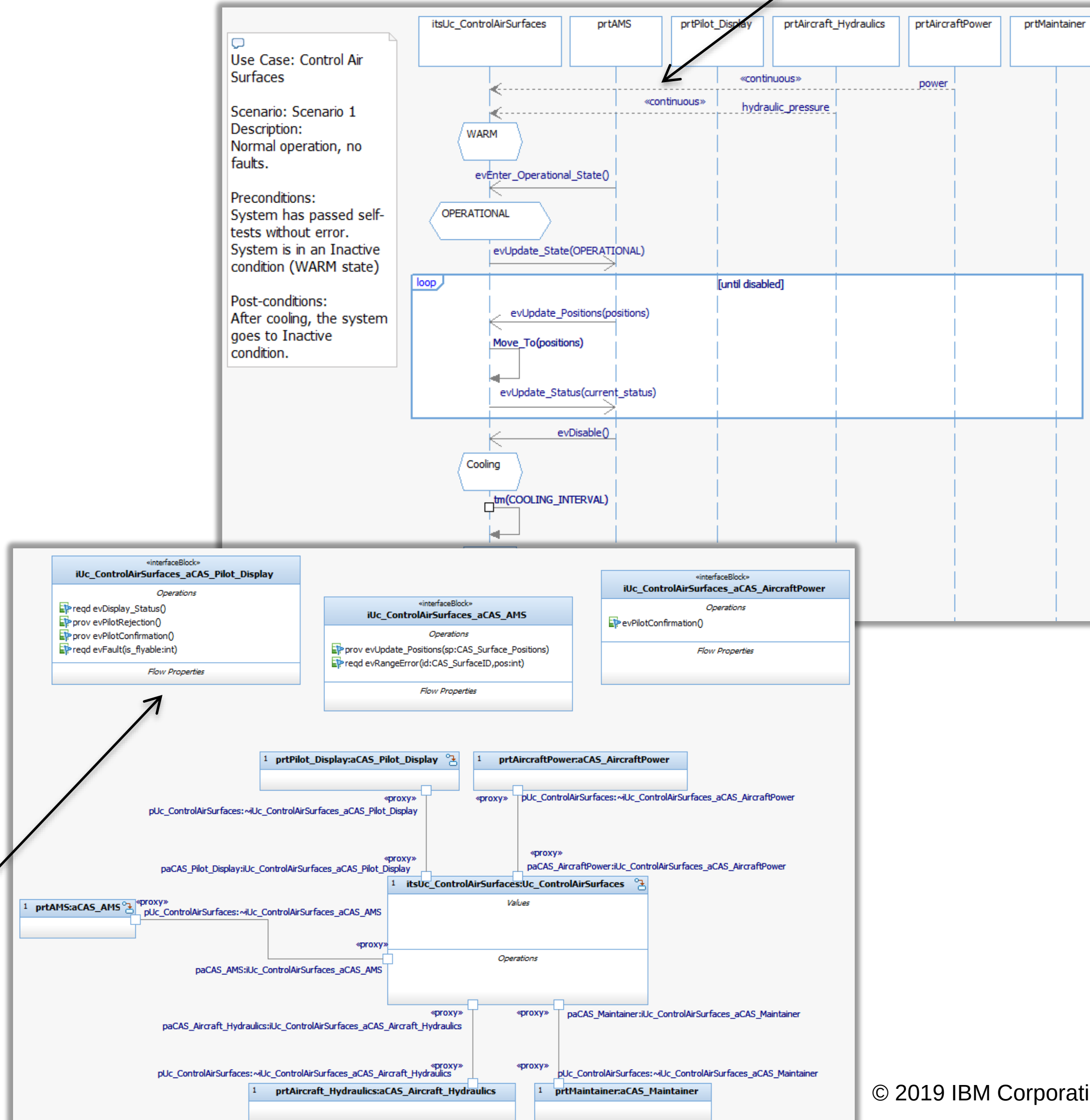
- Logical
  - Functional Analysis
  - Logical Data Schema
  - Architectural Design (Allocation Stage)
- Physical
  - Architectural Design (Realization Phase)
  - Deployment Architecture
  - Distribution Architecture

