The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

XML TELEMETRIC AND COMMAND EXCHANGE (XTCE)

INFORMATIONAL REPORT

CCSDS 660.2-G-2

GREEN BOOK

February 2021

The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

XML TELEMETRIC AND COMMAND EXCHANGE (XTCE)

INFORMATIONAL REPORT

CCSDS 660.2-G-2

GREEN BOOK

February 2021

AUTHORITY

Issue:	Informational Report, Issue 2
Date:	February 2021
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4).

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Report is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the email address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Science Policy Office (BELSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Netherlands Space Office (NSO)/The Netherlands.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 660.0-G-1	XML Telemetric and Command Exchange (XTCE), Informational Report, Issue 1	July 2006	Original issue, superseded
CCSDS 660.2-G-2	XML Telemetric and Command Exchange (XTCE), Informational Report, Issue 2	February 2021	Current issue

CONTENTS

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

This CCSDS Green Book introduces the main concepts of XML Telemetric and Command Exchange (XTCE), a telemetry and telecommand database format for spacecraft monitoring and control. It has been prepared by the Spacecraft Monitor and Control (SM&C) working group of the CCSDS Mission Operations and Information Management Systems (MOIMS) area.

The purpose of this document is to describe the basic need for XTCE, describe its various use cases, and provide a high-level description of its major elements.

The scope of this document is limited to the typical industry use of the satellite telemetry and commanding database format during the development and operation phases of a mission; it addresses general XTCE concepts and terminology. Finally, it provides a high-level overview of XTCE itself.

1.2 DOCUMENT STRUCTURE

Section 1 specifies the purpose, scope, and content of this document and specifies references to other documents that contain relevant material.

Section 2 provides a general description of XTCE. This includes the following XTCE topics: overview, philosophy, concepts, and services.

Section 3 specifies the high-level schema organization and structure.

Section 4 contains an example using major elements.

Annex A shows a complete XTCE example.

Annex B is the glossary that contains definitions of terms and acronyms.

1.3 REFERENCES

The following publications are referenced in this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

Three CCSDS Books address the XTCE standards and its use:

[1] *XML Telemetric and Command Exchange—Version 1.2*. Issue 1.2. Draft Recommendation for Space Data System Standards (Draft Blue Book), CCSDS 660.0-B-1.2. Washington, D.C.: CCSDS, December 2019.

[2] *XML Telemetric and Command Exchange (XTCE)—Element Description*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 660.1-G-1. Washington, D.C.: CCSDS, May 2012.

[3] *XML Telemetric and Command Exchange (XTCE)*. Issue 1.1. Draft Report Concerning Space Data System Standards (Draft Green Book), CCSDS 660.2-G-1.1. Washington, D.C.: CCSDS, October 2019 (this document).

The Object Management Group (OMG) coordinates updates to the XTCE specification. Related OMG websites include:

[4] "OMG XTCE Schema." Object Management Group. https://www.omg.org/spec/XTCE/20180204/SpaceSystem.xsd.

[5] "OMG Space Domain Task Force (Space DTF)." Object Management Group. https://www.omg.org/space/.

[6] "Space Category—Specifications Associated." Object Management Group. https://www.omg.org/spec/category/space/.

[7] "OMG Bug/Issue Reporting Form." Object Management Group. https://issues.omg.org/issues/create-new-issue.

[8] "XTCE US Govt Satellite Conformance Profile." Object Management Group. https://www.omg.org/spec/XUSP/.

The packet utilization standard is also referenced in this document:

[9] *Telemetry and Telecommand Packet Utilization*. ECSS-E-ST-70-41C. Noordwijk, The Netherlands: ECSS Secretariat, 15 April 2016.

In addition, there are many compliant vendor products and open source tools available. It is best to search the internet for the latest listings.

2 CONTEXT AND OVERVIEW

2.1 INTRODUCTION

This document presents a non-proprietary format called XTCE for defining a telemetry and telecommand (command) mission operations database. This format may be used by any space element that needs to describe its telecommanding or/and telemetry, or that wishes to provide that description to another space element. The XTCE standard is the result of years of collaborative work by the OMG Space Domain Task Force, the CCSDS MOIMS Spacecraft Monitor and Control (SM&C) working group, vendors, and many national space institutions and affiliates.

NOTE – The term 'telecommand' is often shortened to 'command' or 'commanding' in common practice and may be substituted in the text below.

2.2 VERSIONS

2.2.1 GENERAL

There are several versions of XTCE that have been published. The first version was XTCE 1.0 published in 2006. Version XTCE 1.1 was published in 2008 after adjustments following a CCSDS review. XTCE 1.2 was published in 2019 after years of accumulated knowledge by end users were reviewed and adjudicated. An associated XTCE schema is also managed by the OMG and is posted on their website (see reference [4]).

This documents generally pertains to all the versions, but specifically version 1.2.

2.2.2 CHANGES FROM XTCE 1.1 TO XTCE 1.2

XTCE 1.2 contains many important changes but is largely backwards compatible syntactically with XTCE 1.1. Even so, some syntax has changed slightly.

The largest syntax change has to with the way arrays may be defined, which in XTCE 1.2, also allows for the indices to defined in the array definition itself.

Another large change, invisible to the end user but apparent to the programmer, is that many of the underlying schema types have been cleaned up to produce a better programmatic mapping to class definitions.

A list of major changes in XTCE 1.2 are as follows:

- The SpaceSystem namespace is updated;
- Various enumerated strings values are normalized;
- Byte order is changed;

- Exponents are integers;
- Specifying string length is updated;
- Float encoding sizes are modified;
- Epoch time/date data type updated;
- Argument area supports both an argumentRef and parameterRef;
- Arrays are updated so that their dimension sizes can be specified in the definition;
- Many annotations have been updated;
- Numerous schema types are improved or modified.

2.3 WHY USE XTCE

XTCE is non-proprietary mission operations database format in an industry filled with proprietary solutions. It can be used to exchange telemetry and telecommand databases during the development and operation phases of a mission, or as a native format, for those who want to support it that way. It can be used to exchange a database between spacecraft manufacturers, instrument manufacturers, and different systems of the ground segment, as seen in figure 2-1. It can also be used as an exchange mechanism between different development teams or between missions, which enhances database reuse.

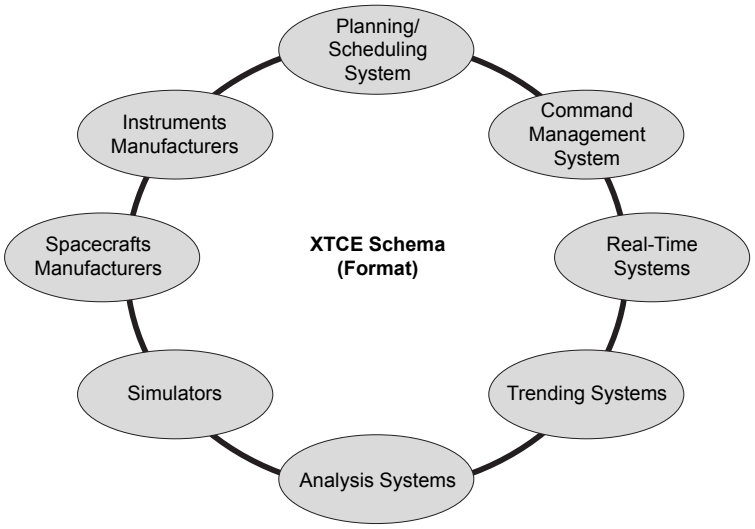

Figure 2-1: XTCE Exchange Concept

XTCE also takes a general approach and is not tied specifically to any existing telemetry format such as CCSDS Space Packets or Major/Minor frames, two common telemetry and command approaches used in the industry.

The use of XTCE can reduce mission costs for several reasons. First, there is no dependency on proprietary tools to convert and edit the database. Legacy telemetry and telecommand databases are readily converted to the XTCE format, and those converters and XML technologies are widely supported within the technology industry. In addition, the use of a standard exchange format from the early stages of spacecraft development through mission operation will reduce life-cycle cost. Finally, XTCE will enable the user to define one unique, non-proprietary format to describe a telemetry and telecommand database; this enables a mix of tools from various vendors or internal groups.

2.4 XTCE SCOPE

The scope of the XTCE is largely from the point of view of the ground system users and consists of major command and telemetry areas and common areas for algorithms, streams, and services organized into a namespace device called a SpaceSystem. Using XTCE, multiple SpaceSystems can be defined into a hierarchical tree, and describing operations database information is convenient and flexible. Several SpaceSystems can be defined in several files and knitted together using XInclude.

As XTCE has garnered much use over the years, there is a growing list of information the user may find with relative ease on the internet. Much of this information can now be found on the Internet.

2.5 STYLISTIC

XTCE is an XML Schema containing many elements (tags) and attributes. When these names are used in the text of this document (including section or figure titles), they are always written out as they appear in the schema. For example, Parameter is a major element in XTCE. Likewise, the attributes are always written as they are in the schema, which is typically in a camel-case style starting with a lowercase letter, such as dataSource. Likewise, if the value of one of these items is an enumerated set of strings, then the value is given in the text as it appears in XTCE, such as: IEEE754_1985.

Many of XTCE's elements appear in sets or lists with a similar name to the major element; for example, Parameter is in ParameterSet. In the text, sometimes the plural form of the major element is used as a convenient stand-in for the set, such as Parameters.

