SR-22-004

DATA SCIENCES GROUP

INTERFACE CONTROL DOCUMENT STANDARDS INVESTIGATION

ABERDEEN TEST CENTER
DUGWAY PROVING GROUND
ELECTRONIC PROVING GROUND
REAGAN TEST SITE
REDSTONE TEST CENTER
WHITE SANDS TEST CENTER
YUMA PROVING GROUND

NAVAL AIR WARFARE CENTER AIRCRAFT DIVISION PATUXENT RIVER
NAVAL AIR WARFARE CENTER WEAPONS DIVISION CHINA LAKE
NAVAL AIR WARFARE CENTER WEAPONS DIVISION POINT MUGU
NAVAL SURFACE WARFARE CENTER DAHLGREN DIVISION
NAVAL UNDERSEA WARFARE CENTER DIVISION KEYPORT
NAVAL UNDERSEA WARFARE CENTER DIVISION NEWPORT
PACIFIC MISSILE RANGE FACILITY

96TH TEST WING
412TH TEST WING
ARNOLD ENGINEERING DEVELOPMENT CENTER

SPACE LAUNCH DELTA 30
SPACE LAUNCH DELTA 45

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

This page intentionally left blank.

SR-22-004

INTERFACE CONTROL DOCUMENT STANDARDS INVESTIGATION

June 2022

Prepared by

DATA SCIENCES GROUP

Published by

Secretariat
Range Commanders Council
U.S. White Sands Missile Range
New Mexico 88002-5110

This page intentionally left blank.

Table of Contents

This page intentionally left blank.

Preface

This document is the product of a task to address shortcomings related to interface control documents. The objectives of the task are for the Data Sciences Group of the Range Commanders Council to:

- formally assess the need for a human- and machine-readable interface control document standard;
- complete a survey to identify existing candidate solutions; and
- provide a recommendation as to whether the Data Sciences Group should develop a standard for the Range Commanders Council.

The Data Sciences Group developed this task in three phases: requirements and use case development; market research; and drafting a white paper (this document).

For questions regarding this document, please contact the Range Commanders Council Secretariat.

Secretariat, Range Commanders Council
ATTN: TEWS-TDR
1510 Headquarters Avenue
White Sands Missile Range, New Mexico 88002-5110
Telephone: (575) 678-1107, DSN 258-1107
E-mail: rcc-feedback@trmc.osd.mil

This page intentionally left blank.

Acronyms

DFDL	Data Format Description Language
DSG	Data Sciences Group
ICD	Interface Control Document
JSON	JavaScript Object Notation
MDL	Metadata Description Language
OEM	original equipment manufacturer
RCC	Range Commanders Council
SPC	Standard Parameter Class
XML	eXtensible Markup Language
XSD	XML Schema Document

This page intentionally left blank.

1. Executive Summary and Recommendation

There is no standard form used to define binary messages from sources such as 1553, Ethernet, fiber, etc. Government software teams, contractor software teams, vendors, and original equipment manufacturers (OEMs) all use a variety of solutions including eXtensible Markup Language (XML), Access databases, Excel spreadsheets, Word documents, PDF files, etc. to document the binary message. Government test tools must convert these formats into intermediate formats, software header files, etc. to facilitate software interpreting the data. Government test team software engineers are often required to spend hours scrutinizing documents to find changes from vendor updates. The teams then spend more hours updating software and testing to ensure that all changes were identified and properly implemented. Program offices have repeatedly asked test ranges what format is desired for OEMs to deliver Interface Control Documents (ICDs) in, and currently we have no standard to point to. In the future, programs could identify the Range Commanders Council (RCC) ICD standard in contracts. Test teams could write software to one standard instead of multiple formats. Multiple, if not all, ranges could benefit from this future standard, including all the ranges represented on the task committee: Naval Air Warfare Center Aircraft Division, Naval Air Warfare Center Weapons Division, 412th Test Wing, and Redstone Test Center.

The objectives of this task were for the Data Sciences Group (DSG) to formally assess the need for a human- and machine-readable RCC ICD standard, complete a survey to identify existing candidate solutions, and provide a recommendation as to whether the DSG should proceed with developing an RCC standard.

The committee formally assessed the need for a human- and machine-readable RCC ICD standard and determined that the DoD and range community will achieve significant cost avoidance and be able to develop software with fewer bugs on shorter timelines if an RCC ICD standard existed. The committee developed a draft ICD Standards Requirements document and developed draft user stories that ICD standard should meet.

The committee completed a market survey and identified three primary candidates: the Data Format Description Language (DFDL), Standard Parameter Class (SPC), and RCC 106 Chapter 23[1]; all of which have active user communities.