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

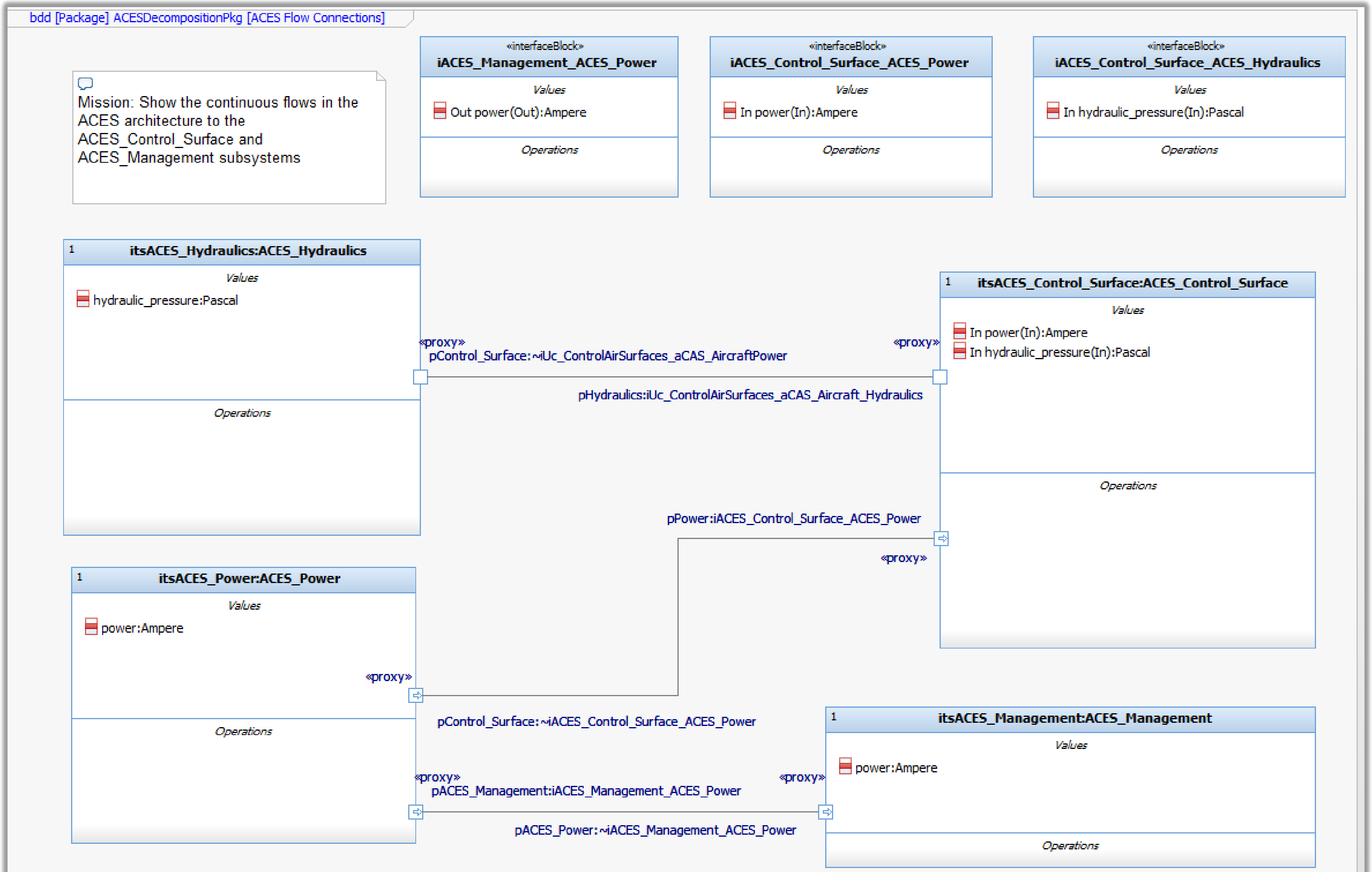




Use case-actor associations indicate the existence of interfaces

Resulting interfaces

Architecture as a Source of Interfaces



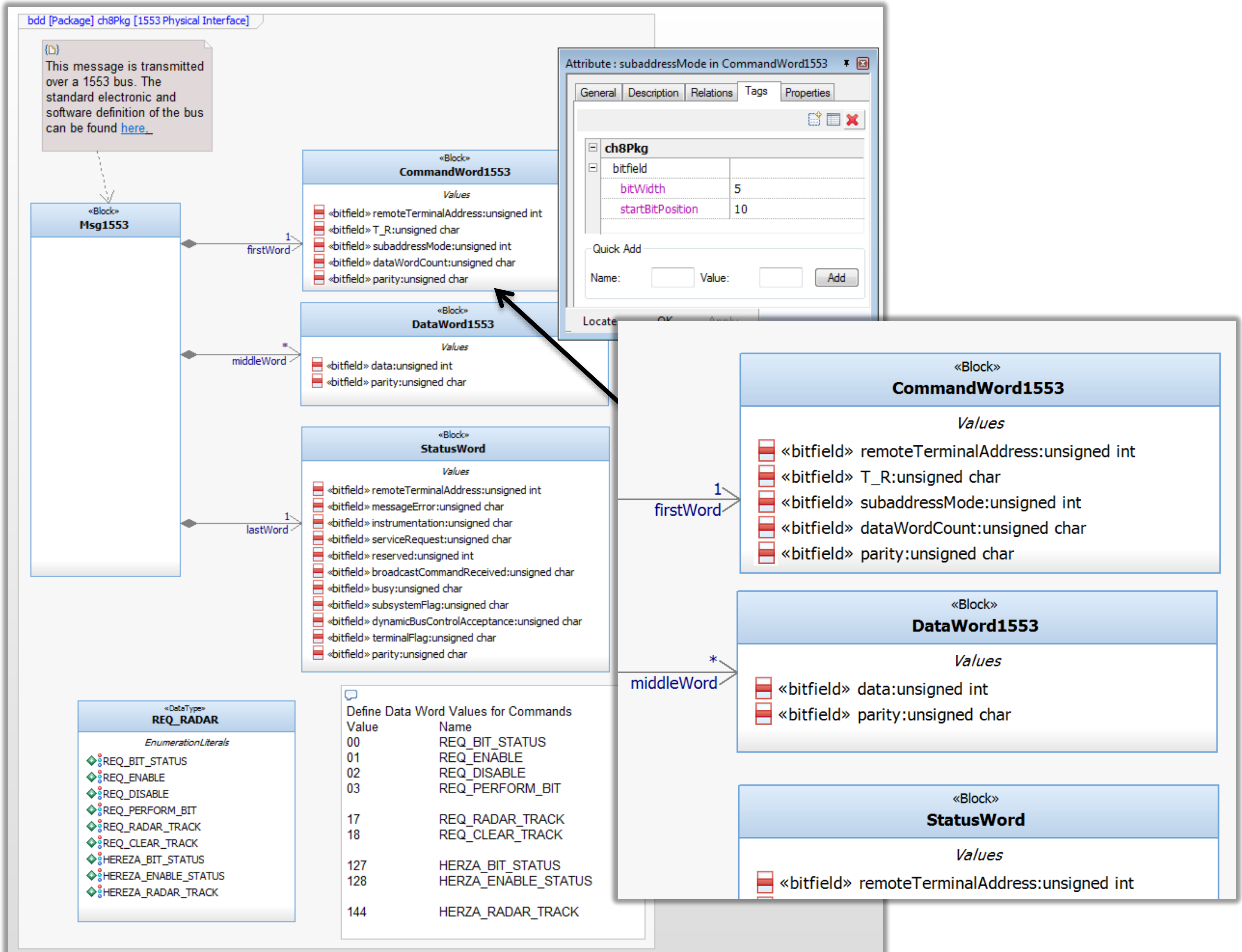
Data Schema Define the Data on Interfaces

Blocks and Classes can model different message fields.

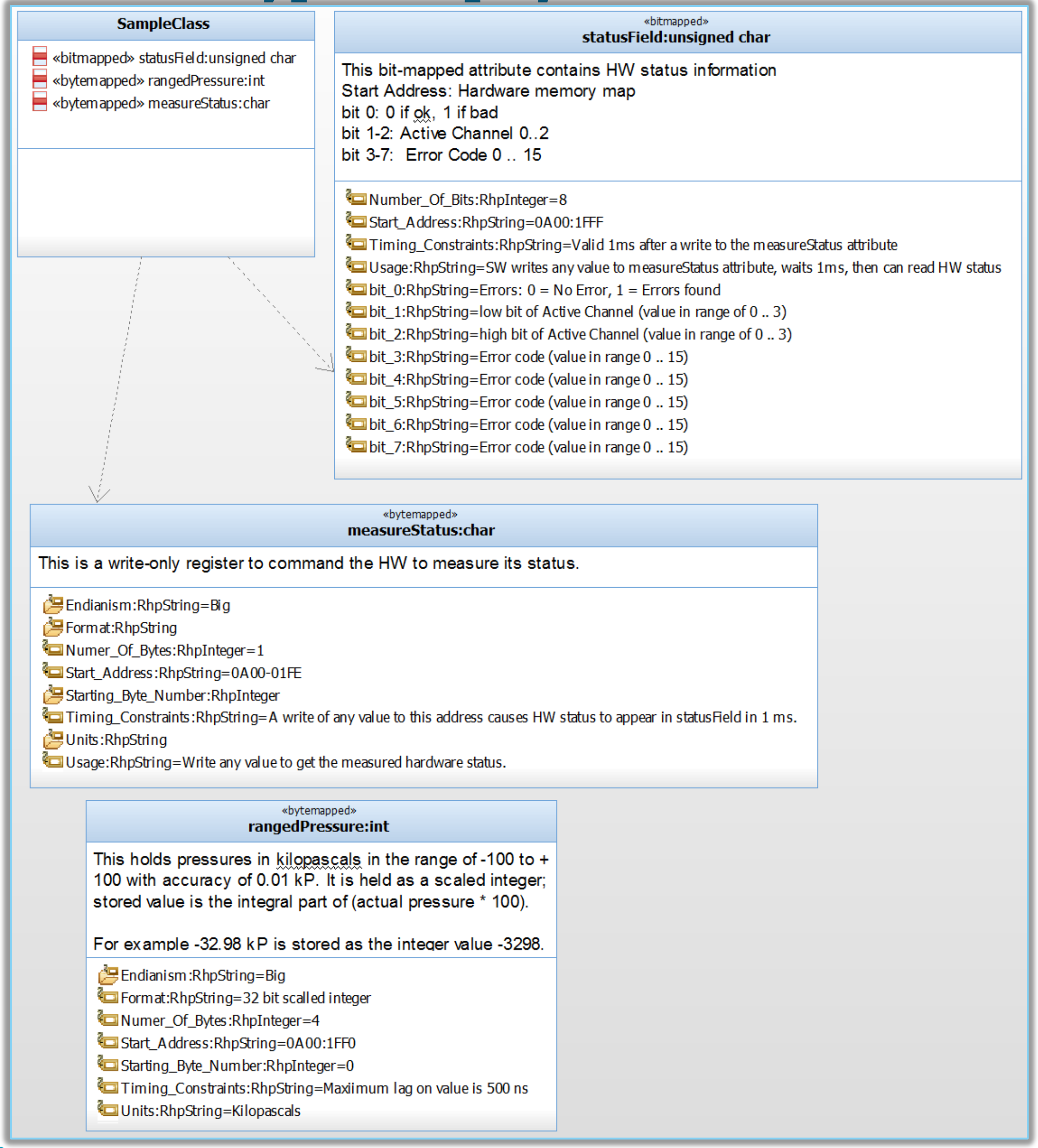
Example: **Msg1553** contains three types as parts:

