XML: A GLOBAL STANDARD FOR THE FLIGHT TEST COMMUNITY

Alan Cooke
ACRA CONTROL

Diarmuid Corry
ACRA CONTROL

ABSTRACT

Much effort has been spent on developing physical layer standards to ease multi-vendor inter-operability. However as anyone familiar with real-life system integration knows a large gap exists in defining system configuration and set-up, not just between vendors but also between different groups on the base.

Different solutions to this problem have been attempted (for example TMATS). However, the emergence of XML (eXtensible Markup Language) as a commercial standard presents a new opportunity to define a powerful and extensible tool for data-interchange between different systems.

This paper introduces the self-documenting standard for information exchange that is XML. A generic model for flight test data acquisition is presented. Finally, an XML vocabulary (or schema) based on this model is proposed. This schema could form the basis for an industry wide XML standard to simplify the problem of data interchange between vendors, between programs, even between different databases in the same organisation.

KEY WORDS

XML, XML Schema, XIDML

INTRODUCTION

At any given flight test facility a broad range of activities contribute to the overall goal of testing aircraft and analysing the resulting data. These activities range in character from instrumenting aircraft to ground-station data analysis and typically involve personnel from a wide variety of organisations and groups.

The task of achieving the overall goal is further complicated by the fact that the equipment and software used on a project are usually derived from a multitude of different vendors. Vendors in turn have their own unique requirements that typically employ proprietary software and data formats. The process of integrating many heterogeneous systems can become a major impediment to the successful completion of a project.

What is needed is a recognised standard that caters for the requirements of a typical flight test project. A standard that captures and models the different elements of a project and that can be used when data needs to be exchanged between the different groups participating in a project.

There have been several attempts to define such a generic standard, most notably the Telemetry Attributes Transfer Standard (TMATS). However the rapid acceptance of XML as a generic data interchange standard presents the opportunity to develop a modern, flexible and extensible standard for the future. This paper proposes just such a vocabulary for the flight test community.

It is not the intention of this paper to describe what XML is, or how it is applied in a generic solution. This ground has been covered before[1]. Instead, this paper proposes a vocabulary (or in XML terms, a "schema") that allows XML to be used to exchange data between the disparate groups and functions during flight test.

WHY XML?

If an XML schema is designed properly then XML offers the following advantages over TMATS and other mechanisms.

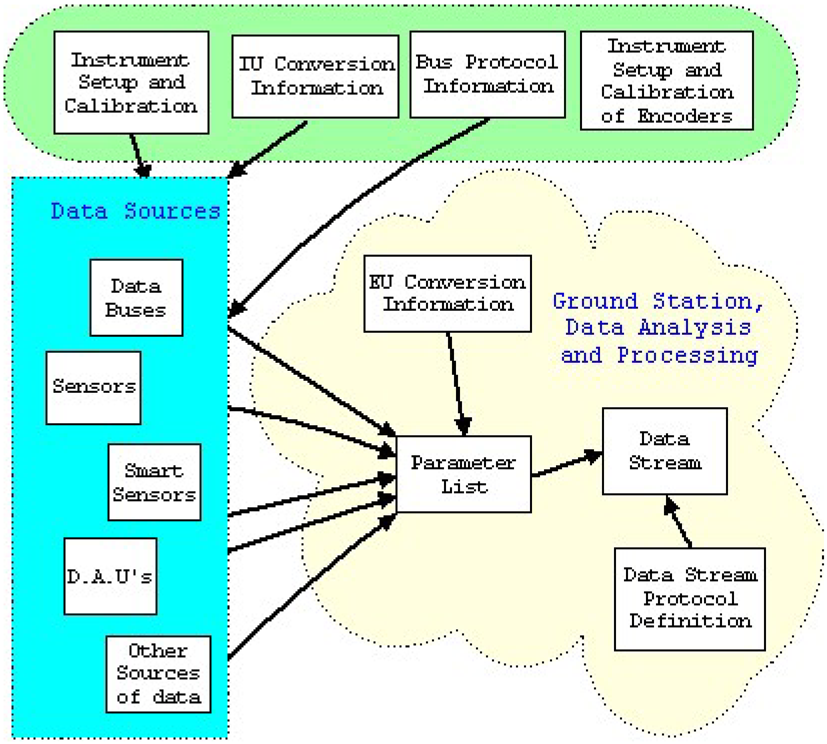
- XML is the de facto commercial standard for data exchange.
- XML is an open, non-proprietary standard. There is no need to pay any royalties, license fees or for any other services in order to use it.
- A large number of tools and utilities exist for the processing and creation of XML documents. Many of these are available for free.
- A large body of XML knowledge already exists. Familiarity with FTI and related areas is not a prerequisite for using XML.
- XML allows a standard for the FTI industry to be defined that is both rigorous and flexible enough to cater for any future requirements.