The authors of this document recommend that the RCC proceed forward in developing an ICD Standard in a follow-on omnibus funded task.

2. Example Requirement and User Stories

The following sections are starting points for what requirements and user stories an ICD standard should meet. The requirements and user stories sections are not complete.

2.1 Requirements

1. Overarching Requirements

[1] Range Commanders Council. “Metadata Configuration.” In *Telemetry Standards*. RCC 106-22. May 2022. May be superseded by update. Retrieved 9 May 2022. Available at https://www.trmc.osd.mil/wiki/display/publicRCC/106+Telemetry+Standards.

a. The standard shall provide the capabilities that enable the user to encode and decode all messages from the following architectures.

(1) MIL-STD-1553
(2) ARINC-429
(3) Ethernet
(4) Fibre Channel
(5) IEEE-1394
(6) RS-232
(7) RS-422
(8) ARINC-664

b. The standard shall be machine (software) readable.
c. The standard shall be human (ASCII) readable.

2. The standard shall support the following data types.
   a. Integer
   b. Unsigned Integer
   c. Real
   d. Enumeration
   e. String
   f. Bit
3. The standard shall support the following bit backing.
   a. Bit packed
   b. Byte aligned
4. The standard shall support the following byte orders.
   a. Big Endian
   b. Little Endian
   c. Middle Endian
5. The standard shall support the following EU Conversion Types (conversion types to be determined).
6. The standard shall support for following floating point reals.
   a. Scaled (multiplier)
   b. IEEE Single Precision
   c. IEEE Double Precision
   d. VAX Floating Point
   e. MIL-STD-1750[2] - 48-bit float
   f. IRIG-106 Appendix 9-D[3] (contain 12 different floating point types)
7. The standard shall support the following angle categories.
   a. 360 degrees (0 - 360)

[2] Department of Defense. *Sixteen-Bit Computer Instruction Set Architecture*. MIL-STD-1750A. 2 July 1980. Inactivated 31 July 1996. Retrieved 5 May 2022. Available at https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=37110.
[3] Range Commanders Council. "Floating Point Formats." Appendix 9-D in *Telemetry Standards*. RCC 106-22. May 2022. May be superseded by update. Retrieved 9 May 2022. Available at https://www.trmc.osd.mil/wiki/display/publicRCC/106+Telemetry+Standards.

   b. 180 degrees (−180 - 180)
   c. 2pi radians (0 - 2π)
   d. pi radians (−π - π)
8. The standard shall allow the following for each message definition.
   a. Name (string)
   b. Description (string)
   c. Length (int)
   d. Sub-Messages
   e. Others pending future development of new message definitions
9. The standard shall allow the following for each field within a message.
   a. Name (string)
   b. Description (string)
   c. Bit Length (int)
   d. ???
10. The standard shall allow specialized formats for each field within a message.
   a. Name (string)
   b. Algorithm or formula for the encoding/decoding of those field (string)
   c. Description (string)
   d. Bit Length (int)

2.2 User Stories

1. As a user, I would like to use the ICD standard to enable my software to decode and encode the following types of data.
   a. MIL-STD-1553
   b. ARINC-429
   c. Ethernet
   d. Fibre Channel
   e. IEEE-1394
   f. RS-232
   g. RS-422
   h. ARINC-664 (AFDX)
   i. PCM
   j. CSV files
   k. XML files
   l. ...and other digital formats
2. As a user, I would like messages/buses/files definitions defined using the ICD standard to be machine (software) readable, enabling my software to encode and decode binary data.
3. As a user, I would like messages/buses/files definitions defined using the ICD standard to be human (ASCII) readable.
4. As a user, I would like to be able to read the human-readable file as a stand-alone document that does not need to be supplemented by accompanying documentation.
5. As a user, I would like to be able to leverage the same ICD for performing post-processing persisted raw file captures of the real-time data streams.

6. As a user, I would like to be able to extend the ICD through custom markup if needed. To re-word things: the ICD format should be open for custom extensions.
7. As a user, I would like to be able to deliver the ICD standard to an OEM aircraft vendor and receive their ICD in the standard format.

3. Candidate Solutions

Presented in alphabetical order in this document the following three candidate solutions were identified and examined: DFDL, RCC 106 Chapter 23, and SPCs.

4. Data Format Description Language (DFDL)

4.1 Overview

The DFDL is a general-purpose modeling language that aims to describe general text and binary data in a standard way. It does this by building on top of the existing XML schema standard - only a subset of XML schema is used - and then extending it though a set of custom XML elements and attributes that are used to describe things such as memory layout, ordering, and data validation.

At its core, DFDL could support the needed information for describing a dataset, though it would not necessarily support additional features discussed (such as semantic field tagging). Being an XML-based schema means that additional functionality to support such features could be done through additional XML namespaces.