- Command word
- Data word
- Status word

And the bit-structure of these parts is identified using «bitfield» stereptype



Example use of stereotypes in physical interface definition



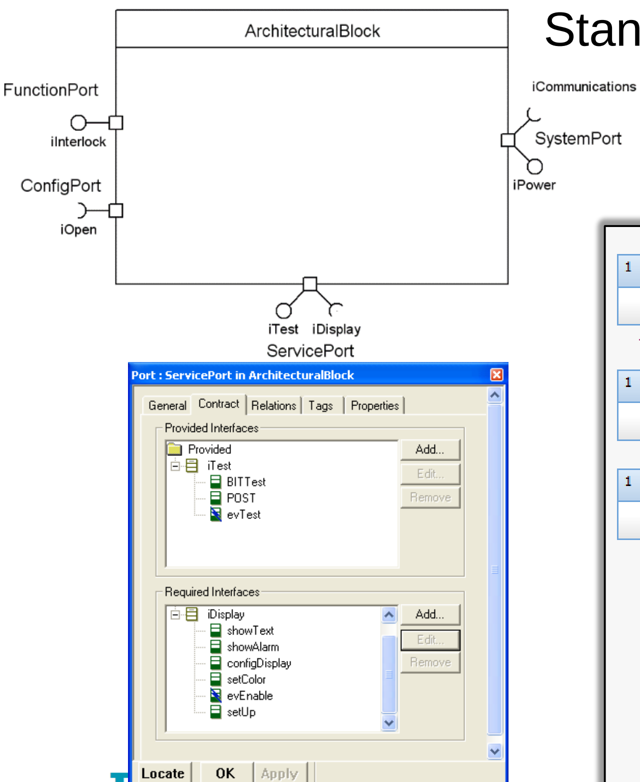
Agenda

- What's an ICD?
- What's in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- Interfaces in SysML
- Tables and matrices
- Example

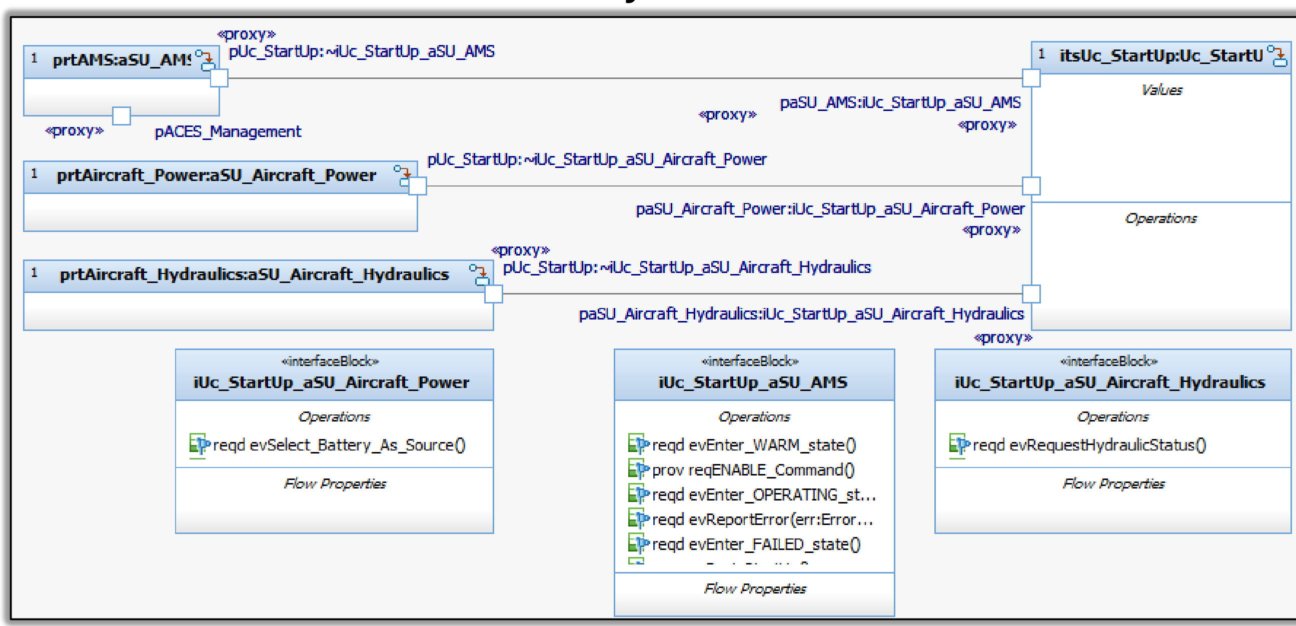
Interfaces in SysML

- 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.

Standard Ports



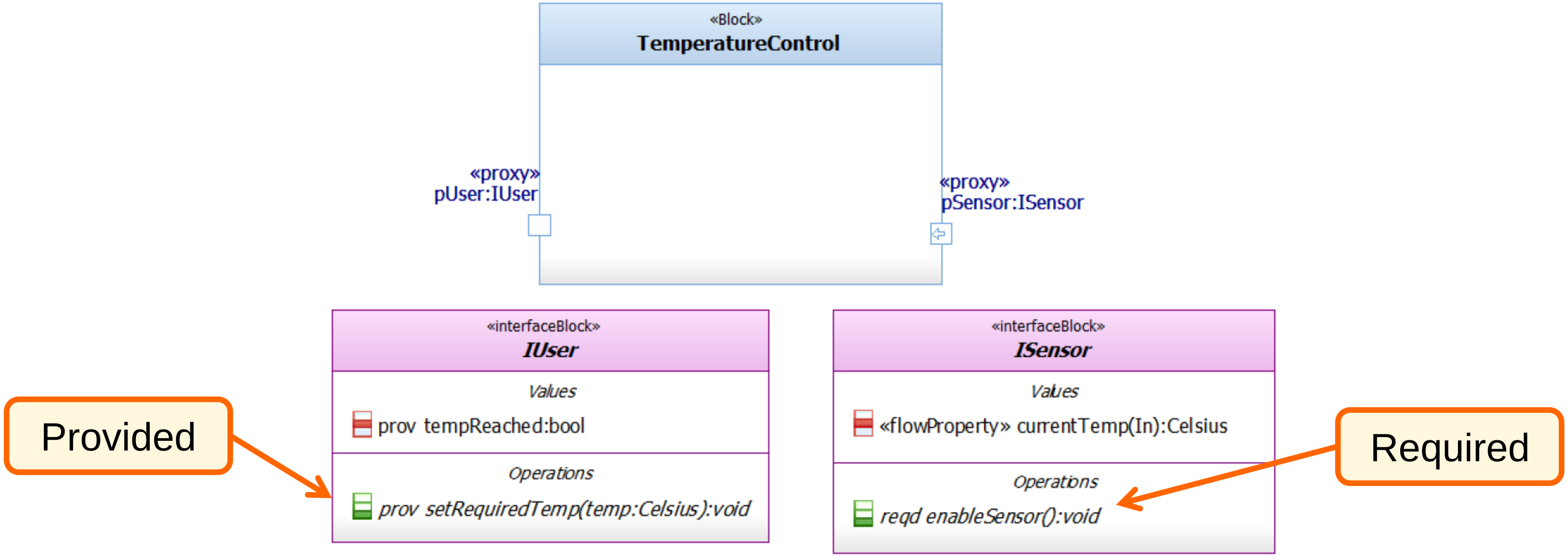
Proxy Ports



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)

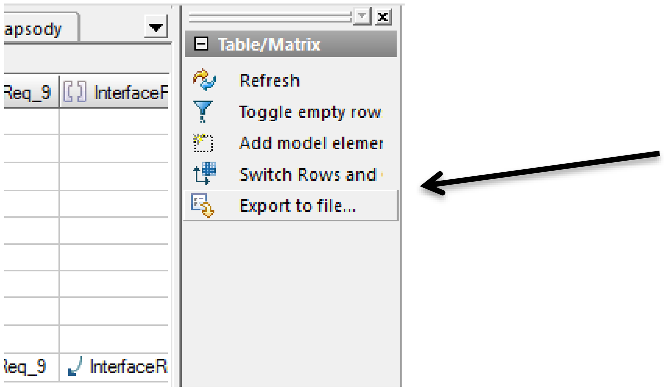


Agenda

- What's an ICD?
- What's in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- Interfaces in SysML
- Tables and matrices
- Example