In a few cases, the formal name of the element also has an informal but widely used variant; these are used interchangeably here. The principal example of this is Container for SequenceContainer, the latter of which is the actual syntax.

Finally, XTCE itself may be referred to as the XTCE Schema, or XML Schema.

2.6 CORE XTCE CONCEPTS

2.6.1 SPACESYSTEMS

The SpaceSystem element in XTCE has no formal meaning, so it can be any part of a spacecraft (e.g., an instrument), the spacecraft itself, a constellation of spacecraft, or a ground station. This concept allows for the division of the information into smaller distinct components.

An example use of this concept, for satellite development, would be to create a separate telemetry and telecommand database in XTCE for each instrument on board the spacecraft, and another one for the spacecraft bus itself (for navigation, attitude control, etc.).

XTCE allows assembling these separate databases inputs into one large database (for instance, the satellite database) using XInclude. In this way, each dedicated database can describe its own information for telemetry or/and telecommanding. *XML Telemetric and Command Exchange (XTCE)—Element Description* (reference [2]) provides examples, and the *XML Telemetric and Command Exchange—Version 1.2* (reference [1]) details the XTCE Schema elements and attributes.

2.6.2 TELEMETRY

XTCE allows one to describe features of packets and packet structure, telemetered values to engineering ground values, calibrators, alarms, and telemetry-related algorithms for the purposed of ground processing.

Telemetry packets are described using the XTCE Container element, which allows users to describe how their telemetry is packaged. XTCE uses Containers to describe a sequence of bits. In CCSDS, this packaging takes the form of a packet or even a part of a packet. For TDM, this packaging is likely a minor frame. In order to complete the packet description, the XTCE Container provides an inheritance mechanism between Containers. For example, a CCSDS Packet Container could be the parent of all specific mission packets that extend it in a particular XTCE file for that mission.

XTCE Containers list their mnemonic entry fields or Parameters, which are grouped together to build Containers. Entries (EntrySet) are addressable and may have other information associated them (such as a conditional for inclusion, and so on).

Once Containers are described in XTCE, the various Parameter (mnemonic) properties should be completed as well.

This would include information such as the name, packet encoding, and all information needed by ground processing system to the fully transform those bits into ground engineering data type values.

Alarms (limits) and calibration information are also included in this area.

2.6.3 TELECOMMANDING

XTCE supports commanding in two ways: it has a high-level command description area (MetaCommand, aka 'command') that includes Arguments, and a 'packaging' area, similar to the TelemetryMetaData Container above but local to it, called CommandContainer.

The additional command definition elements are:

- TransmissionConstraint – constrains the current command;
- VerifierSet – verifiers, that is, verifies the current command;
- Interlock – constrains the next command;
- DefaultSignificance (also ContextSignificance) – level of access to the command;
- ParameterToSetList – Parameters to set after the command is sent;
- ParametersToSuspendAlarmsOnSet – suspend alarms on this Parameters after the command is set.

In XTCE, command Arguments are local to a command and cannot be reused between commands.

However, commands may be inherited. Namely, a command may be derived from another command using the BaseMetaCommand element.

There is also an element used to describe a sequence of commands called BlockMetaCommand.

2.6.4 SERVICES

Services represent an advanced concept of selection in XTCE. This concept allows the grouping of Containers under a given label. For example, all that are related to the onboard 'memory dump' function could be grouped under a service in the Service section with the same name.

This is also an underutilized area of XTCE.

2.7 WHEN XTCE SEEMS TOO COMPLEX

For many users, XTCE will have far more features than are needed. Sometimes this is a stumbling block to adoption. To address this, the standards community put out a 'trimmed' version of XTCE colloquially called GovSat (published as OMG's XTCE US Govt Satellite Conformance Profile, XUSP™). While GovSat is not a new XML Schema, it is a set of rules that restricts various parts of XTCE. Those rules are executable and could be used in a validation program after an XTCE file has been XML validated.

One restriction of the rules is reducing the number of SpaceSystems to one. Others restrict the number of coefficients in a polynomial, provide a character-length limit on mnemonics, and so on.

Another important area is the definition of a standard CCSDS header and related Parameters.

Organizations can also modify the rules to meet their needs, cognizant of the fact this will make compatibility harder for other XUSP adopters unless they agree as a group to relax part of it.

For example, many organizations now wish to support more than one SpaceSystem.

2.8 WHEN XTCE DOESN'T HAVE ALL THE FEATURES NEEDED

In a somewhat contradictory way, many users will find XTCE to be both too large and too small for their needs. Typically, a missing 'must have' feature is not present in XTCE but is a local practice that is not common across the industry. In some cases, this missing feature is considered critical and must be supported.

There are several ways to handle this problem:

a) Use the AncillaryData Element.

   The AncillaryData element is sprinkled about the schema. It gives the user the ability to define a name/value pair. In many cases, there's an AncillaryData right near where one needs the extension, and the extension itself is simple enough to be encoded as name/value pairs. This is certainly by far one of the easiest ways to solve this kind of problem, and the files remain compatible with the XTCE schema.

b) Modify XTCE locally.

   Another approach is to simply modify the schema directly. Once again, one has to weigh the advantages and disadvantages of this approach. It may make sense if the AncillaryData element isn't nearby in the syntax tree or the item is complex enough in nature that the AncillaryData element doesn't appear to be a good fit, but one is 'breaking the standard' by modifying it in this way, and there is a cost to breaking it.

c) Modify XTCE but with an external schema that is included in XTCE.

   This is a variation on the approach above but has some further benefits. In this variation, the modification is held in a separate schema. Typically, that schema has its own namespace. The original XTCE is modified with care so that the element or attribute in question is included and new syntax appears at the modification point (see XInclude and 'any'). The end result in XTCE files is that the bulk of the file will be in the XTCE namespace, while extensions will be in their own separate namespace. This allows for the easy separation of the two areas, which can help in a variety of ways in managing the issue.

d) Use an external schema and external file to hold the needed information.

   This is somewhat similar to the above two items, but software needs to know how to integrate the information with XTCE. The original XTCE file remains compatible with the schema, but there are now two separate files to manage.

e) Overload or misuse elements or attributes in XTCE for some other purpose than they were originally intended.

   Depending on the elements, this may break existing software. Likewise, your software needs to know of and readily recognize these special cases. This is possibly the least preferred option.

f) Submit an XTCE change request to the OMG for consideration.

   If the capability has broad applicability and could be of value to other organizations and product vendors, then an update to the XTCE specification may be considered. However, this can be a very lengthy process, and the above options may be preferred in the shorter term.

3 XTCE SCHEMA ORGANIZATION AND STRUCTURE

3.1 OVERVIEW

The following sections give a brief overview of the major XTCE elements. For a more detailed explanation of all the XTCE elements, please refer to the XTCE Element Description Green Book (reference [2]).

3.2 SPACESYSTEM

3.2.1 GENERAL

The XTCE Schema root element is SpaceSystem. A hierarchy of SpaceSystems may be formed. Another way to think of this is that the SpaceSystems are namespaces. For example, the 'path' syntax is adopted for XTCE SpaceSystems so that a file may contain something like the following, each with the appropriate definitions for that 'level':

/MyMission

/MyMission/SubSystem1

/MyMission/SubSystem1/Power

/MyMission/SubSystem1/Power/Battery1

And so forth.

Multiple SpaceSystems in a single XTCE file may be advantageous to some organizations. Others will be content with using a single SpaceSystem.

These SpaceSystems describe the related telemetry and telecommand metadata as well as containing a dedicated header and services (figure 3-1), which will be covered later. In the following figures, dark colors mean parent elements, whereas lighter colors mean child elements.

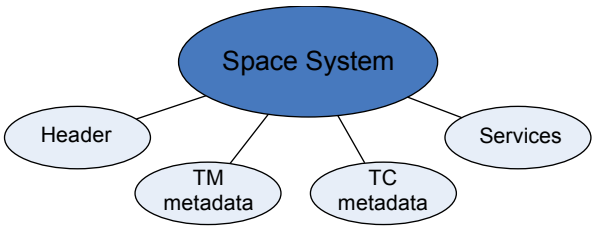

Figure 3-1: A Space System

All implementations of XTCE start out with a root SpaceSystem element and any number of child SpaceSystems can be added as is needed (see figure 3-2).

In XTCE 1.2, the support for XInclude means SpaceSystems could be combined into one file from many to build the entire description for that mission. This system enables the user to describe spacecraft telemetry and commanding step by step and regroup telemetry and telecommand data under logical systems.

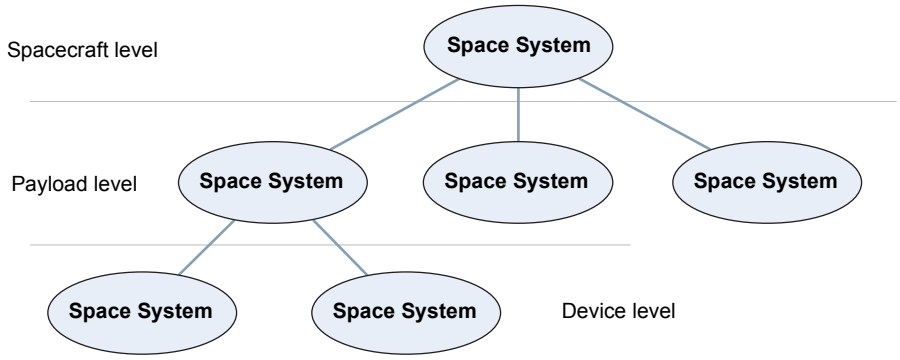

Figure 3-2: Hierarchy of Space Systems

After the root SpaceSystem, a Header element is typically provided, and then the contents of the major elements TelemetryMetaData, CommandMetaData, and ServiceSet follow.

