XTCE US Government Satellite Conformance Profile (XUSP)

Version 1.0 – FTF Beta ~~1~~2

Comment [blk1]: editorial, also changes to document name in footer to match spec name

OMG Document Number: dtc/2014-09-11~~8 xx~~
Standard document URL: http://www.omg.org/spec/XUSP/1.0
Associated Machine-readable files:~~xxxxx~~
http://www.omg.org/spec/XUSP/20140801/XUSProfileRules.csv (normative)
http://www.omg.org/spec/XUSP/20140801/XUSPTemplate.xtce (normative)

Comment [blk2]: editorial

Comment [blk3]: 18369 & 18372, both changes replaced by 19594

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully- paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, ormechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES

LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.

IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227- 7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 140 Kendrick Street, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management GroupTM, OMGTM , Unified Modeling LanguageTM, Model Driven Architecture LogoTM, Model Driven Architecture DiagramTM, CORBA logosTM, XMI LogoTM, CWMTM, CWM LogoTM, IIOPTM , IMMTM, MOFTM , OMG Interface Definition Language (IDL)TM , and OMG Systems Modeling Language (OMG SysML)TM are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page *http:// www.omg.org*, under Documents, Report a Bug/Issue (http://www.omg.org/technology/agreement.

Table of Contents

Comment [blk4]: 18370 eliminated one subsection.

Table of Figures

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling LanguageTM); CORBA® (Common Object Request Broker Architecture); CWMTM (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/.*

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A catalog of all OMG Specifications is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

Specifications within the Catalog are organized by the following categories:

Business Modeling Specifications

- Business Rules and Process Management Specifications

Language Mappings

- IDL/Language Mapping Specifications
- Other Language Mapping Specifications

Middleware Specifications

- CORBA/IIOP
- CORBA Component Model
- Data Distribution
- Specialized CORBA

Modeling and Metadata Specifications

- UML
- MOF
- XMI
- CWM

- Profile specifications.

Modernization Specifications

- KDM

Platform Independent Model (PIM), Platform Specific Model (PSM), and Interface Specifications

- CORBAservices
- CORBAfacilities
- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) All specifications are available in PostScript and PDF format and may be obtained from the Specifications Catalog cited above. Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

OMG Contact Information

OMG Headquarters
140 Kendrick Street Building A, Suite 300
Needham, MA 02494 USA

Tel: +1-781-444-0404
Fax: +1-781-444-0320
http://www.omg.org/
Email: *pubs@omg.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax

elements. **`Courier - 10 pt. Bold:`** Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/technology/agreement.html*.

Introduction to Specification

This XML Telemetric and Command Exchange (XTCE) Government Satellite (GovSat) Tailoring Guide Specification defines a specialization of XTCE typical of United States (US) based space missions. XTCE is too broad for these missions and only a subset is necessary for successful use. The tailoring is necessary to reduce both implementation and cost for users and COTS vendors of XTCE. This tailoring guide defines the GovSat subset of XTCE 1.1 for US missions that are CCSDS compliant. In the event of conflict or ambiguity in this tailoring guide, the *XTCE 1.1 Specification* takes precedence.

The normative portion (section 6) of this specification is presented as a table of rules. To be XML 1.1 GovSat compliant, the rules must be met in addition to being a valid XTCE 1.1 document.

1 Scope

This specification addresses the need for a subset of XTCE 1.1 called GovSat for United States of America (USA) missions that are CCSDS compliant.

Missions with the following telemetry and command features will find this specialization applicable:

- Uses the CCSDS packet format
- Supports packet identification using from one to four items
- Supports some or all the following data types in telemetry and command: integer, float, string, enumeration, array and structure
- Supports three-levels of alarms/limits
- Supports polynomial calibration with no more than 10 terms
- Supports linear calibration with no more than 100 points

2 Conformance

Conformance to the tailoring consists of two parts: the XTCE document in question is valid against XTCE 1.1, and the rules in the table of section 6 as applied against the document are all true.

3 Normative References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

http://www.w3.org/	XPath 2.0
http://www.w3.org/TR/xmlschema-0/	XML Schema Part 0: Primer

4 Terms and Definitions

For the purposes of this specification, the following terms and definitions apply.

Telemetering

(IEEE Std 100-1996 [1996]) “Measurement with the aid of intermediate means that permit the measurement to be interpreted at a distance from the primary detector.” Measurements on board the spacecraft are transmitted via one or more telemetry streams to spacecraft monitoring systems. Telemetry as used here refers to these measurements originating from both the spacecraft and from systems (such as ground system components) used to support the spacecraft. Most telemetry measurements will require engineering unit conversion and measurements will have associated validation ranges or lists of acceptable values.

Commands

Commands are messages that initiate actions on a remote system. Commands as used here may mean both commands destined for the spacecraft and to the systems used to support the spacecraft. Spacecraft commanding usually implies coding and packaging of the command information, validation and verification, as well as authorization to perform. Telemetry and Commanding data are necessarily related to one another, with some command information originating from telemetry and commands relating to particular telemetry measurements. Therefore, the ability to relate individual telemetry with one another and to commands is a very important part of this specification.