Tables and Matrices

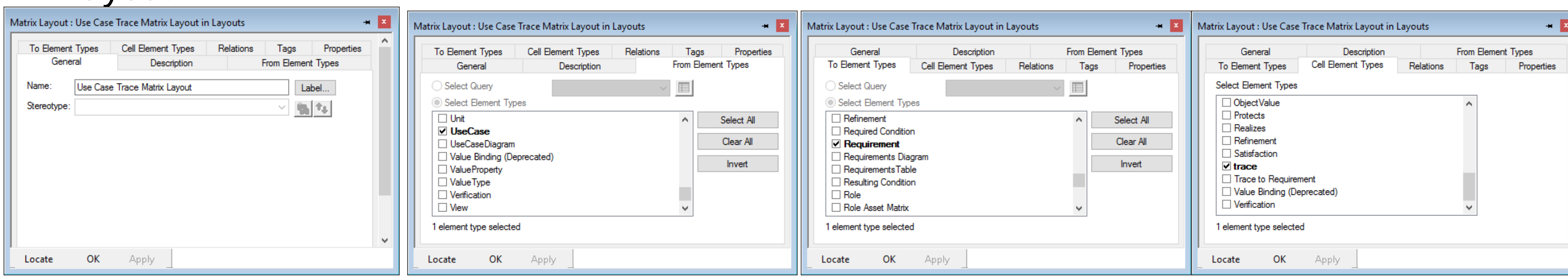
- Tables and Matrices are primarily present to provide a textual summary view of model metadata
- Both provide flexible and powerful ways to view information in the model
- Both provide the ability to create model elements and relations within the context of the table that are added into the model
  - 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)



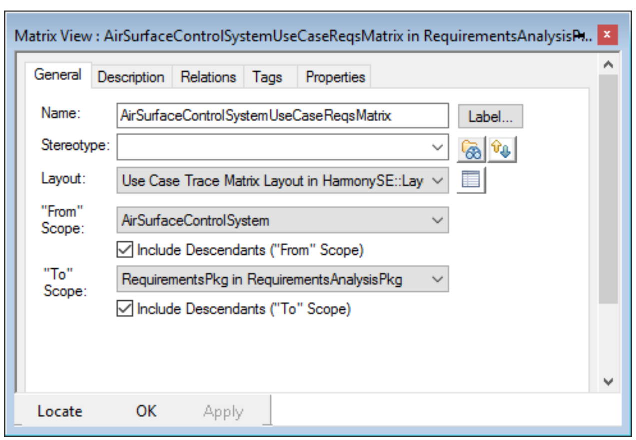
Rhapsody Matrices

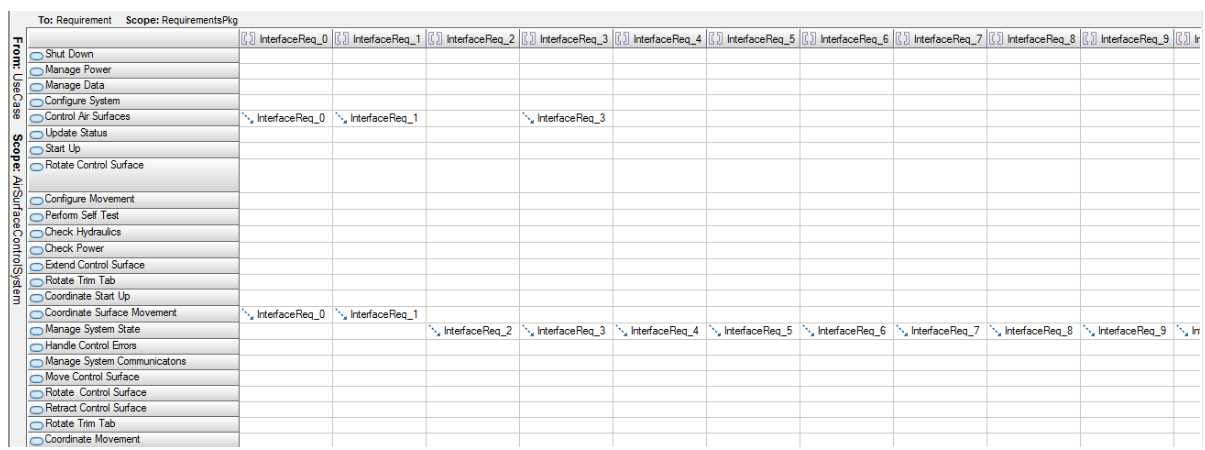
- Matrices are primarily use to show specifically related model elements of different kinds

- Layout



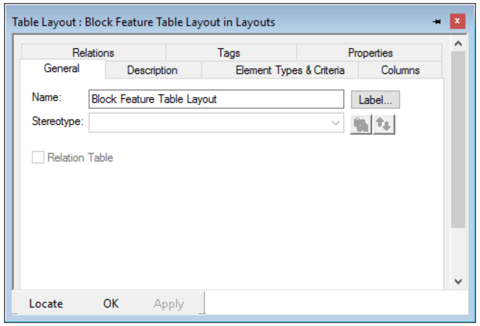
- View

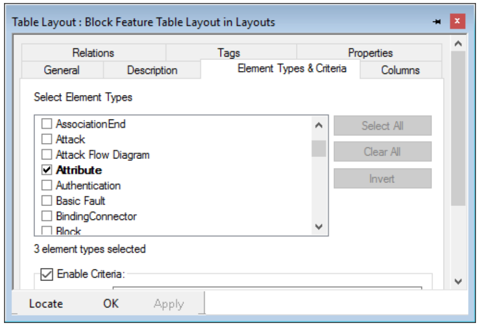


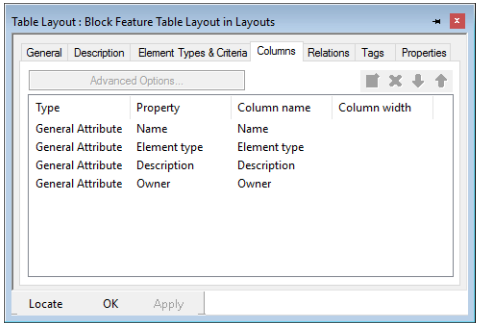


Rhapsody Tables

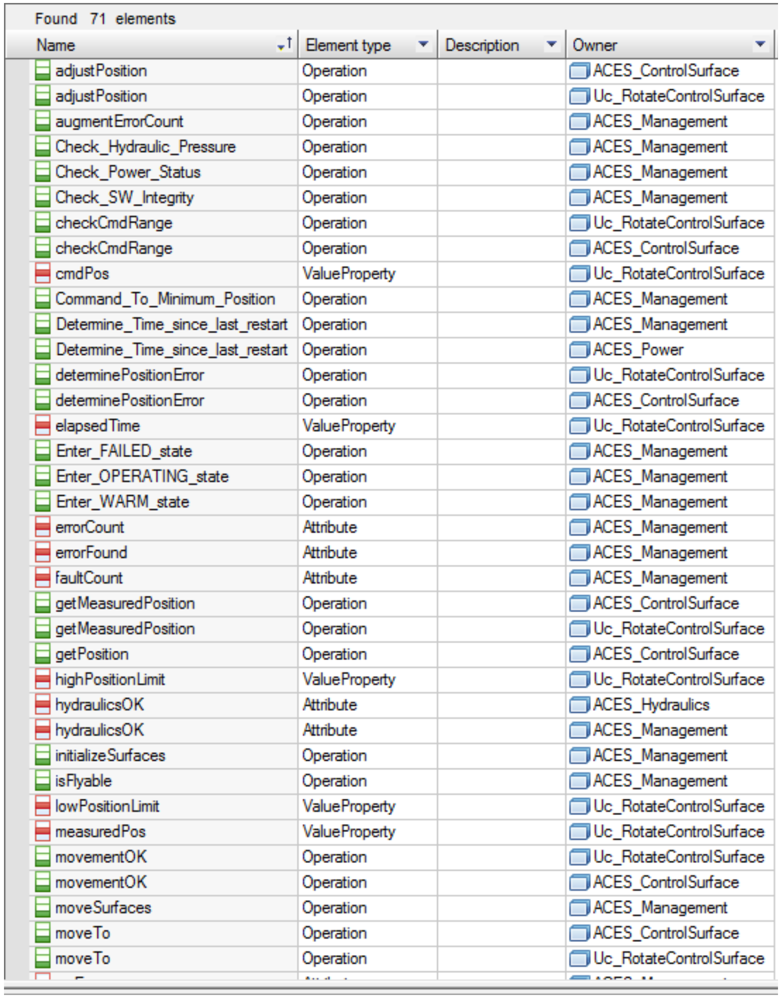
- Tables are primarily used to show metadata around elements of a common kind
- Layout