Each of these will be discussed briefly below, although it should be noted that most of XTCE's elements are optional.

3.2.2 HEADER

The Header element describes information related to the XTCE file itself. It contains information about:

- date, version, classification, and validation status (XML is validated, tested, or drafted);
- a list of authors with names and surnames of contributors;
- a list of notes; any textual notes can be written in the header;
- a list of history records that can be used for configuration-management purposes.

3.2.3 TELEMETRYMETADATA

3.2.3.1 General

Telemetry-related descriptions are in the TelemetryMetaData element that has the major child elements as follows:

- ParameterTypeSet – holds ParameterTypes associated with Parameters;
- ParameterSet – hold the set of Parameter descriptions;

– ContainerSet – descriptions for package, such as packets or minor frames;
– MessageSet – holds Messages that are an alternative to Containers;
– StreamSet – bit stream description area;
– AlgorithmSet – Algorithms associated with telemetry process.

These are discussed further below.

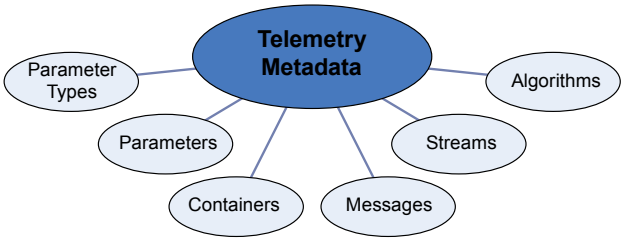

Figure 3-3: XTCE Telemetry

The Parameter element is the description area for mnemonics, the individual value fields in a packet or minor frame being telemetered to the ground.

In XTCE, a Parameter has a kind of data type. This is represented in the ParameterType area, and there are ten of these, discussed below.

The ContainerSet element (in ContainerSet) describes how packet or minor frame structures would be defined. This element refers to or points to Parameter definitions.

The MessageSet is an alternative to the ContainerSet. Whereas the Container concept uses inheritance to build complete Packet descriptions, the Message uses a 'has a' construction technique. This element is another underutilized area of XTCE.

The StreamSet is meant to hold information related to the bit-stream, typically, for example, what occurs from the framesync and any error correcting codes. As discussed above, this is underused part of XTCE.

Finally, AlgorithmSet holds 'algorithm' information; for example, derived or pseudo telemetry conversions may be defined here, or any other algorithm associated with this SpaceSystem's telemetry processing, as desired.

3.2.3.2 ParameterTypeSet

A Parameter has a ParameterType. (In theory, a ParameterType may be referred to by more than one Parameter, but this is not common practice.)

A ParameterType has two forms. The first and most common form represents the parameter value from the spacecraft data type to a ground data type. These contain the specific ParameterType element and a child DataEncoding element.

The second form describes ground side data types. These have a top-level ParameterType element but no DataEncoding element (and the Parameter should set the attribute dataSource to 'local').

The following ParameterTypes are defined in XTCE:

- IntegerParameterType – telemetered to the ground as an integer;
- FloatParameterType – telemetered to the ground as a float or calibrated integer;
- EnumeratedParameterType – an enumerated integer (e.g., ON=1, OFF=2);
- StringParameterType – telemetered to the ground as a string (e.g., a byte array of ASCII);
- BooleanParameterType – a special case of enumerated types above;
- BinaryParameterType – a 'blob' catch-all type;
- RelativeTimeParameterType – durations;
- AbsoluteTimeParameterType – clock times;
- ArrayParameterType – an array of one of the other ParameterTypes;
- AggregateParameterType – a 'structure' or record).

A Parameter will always need to refer to a ParameterType.

Each ParameterType has numerous properties. Some are as follows:

- Telemetered ParameterType:
  - Has a DataEncoding containing specific details of the telemetered value so it can be interpreted properly on the ground,
  - Calibrators are defined in this section if appropriate;
- Ground side data type:
  - No DataEncoding;
- Descriptive information;

- Ancillary and Alias info, if applicable;
- Units;
- ValidRange, applicable to some IntegerParameterType and FloatParameterTypes;
- Alarms (limits), which vary somewhat by ParameterType.

There are four types of DataEncoding elements within all the ParameterTypes except Array and Aggregate:

- IntegerDataEncoding – the telemetered items is an integer;
- FloatDataEncoding – the telemetered item is a float;
- StringDataEncoding – the telemetered items is a string;
- BinaryDataEncoding – a special case.

There are many attributes and child elements in DataEncoding; only a few will be discussed here.

The two 'numeric' DataEncodings, IntegerDataEncoding and FloatDataEncoding, have various kinds of calibrators: spline (linear), polynomial, and a more complicated math expression (MathOperationCalibration).

If a more complicated calibration algorithm is needed, the suggested course of action is to use the BinaryDataEncoding and supply a 'FromBinaryTransformationAlgorithm'.

And because most of the ParameterType have the four DataEncoding, only common combinations are typically used because they match real world scenarios. The following are just three examples:

- A calibrated float calculated from a telemetered integer count would be a FloatParameterType with an IntegerDataEncoding and (typically) a PolynomialCalibrator;
- A plain integer (a telemetered integer to a ground integer value) would be an IntegerParameterType with IntegerDataEncoding;
- A plain float (a telemetered float to ground float) would be FloatParameterType with FloatDataEncoding.

Further details are discussed at length in the XTCE Element Description Green Book (reference [2]).

3.2.3.3 ParameterSet

The ParameterSet is the collection of Parameter definitions for SpaceSystem. It should be noted that from a telemetry point of view, a Parameter is a mnemonic value from the spacecraft to the ground.

Parameters have many attributes and child elements, but must also have a ParameterTypeRef to its ParameterType; a Parameter 'has a' ParameterType.

It is common practice, although not strictly necessary, for each Parameter to have its own ParameterType definition.

This makes it easier to ensure changes in a ParameterType do not unintentionally propagate to other Parameters that share the same ParameterType. For example, given that alarms and calibrators are defined in the ParameterType, changing these in a shared ParameterType could result in a disastrous mistake. Even so, a one-to-one relationship with ParameterType and Parameter results in a larger XTCE file, which may be undesirable for some users.

Of the items selectable, the dataSource attribute is often overlooked. It defaults to the value 'telemetered'.

3.2.3.4 ContainerSet

The ContainerSet is a collection of SequenceContainers (Container). SequenceContainers are used to describe blocks of information, such as packets.

There are several kinds of Containers: plain Containers, abstract Containers, and so-called 'instance Containers', which are part of a Container inheritance chain with other Containers.

Containers refer to Parameters in an entry; there are several kinds of modifiers to the entry, such as an address (LocationInContainerInBits) and a conditional check for inclusion (IncludeCondition).

Container inheritance may also be defined. This is the preferred method for defining packets, and there are examples in the XTCE Elements Green Book 660.1.G.

Further, there are several kinds of Entries. The most often used is the ParameterRefEntry, which refers to a Parameter in a SpaceSystem. Another often-used kind of entry is the ArrayParameterRefEntry, which is used to refer to a Parameter that has an ArrayParameterType.

In XTCE 1.1, the size of an array's indices was defined in Container in the ArrayParameterRefEntry, while the number of dimensions was defined in a Parameter's ArrayParameterType.

XTCE 1.2 consolidates the above functions so that the indices can be fully defined in ArrayParameterType. The ArrayParameterRefEntry still retains the original syntax as an option for backwards compatibility and should be used in EntrySet in either case.

Figure 3-4 represents the concepts exposed above with the use of CCSDS Recommended Standards for telemetry (CCSDS 132.0-B-2 and 133.0-B-1).

It is generally recommended and a common practice to define in XTCE the packet definitions and not the framing information.

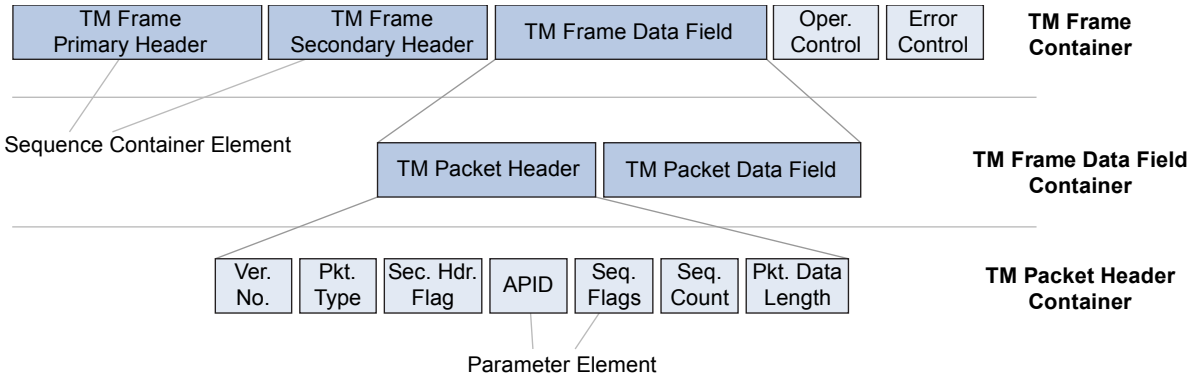

Figure 3-4: XTCE TM Packet Header and Field Applicability

3.2.3.5 MessageSet

The MessageSet is an alternative method to Container inheritance for describing packets. Its EntryList is the same as the one in Containers. However, the construction is a 'has a' relationship instead of the 'is a' relationship in Container inheritance.