4.2 Existing Parser Implementations

Currently there are only two big players with regards to implementations: An Apache Daffodil implementation and an IBM implementation that is used in their IBM Integration Bus, which is an Enterprise Service Bus implementation.

The Apache Daffodil implementation is free to download and open source through the Apache v2 License. The sample implementation comes as a command line tool which will convert a source file using a defined DFDL schema into either an XML or a JSON file. In theory you could extend the source to support a custom output handler.

This implementation has a handful of unsupported features, which is functionality supported by the DFDL specification but not handled by the Daffodil specification.

During some testing with this implementation some errors occurred with regards to running out of heap space. This could have been user error during exploration or the tooling itself and definitely would not indicate a fault of the specification itself.

5. IRIG 106 Chapter 23 – Metadata Configuration

Chapter 23 of RCC 106 presents an XML-based Metadata Description Language (MDL) that can be used to describe an entire test configuration from the test article to the ground station and including the RF network. This chapter includes an XML schema document (XSD) along with examples of specific implementations to capture common aircraft data such as MIL STD 1553 and ARINC-429.

5.1 Pros

- Chapter 23 makes provision for incorporating IRIG 106 Chapter 9 Telemetry Attributes Transfer Standard (TMATS)[4] within the standard. This provides a bridge between existing data definitions using the TMATS.
- It is already a part of an existing RCC standard that will promote interoperability between vendors.
- The MDL makes it possible to describe data structures representing specific data types within the payload of the carrier. This could be any data source such as Ethernet, Fibre Channel, MIL STD 1553 or ARINC-664.

5.2 Cons

- Any linkage to a standard maintained by another committee could cause delays in implementing changes or updates.
- The RCC standards are not supported with software as effectively as public open-source standards.

5.3 Example

Appendix A of the 106 Chapter 23 provides a link to examples in a single zip file. These examples include all-bus capture for ARINC-429 and MIL-STD 1553 plus how to define individual parameters from these busses.

Defining the primary information for a MIL-STD 1553 message uses easy-to-read XML tags as illustrated here.

```
  <MILSTD1553Messages>
    <MILSTD1553Message ID="MS1553_Bus_NAV_MSG1">
      <Name>unused</Name>
      <Description>unused</Description>

<MILSTD1553MessageType>RemoteTerminalToRemoteTerminal</MILSTD1553Messa
geType>
      <MILSTD1553RemoteTerminal>13</MILSTD1553RemoteTerminal>
      <MILSTD1553SubaddressOrMode>7</MILSTD1553SubaddressOrMode>
      <MILSTD1553TransmitReceive>Receive</MILSTD1553TransmitReceive>

<MILSTD1553WordCountOrModeCode>30</MILSTD1553WordCountOrModeCode>
```

The examples from Chapter 23 Appendix A also include how to define individual measurements. The following is an example of how a data conversion or data operation would be defined for a specific measurement.

```
 <DataOperation ID="AIRSPD_EU_CONV">
    <Name>Airspeed EU Conversion</Name>
    <Description>TBD description</Description>
```

[4] Range Commanders Council. "Telemetry Attributes Transfer Standard." In *Telemetry Standards*. RCC 106-22. May 2022. May be superseded by update. Retrieved 9 May 2022. Available at https://www.trmc.osd.mil/wiki/display/publicRCC/106+Telemetry+Standards.

```
    <DataOperationType>CountstoEU</DataOperationType>
    <Method ID="AirspeedMethod">
      <Name>TBD name</Name>
      <Description>TBD description</Description>
      <MathematicalExpression>
        <PolynomialNumerator>
          <Term>
            <Coefficient>0.5</Coefficient>
            <Exponent>1</Exponent>
          </Term>
        </PolynomialNumerator>
      </MathematicalExpression>
    </Method>
    <InputUnits>
      <SIUnits>Counts</SIUnits>
    </InputUnits>
    <OutputUnits>
      <SIUnits>Knots</SIUnits>
    </OutputUnits>
  </DataOperation>
```

The 106 Chapter 23 also includes a link to the most current XSD posted to the RCC library site.

6. Standard Parameter Classes (SPCs)

6.1 Overview

Developed as government off-the-shelf by the Atlantic Test Ranges, the SPC library is an application programming interface that provides an easy way for creating and interpreting data messages and parameters contained within those messages. Built in order to become more agile and responsive to evolving range requirements, SPC defines message and parameter content using a collection of XML files instead of using hardcoded data structures. These XML definitions can then be easily shared, managed, and utilized across various applications across the range. The SPC library is provided in C++ source code format and is written in a portable, cross-platform manner.