View



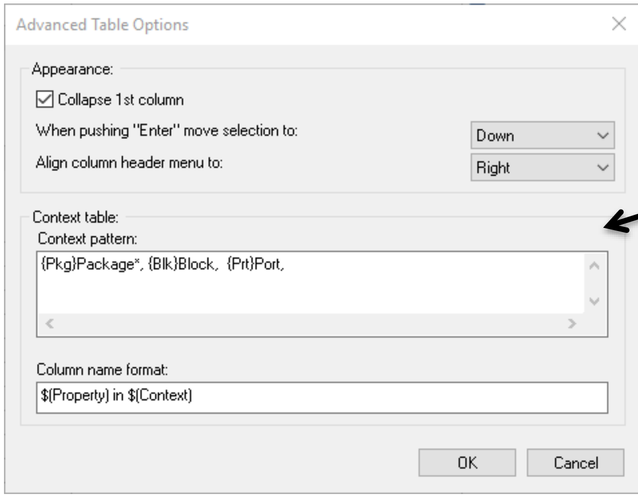
Found 71 elements

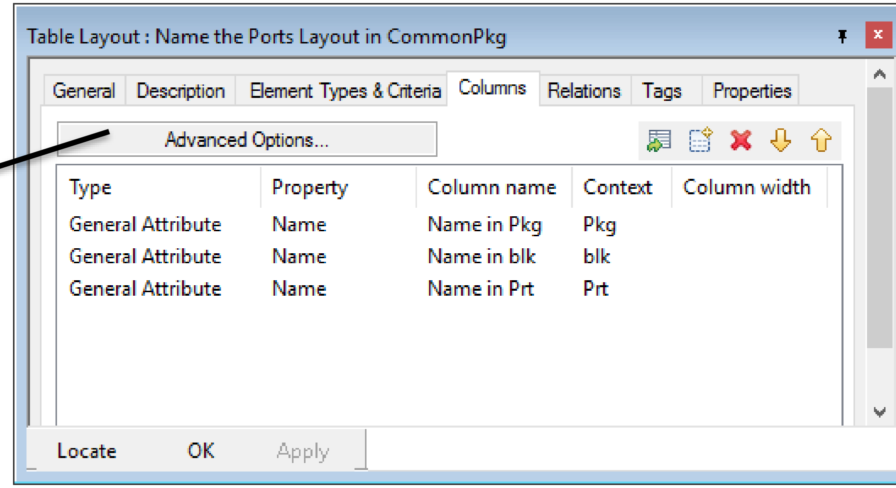
Name	Element type	Description	Owner
adjustPosition	Operation		ACES_ControlSurface
adjustPosition	Operation		Uc_RotateControlSurface
augmentErrorCount	Operation		ACES_Management
Check_Hydraulic_Pressure	Operation		ACES_Management
Check_Power_Status	Operation		ACES_Management
Check_SW_Integrity	Operation		ACES_Management
checkCmdRange	Operation		Uc_RotateControlSurface
checkCmdRange	Operation		ACES_ControlSurface
cmdPos	ValueProperty		Uc_RotateControlSurface
Command_To_Minimum_Position	Operation		ACES_Management
Determine_Time_since_last_restart	Operation		ACES_Management
Determine_Time_since_last_restart	Operation		ACES_Power
determinePositionError	Operation		Uc_RotateControlSurface
determinePositionError	Operation		ACES_ControlSurface
elapsedTime	ValueProperty		Uc_RotateControlSurface
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_RotateControlSurface
getPosition	Operation		ACES_ControlSurface
highPositionLimit	ValueProperty		Uc_RotateControlSurface
hydraulicsOK	Attribute		ACES_Hydraulics
hydraulicsOK	Attribute		ACES_Management
initializeSurfaces	Operation		ACES_Management
isFlyable	Operation		ACES_Management
lowPositionLimit	ValueProperty		Uc_RotateControlSurface
measuredPos	ValueProperty		Uc_RotateControlSurface
movementOK	Operation		Uc_RotateControlSurface
movementOK	Operation		ACES_ControlSurface
moveSurfaces	Operation		ACES_Management
moveTo	Operation		ACES_ControlSurface
moveTo	Operation		Uc_RotateControlSurface