However, while it still exists in the XTCE for backwards compatibility, it is proven to be one of lesser used areas of the schemas.

For this reason, one may want to consider it 'deprecated'.

3.2.3.6 StreamSet

The StreamSet describes downlink characteristics, typical information for so-called front-end processing.

This is another under-used portion of XTCE, and many in the community feel it is too simplistic for their needs of front end configuration. Even so, there are some within the XTCE community that have expressed an interest in 'fixing it', and that is likely future work.

A Stream represents a communication link. In the context of a spacecraft, the stream will describe the data link from the spacecraft to the ground. There can be any number of stream definitions for telemetry, including none at all.

A Stream defines the processing needed to extract information out of the physical channel, such as bit patterns for synchronization, expected rate, and expected Container (i.e., frame).

Streams can be of three types:

– FixedFrameStream – CCSDS frames would be in this section;

– VariableFrameStream;

– CustomFrameStream.

The first two Streams are the most common, but if it is not enough, the user can define their own stream via the custom frame stream element.

A Stream can be bound to a Container or indirectly to a service (which itself is bound to some Containers, see 3.3). Figure 3-5 presents the two possibilities to create a Stream for Virtual Channel 0.

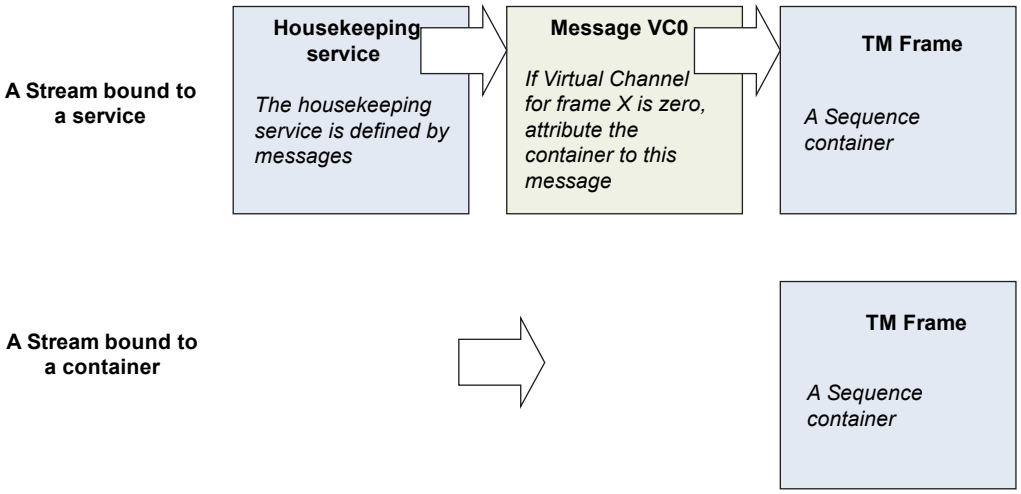

Figure 3-5: XTCE Stream Association

The Stream definition contains the information to assemble and disassemble information transmitted through the channel.

3.2.3.7 AlgorithmSet

The AlgorithmSet in XTCE is often overlooked by a new user, but 'algorithms' are an important part of ground database processing. There are many kinds of algorithms that may appear in mission operations databases, although this is somewhat tool dependent. There are scripts, derivation/pseudos, and support programs of all kinds.

The AlgorithmSet contains a collection of algorithms that may be needed by space-ground systems to perform some specialized processing to process the telemetry.

There are two kinds of algorithms in the AlgorithmSet: the CustomAlgorithm and the MathAlgorithm.

The CustomAlgorithm is very general in a nature; there is not a way to encode the instruction set of an algorithm in the element in an independent manner. Instead, the 'outer' entry and exit points (algorithm 'Arguments' and return results) may be defined along with any other descriptive and naming information. Further, an algorithm may have a TriggerSet, a set of conditions when it should be invoked. Finally, there are two options for the text of algorithm: one is to copy the text into the provided AlgorithmText area; the other is to provide a URL to refer to it externally.

Regarding derived or pseudo Parameters, they would typically be defined as a Parameter with ParameterType without a DataEncoding. The Parameter's dataSource attribute would be set to 'derived'. Then in AlgorithmSet, an algorithm for the derivation could be defined with the named Parameter in the OutputSet.

3.2.4 COMMANDMETADATA

3.2.4.1 General

The CommandMetaData element describes command side descriptive information in the database. While the majority of the syntax is very similar to the previous XTCEversion and shares much of the same feature set with that of TelemetryMetaData, many of the underlying schema types have been reworked and split off from their telemetry schema-type ancestry in XTCE 1.2, so they are now more command specific.

CommandMetaData contains the following major child elements: ParameterTypeSet, ParameterSet, ArgumentTypeSet, MetaCommandSet, CommandContainerSet, StreamSet, and AlgorithmSet.

There are some conceptual differences with some of these types even though they share much of their construction with TelemetryMetaData, as is discussed below. Perhaps the simplistic way to think of it is that the information describes the 'reverse' of TelemetryMetaData, that is, the ground-to-space transformation.

Likewise, the CommandContainer, and the private-but-similar-in-construction MetaCommand/CommandContainer describes how the Arguments and command Parameters are packaged into packets uplinked to the space system.

Figure 3-6 is a representation of the child elements of CommandMetaData.

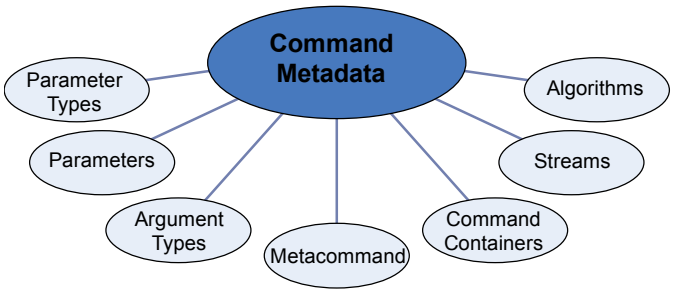

Figure 3-6: XTCE CommandMetaData

3.2.4.2 ParameterSet and ParameterTypeSet

The construction of these is identical to those in TelemetryMetaData.

In regards to the conceptual difference in command for Parameters and Arguments, the dividing line is that a command Parameter in XTCE is something that would go into a command packet but not be a settable controlling value by an end user, which would be an Argument.

But in some cases, this dividing line is not so clear to certain users because they have complete visibility to the command 'load', while other users have specific command Arguments visible only to them, and no knowledge of the underlying command packaging at all.

Even so, XTCE provides the syntax to differentiate these items if it is applicable.

For ParameterTypes and the similar ArgumentType, the DataEncoding is how the ground-supplied data type's value will be encoded for uplink; in a sense, it is the 'reverse', from a conceptual point of view, of TelemetryMetaData's ParameterType.

3.2.4.3 ArgumentTypeSet

The ArgumentTypeSet holds, similarly to ParameterTypeSet, the data types of all MetaCommand Arguments in this SpaceSystem, even though the Argument itself is local to each MetaCommand only.

Once again it is possible for Arguments to share ArgumentTypes in a very similar manner to the ParameterTypes, but in practice, most Arguments have their own ArgumentType (this is easier to manage with the downside that the XTCE files are larger).

So the ArgumentType and ParameterType share very similar features and syntax, but there are some differences.

An important difference in XTCE 1.2 is that some of the underlying schema types have been refactored and split off from those of ParameterType. Specifically, this is to address syntactic issues in locations where there is a parameterRef or parameterInstanceRef.

In XTCE 1.2, this has been changed to allow the user to refer to either an argumentRef/argumentInstanceRef or a parameterRef/parameterInstanceRef.

A final important difference is that ArgumentTypes do not have alarm definition areas like ParameterTypes.

3.2.4.4 MetaCommandSet

3.2.4.4.1 General

The MetaCommandSet holds MetaCommand (command) descriptions, the 'about' part of a command definition that includes its name, descriptive information, Arguments, and a private Container area for describing how the command is 'packaged' (packetized) for uplink. There are other elements: TransmissionConstraints, Significance, Interlock, ParametersToSet, and Verifiers (figure 3-7); these areas are discussed in more detail below.

Sequences of MetaCommand may be specified in the BlockMetaCommand.

