ARINC 429 data buses in AADL

Roger Champagne
Dept. of Software and IT Engineering
École de technologie supérieure (ÉTS)
Montreal, Canada
roger.champagne@etsmtl.ca

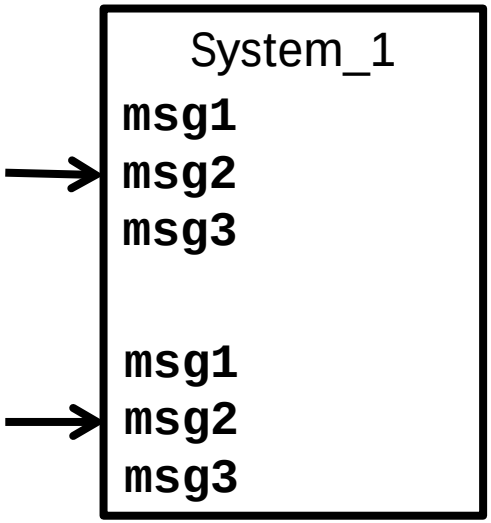
ÉTS
Le génie pour l'industrie
Dept. of Software and IT Engineering

Context

- Research project → Avionics systems integration
  - Interface Control Document (ICD) Management
    - Goal: to capture ICDs of avionics systems in a systematic way
- Assumptions
  - Specifications of avionics systems can be captured in AADL
  - ICDs can be generated from AADL model
    - For both federated and IMA
- This presentation
  - How to model ARINC 429 messages in AADL?
  - Just enough so that we can
    - Validate sending and receiving ends understand the same messages
    - Generate the ICDs

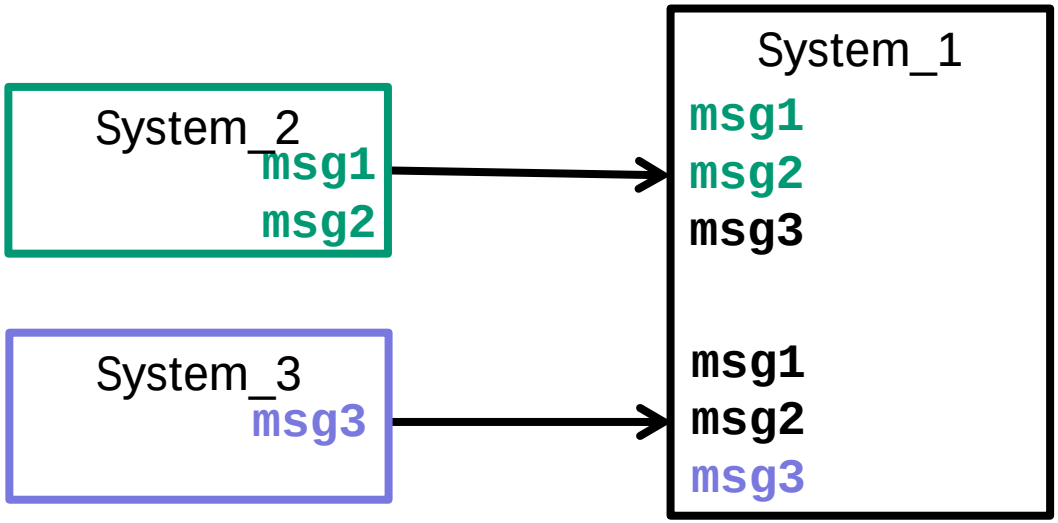
Intention [1/]

- Context diagram, informal notation



- Can receive 3 different input messages
- 2 input ports, both understand 3 messages

Intention [2/]



Different systems might be sending the 3 different messages that System_1 understands

Q1: How to model the individual messages?

Q2: How to bind messages to ports?