Acronyms

GovSat	Government Satellite
List	An ordered collection, for example an ArgumentList is an ordered collection of arguments.
Meta	Is a description, for example a MetaCommand is a command description.
NASA	National Aeronautics and Space Administration
Para	An abbreviation sometimes used for Parameter
Ref	A reference (by name) to an object defined elsewhere in the XML document, for example an ArgumentRef is a named reference to an Argument defined elsewhere.
Set	An unordered collection, for example a MetaCommandSet is an unordered collection of command descriptions.
UCS	Universal Character Set
UTF	UCS Transformation Format
W3C	World Wide Web Consortium
XTCE	XML Telemetric and Command Exchange format

5 Additional Information

5.1 Acknowledgements

The following organizations submitted and/or supported parts of this specification:

- NASA Goddard Space Flight Center (GSFC)

6 GovSat Tailoring Guide

6.1 Rules Table

The table consists of a set of rules that must be met in order for an XTCE document to be considered GovSat compliant.

The rules are written in XPath 2.0.

The GovSat Tailoring Guide is comprised of two parts:

- XTCE 1.1 GovSat Tailoring Guide Specification (this document) – rationale & description
- XTCE 1.1 GovSat Tailoring Guide Rules Table (separate spreadsheet) – defines rules used by implementers
  - http://www.omg.org/spec/XUSP/20140801/XUSProfileRules.csv~~Excel file name: XTCE_v1.1_GovSat Tailoring Guide_Rules Table~~

Comment [blk5]: 18369 change superseded by 19594

6.1.1 Table Format

The rules table spreadsheet is defined with five mandatory columns plus additional optional columns if needed:

- **5 Mandatory**
  - Column 1 Title: **"Title"** of Spreadsheet - include Title Name, version of XTCE, date
  - Column 2 Title: **"Element"** – full XPath 2.0 of all XTCE elements (includes children)
  - Column 3 Title: **"Tailoring"** – entries are **"Supported"** and **"X"** (not supported)
  - Column 4 Title: **"Rule Description"** – textual description of the rule and restrictions
  - Column 5 Title: **"Rule Specification"** – XPath 2.0 expression of the rule.
- **Optional**
  - Add additional columns if needed for a mission (i.e. "Source")

~~6.1.2 Table Summary~~

Comment [blk6]: 18370

~~The following table summarizes the rules from the complete rules table (Excel file). This table only contains 2 of the 5 mandatory columns. The Excel spreadsheet contains all 5 mandatory columns.~~

~~6.1.2.1 Rules Table Summary~~

~~Rule Description~~	~~Rule Specification~~
~~Only one SpaceSystem is allowed~~	~~count(//xtce:SpaceSystem)=1~~
~~Names may be from 1 to 64 characters~~	~~count(//*/@name[string-length() < 1]| //*/@name[string-length() > 64])=0~~
~~Short descriptions may be from 1 to 128 characters~~	~~count(//*/@shortDescription[string-length() < 1]| //*/@shortDescription[string-length() > 128])=0~~
~~Long descriptions may be from 1 to 1024 characters~~	~~not((for $x in //xtce:LongDescription/text() return string-length($x)) < 1)~~
~~FileType must be defined in the SpaceSystem's AncillaryDataSet and it must be set to a value of GovSat~~	~~count(//xtce:SpaceSystem/xtce:AncillaryDataSet/xtce:AncillaryData[@name='FileType' and text()='GovSat'])=1~~
~~Restriction patterns for StringParameterType are not supported~~	~~count(//*/@restrictionPattern)=0~~
~~ParameterType inheritance is not supported~~	~~count(//*/@baseType)=0~~
~~Only one Unit is supported in UnitSet~~	~~count(//xtce:UnitSet/xtce:Unit[position()=2])=0~~

Ignore Unit's power attribute only its default value is allowed	count(//xtce:Unit[@power>1])=0
Ignore Unit's factor attribute only its default value is allowed	count(//xtce:Unit[@factor>1])=0
BinaryDataEncoding is not supported for StringParameterType	count(//xtce:StringParameterType/xtce:BinaryDataEncoding)=0
FloatDataEncoding is not supported for StringParameterType	count(//xtce:StringParameterType/xtce:FloatDataEncoding)=0
IntegerDataEncoding is not supported for StringParameterType	count(//xtce:StringParameterType/xtce:IntegerDataEncoding)=0
Error detection and correction are not supported	count(//xtce:ErrorDetectCorrect)=0
A fixed length string based on a dynamic value look up is not supported	count(xtce:StringDataEncoding/xtce:SizeInBits/xtce:Fixed/xtce:DynamicValue)=0
An array dimension size starting index that is fixed must be non-negative	count(//xtce:SizeInBits/xtce:FixedValue[text() < 0])=0
StringParameterTypes based on strings with a certain terminating character are not supported	count(xtce:TerminationChar)=0
DiscreteLookupList is not supported	count(//xtce:DiscreteLookupList)=0
StringParameterType alarms are not supported	count(//xtce:StringParameterType/xtce:ContextAlarmList)=0
StringParameterType alarms are not supported	count(//xtce:StringParameterType/xtce:DefaultAlarm)=0
BinaryDataEncoding of EnumeratedParameterTypes is not supported	count(//xtce:EnumeratedParameterType/xtce:FloatDataEncoding)=0
BinaryDataEncoding of EnumeratedParameterTypes is not supported	count(//xtce:EnumeratedParameterType/xtce:BinaryDataEncoding)=0
Custom algorithm is not supported	count(//xtce:CustomAlgorithm)=0
Calibrators are not supported for EnumeratedParametersTypes	count(//xtce:EnumeratedParameterType/xtce:IntegerDataEncoding/xtce:DefaultCalibrator)=0
Linear calibrators from 2 points to 100 points are supported	not((for $x in //xtce:SplineCalibrator return count($x/xtce:SplinePoint)>1 and count($x/xtce:SplinePoint)<100)=false())
The @order attribute is a schema bug- only the default value is allowed and it should be ignored	count(//xtce:SplineCalibrator[@order!=1])=0
Polynomial calibrators from 1 to 10 terms are supported	not((for $x in //xtce:PolynomialCalibrator return count($x/xtce:Term)> 0 and count($x/xtce:Term) < 11)=false())
The order attribute in the SplinePoint element is a typo in the XTCE Schema and may not be specified	count(//xtce:SplinePoint/@order)=0
BinaryDataEncoding of EnumeratedParameterTypes is not supported	count(//xtce:EnumeratedParameterType/xtce:StringDataEncoding)=0
The math operation calibrator is not supported	count(//xtce:MathOperationCalibrator)=0
Distress alarm is not supported	count(//xtce:DistressAlarm)=0
Critical alarm is not supported	count(//xtce:CriticalAlarm)=0
Only three alarm levels are supported: normal warning and severe. Watch distress and critical are not supported.	count(//*[@defaultAlarmLevel='watch']|//*[@defaultAlarmLevel='distress']|//*[@defaultAlarmLevel='critical'])=0
Watch alarm is not supported	count(//xtce:WatchAlarm)=0
BinaryDataEncoding of IntegerParameterTypes is not supported	count(//xtce:IntegerParameterType/xtce:BinaryDataEncoding)=0
FloatDataEncoding of IntegerParameterTypes is not supported	count(//xtce:IntegerParameterType/xtce:FloatDataEncoding)=0
Custom alarm is not supported	count(//xtce:CustomAlarm)=0
Only three alarm levels are supported: normal warning and severe. Watch distress and critical are	count(//*[@alarmLevel='watch']|//*[@alarmLevel='distress']|//*[@alarmLevel='critical'])=0