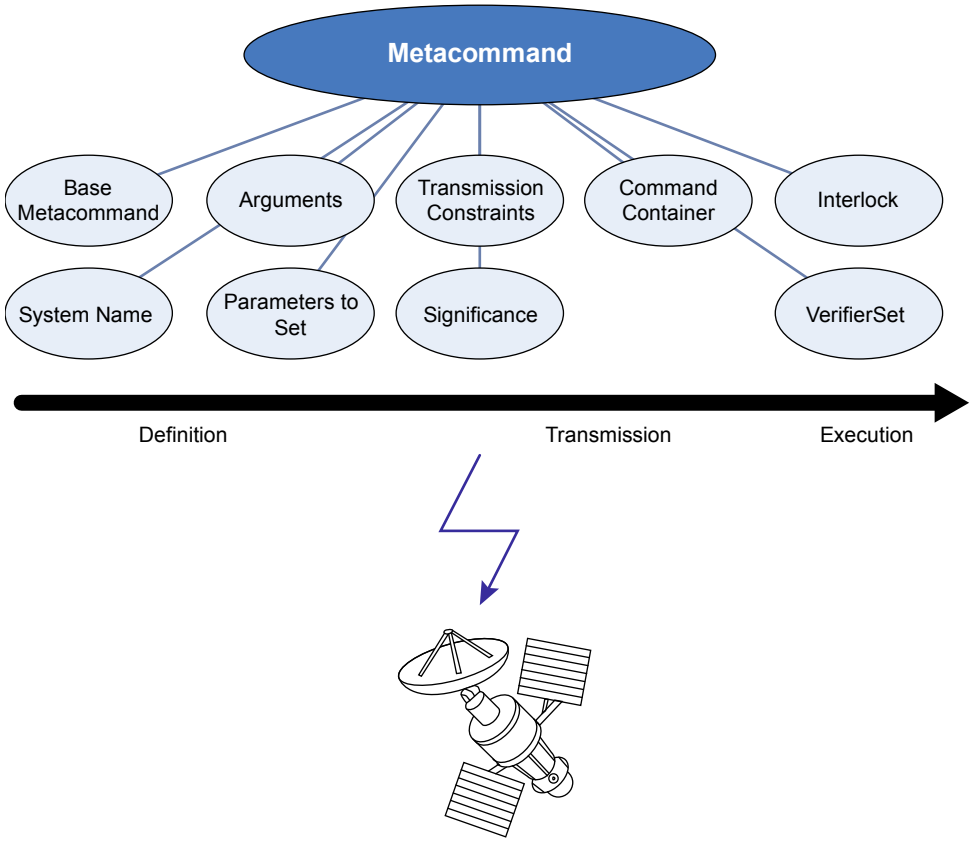

Figure 3-7: Overview of a Meta-Command

3.2.4.4.2 BaseMetaCommand

A MetaCommand (child) can extend another MetaCommand (parent), using this element. Various parts of the parent are inherited in the child, such as Arguments and Significance. But the local CommandContainer has its own explicit syntax for this mechanism.

It should be noted that Arguments may be assigned values here also.

3.2.4.4.3 ArgumentList

The ArgumentList of the MetaCommand describes Arguments used by the command. If the MetaCommand extends another MetaCommand, the complete list of Arguments is formed through the inheritance.

3.2.4.4.4 MetaCommand's CommandContainer

The local CommandContainer describes how the command is packaged. The concept here is similar to the one used with telemetry packaging, except that in this case, CommandContainers may also have Argument entries (EntrySet) that are not allowed in a SequenceContainer (see TelemetryMetaData/ContainerSet and CommandMetaData/CommandContainerSet). Moreover, this CommandContainer can also support fixed value entries (FixedValueEntry).

The inheritance mechanism also applies these CommandContainers, but the syntax must be explicitly spelled out.

Generally, most users will define a top-level abstract MetaCommand and from it define one or more levels, finally reaching specific command instances. However, the CommandContainer could reference outside of the MetaCommand and into either TelemetryMetaData/ContainerSet or CommandMetaData/CommandContainerSet, although there are some differences with the syntax; in particular, there is no ArgumentRefEntry.

This is discussed in more detail in the XTCE Element Description Green Book (reference [2]).

3.2.4.4.5 TransmissionConstraintList

TransmissionConstraints are constraints against the current command. Usually constraints consist of comparing telemetry Parameter values. All conditions defined in a transmission list must be met prior to the actual transmission of the command.

The constraints can also be represented as a CustomAlgorithm, invoking any proprietary processing before the command transmission (so constraints are not only limited to Parameter comparisons).

3.2.4.4.6 DefaultSignificance and ContextSignificanceList

Some command and control systems may require special user access or confirmations before transmitting commands. The Significance includes the name of the space system at risk and a significance level.

Significance levels are: none, watch, warning, distress, critical, and severe.

3.2.4.4.7 Interlock

An Interlock is a constraint for commands following the current command being executed. It will block successive commands until the current command has reached a certain stage (through verification). Interlocks are scoped to a space system basis. The verification to wait for (to release the lock) is linked to the status of the command. Examples of stages are: transferred to range, received, accepted, queued, executing, and complete.

3.2.4.4.8 VerifierSet

A verifier is a conditional check on the telemetry down-linked from the space system that provides verification of the execution of a command. There are eight different verifiers, each associated with different stages in command completion*:* Transferred to range, Transferred from range, Received, Accepted, Queued, Execution, Complete, and Failed. Verifiers are groups in a set for the current telecommand being defined.

3.2.4.5 ParameterToSetList

ParameterToSetList is a list that contains Parameters whose values are expected to be set after the current command has reached a given stage.

3.2.4.6 ParametersToSuspendAlarmsOnSet

ParametersToSuspendAlarmsOnSet is a list that indicates Parameters whose alarms should be suspended after the command has been sent.

3.2.5 COMMANDCONTAINERSSET, STREAMSET, AND ALGORITHMSET

CommandContainersSet, StreamSet, and AlgorithmSet are similar to their conterparts in TelemetryMetaData.

It should be noted that the element CommandContainersSet is for the ‘pieces/parts’ of MetaCommand Containers (see ContainerEntryRef) that may be shared among a number of commands.

3.3 SERVICESET

The ServiceSet is a collection of services. Services are logical groupings of Containers or Messages.

The concept of services has been introduced in XTCE for the European space organizations that deal with services. *Telemetry and Telecommand Packet Utilization* (reference [9]), compliant with CCSDS standards, goes a little bit deeper in packet composition and introduces the notion of services. Please refer to that standard in order to get examples and detailed definitions of services.

3.4 ORGANIZATION OF XTCE MAJOR ELEMENTS

Figure 3-8 diagrammatically illustrates the hierarchy of the XTCE schema. All major elements are represented up to a depth of 3 (one color per depth level). This is sufficient to understand the overall structure of XTCE Schema.

In the figure, many items are represented in a syntactically similar way in both TelemetryMetadData and CommandMetaData.

In some cases, the underlying schema types are identical.

3.5 XTCE XML TREE OVERVIEW

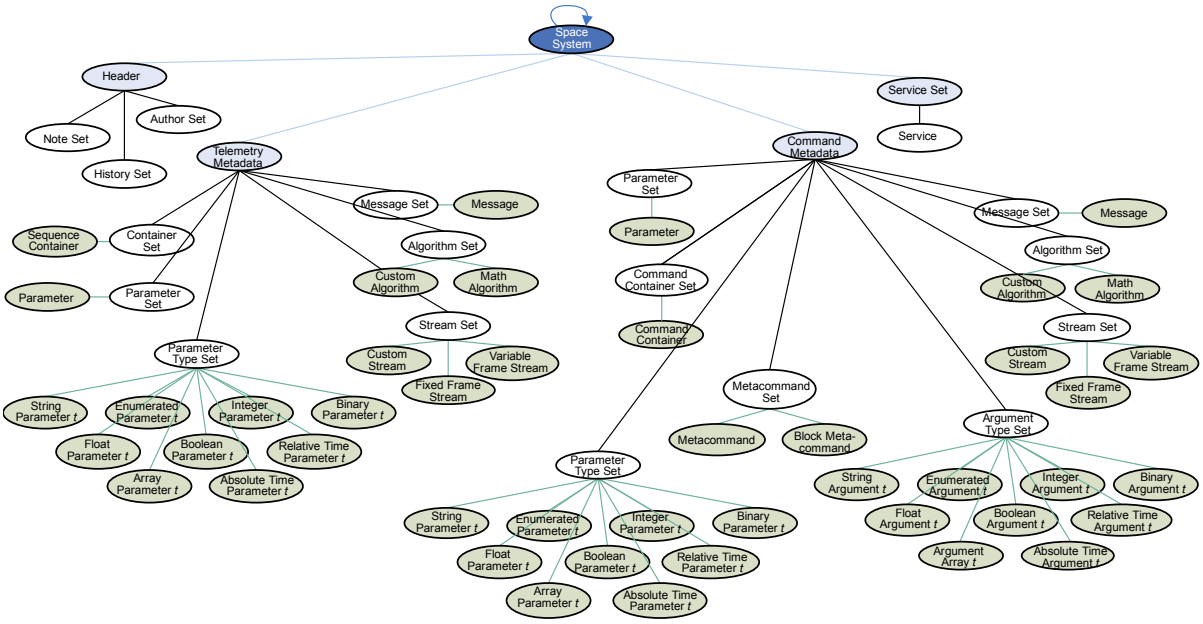

Figure 3-8: XTCE Tree Overview

4 USAGE EXAMPLE OF XTCE

This section addresses the issue of a Spacecraft Project Manager who receives various onboard subsystems and payloads for their spacecraft that are developed by various contractors.