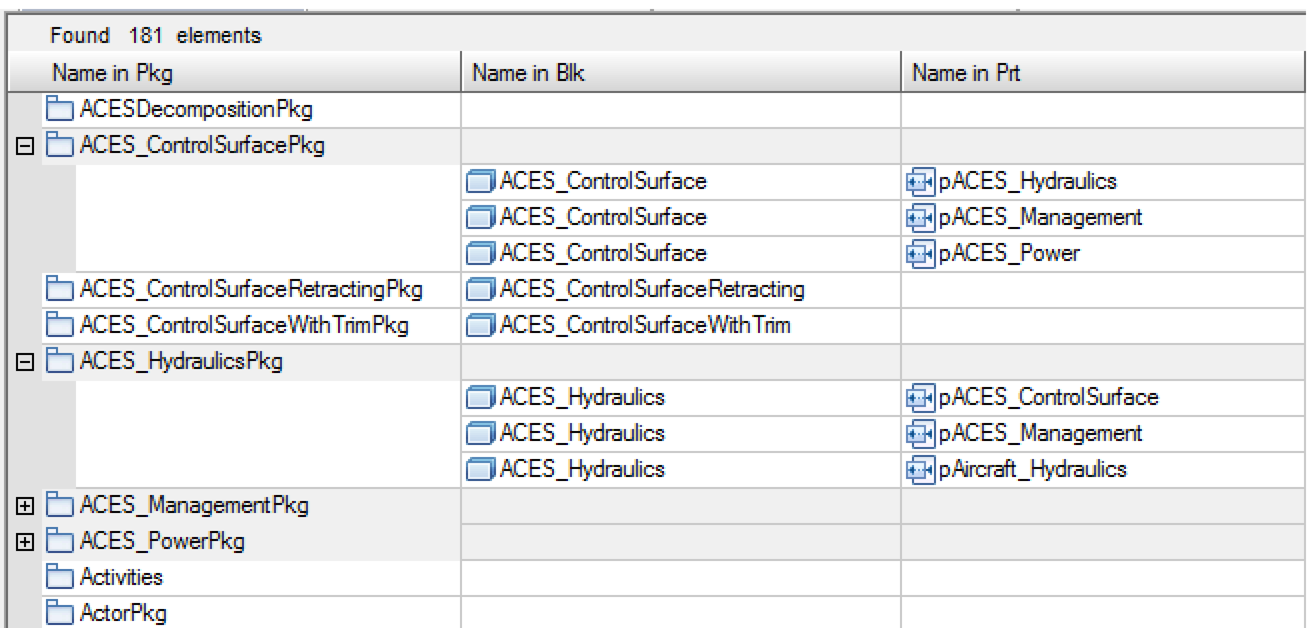
Rhapsody Tables and Context Patterns

- 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



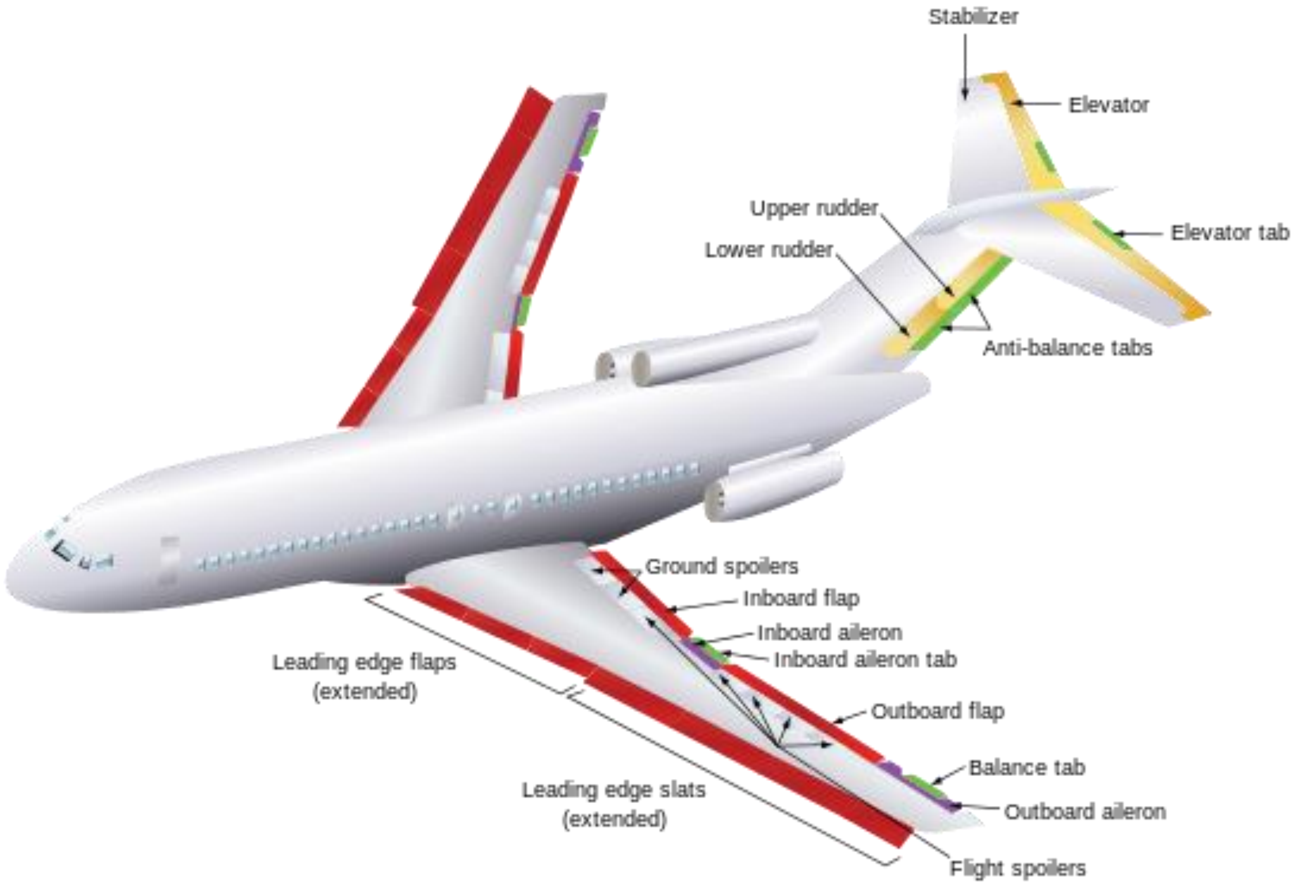




Agenda

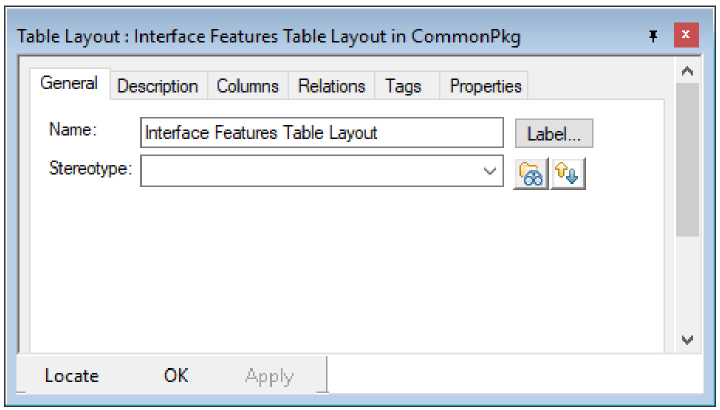
- What’s an ICD?
- What’s in an Interface?
- Common ICD Forms
- Where to interfaces come from?
- Interfaces in SysML
- Tables and matrices in Rhapsody
- Example

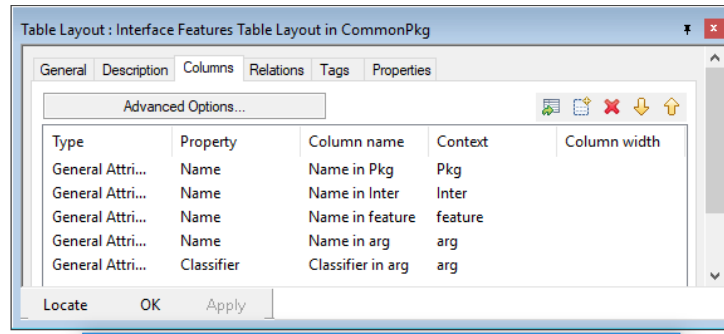
The Air Control Surfaces Enactment System (ACES)

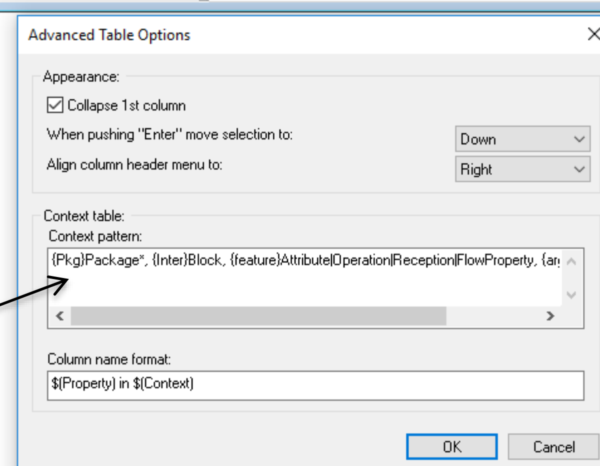


ACES Example: Logical Interfaces

- 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







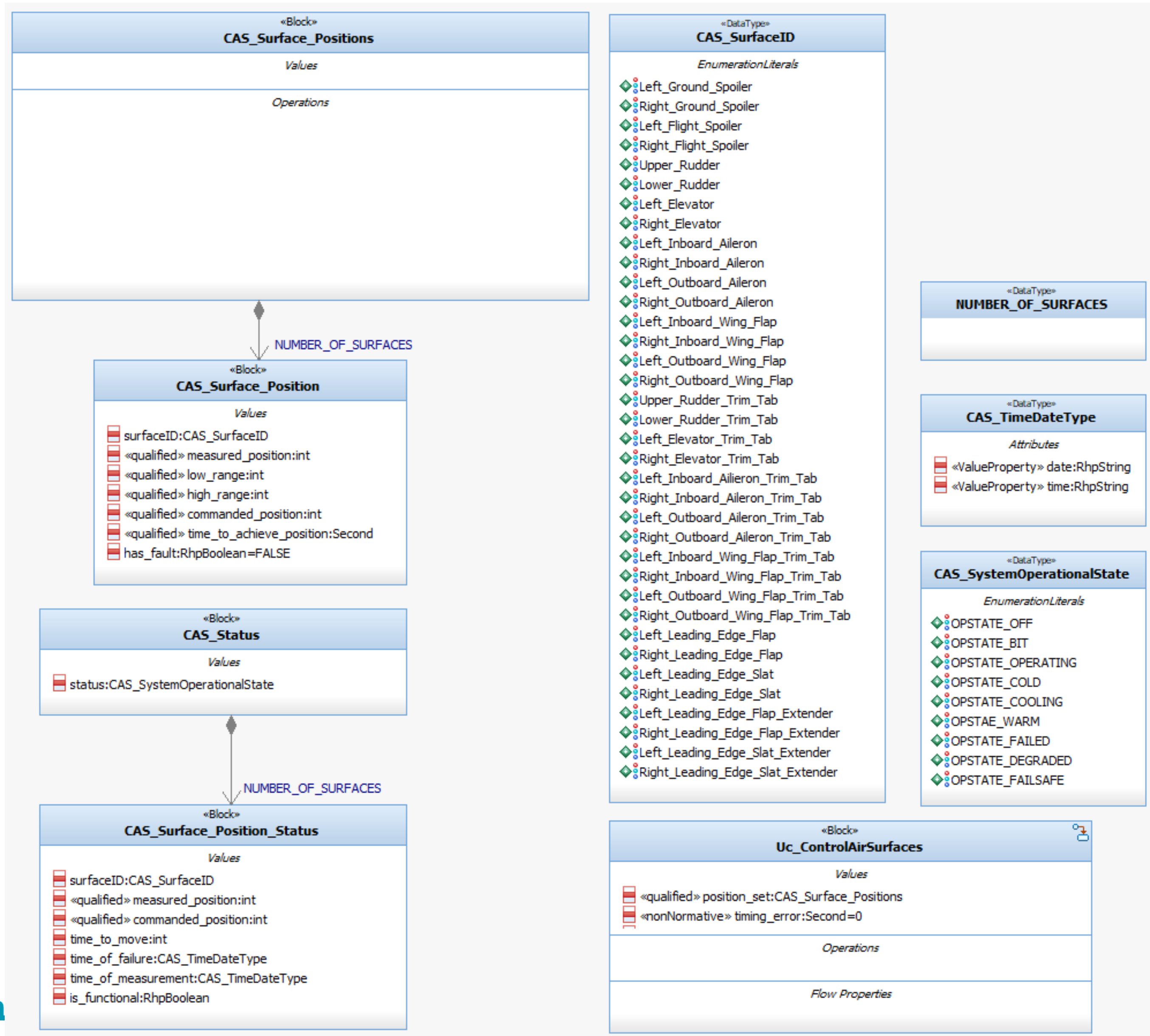
{Pkg}Package*, {Inter}Block, {feature}Attribute|Operation|Reception|FlowProperty, {arg}Argument