not supported.	
Watch range is not supported	count(//xtce:WatchRange)=0
Distress range is not supported	count(//xtce:DistressRange)=0
StringDataEncoding of IntegerParameterTypes is not supported	count(//xtce:IntegerParameterType/xtce:StringDataEncoding)=0
ToString is not supported	count(//xtce:ToString)=0
ToBinaryTransformAlgorithm is not supported	count(//xtce:ToBinaryTransformAlgorithm)=0
FloatDataEncoding of BinaryParameterTypes is not supported	count(//xtce:BinaryParameterType/xtce:FloatDataEncoding)=0
Critical range is not supported	count(//xtce:CriticalRange)=0
FromBinaryTransformAlgorithm is not supported	count(//xtce:FromBinaryTransformAlgorithm)=0
FloatDataEncoding of MIL-1750A is not supported	count(//xtce:FloatDataEncoding[@encoding='MILSTD_1750A'])=0
BinaryDataEncoding of FloatParameterTypes is not supported	count(//xtce:FloatParameterType/xtce:BinaryDataEncoding)=0
StringDataEncoding of BinaryParameterTypes is not supported	count(//xtce:BinaryParameterType/xtce:StringDataEncoding)=0
IntegerDataEncoding of BinaryParameterTypes is not supported	count(//xtce:BinaryParameterType/xtce:IntegerDataEncoding)=0
Default value should be ignored	count(//xtce:Encoding/@scale)=0
Default value should be ignored	count(//xtce:Encoding/@units)=0
BooleanParameterType is not supported	count(//xtce:BooleanParameterType)=0
StringDataEncoding of FloatParameterTypes is not supported	count(//xtce:FloatParameterType/xtce:StringDataEncoding)=0
StringDataEncoding of RelativeTimeParameterType is not supported	count(//xtce:RelativeParameterType/xtce:Encoding/xtce:StringDataEncoding)=0
FloatDataEncoding of RelativeTimeParameterType is not supported	count(//xtce:RelativeParameterType/xtce:Encoding/xtce:FloatDataEncoding)=0
BinaryDataEncoding of RelativeTimeParameterType is not supported	count(//xtce:RelativeParameterType/xtce:Encoding/xtce:BinaryDataEncoding)=0
Default value should be ignored	count(//xtce:Encoding/@offset)=0
ParameterRef is not supported	count(//xtce:ParameterRef)=0
The ParameterType name must consist of the name of its Parameter + Type	not((for $x in //xtce:ParameterSet/*/@name return concat($x 'Type')=//xtce:ParameterTypeSet/*/@name)=false())
OffsetFrom is not supported	count(//xtce:OffsetFrom)=0
ParameterSegmentRefEntry is not supported	count(//xtce:ParameterSegmentRefEntry)=0
ContainerSegmentRefEntry is not supported	count(//xtce:ContainerSegmentRefEntry)=0
StreamSegmentEntry is not supported	count(//xtce:StreamSegmentEntry)=0
IndirectParameterRefEntry is not supported	count(//xtce:IndirectParameterRefEntry)=0
DefaultRateInStream is not supported	count(//xtce:DefaultRateInStream)=0
RateInStreamSet is not supported	count(//xtce:RateInStreamSet)=0
An offset may not be dynamic	count(//xtce:Offset/xtce:DynamicValue)=0
NextContainer is not supported	count(//xtce:NextContainer)=0
MessageSet is not supported	count(//xtce:MessageSet)=0
StreamSet is not supported	count(//xtce:StreamSet)=0
AlgorithmSet is not supported	count(//xtce:AlgorithmSet)=0
Attribute lastEntryForThisArrayInstance is not allowed	count(//@/@lastEntryForThisArrayInstance)=0
A starting index may not be dynamic	count(//xtce:StartingIndex/xtce:DynamicValue)=0
CCSDSTelmetryPacket must extend CCSDSPacket and mission packet bodies must extend CCSDSTelemetryPacket	count(//xtce:SequenceContainer/xtce:BaseContainer[@containerRef='CCSDSPacket']|//xtce:SequenceContainer/xtce:BaseContainer[@containerRef='CCSDSTelemetryPacket'])>1