From each of them, the Spacecraft Project Manager receives an XTCE XML file containing telecommand and telemetry definitions for that subsystem/payload.

Those files are merged into a single XTCE Spacecraft Source Database (SDB), which is then converted to run-time databases for use in Assembly, Integration, and Verification (AIV), also called Integration and Test (I&T).

During a typical mission lifecycle, there are several levels of database integration, translation, and re-translation. The spacecraft manufacturer integrates many systems, sub-systems. These sub-systems are developed by vendors or subcontractors from around the world. Each of these sub-systems will generate telemetry data and receive commands that must be described in a database.

The spacecraft manufacturer must include all sub-system data in a master spacecraft database (Source Database).

This master spacecraft Source Database will then be sent to the mission readiness team for inclusion or translation into many different databases used for operations (Operational Database).

Figure 4-1 depicts a representative mission database flow from AIV (I&T) to Operations.

Using XTCE as the program's standard or common telemetry and command database exchange format will reduce mission cost, schedule, and risk by eliminating custom database translations.

XTCE is the exchange of database information with other databases, not the direct interface for sending commands or receiving telemetry between systems and subsystems.

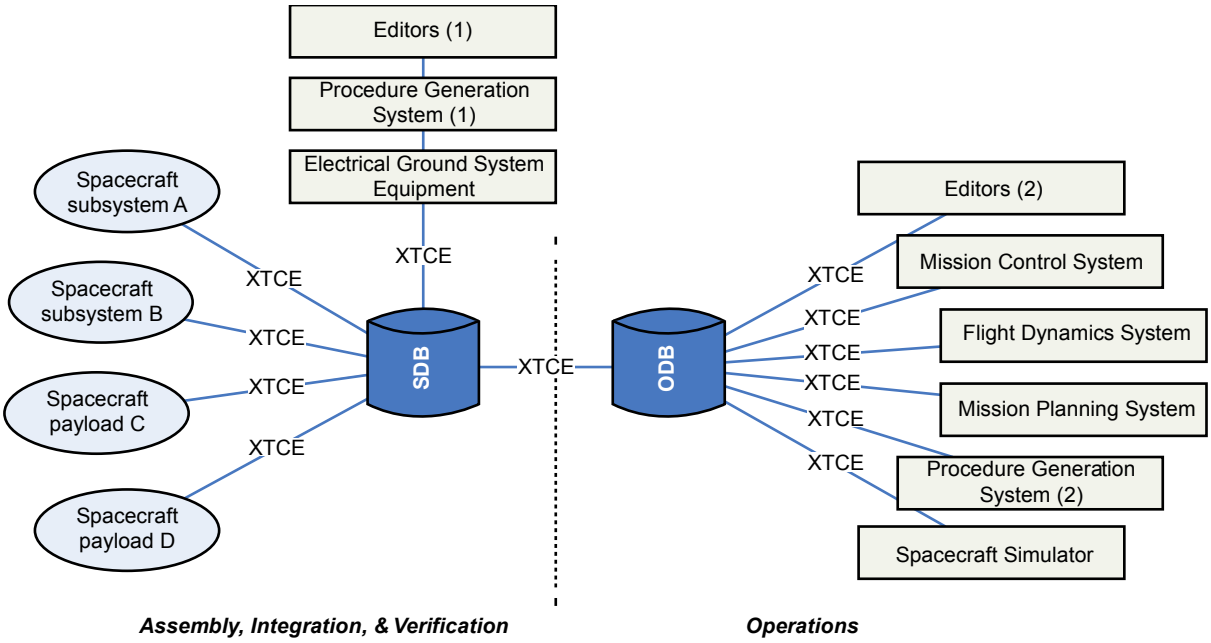

Figure 4-1: XTCE Usage Scenario

ANNEX A

XTCE EXAMPLE

A1 OVERVIEW

The following short example contains two commands and telemetry Parameters. There are longer examples in the XTCE Element Description Green Book (reference [2]) as well as an extended GovSat example. This example uses one packet and one command from it.

A2 SPACESYSTEM

A2.1 GENERAL

The SpaceSystem is the root element of an XTCE XML.

The following code fragment represents the overall structure of the file.

```
<xtce:SpaceSystem xmlns:xtce="http://www.omg.org/spec/XTCE/20180204"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="GovSat"
shortDescription="A GovSat example"
xsi:schemaLocation="http://www.omg.org/spec/XTCE/20180204 XTCE12.xsd">

      <xtce:LongDescription>This file is intended to give a quick
overview of XTCE for CCSDS based mission. This example does not cover all
possibilities of XTCE but tries to give relevant examples of the main
points covered by the standard. </xtce:LongDescription>

      <xtce:Header validationStatus="Validated" date="2019-04-04"
version="0.2"/>   <TelemetryMetaData>
            <ParameterTypeSet>[…]</ParameterTypeSet>
            <ParameterSet>[…]</ParameterSet>
            <ContainerSet>[…]</ContainerSet>

      </TelemetryMetaData>
      <CommandMetaData>
            <ArgumentTypeSet>[…]</ArgumentTypeSet>
            <MetaCommandSet>[…]</MetaCommandSet>

      </CommandMetaData>
</SpaceSystem>
```

A2.2 TELEMETRYMETADATA

A2.2.1 Overview

This step defines the three empty elements of the telemetry metadata element: ParameterTypeSet, ParameterSet and ContainerSet.