---

[1] In particular, see the paper RDBMS AND XML FOR TELEMETRY ATTRIBUTES from ITC 2003.

A GENERIC DATA ACQUITION SYSTEM

The following figure outlines, in broad terms, the various elements, inputs and outputs of a generic data acquisition system.



Elements of a Data Acquisition System

Figure 1

In general, a typical data acquisition system consists of the following.

1. Physical hardware used to retrieve raw data. Examples include sensors, smart sensors, data buses and various types of data acquisition units.
2. Information used to interpret data from these systems. Examples include information on the protocol used by a data bus, calibration information for the data acquisition modules or sensors and the information used to transform data from instrumentation units to engineering units.
3. A data acquisition system encodes the acquired data into one or more data streams in some particular format for transmission or storage. An example would be the encoding of data into an IRIG-106 PCM data stream.

Depending on the circumstances, only a subset of this information may be of interest at any particular time. For example, a ground station vendor may only be interested in the data stream in which the acquired data is transmitted, the transmission protocol and how the transmitted data should be interpreted.

SCHEMA DESIGN GOALS

Given the generic data acquisition system outlined above. An XML schema should meet the following criteria

- The schema should be vendor neutral.
- The schema should capture as many of the recognised elements of a generic data acquisition system as possible in order to be useful to as many potential consumers or producers of flight-test data as possible
- The layout of the schema should be logical and easy to understand and therefore lend itself easily to the extraction of information.
- The schema should be extensible and future proofed.

ACRA CONTROL has created a schema, called XIDML, that we believe meets these goals.

FEATURES OF XIDML

Generic and Multi-purpose

XIDML has been designed to be generic and vendor independent and has been constructed to be as useful as possible to as many groups and organisations that work in the flight test area. Furthermore, wherever possible, use of existing established standards, MathML for example, have been included as part of the schema.

Data Reuse

One of the main features of the XIDML schema design that should be noted is the emphasis on the reuse of data and the decrease in data redundancy. For example, an IRIG PCM frame description can be reused more than once. The same can also be said of other protocols such as the MIL-STD 1553 and ARINC-429 bus protocols. Where it makes sense, this ability to reuse data has been built into other areas of the schema.

Atomic Structure

The XIDML schema has been designed to group data logically and in ways that make sense to those who work in the flight test area. By grouping, for example, all parameter definitions together and all protocols definitions together and so on, users of XIDML based data need only look at the information *they* are specifically interested in. This is a feature that has been leveraged directly from the XML standard.

Extensibility

Although it is not possible in this paper to rigorously describe the extensibility mechanisms that have been built into XIDML, the schema has been designed with the future in mind. XIDML fully utilises the features of the XML Schema[2] standard that facilitate the extension of XML schemas This means for example, new types of protocol definitions can been included trivially, TEDS, FireWire and other standards

[2] These features include Substitution Groups, Type Substitution and Type Derivation.

can also be added readily. Importantly, where the need arises, vendor specific information can also be included in the schema at will.

XIDML SCHEMA OVERVIEW

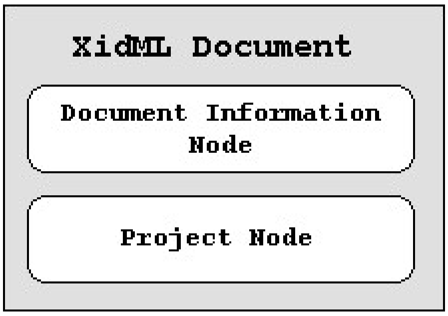


Figure 2

At the highest level of the document are the *Project* and *Document Information* nodes. Figure 2 shows the high level structure of a XIDML document. The *Document Information* section contains general information about the XML document such as when the document was created and a description of the project. An optional reference to more detailed information outside the document can also be included

PROJECTS

The *Project* section contains all information about a flight test project such as its name, a description of the project as well as optional references to more detailed information about it in another document. Most importantly however, a *Project* also contains the definitions for all **data acquisition systems** that are part of a flight test project. There can be more than one *Acquisition System* node defined in a project depending upon the nature of the flight test project being modelled.