RestrictionCriteria must have at least comparisons to CCSDSAPID	//xtce:RestrictionCriteria//@parameterRef='CCSDSAPID'
BinaryDataEncoding of EnumeratedArgumentTypes is not supported	count(//xtce:EnumeratedArgumentType/xtce:BinaryDataEncoding)=0
IntegerDataEncoding is not supported for StringArgumentType	count(//xtce:StringArgumentType/xtce:IntegerDataEncoding)=0
Calibrators are not supported for EnumeratedArgumentsTypes	count(//xtce:EnumeratedArgumentType/xtce:IntegerDataEncoding/xtce:DefaultCalibrator)=0
BinaryDataEncoding of EnumeratedArgumentTypes is not supported	count(//xtce:EnumeratedArgumentType/xtce:FloatDataEncoding)=0
Alarms are not supported in commanding	count(//xtce:CommandMetaData/*/xtce:ContextAlarmList
Alarms are not supported in commanding	count(//xtce:CommandMetaData/*/xtce:DefaultAlarm
FloatDataEncoding is not supported for StringArgumentType	count(//xtce:StringArgumentType/xtce:FloatDataEncoding)=0
BinaryDataEncoding is not supported for StringArgumentType	count(//xtce:StringArgumentType/xtce:BinaryDataEncoding)=0
IntegerDataEncoding of BinaryArgumentTypes is not supported	count(//xtce:BinaryArgumentType/xtce:IntegerDataEncoding)=0
FloatDataEncoding of BinaryArgumentTypes is not supported	count(//xtce:BinaryArgumentType/xtce:FloatDataEncoding)=0
BinaryDataEncoding of FloatArgumentTypes is not supported	count(//xtce:FloatArgumentType/xtce:BinaryDataEncoding)=0
StringDataEncoding of BinaryArgumentTypes is not supported	count(//xtce:BinaryArgumentType/xtce:StringDataEncoding)=0
BinaryDataEncoding of IntegerArgumentTypes is not supported	count(//xtce:IntegerArgumentType/xtce:BinaryDataEncoding)=0
BinaryDataEncoding of EnumeratedArgumentTypes is not supported	count(//xtce:EnumeratedArgumentType/xtce:StringDataEncoding)=0
StringDataEncoding of IntegerArgumentTypes is not supported	count(//xtce:IntegerArgumentType/xtce:StringDataEncoding)=0
FloatDataEncoding of IntegerArgumentTypes is not supported	count(//xtce:IntegerArgumentType/xtce:FloatDataEncoding)=0
There should be one CCSDSPacket in BaseContainer of MetaCommand/CommandContainers	count(//xtce:MetaCommand//xtce:CommandContainer/xtce:BaseContainer[@containerRef='CCSDSPacket'])=1
TransferredToRangeVerifier is not supported	count(//xtce:TransferredToRangeVerifier)=0
StringDataEncoding of AbsoluteTimeArgumentType is not supported	count(//xtce:RelativeArgumentType/xtce:Encoding/xtce:StringDataEncoding)=0
BinaryDataEncoding of AbsoluteTimeArgumentType is not supported	count(//xtce:RelativeArgumentType/xtce:Encoding/xtce:BinaryDataEncoding)=0
FloatDataEncoding of AbsoluteTimeArgumentType is not supported	count(//xtce:RelativeArgumentType/xtce:Encoding/xtce:FloatDataEncoding)=0
StringDataEncoding of FloatArgumentTypes is not supported	count(//xtce:FloatArgumentType/xtce:StringDataEncoding)=0
BooleanArgumentType is not supported	count(//xtce:BooleanArgumentType)=0
FailedVerifier is not supported	count(//xtce:FailedVerifier)=0
MetaCommandRef is not supported	count(//xtce:MetaCommandRef)=0
ExecutionVerifier is not supported	count(//xtce:ExecutionVerifier)=0
CheckWindowAlgorithms is not supported	count(//xtce:CheckWindowAlgorithms)=0
AcceptedVerifier is not supported	count(//xtce:AcceptedVerifier)=0
QueuedVerifier is not supported	count(//xtce:QueuedVerifier)=0
SentFromRangeVerifier is not supported	count(//xtce:SentFromRangeVerifier)=0
ReceivedVerifier is not supported	count(//xtce:ReceivedVerifier)=0

~~ServiceSet is not supported~~	~~count(//xtce:ServiceSet)=0~~

6.2 Additional Rules

The following section describes additional rules that are not captured completely in the rules table for various reasons. These rules are normative.

6.2.1 Virtual Channel Identifiers

Virtual channel identifiers (VCIDs) associated with each packet, telemetry or command shall be held in an AncillaryData element with each packet container using the name "VCID". One or more VCIDs values may be specified in a comma-delimited list of values in element container. A value may be from 0 to 63, and a range of VCID values may be specified using a "#-#" pattern.