A2.2.2 Telemetry ParameterTypeSet

The following section defines ParameterTypes that will be used by the Parameters that appear in the Containers.

```
<xtce:ParameterTypeSet>
                        <xtce:IntegerParameterType signed="false" name="CCSDSSCIDType"/>
                        <xtce:IntegerParameterType signed="false" name="CCSDSVCIDType"/>
                        <xtce:IntegerParameterType signed="false" name="CCSDSVersionType">
                                <xtce:IntegerDataEncoding sizeInBits="3"/>
                        </xtce:IntegerParameterType>
                        <xtce:IntegerParameterType signed="false" name="CCSDSTypeType">
                                <xtce:IntegerDataEncoding sizeInBits="1"/>
                        </xtce:IntegerParameterType>
                        <xtce:IntegerParameterType signed="false" name="CCSDSSecHType">
                                <xtce:IntegerDataEncoding sizeInBits="1"/>
                        </xtce:IntegerParameterType>
                        <xtce:IntegerParameterType signed="false" name="CCSDSAPIDType">
                                <xtce:IntegerDataEncoding sizeInBits="11"/>
                        </xtce:IntegerParameterType>
                        <xtce:IntegerParameterType signed="false" name="CCSDSGroupFlagsType">
                                <xtce:IntegerDataEncoding sizeInBits="2"/>
                        </xtce:IntegerParameterType>
                        <xtce:IntegerParameterType signed="false"
name="CCSDSSourceSequenceCountType">
                                <xtce:IntegerDataEncoding sizeInBits="14"/>
                        </xtce:IntegerParameterType>
                        <xtce:IntegerParameterType signed="false" name="CCSDSPacketLengthType">
                                <xtce:IntegerDataEncoding sizeInBits="16"/>
                        </xtce:IntegerParameterType>
                        <xtce:AbsoluteTimeParameterType name="TimeStampType"
shortDescription="16b: pfld, 32b: secs epoch, 16b: (1 mil u/ffff) 15.2587890625 usec per
tick">
                                <xtce:Encoding units="seconds" scale="1.0" offset="0.0">
                                        <xtce:IntegerDataEncoding encoding="unsigned"
sizeInBits="64" bitOrder="mostSignificantBitFirst"/>
                                </xtce:Encoding>
                                <xtce:ReferenceTime>
                                        <xtce:Epoch>TAI</xtce:Epoch>
                                </xtce:ReferenceTime>
                        </xtce:AbsoluteTimeParameterType>
                        <xtce:FloatParameterType sizeInBits="64" name="PowmX0EP_VoltageType">
                                <xtce:UnitSet>
                                        <xtce:Unit power="1.0" factor="1"
description="ScaleUnits">V</xtce:Unit>
                                </xtce:UnitSet>
                                <xtce:IntegerDataEncoding encoding="twosComplement"
sizeInBits="16" bitOrder="mostSignificantBitFirst">
                                        <xtce:DefaultCalibrator>
                                                <xtce:PolynomialCalibrator>
                                                        <xtce:Term coefficient="0.0"
exponent="0"/>
                                                        <xtce:Term coefficient="0.1"
exponent="1"/>
                                                </xtce:PolynomialCalibrator>
                                        </xtce:DefaultCalibrator>
                                </xtce:IntegerDataEncoding>
                        </xtce:FloatParameterType>
                        <xtce:FloatParameterType sizeInBits="64" name="PowmX0EP_CurrentType">
                                <xtce:UnitSet>
                                        <xtce:Unit power="1.0" factor="1"
description="ScaleUnits">A</xtce:Unit>
                                </xtce:UnitSet>
                                <xtce:IntegerDataEncoding encoding="twosComplement"
sizeInBits="16" bitOrder="mostSignificantBitFirst">
                                        <xtce:DefaultCalibrator>
                                                <xtce:PolynomialCalibrator>
                                                        <xtce:Term coefficient="0.0"
exponent="0"/>
                                                        <xtce:Term coefficient="0.1"
exponent="1"/>
                                                </xtce:PolynomialCalibrator>
```

```
                                        </xtce:DefaultCalibrator>
                                </xtce:IntegerDataEncoding>
                        </xtce:FloatParameterType>
                        <xtce:IntegerParameterType sizeInBits="32" signed="false"
name="PowmX0PPS SourceType">
                                <xtce:IntegerDataEncoding encoding="unsigned" sizeInBits="8"
bitOrder="mostSignificantBitFirst"/>
                        </xtce:IntegerParameterType>
                        <xtce:FloatParameterType sizeInBits="64" name="PowmX0EP_MaxCurType">
                                <xtce:UnitSet>
                                        <xtce:Unit power="1.0" factor="1"
description="ScaleUnits">A</xtce:Unit>
                                </xtce:UnitSet>
                                <xtce:IntegerDataEncoding encoding="twosComplement"
sizeInBits="16" bitOrder="mostSignificantBitFirst">
                                        <xtce:DefaultCalibrator>
                                                <xtce:PolynomialCalibrator>
                                                        <xtce:Term coefficient="0.0"
exponent="0"/>
                                                        <xtce:Term coefficient="0.1"
exponent="1"/>
                                                </xtce:PolynomialCalibrator>
                                        </xtce:DefaultCalibrator>
                                </xtce:IntegerDataEncoding>
                        </xtce:FloatParameterType>
                        <xtce:FloatParameterType sizeInBits="64" name="PowmX0Bus_VoltageType">
                                <xtce:UnitSet>
                                        <xtce:Unit power="1.0" factor="1"
description="ScaleUnits">V</xtce:Unit>
                                </xtce:UnitSet>
                                <xtce:IntegerDataEncoding encoding="twosComplement"
sizeInBits="16" bitOrder="mostSignificantBitFirst">
                                        <xtce:DefaultCalibrator>
                                                <xtce:PolynomialCalibrator>
                                                        <xtce:Term coefficient="0.0"
exponent="0"/>
                                                        <xtce:Term coefficient="0.1"
exponent="1"/>
                                                </xtce:PolynomialCalibrator>
                                        </xtce:DefaultCalibrator>
                                </xtce:IntegerDataEncoding>
                        </xtce:FloatParameterType>
                        <xtce:EnumeratedParameterType initialValue="Off"
name="PowmX0EP_PowerStateType">
                                <xtce:IntegerDataEncoding encoding="unsigned" sizeInBits="8"
bitOrder="mostSignificantBitFirst"/>
                                <xtce:EnumerationList>
                                        <xtce:Enumeration value="1" label="Off"/>
                                        <xtce:Enumeration value="2" label="On"/>
                                        <xtce:Enumeration value="3" label="Tripped"/>
                                        <xtce:Enumeration value="4" label="NotWorking"/>
                                </xtce:EnumerationList>
                        </xtce:EnumeratedParameterType>
                        <xtce:FloatParameterType sizeInBits="64" name="PowmX0EP_SetPointType">
                                <xtce:UnitSet>
                                        <xtce:Unit power="1.0" factor="1"
description="ScaleUnits">A</xtce:Unit>
                                </xtce:UnitSet>
                                <xtce:IntegerDataEncoding encoding="twosComplement"
sizeInBits="16" bitOrder="mostSignificantBitFirst">
                                        <xtce:DefaultCalibrator>
                                                <xtce:PolynomialCalibrator>
                                                        <xtce:Term coefficient="0.0"
exponent="0"/>
                                                        <xtce:Term coefficient="0.1"
exponent="1"/>
                                                </xtce:PolynomialCalibrator>
                                        </xtce:DefaultCalibrator>
                                </xtce:IntegerDataEncoding>
                        </xtce:FloatParameterType>
                </xtce:ParameterTypeSet>
                <xtce:ParameterSet>
                        <xtce:Parameter parameterTypeRef="CCSDSSCIDType" name="CCSDSSCID">
                                <xtce:ParameterProperties dataSource="local"/>
                        </xtce:Parameter>
```

```
                        <xtce:Parameter parameterTypeRef="CCSDSVCIDType" name="CCSDSVCID">
                                <xtce:ParameterProperties dataSource="local"/>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="CCSDSVersionType"
name="CCSDSVersion"/>
                        <xtce:Parameter parameterTypeRef="CCSDSTypeType" name="CCSDSType"/>
                        <xtce:Parameter parameterTypeRef="CCSDSSecHType" name="CCSDSSecH"/>
                        <xtce:Parameter parameterTypeRef="CCSDSAPIDType" name="CCSDSAPID"/>
                        <xtce:Parameter parameterTypeRef="CCSDSGroupFlagsType"
name="CCSDSGroupFlags"/>
                        <xtce:Parameter parameterTypeRef="CCSDSSourceSequenceCountType"
name="CCSDSSourceSequenceCount"/>
                        <xtce:Parameter parameterTypeRef="CCSDSPacketLengthType"
name="CCSDSPacketLength"/>
                        <xtce:Parameter parameterTypeRef="TimeStampType" name="TimeStamp"/>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_VoltageType"
name="PowmX0EP_Voltage">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_CurrentType"
name="PowmX0EP_Current">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0PPS_SourceType"
name="PowmX0PPS Source">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_MaxCurType"
name="PowmX0EP_MaxCur">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0Bus_VoltageType"
name="PowmX0Bus_Voltage">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_PowerStateType"
name="PowmX0EP PowerState">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_SetPointType"
name="PowmX0EP_SetPoint">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                </xtce:ParameterSet>
```

A2.2.3 Telemetry ParameterSet

The Parameters that appear in the Containers refer to the various ParameterTypes with a one to one mapping.

```
<xtce:ParameterSet>
                        <xtce:Parameter parameterTypeRef="CCSDSSCIDType" name="CCSDSSCID">
                                <xtce:ParameterProperties dataSource="local"/>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="CCSDSVCIDType" name="CCSDSVCID">
                                <xtce:ParameterProperties dataSource="local"/>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="CCSDSVersionType"
name="CCSDSVersion"/>
                        <xtce:Parameter parameterTypeRef="CCSDSTypeType" name="CCSDSType"/>
                        <xtce:Parameter parameterTypeRef="CCSDSSecHType" name="CCSDSSecH"/>
                        <xtce:Parameter parameterTypeRef="CCSDSAPIDType" name="CCSDSAPID"/>
                        <xtce:Parameter parameterTypeRef="CCSDSGroupFlagsType"
name="CCSDSGroupFlags"/>
                        <xtce:Parameter parameterTypeRef="CCSDSSourceSequenceCountType"
name="CCSDSSourceSequenceCount"/>
                        <xtce:Parameter parameterTypeRef="CCSDSPacketLengthType"
name="CCSDSPacketLength"/>
                        <xtce:Parameter parameterTypeRef="TimeStampType" name="TimeStamp"/>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_VoltageType"
name="PowmX0EP_Voltage">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_CurrentType"
name="PowmX0EP_Current">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0PPS_SourceType"
name="PowmX0PPS_Source">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_MaxCurType"
name="PowmX0EP MaxCur">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0Bus_VoltageType"
name="PowmX0Bus_Voltage">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_PowerStateType"
name="PowmX0EP_PowerState">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                        <xtce:Parameter parameterTypeRef="PowmX0EP_SetPointType"
name="PowmX0EP SetPoint">
                                <xtce:ParameterProperties readOnly="false">
                                        <xtce:SystemName>PowmX0</xtce:SystemName>
                                </xtce:ParameterProperties>
                        </xtce:Parameter>
                </xtce:ParameterSet>
```

A2.2.4 Telemetry ContainerSet

The ContainerSet defines one packet using inheritance. It should be noted that it is the Container with name PowmX0EP_StatusPacket:

```
<xtce:ContainerSet>
                        <xtce:SequenceContainer abstract="true" name="CCSDSPacket">
                                <xtce:LongDescription>Super-container for all CCSDS telemetry
and command packets</xtce:LongDescription>
                                <xtce:EntryList>
                                        <xtce:ParameterRefEntry parameterRef="CCSDSVersion"/>
                                        <xtce:ParameterRefEntry parameterRef="CCSDSType"/>
                                        <xtce:ParameterRefEntry parameterRef="CCSDSSecH"/>
                                        <xtce:ParameterRefEntry parameterRef="CCSDSAPID"/>
                                        <xtce:ParameterRefEntry parameterRef="CCSDSGroupFlags"/>
                                        <xtce:ParameterRefEntry
parameterRef="CCSDSSourceSequenceCount"/>
                                        <xtce:ParameterRefEntry
parameterRef="CCSDSPacketLength"/>
                                </xtce:EntryList>
                        </xtce:SequenceContainer>
                        <xtce:SequenceContainer abstract="true" name="CCSDSTelemetryPacket">
                                <xtce:LongDescription>Super-container for all CCSDS telemetry
packets.</xtce:LongDescription>
                                <xtce:EntryList/>
                                <xtce:BaseContainer containerRef="CCSDSPacket">
                                        <xtce:RestrictionCriteria>
                                                <xtce:ComparisonList>
                                                        <xtce:Comparison value="0"
parameterRef="CCSDSVersion"/>
                                                        <xtce:Comparison value="0"
parameterRef="CCSDSType"/>
                                                </xtce:ComparisonList>
                                        </xtce:RestrictionCriteria>
                                </xtce:BaseContainer>
                        </xtce:SequenceContainer>
                        <xtce:SequenceContainer idlePattern="0x0" name="PowmX0EP_StatusPacket">
                                <xtce:AncillaryDataSet>
                                        <xtce:AncillaryData name="VCID"
mimeType="text/plain">6</xtce:AncillaryData>
                                        <xtce:AncillaryData name="SystemName"
mimeType="text/plain">PowmX0</xtce:AncillaryData>
                                </xtce:AncillaryDataSet>
                                <xtce:EntryList>
                                        <xtce:ParameterRefEntry parameterRef="TimeStamp"/>
                                        <xtce:ParameterRefEntry
parameterRef="PowmX0EP_Voltage"/>
                                        <xtce:ParameterRefEntry
parameterRef="PowmX0EP_Current"/>
                                        <xtce:ParameterRefEntry
parameterRef="PowmX0PPS_Source"/>
                                        <xtce:ParameterRefEntry parameterRef="PowmX0EP_MaxCur"/>
                                        <xtce:ParameterRefEntry
parameterRef="PowmX0Bus_Voltage"/>
                                        <xtce:ParameterRefEntry
parameterRef="PowmX0EP_PowerState"/>
                                        <xtce:ParameterRefEntry
parameterRef="PowmX0EP_SetPoint"/>
                                </xtce:EntryList>
                                <xtce:BaseContainer containerRef="CCSDSTelemetryPacket">
                                        <xtce:RestrictionCriteria>
                                                <xtce:Comparison comparisonOperator="=="
value="47" instance="0" useCalibratedValue="true" parameterRef="CCSDSAPID"/>
                                        </xtce:RestrictionCriteria>
                                </xtce:BaseContainer>
                        </xtce:SequenceContainer>
                </xtce:ContainerSet>
```