Resulting Logical Interfaces Table

Found 42 elements

Name in Pkg	Name in Inter	Name in feature	Name in arg	Classifier in arg
SubsystemInterfacesPkg				
	iACES_Control_Surface_ACES_Hydraulics	hydraulic_pressure		
	iACES_Control_Surface_ACES_Power	power		
	iACES_Hydraulics_Aircraft_Hydraulics	evRequest_Hydraulic_Status		
	iACES_Hydraulics_Aircraft_Hydraulics	hereza_Hydraulic_Pressure	hs	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	SW_Status	id	SurfaceID
	iACES_Management_ACES_Control_Surface	SW_Status	sw_status_msg	SW_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	Check_Hydraullics		
	iACES_Management_ACES_Hydraulics	Request_SW_Integrity_Check		
	iACES_Management_ACES_Hydraulics	SW_Status	id	SurfaceID
	iACES_Management_ACES_Hydraulics	SW_Status	sw_status_msg	SW_Status
	iACES_Management_ACES_Hydraulics	Update_Hydraulic_Status	h_status	Hydraulic_Status1
	iACES_Management_ACES_Power	Request_SW_Integrity_Check		
	iACES_Management_ACES_Power	SW_Status	id	SurfaceID
	iACES_Management_ACES_Power	SW_Status	sw_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	evDisable		
	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	reqENABLE_Command		
	iACES_Power_Aircraft_Power	Operation_1	x	int
	iACES_Power_Aircraft_Power	Operation_1	y	double
	iACES_Power_Aircraft_Power	Select_Power_Source	source	POWERSOURCE...
	iACES_Power_Aircraft_Power	evRequest_Power_Status	source	POWERSOURCE...
	iACES_Power_Aircraft_Power	evSelect_Battery_As_Source		
	iACES_Power_Aircraft_Power	hereza_Power_Status	ps	Power_Status1
	iACES_Power_Aircraft_Power	power		

Logical Data Schema (Graphical View)

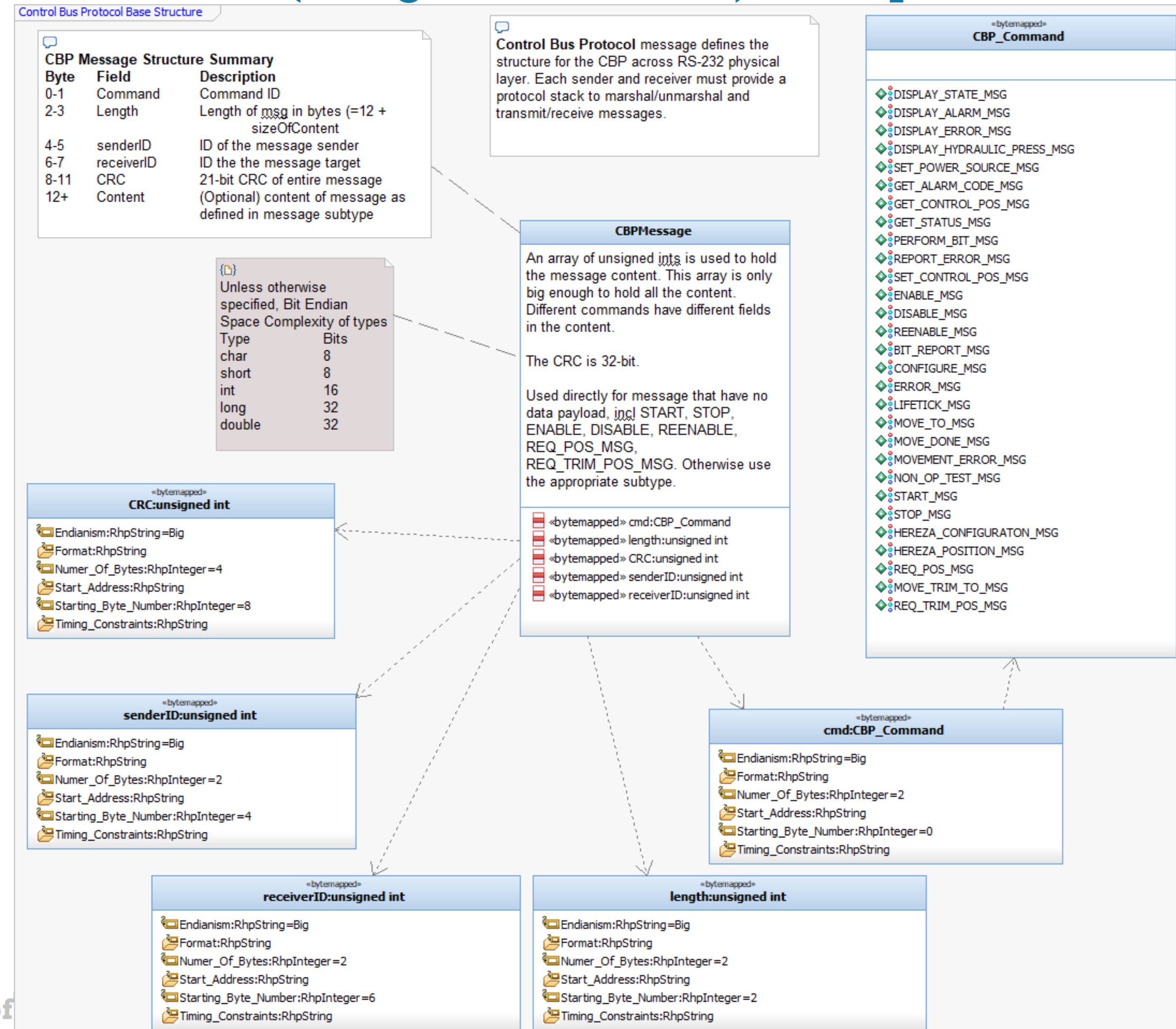


Logical Data Schema (Tabular)