For example the following specifies VCIDs as 0, 8, 9, 10, 11, 12 and 20.

```
<xtce:SequenceContainer name="MyPacket">
    <xtce:AncillaryDataSet>
        <xtce:AncillaryData name="VCID">0, 20, 8-12</xtce:AncillaryData>
    </xtce:AncillaryDataSet>
    <xtce:EntryList>
        <xtce:ParameterRefEntry parameterRef="TimeStamp"/>
        <xtce:ParameterRefEntry parameterRef="NumImagers"/>
    </xtce:EntryList>
</xtce:SequenceContainer>
```

6.2.2 Telemetry Packet Pattern

The telemetry packet pattern describes several container constructs in TelemetryMetaData for consistently defining a CCSDS format based mission packet. Two of the container constructs are fixed and supplied with each GovSat XTCE file, CSDSPacket and CCSDSTelemetryPacket. All packet definitions refer to these items through XTCE's container inheritance mechanism.

- the root CCSDSPacket container is an abstract container describing the CCSDS header.
- the common CCSDSTelemetryPacket container extends CCSDSPacket
- each mission specific packet body container extends CCSDSTelemetryPacket

6.2.2.1 Root CCSDS Packet Container

The purpose of the CCSDSPacket container is to supply all the fields for the CCSDS primary header in a single container construction.

6.2.2.2 Common CCSDSTelemetryPacket Container

It supplies two constraints: CCSDSType and CCSDSVersion.

It has no EntryList, supplying no further information to the container hierarchy and final parameter list.

6.2.2.3 Mission Specific Packet Body Container

Each mission telemetry packet container extends the CCCSDTelemetryPacket container; each supplies up to four identifying fields in the RestrictionCriteria to uniquely identify the description. These containers shall not be abstract. The identifying fields would typically consist of at least APID for that packet and up to three other conditions for other identifying fields if they are present in the packet, such as any additional secondary headers. For some organizations this will not be the case and the APID will be sufficient by itself.

The full pattern is as follows.

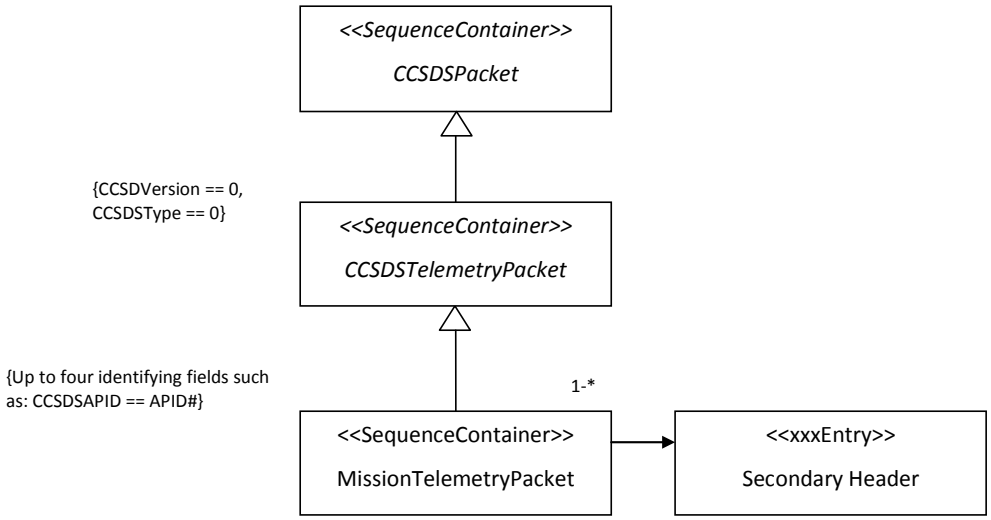

Figure 1 – CCSDS Telemetry Packet Container Pattern

6.2.2.3.1 Secondary Header

Often mission telemetry packet will include a secondary header. For many missions the secondary header will be a time stamp that would simply be a ParameterRefEntry to a Parameter that is AbsoluteTimeParameterType.

Some missions may define more complex structures for secondary headers that may be defined inline or as another container that is included in the packet using a ContainerRefEntry.

6.2.2.3.2 Telemetry Packet Identifying Keys

When specifying the RestrictionCriteria of the MissionTelemetryPacket the Comparison or ComparisonList element shall be used:

- Comparison: for convenience, a single Comparison element if this is the only item needed to uniquely identify the packet, otherwise use a ComparisonList.
- ComparisonList: Some mission formats may require more identifying fields than just the APID field in the CCSDS primary header, use the ComparisonList to specify one or more of them.

The total number of indentifying fields that maybe defined is four.

6.2.2.4 Telemetry Packet Body

Additional explanatory information is provided for these EntryList items.

- ArrayParameterRefEntry
  - Only 1D and 2D arrays are supported
- Entry Modifiers
  - IncludeCondition

- RepeatEntry
- LocationInContainerInBits

Various aspects of these items have further restrictions.

6.2.2.4.1 ArrayParameterRefEntry

The dimension sizes are set here by using the child element DimensionList, while the number of dimensions (1-D and 2-D in GovSat) is set in the ArrayParameterType. The size of the dimensions may be fixed, or dynamic.

6.2.2.4.2 IncludeCondition

Only Comparison and ComparisonList are supported.

6.2.2.4.3 Repeat

Only the dynamic and fixed Count form are supported.

The Repeat element has a child element called Offset, this is disallowed in GovSat.