A2.3 COMMANDMETADATA

A2.3.1 Overview

The CommandMetaData element example below is composed of Arguments, MetaCommands, and CommandContainers. There is one command defined.

A2.3.2 Command ArgumentTypes

Arguments refer to ArgumentTypes, similar to Parameters and ParameterTypes.

```
<xtce:ArgumentTypeSet>
                        <xtce:IntegerArgumentType sizeInBits="32" signed="false"
name="PM1Msg TypeType">
                                <xtce:IntegerDataEncoding encoding="unsigned" sizeInBits="8"
bitOrder="mostSignificantBitFirst"/>
                        </xtce:IntegerArgumentType>
                        <xtce:IntegerArgumentType sizeInBits="32" signed="false"
name="PM1AddressType">
                                <xtce:IntegerDataEncoding encoding="unsigned" sizeInBits="32"
bitOrder="mostSignificantBitFirst"/>
                        </xtce:IntegerArgumentType>
                        <xtce:IntegerArgumentType sizeInBits="32" signed="false" name="PM1PortType">
                                <xtce:IntegerDataEncoding encoding="unsigned" sizeInBits="16"
bitOrder="mostSignificantBitFirst"/>
                        </xtce:IntegerArgumentType>
                        <xtce:IntegerArgumentType sizeInBits="32" signed="false"
name="PM1Sensor_IDType">
                                <xtce:IntegerDataEncoding encoding="unsigned" sizeInBits="32"
bitOrder="mostSignificantBitFirst"/>
                        </xtce:IntegerArgumentType>
                </xtce:ArgumentTypeSet>
```

A2.3.3 MetaCommand

The following is a command take from GovSat. It uses inheritance to define one command.

```
<xtce:MetaCommandSet>
                        <xtce:MetaCommand abstract="true" name="CCSDSCommand">
                                <xtce:LongDescription>Super-Command for all CCSDS
commands.</xtce:LongDescription>
                                <xtce:CommandContainer name="CCSDSCommandPacket">
                                        <xtce:LongDescription>Super-container for all CCSDS command
packets.</xtce:LongDescription>
                                        <xtce:EntryList/>
                                        <xtce:BaseContainer containerRef="CCSDSPacket">
                                                <xtce:RestrictionCriteria>
                                                        <xtce:ComparisonList>
                                                                <xtce:Comparison value="0"
parameterRef="CCSDSVersion"/>
                                                                <xtce:Comparison value="1"
parameterRef="CCSDSType"/>
                                                        </xtce:ComparisonList>
                                                </xtce:RestrictionCriteria>
                                        </xtce:BaseContainer>
                                </xtce:CommandContainer>
                        </xtce:MetaCommand>
                        <xtce:MetaCommand abstract="false" name="PM1Enable Logging">
                                <xtce:BaseMetaCommand metaCommandRef="CCSDSCommand"/>
                                <xtce:SystemName>PM1</xtce:SystemName>
                                <xtce:ArgumentList>
                                        <xtce:Argument argumentTypeRef="PM1Msg_TypeType"
name="PM1Msg_Type"/>
                                        <xtce:Argument argumentTypeRef="PM1AddressType"
```

```
name="PM1Address"/>
                                        <xtce:Argument argumentTypeRef="PM1PortType"
name="PM1Port"/>
                                        <xtce:Argument argumentTypeRef="PM1Sensor_IDType"
name="PM1Sensor ID"/>
                                </xtce:ArgumentList>
                                <xtce:CommandContainer name="PM1Enable_LoggingPacket">
                                        <xtce:AncillaryDataSet>
                                                <xtce:AncillaryData name="VCID"
mimeType="text/plain">27</xtce:AncillaryData>
                                        </xtce:AncillaryDataSet>
                                        <xtce:EntryList>
                                                <xtce:ArgumentRefEntry argumentRef="PM1Msg Type"/>
                                                <xtce:ArgumentRefEntry argumentRef="PM1Address"/>
                                                <xtce:ArgumentRefEntry argumentRef="PM1Port"/>
                                                <xtce:ArgumentRefEntry argumentRef="PM1Sensor_ID"/>
                                        </xtce:EntryList>
                                        <xtce:BaseContainer containerRef="CCSDSCommandPacket">
                                                <xtce:RestrictionCriteria>
                                                        <xtce:Comparison comparisonOperator="=="
value="130" instance="0" useCalibratedValue="true" parameterRef="CCSDSAPID"/>
                                                </xtce:RestrictionCriteria>
                                        </xtce:BaseContainer>
                                </xtce:CommandContainer>
                        </xtce:MetaCommand>
                </xtce:MetaCommandSet>
        </xtce:CommandMetaData>
```

A2.3.4 CommandContainers

There is no specific CommandContainerSet in this example. The CCSDS header Containers are shared in and defined in the TelemetryMetaData area, which is legal XTCE.

All previous code fragments put together give a complete XML files that is valid against XTCE Schema 1.2.

ANNEX B

GLOSSARY

B1 DEFINITION OF TERMS

Here are definitions for specific terms used in this document and their meaning in the XTCE context:

Argument A value that is part of a telecommand packet. This value is in general set by a human actor, usually a spacecraft controller. Arguments are only used in the context of commanding, never for telemetry purposes.

command An instruction for a remote system. Spacecraft command definition usually implies information about coding and packaging the command, validation and verification checks, as well as command execution control.

Container A Container describes a list of items, intended as a general way to describe packets, or frames, or any other 'packaging' of telemetered or command data.

element An XML document is composed of a tree of elements. An element is a named tag, which can have associated attributes or not. A tag starts with '<' and end with '>'. An element can be empty or can contain text or other elements. The tags defining an element come in pairs. Example: <element attribute="I am an attribute"> some text </element>

list A list is an ordered collection of elements. For example, an Argument list is an ordered collection of Arguments.

meta Meta is a prefix used for descriptions. Added to a name, meta declares that the content is a high-level description of the name, not the name itself. For example, a Metacommand is a description of the command, not the command itself (i.e., packaging or packet).

Parameter A Parameter is a value received or set by a machine. Parameters can be telemetered, derived, or constant. Often called mnemonic, mnemonic value, telemetered data point, or similar terms throughout the industry.

Ref A Ref is a named reference to an element defined elsewhere in the XML document. For example, an argumentRef is a named reference to an Argument defined elsewhere.

Set A Set is an unordered collection of elements. For example, a MetaCommandSet is an unordered collection of command descriptions.

telecommand A telecommand is a command.

telemetry Measurement with the aid of intermediate means that permit the measurement to be interpreted at a distance from the primary detector. All measurements on board the spacecraft are transmitted to the ground system in a telemetry stream. Most telemetry measurements will require engineering unit conversion, and measurements will have associated validation ranges or lists of acceptable values.

B2 ABBREVIATIONS AND ACRONYMS

AIV	assembly, integration, and verification
CCSDS	Consultative Committee for Space Data Systems
DB	database
DTD	document type definitions
EGSE	electrical ground system equipment
FDS	flight dynamics system
I&T	integration and test
MCS	mission control system
MOIMS	mission operations and information management systems
MF	minor frame
MPS	mission planning system
MSB	most significant bit
ODB	operational database
OMG	Object Management Group
PGS	procedure generation system
S/C	spacecraft
SDB	source database
SDTF	Space Domain Task Force
SIM	spacecraft simulator
SM&C	Spacecraft Monitor and Control
T&C	telemetry and command
TDM	time division multiplexing
TC	telecommand
TM	telemetric
VC	virtual channel
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
XSD	XML schema definition
XTCE	XML Telemetric and Command Exchange