ACQUISITION SYSTEMS

An *Acquisition System* is the most fundamental and important construct in the XIDML schema. In the context of XIDML, an Acquisition System is defined as a system that is used to retrieve and/or process telemetric data. Table 1 lists some of the main components of an Acquisition System as understood in XIDML. All of the sections in Table 1 are *optional* and only need to be included if they make sense in the context of the Acquisition System being modelled.

Figure 3 gives a high level overview of some of the main components of an Acquisition System as understood in XIDML. All of the sections shown in Figure x are *optional* and only need to be included if they make sense in the context of the Acquisition System being modelled.

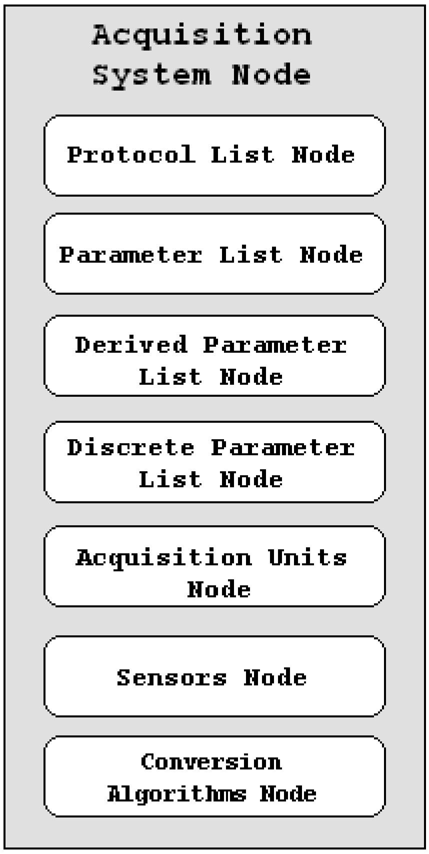


Figure 3

Table 1: Main sections in the Acquisition System node

Node	Description	Observations
Protocols List	Contains descriptions of the PCM frame, MIL-STD 1553, ARINC-429, Ethernet packets and other protocols used by an acquisition system.	A protocol definition may be reused many times and across multiple projects. Any type of protocol definition can be added to the Protocol List section[3].
Parameters List	Lists all the parameters or signals sourced in, or used by, an acquisition system. Each one is identified by a unique name.	Each definition can include information such as, units used, data format and mechanisms for the conversion from raw data to IU or to EU
Derived Parameters List	Defines how a parameter is derived from one or more other parameters.	The MathML XML vocabulary is used to describe any mathematical relationships.
Discrete Parameters List	Defines how a parameter is composed of one or more discrete subset(s) of other parameter(s)	
Acquisition Units List	Lists the uniquely named DAUs to be used to describe the set-up, configuration and so on.	A DAU can be from any vendor.
Sensors List	Lists the EU range (gain/offset), defines the conversion to IU, filter cut-off and excitation	A Sensor can be from any vendor.
Conversion Algorithms List	Describes how raw data is converted to IU and EU. Each algorithm is identified by a unique name.	Each conversion algorithm can be reused in more than parameter definition or in Sensor and DAU descriptions.
Wiring	Describes the connections between data sources and sinks	

VENDOR INFORMATION

XIDML has been designed with flexibility and extensibility in mind. To this end, one or more *Vendor* sections can be added to (or referenced from) the XIDML file if vendor specific information needs to be included. This allows, for example, information required by ACME Inc. to be in the XIDML document alongside that of ACRA Control and other third party vendors without the structure and integrity of the rest of the XIDML file being affected.

SOME EXAMPLES

The following examples illustrate how a simple IRIG PCM frame containing 3 placed parameters can be defined. Note that a complete and detailed schema is available which provides a more comprehensive description of XIDML. The examples further show how to specify and define IU to EU conversion for a parameter as well as showing how a derived parameter is specified.