6.2.2.4.4 LocationInContainerInBits

Only absolute addressing (ContainerStart) and relative addressing (PreviousEntry are supported (the default is PreviousEntry if not explicitly specified). In addition only the FixedValue and DynamicValue forms are supported.

6.2.3 Command and Command Packet Pattern

XTCE Commands and packet descriptions are in MetaCommand and its CommandContainer. The pattern is similar to the telemetry pattern, except the packet related containers are inside the MetaCommand.

- the root CCDSCommand has a root CCSDSCommandPacket.
- the CCSDSCommandPacket extends CCSDSPacket which sets restrictions for Version and Type.
- Each MissionCommand extends the CCSDSCommand providing any specific command arguments. Its

MissionCommandPacket extends CCSDSCommand/CCSDSCommandPacket.

A restriction for APID is provided, missions may wish to incorporate an opcode in the restrictions or use FixedValueEntry to add an opcode.

VCID(s) and packet length are handled in a similar manner to telemetry packet descriptions, the VCIDs are held in AncillaryData and the packet length can be calculated from the construction if it is needed.

MissionCommands (and their packet containers) may be extended by other mission commands in certain cases of derived command.

In a derived command, a base command is extended one or more times and certain arguments are fixed to provide certain behaviors.

The base mission command here may need to be marked as abstract if it itself is not a command that will ever be sent itself.

The derived command is shown in “dashed lines” to signify that it may not be used on every command or even any mission commands.

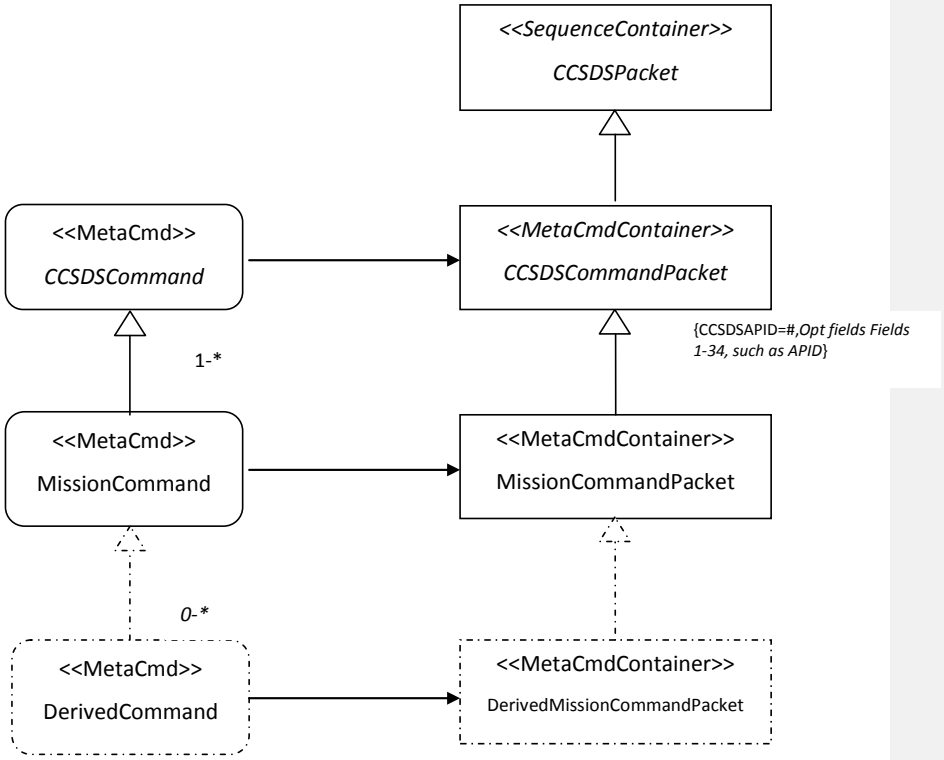

Figure 2 – Mission Commands and Packets

6.2.3.1 Command Packet Identifying Keys

6.2.3.1.1 One Identifying Key

Similar to telemetry, if there is only one parameter to check such as the APID, use a comparison. Put VCIDs in the AncillaryData of CommandContainer.

Note, VCIDs are cumulative in command inheritance. If a derived command could travel over other VCIDs, these specific VCIDs only have to be added in derived command CommandContainer. Also, they will be combined with any specified in its MissionCommand's AncillaryData.

There is no way to remove VCIDs through inheritance. If a derived command is more restrictive in this area, there isn't a way to remove any VCIDs it cannot use. If that's the case, do not put any VCIDs in the base MissionCommand and let each derived command specify the VCIDs that are legal for it. This works in the case where the MissionCommand is abstract.

If the base MissionCommand is also a valid command and its derived command needs to further restrict its VCIDs, then the only approach is to define each command separately. Replicate the CommandContainer in each independent command.

6.2.3.1.2 Multiple Identifying Keys

Similar to the telemetry packet pattern, more than one key can be used to identify the command packet. Up to four total comparisons can be defined using the ComparisonList element.

6.2.3.2 Derived Commands

A derived command occurs when a base command is defined and specific commands are defined from it. Although these are separate commands, they all share the same packet (and hence CommandContainer).

Usually the derived command sets specific argument values in the more general base command to enforce certain behaviors in it.

For example, suppose a general purpose "SetRelays" command allows one to specify the state of several onboard relays (on, off, or no change). Then specific derived commands could be created to turn all the relays on, off.

In GovSat deriving MetaCommands from the MissionCommand (e.g. SetRelays) will be used for this purpose. It should be noted, the MissionCommand may be set to abstract if it should itself not be sent by the user.