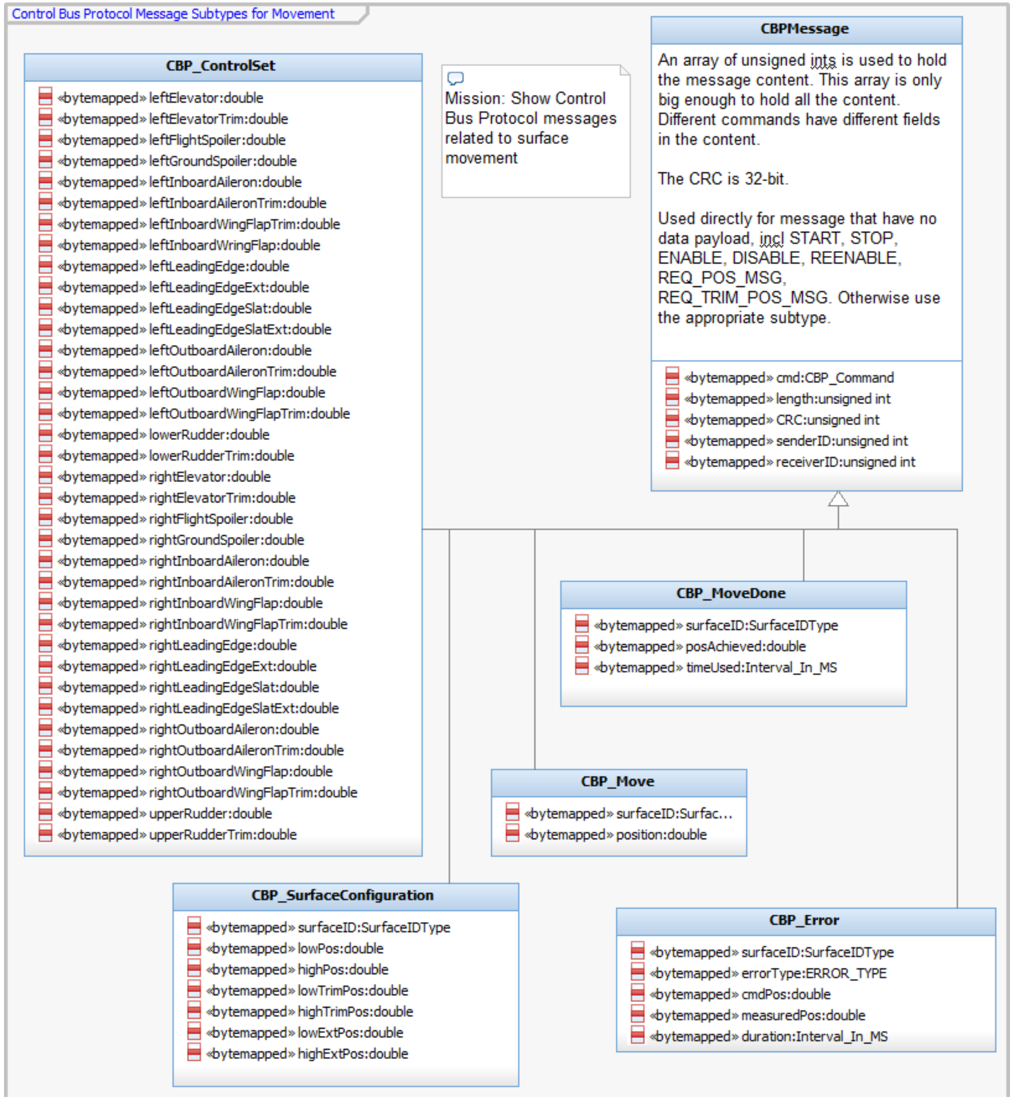
Found 56 elements

Name in pkg	Name in type	Name in feature	Classifier in feature
DataTypesPkg			
	ERROR_TYPE		
	Error_Log		
	Error_Report	date_time	TimeDateType
	Error_Report	error	ERROR_TYPE
	Error_Report	source	POWERSOURCE_TYPE
	Error_Report	surfaceID	SurfaceID
	Hydraulic_Status1	has_faults	RhpBoolean
	Hydraulic_Status1	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...		
	SURFACE_POSITION_TOLERANCE		
	Status	status	SystemOperationalState
	SurfaceID		
	Surface_Configuration_Type	high_limit	int
	Surface_Configuration_Type	low_limit	int
	Surface_Position	commanded_position	int
	Surface_Position	has_fault	RhpBoolean
	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	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	getFault_Status	
	Surface_Positions	getSurface_Position	
	Surface_Positions	isIn_Range	
	Surface_Positions	setFault_Status	
	Surface_Positions	setMeasured_Position	
	Surface_Positions	setSurfaceID	
	Surface_Positions	setSurface_Position	
	Surface_Positions	setSurface_Range	
	Surface_Positions	setTime_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	testID	TEST_TYPE

Physical Data Schema (using CBP Protocol) – Graphical



Physical Schema – Message Types



Physical – Logical Trace Table

Name in pkg	Name in PhysicalRepresentation	To element in LogicalRepresenta...
MessageTypesPkg		
	CBPMessage	checkPower
	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	evMoveTo
	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	ReportError
	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	evGotoFailedState
	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	Numer_Of_Bytes	2
CBP_Error	measuredPos	double	Usage	Measured position after movement completes
CBP_Error	measuredPos	double	Numer_Of_Bytes	4
CBP_Error	measuredPos	double	Format	4-byte IEEE floating point format
CBP_Error	measuredPos	double	Starting_Byte_Number	7
CBP_Error	when	TimeDate_Type		
CBP_Error	cmdPos	double	Starting_Byte_Number	3
CBP_Error	cmdPos	double	Usage	Commanded position
CBP_Error	cmdPos	double	Format	4-byte IEEE floating point format
CBP_Error	cmdPos	double	Numer_Of_Bytes	4
CBP_Error	surfaceID	SurfaceIDType	Start_Address	0
CBP_Error	surfaceID	SurfaceIDType	Numer_Of_Bytes	1
CBP_Error	errorType	ERROR_TYPE	Numer_Of_Bytes	2
CBP_Error	errorType	ERROR_TYPE	Start_Address	1
CBP_Error	duration	Interval_In_MS	Starting_Byte_Number	11
CBP_HydraulicStatus	status	HydraulicStatus		
CBP_Move	position	double	Numer_Of_Bytes	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	0
CBP_MoveDone	posAchieved	double	Format	4-byte IEEE floating point format
CBP_MoveDone	posAchieved	double	Endianism	Big
CBP_MoveDone	surfaceID	SurfaceIDType	Numer_Of_Bytes	1
CBP_MoveDone	surfaceID	SurfaceIDType	Usage	ID of the referenced control surface
CBP_MoveDone	posAchieved	double	Numer_Of_Bytes	4
CBP_MoveDone	posAchieved	double	Usage	The measured position achieved in movement
CBP_MoveDone	posAchieved	double	Starting_Byte_Number	1
CBP_MoveDone	timeUsed	Interval_In_MS	Numer_Of_Bytes	4
CBP_MoveDone	timeUsed	Interval_In_MS	Endianism	Big
CBP_MoveDone	surfaceID	SurfaceIDType	Endianism	Big
CBP_MoveDone	timeUsed	Interval_In_MS	Starting_Byte_Number	5
CBP_MoveDone	timeUsed	Interval_In_MS	Usage	Duration 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:

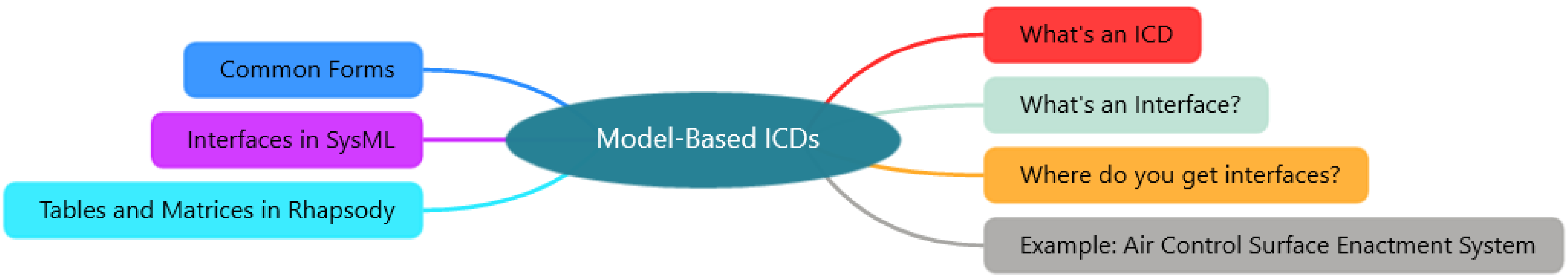
Overview
- Reference model
- Scope
- Date Created

Interfaces & Interface blocks, each with
- Name
- **Value and Flow Properties**
  - Name
  - Value description
  - Type
  - Tags
- **Services**
  - Name
  - Service Description
  - Parameters
    - Parameter type
  - Return type
  - Tags
- Description

Flow ports
- Name
- Owning Classifier
- Type
- Tags

Types and Classifiers
- **Blocks and Classes**
  - Name
  - **Value and Flow Properties**
    - Name
    - Description
    - Type
    - Tags
  - **Services**
    - Name
    - Service Description
    - Parameters
      - Parameter type
    - Return type
    - Tags
- **Types, Value Types, and Data Types**
  - Name
  - Description
  - Based Type
    - If Enumeration, then list enumerations

Summary


Model-Based ICDs
Common Forms
Interfaces in SysML
Tables and Matrices in Rhapsody
What's an ICD
What's an Interface?
Where do you get interfaces?
Example: Air Control Surface Enactment System

Real-Time Agile Systems and Software Development

Welcome to **www.bruce-douglass.com**

You've found yourself on **www.bruce-douglass.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 About 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.