[3] As long as it meets some minimal requirements (see the PCM Frame Definition section below).

PCM Frame Definition

The example below shows the definition of a simple PCM frame with one minor frame and three placed parameters. This frame definition can be included inline within the main XIDML file or be located in an external file and just referenced in the XIDML file[4]. A PCM frame definition can be used more than once which increases reuse and reduces data redundancy.

```
<ProtocolList>
   <IRIG106FrameDefinition Name="MySampleFrame" LayoutRevision="1.0">
      <PCMFrameStructure>
            <MajorframeProperties>
                 <NumberOfMinorFrames>1</NumberOfMinorFrames>
                 <NumberOfMinorFrameWords>4</NumberOfMinorFrameWords>
                      <MinorFrameStartIndex>0</MinorFrameStartIndex>
                      <SyncWordReference>0</SyncWordReference>
                      <FillPattern>AAAA</FillPattern>
                      <Justification>LEFT</Justification>
                      <Parity>NONE</Parity>
                      <BitsPerWord>16</BitsPerWord>
            </MajorframeProperties>
            <SynchronisationStrategy>
                  <SyncPattern>1110101110010000</SyncPattern>
                  <SyncMask> 1111111111111111 </SyncMask>
            </SynchronisationStrategy>
      </PCMFrameStructure>
            <FramePlacement>
                  <PlacedParameter>
                        <Name>LeftWingTemperature</Name>
                        <WordIndex>1</WordIndex>
                        <Frame>0</Frame>
                        <Samples>1</Samples>
                        <BitMask>FFFF</BitMask>
                        <MSB>First</MSB>
                  </PlacedParameter>
                  <PlacedParameter>
                        <Name>RightWingTemperature</Name>
                        <WordIndex>2</WordIndex>
                        <Frame>0</Frame>
                        <Samples>1</Samples>
                        <BitMask>FFFF</BitMask>
                        <MSB>First</MSB>
                  </PlacedParameter>
                  <PlacedParameter>
                        <Name>AverageWingTemperature</Name>
                        <WordIndex>3</WordIndex>
                        <Frame>0</Frame>
                        <Samples>1</Samples>
                        <BitMask>FFFF</BitMask>
                        <MSB>First</MSB>
                  </PlacedParameter>
            </FramePlacement>
      </IRIG106FrameDefinition>
                  .
                  .
<ProtocolList>
```

[4] Data located in external files is referenced using the XLink and XPointer standards.

It should be noted that all protocols belonging to the *ProtocolList* above must have the same basic structure. In XML schema parlance, they must be the same type and belong to same substitution group. In this case, a protocol must have a (unique) name[5] and a version number. Any protocol conforming to these rules can be included legally as a protocol in a XIDML file.

Parameter Definition

This example shows how a parameter can be defined. Each parameter definition references an EU conversion algorithm named *StandardLinearConversion*. This algorithm is used to convert IU to EU. An example of how to specify a conversion algorithm is show later.

```
<ParameterList>
                  .
                  .
   <Parameter name="LeftWingTemperature">
                  .
      <Units>Centigrade</Units>
      <EUConversionRef>StandardLinearConversion</EUConversionRef>
                  .
   </Parameter>
   <Parameter name="RightWingTemperature">
                  .
      <Units>Centigrade</Units>
      <EUConversionRef>StandardLinearConversion</EUConversionRef>
                  .
   </Parameter>
                  .
                  .
</ParameterList>
```

Derived Parameter Definition

This example shows how a derived parameter can be defined and illustrates how to get the average of two values. In this case, the derived parameter takes in two parameter values, *LeftWingTemperature* and *RightWingTemperature*, adds the two values together, and then divides the value by two and assigns the resultant value to the derived parameter called *AverageWingTemperature*. All mathematical operations are defined using MathML. MathML is the internationally recognised standard for the definition and display of mathematical expressions.

```
<DerivedParameter>
      <Name>AverageWingTemperature</Name>
      <Composition>
            <DerivedParameterMathAlgoritm>
                  <InputParameters>
                        <ParameterRef>
                              <Name>LeftWingTemperature</Name>
                        </ParameterRef>
                        <ParameterRef>
                              <Name>RightWingTemperature</Name>
                        </ParameterRef>
                  </InputParameters>
```