ILS receiver ARINC 429 messages

Parameter	Label		ILS Operational Range	Max Transmit Interval msec	BNR Word Range	Binary Output		BCD Output	
	Type	No.				Sig Bits	Resolution Units	Sig Digs	Least Signif Digits
Localizer Deviation	BNR	173	±0.155 DDM (min)	62.5	±0.4	12	0.0001 DDM	--	--
Glide Slope Deviation	BNR	174	±0.22 DDM (min)	62.5	±0.8	12	0.0002 DDM	--	--
ILS Frequency	BCD	033	108.00 - 111.95	200	--		--	4	0.01 MHz
Runway Heading	BCD	017	0 - 359.9°	200	--	--	--	--	--
Runway Heading	BNR	105	0 - 359.9°	50	±180°	11	0.09°	4	0.1°
ILS Ground Station Ident	Special	263 & 264	--	200	--	--	--	--	--

Table 3-2 BNR Deviation Data Coding Examples (Twos Complement Fractional Notation)

Bit No.	32	31 30 29	28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11	10 9	8 7 6 5 4 3 2 1
Function	Py (2)	SSM	DATA FIELD (3)	SDI	LABEL
Loc + 0.021 DDM	1	1 1 0	0 0 0 0 1 1 0 1 1 0 0 0 P P P P P (1)	0 0	1 1 0 1 1 1 1 0
G.S.- 0.125 DDM	1	1 1 1	1 1 0 1 1 0 0 0 0 0 0 0 P P P P P (1)	0 0	0 0 1 1 1 1 1 0

Notes:
(1) Bit no. 11 takes on the binary state "one" to annunciate that the ILS receiver is in the "tune/functional test inhibit" condition. Section 3.4 of this document refers
(2) Bit no. 32 is set to render word parity odd
(3) P denotes pad bit (binary "zero" or valid data)

Source: Aeronautical Radio Inc., Mark 2 airborne ILS receiver, ARINC characteristic 710-10, 1997.

Attempts: ARINC 429 messages [1/]

- Data components w/ extended OSATE ARINC 429 property set

```
property set ARINC429_mod is
  WordID : aadlinteger applies to (data);
  FirstBit : aadlinteger 11 .. 28 applies to (data);
  NumberBits : aadlinteger 1 .. 18 applies to (data);
  Encoding: enumeration (BNR, BCD) applies to (data);
  DataRange: aadlreal applies to (data);
  MessageNumberOfWords: aadlinteger applies to (data);
  Label: aadlinteger applies to (data); -- octal...
  DataUnits: aadlstring applies to (data);
end ARINC429_mod;
```

Attempts: ARINC 429 messages [2/]

```
-- abstract to prevent anyone instantiating this type. Intent is to
-- define a template for all ARINC 429 messages exchanged in modeled
-- avionics subsystems
abstract basic_message
    properties
        Data_Size => 32 bits;
end basic_message;

-- abstract implementation with default values
abstract implementation basic_message.impl
    properties
        ARINC429_mod::Encoding => BNR;
        ARINC429_mod::DataRange => 0.0;
        ARINC429_mod::NumberBits => 1;
        ARINC429_mod::Label => 0;
        ARINC429_mod::MessageNumberWords => 1;
        ARINC429_mod::DataUnits => "";
        ARINC429_mod::FirstBit => 11;
end basic_message.impl;
```

Attempts: ARINC 429 messages [3/]

```
-- ILS frequency is input to the Instrument Landing System (ILS)
-- receiver. It can come from various sources. Under certain conditions,
-- it must also be relayed to ILS outputs. See ARINC 710-10.
data ils_frequency extends basic_message
end ils_frequency;

data implementation ils_frequency.impl extends basic_message.impl
    properties
        ARINC429_mod::Encoding => BCD;
        ARINC429_mod::DataRange => 0.0;
        ARINC429_mod::NumberBits => 15;
        ARINC429_mod::Label => 0;
        ARINC429_mod::MessageNumberWords => 1;
        ARINC429_mod::DataUnits => "MHz";
        ARINC429_mod::FirstBit => 15;
end ils_frequency.impl;
```

Binding messages to ports

- Goal: validation at both ends of a connexion
- Tried to bind aggregate data components to ports
  - Semantics: exactly the same aggregate at both ends
  - No "subset" property, exact match only
- Alternative solution to explore: add H/W
  - Bind connexion between systems to bus
  - Virtual buses as subcomponents of bus
    - 1 virtual bus per message
  - Bind SW components to appropriate VB at both ends to type match specific messages

Eventually

- Capture other variable aspects related to messages in AADL model
  - Message content depends on certain pins being connected certain ways (modes?)
  - ARINC 429 outputs as time multiplexed sequences of multiple messages at various rates
  - …

Thank you!