In the following example, relay "state" is defined using an EnumeratedArgumentType. A MissionCommand is defined called SetRelay which allows the state of four relays to be set. From that, two derived commands are created. One that sets all relays to on and the other sets all relays to off.

```
<xtce:ArgumentTypeSet>
  <xtce:EnumeratedArgumentType name="RelayStateType">
   <xtce:LongDescription>
     The relay is either on, off or unchanged from its current state
   </xtce:LongDescription>
   <xtce:UnitSet/>
   <xtce:IntegerDataEncoding/>
   <xtce:EnumerationList>
    <xtce:Enumeration value="0" label="Off"/>
    <xtce:Enumeration value="1" label="On"/>
    <xtce:Enumeration value="2" label="NoChange"/>
  </xtce:EnumerationList>
  </xtce:EnumeratedArgumentType>
</xtce:ArgumentTypeSet>

The MetaCommand is at the MissionCommand level of the command pattern:
<xtce:MetaCommand abstract="true" name="SetRelays">
  <xtce:LongDescription>
    Set any relay, this command is not directly available to the end user
  </xtce:LongDescription>
  <xtce:BaseMetaCommand metaCommandRef="CCSDSCommand"/>
  <xtce:ArgumentList>
    <xtce:Argument name="Relay1State" argumentTypeRef="RelayStateType"/>
    <xtce:Argument name="Relay2State" argumentTypeRef="RelayStateType"/>
    <xtce:Argument name="Relay3State" argumentTypeRef="RelayStateType"/>
    <xtce:Argument name="Relay4State" argumentTypeRef="RelayStateType"/>
  </xtce:ArgumentList>
  <xtce:CommandContainer name="SetRelayPacket"
              shortDescription="Turn a relay off or on, or unchanged">
    <xtce:AncillaryDataSet>
      <xtce:AncillaryData name="VCID">0</xtce:AncillaryData>
    </xtce:AncillaryDataSet>
    <xtce:EntryList>
```

Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Comment [blk7]: 18371
Formatted ...
Formatted: Font: Not Bold
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...

```
        <xtce:ArgumentRefEntry argumentRef="Relay1State"/>
        <xtce:ArgumentRefEntry argumentRef="Relay2State"/>
        <xtce:ArgumentRefEntry argumentRef="Relay3State"/>
        <xtce:ArgumentRefEntry argumentRef="Relay4State"/>
      </xtce:EntryList>
      <xtce:BaseContainer containerRef="CCSDSCommandPacket">
        <xtce:RestrictionCriteria>
          <xtce:Comparison value="1" parameterRef="CCSDSAPID"/>
        </xtce:RestrictionCriteria>
      </xtce:BaseContainer>
    </xtce:CommandContainer>
</xtce:MetaCommand>
```

Comment [blk8]: 18371

Note that the SetRelays command is set to abstract, this means the user cannot explicitly issue one of these commands directly.

The following example shows the derived command that sets all relays on.

```
<xtce:MetaCommand abstract="false" name="SetAllRelaysOn">
  <xtce:LongDescription>Set all relays to on</xtce:LongDescription>
  <xtce:BaseMetaCommand metaCommandRef="SetRelays">
    <xtce:ArgumentAssignmentList>
        <xtce:ArgumentAssignment argumentName="Relay1State" argumentValue="On"/>
        <xtce:ArgumentAssignment argumentName="Relay2State" argumentValue="On"/>
        <xtce:ArgumentAssignment argumentName="Relay3State" argumentValue="On"/>
        <xtce:ArgumentAssignment argumentName="Relay4State" argumentValue="On"/>
      </xtce:ArgumentAssignmentList>
    </xtce:BaseMetaCommand>
    <xtce:CommandContainer name="SetAllRelaysOnPacket"
          shortDescription="Name a relay and turn it on">
      <xtce:EntryList/>
      <xtce:BaseContainer containerRef="SetRelayPacket"/>
    </xtce:CommandContainer>
</xtce:MetaCommand>
```

Comment [blk9]: 18371

This example shows the derived command that sets all relays off.

```
<xtce:MetaCommand abstract="false" name="SetAllRelaysOff">
  <xtce:LongDescription>Set all relays to off</xtce:LongDescription>
  <xtce:BaseMetaCommand metaCommandRef="SetRelays">
    <xtce:ArgumentAssignmentList>
       <xtce:ArgumentAssignment argumentName="Relay1State" argumentValue="Off"/>
       <xtce:ArgumentAssignment argumentName="Relay2State" argumentValue="Off"/>
       <xtce:ArgumentAssignment argumentName="Relay3State" argumentValue="Off"/>
       <xtce:ArgumentAssignment argumentName="Relay4State" argumentValue="Off"/>
    </xtce:ArgumentAssignmentList>
  </xtce:BaseMetaCommand>
  <xtce:CommandContainer name="SetAllRelaysOffPacket"
         shortDescription="Name a relay and turn it off">
    <xtce:EntryList/>
    <xtce:BaseContainer containerRef="SetRelayPacket"/>
  </xtce:CommandContainer>
```

```
</xtce:MetaCommand>
```

Comment [blk10]: 18371

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

In both cases, the constructions use the ArgumentAssignment element to assign arguments in the SetRelays explicit values.

In addition, the EntryList in each is empty. This is correct for GovSat since the original packet is defined in the SetRelays command and the packet format cannot change for the derived commands. Note, XTCE does not enforce this; additional entries could be supplied here if needed. The GovSat implementer must check that Entrylist in the derived commands is empty.