Rather than a data transmission or network protocol, SPC is a serialization library that can be used in conjunction with other data transmission libraries.

The SPC classes provide a group of easy-to-use methods that enable the programmer to do the following.

- Create messages based on new or existing message definitions.
- Save message definitions.
- Create parameters based on new or existing parameter definitions.
- Save parameter definitions.
- Set or get raw values for a block of data in a message.
- Set or get raw values for individual parameters in a message.
- Get fully converted (engineering units) values for individual parameters in a message.

The following are features that SPC supports.

- Bit-packed and byte-aligned data
- Endian order: Big, little, and middle
- Data types: signed integer, unsigned integer, real, strings, bits, and enumerations
- Unit categories: position, velocity, acceleration, angular position, angular velocity, time, generic integer values, and unitless blobs
- Inherited message definitions
- Simple data arrays
- On-the-fly format switching
- Composite parameters
- Derived parameters using custom user functions

6.2 Example

The following XML snippet represents a Standard Parameter Message Definition that describes the overarching message and points to separate Standard Parameter Definitions that describe the parametric content.

```
<SPMD>
  <Message_Description>
    <Name> Common TSPI Message </Name>
    <SPMD_ID> 1 </SPMD_ID>
    <Packing> ALIGNED </Packing>
    <SPD>
      <Tag> 0 </Tag>
      <Name> Defaults </Name>
      <Byte_Order> BIG_ENDIAN_ORDER </Byte_Order>
    </SPD>
  </Message_Description>
  <SPDs>
    <SPD><Tag> 2000 </Tag></SPD>
    <SPD><Tag> 2001 </Tag></SPD>
    <SPD><Tag> 2002 </Tag></SPD>
    <SPD><Tag> 6675 </Tag> <Bit_Offset> 16 </Bit_Offset> </SPD>
    <SPD><Tag> 6663 </Tag> <Bit_Offset> 15 </Bit_Offset> </SPD>
    <SPD><Tag> 6664 </Tag> <Bit_Offset> 14 </Bit_Offset> </SPD>
    <SPD><Tag> 6665 </Tag> <Bit_Offset> 13 </Bit_Offset> </SPD>
    <!-- ... -->
  </SPDs>
</SPMD>
```

The following XML snippet represents a sample parameter definition.

```
<SPD>
  <Tag>6067</Tag>
  <Name>Weapons Bay Left Inboard Flipper</Name>
  <Bit_Length>64</Bit_Length>
```

```
  <Data_Type>REAL</Data_Type>
  <Byte_Order>BIG_ENDIAN_ORDER</Byte_Order>
  <Bit_Order>NORMAL</Bit_Order>
  <Unit_Category>UNITLESS</Unit_Category>
  <Storage_Units>NONE_DEFINED</Storage_Units>
  <Version>1</Version>
  <Classification>UNRESTRICTED</Classification>
  <Description></Description>
  <Category>All</Category>
  <C0>1</C0>
  <C1>0</C1>
  <Display_Precision>2</Display_Precision>
  <Default_Units>NONE_DEFINED</Default_Units>
  <Data_Min>0</Data_Min>
  <Data_Max>0</Data_Max>
  <Data_Color>000000</Data_Color>
  <Caution_Min>0</Caution_Min>
  <Caution_Max>0</Caution_Max>
  <Caution_Color>ff0000</Caution_Color>
  <Warning_Min>0</Warning_Min>
  <Warning_Max>0</Warning_Max>
  <Warning_Color>ffff00</Warning_Color>
</SPD>
```

APPENDIX A

Citations

Department of Defense. *Sixteen-Bit Computer Instruction Set Architecture*. MIL-STD-1750A. 2 July 1980. Inactivated 31 July 1996. Retrieved 5 May 2022. Available at https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=37110.

Range Commanders Council. "Floating Point Formats." Appendix 9-D in *Telemetry Standards*. RCC 106-22. May 2022. May be superseded by update. Retrieved 9 May 2022. Available at https://www.trmc.osd.mil/wiki/display/publicRCC/106+Telemetry+Standards.

———. "Metadata Configuration." Chapter 23 in *Telemetry Standards*. RCC 106-22. May 2020. May be superseded by update. Retrieved 9 May 2022. Available at https://www.trmc.osd.mil/wiki/display/publicRCC/106+Telemetry+Standards.

———. "Telemetry Attributes Transfer Standard." Chapter 9 in *Telemetry Standards*. RCC 106-22. May 2022. May be superseded by update. Retrieved 9 May 2022. Available at https://www.trmc.osd.mil/wiki/display/publicRCC/106+Telemetry+Standards.

*** * * * END OF DOCUMENT * * ***