[5] The uniqueness constraint is again specified using the XML Schema standard.

```
                  <MathAlgorithm>
                        <mml:apply>
                            <mml:divide/>
                            <mml:apply>
                                <mml:plus/>
                                <mml:ci>LeftWingTemperature</mml:ci>
                                <mml:ci>RightWingTemperature</mml:ci>
                            </mml:apply>
                            <mml:cn>2</mml:cn>
                        </mml:apply>
                  </MathAlgorithm>
            </DerivedParameterMathAlgoritm>
      </Composition>
</DerivedParameter>
```

Conversion Algorithm Definition

The following example shows how the conversion algorithms referenced in the parameter definitions earlier is defined[6].

```
<ConversionAlgorithms>
      <ConversionAlgorithm name="StandardLinearConversion">
            <tc:PolynomialCalibrator>
                  <tc:Term coefficient="0" exponent="0"/>
                  <tc:Term coefficient="1" exponent="1"/>
            </tc:PolynomialCalibrator>
      </ConversionAlgorithm>
</ConversionAlgorithms>
```

Vendor Information

The following example shows how vendor specific information can be included in a XIDML document. It should also be noted that this information could alternatively be included in an external file. This example shows how information for a DAU provided by ACRA Control would be included.

```
<VendorSection>
            .
      <Vendor name="ACRA Control">
                  .
            <KAD_ADC_001 Name="MyADC001" SerialNumber="01234">
                  <InstrumentSetup>
                     <ANALOG SourceBits="16">
                        <SourcedParameterName>LeftWingTemperature
                        </SourcedParameterName>
                        <Gain>1</Gain>
                        <Offset>-10</Offset>
                     </ANALOG>
                     <ANALOG SourceBits="16">
                        <SourcedParameterName>RightWingTemperature
                        </SourcedParameterName>
                        <Gain>1</Gain>
                        <Offset>-10</Offset>
                     </ANALOG>
```

[6] It should be noted here that the algorithm definition uses the standard submitted to the OMG.Space Domain Task Force Telemetric and Command Data Specification Space RFP-1. Reference space/2003-03-01. Issue 1.2, 3 March 2003.

```
                  </InstrumentSetup>
            </KAD_ADC_001>
                  .
      </Vendor>
            .
      <Vendor name="ACME Inc">
            .
            .
      </Vendor>
            .
</VendorSection>
```

CONCLUSION

In this paper some of the problems involved in the successful completion of a flight test project have been outlined and a model for a generic data acquisition system was described. An XML vocabulary, or schema, called XIDML was then presented that is based on the generic model of a data acquisition system. Finally, several reasons were highlighted why it is believed XIDML may form the basis of a standard that could be adopted by the flight test community. Some of the reasons outlined are

- XIDML is designed to be generic and multi-purpose.
- The schema promotes data reuse
- The atomic nature of the schema
- The schema was designed to be flexible, scalable and future proof.

REFERENCES

[1] The XML home page can be found at http://www.w3c.org

[2] The XML Schema homepage can be found at http://www.w3.org/XML/Schema

[3] The home page for MathML can be found here http://www.w3c.org/Math

[4] An introduction to MathML can be found at http://www.dessci.com/en/support/tutorials/mathml/default.htm

[5] Tutorials on various XML standards can be found at http://www.w3schools.com/

GLOSSARY OF TERMS

DAU	Data Acquisition Unit.
DTD	Document Type Definition. A mechanism used to validate XML documents. The predecessor to the XML Schema standard.
EU	Engineering Units
FTI	Flight Test Instrumentation
IU	Instrumentation Units
MathML	Mathematical Markup Language. An XML vocabulary used to describe the definition and display of mathematical expressions.
Substitution Group	An XML Schema construct that represents a class or type of data structure.
TMATS	Telemetry Attributes Transfer Standard
XIDML	Extensible Instrumentation and Definition Markup Language
XML	Extensible Markup Language
XLink	An XML standard used to reference other XML files
XPointer	An XML standard used to reference data contained in specific locations in other XML files
XML Schema	An XML vocabulary used to validate an XML document. It is designed to be the successor to the DTD standard