This means the SetAllRelaysOffPacket and SetAllRelaysOnPacket are identical to SetRelayPackets in terms of format and APID.

6.2.3.3 Additional Command Features

MetaCommand has additional elements related to commanding. The VerifierSet/CommandComplete, and VerifierSet/FailedVerifier are supported by GovSat.

6.2.3.4 Command Significance

Command significance marks a command with one of three levels. The DefaultSignificance is used to map the command to: none, critical and severe

- None – no restrictions
- Critical – requires confirmation
- Severe – the command will not be sent

If unspecified, “no restrictions” for the command is used. Since all attributes are optional, it is possible to have an empty element in this location that is also interpreted as having no restrictions.

The ContextSignificanceList element is used to define significances based on context. A context is mission defined and based on comparisons (expressions). Mission phase or operating modes are examples. They would be created as session variables with enumerations. The variables are then placed in comparisons and evaluated to determine the desired meaning.

For example, a command marked as critical (not allowed to send) during integration and test, unless during thermal vacuum activities.

```
<xtce:MetaCommand name="ThemalControlCmd">
  <xtce:ContextSignificanceList>
    <xtce:ContextSignificance>
        <xtce:ContextMatch>
          <xtce:Comparison parameterRef="SysPhase"
                   value="IntegrationTestInThermVac" comparisonOperator="=="/>
          </xtce:ContextMatch>
        <xtce:Significance consequenceLevel="critical"/>
    </xtce:ContextSignificance>
  </xtce:ContextSignificanceList>
</xtce:MetaCommand>
```

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Comment [blk11]: 18371

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

6.2.3.5 Command Complete and Failed

Command complete checks and failures are support in the VerifierSet/CommandVerifier and VerifierSet/FailedVerifier.

Due to the wide variety of mission specific issues in this area, nothing specific to GovSat is further specified in the rules.

The following example script fragment shows aspects of a command complete and failure check. SBCxxx are telemetry parameters and the tc_xxx and exp_tc are represented in XTCE as system variables. They are counters. PKT0015 is the command response packet.

```
;  Issue command
/SBCNOOP
.
;  Wait for a s/c response
WAIT UNTIL (GBL_PKTCNT_0015 .GT. (pkt15count+1))
WAIT UNTIL ((SBCTCERRORS .NE. tc_err) .OR. (SBCTCRECVD .NE. tc_rcv) .OR. (SBCTCREJECT .NE. tc_rej))
;
;  Check for expected response
IF ((SBCTCERRORS .NE. exp_tc_err) .OR. (SBCTCRECVD .NE. exp_tc_rcv)  .OR. (SBCTCREJECT .NE. exp_tc_rej)) THEN
    ASK (CONCAT("Unexpected command counter response!\n\n",;;
            "Tlm Mnemonic   Expected  Actual  \n",;;
            " SBCTCRECVD       ",exp_tc_rcv,"        ",SBCTCRECVD,"\n",;;
            " SBCTCREJECT      ",exp_tc_rej,"        ",SBCTCREJECT,"\n",;;
            " SBCTCERRORS      ",exp_tc_err,"        ",SBCTCERRORS))
```

The XTCE representation of the above operations script follows.

```
<xtce:MetaCommand name="SBCNOOP">
    < xtce:VerifierSet>
        < xtce:CompleteVerifier shortDescription="WAIT UNTIL PKT015 shows up">
            < xtce:ContainerRef containerRef="PKT015"/>
            < xtce:CheckWindow timeToStopChecking="PT1M" timeToStartChecking="PT10M"/>
        </ xtce:CompleteVerifier>
        < xtce:CompleteVerifier shortDescription="WAIT UNTIL COUNTS are updated">
            < xtce:BooleanExpression>
                < xtce:ORedConditions>
                    < xtce:Condition>
                        < xtce:ParameterInstanceRef parameterRef="SBCTCERRORS "/>
                        < xtce:ComparisonOperator>!=</ComparisonOperator>
                        < xtce:ParameterInstanceRef parameterRef="tc_err"/>
                    </ xtce:Condition>
                    < xtce:Condition>
                        < xtce:ParameterInstanceRef parameterRef="SBCTCRECVD"/>
                        < xtce:ComparisonOperator>!=</ComparisonOperator>
                        < xtce:ParameterInstanceRef parameterRef="tc_rcv"/>
                    </ xtce:Condition>
                    < xtce:Condition>
                        < xtce:ParameterInstanceRef parameterRef="SBCTCREJECT"/>
                        < xtce:ComparisonOperator>!=</ComparisonOperator>
                        < xtce:ParameterInstanceRef parameterRef=" tc_rej"/>
                    </ xtce:Condition>
                </ xtce:ORedConditions>
            </ xtce:BooleanExpression>
            < xtce:CheckWindow timeToStartChecking="PT1M" timeToStopChecking="PT10M"/>
        </ xtce:CompleteVerifier>
        < xtce:FailedVerifier shortDescription="Check the expected counts... ">
            < xtce:BooleanExpression>
                < xtce:ORedConditions>
                    < xtce:Condition>
```

Formatted ... (×50)

Comment [blk12]: 18371

```
< xtce:ParameterInstanceRef parameterRef="SBCTCERRORS "/>
< xtce:ComparisonOperator>!=</ComparisonOperator>
< xtce:ParameterInstanceRef parameterRef="exp_tc_err"/>
</ xtce:Condition>
< xtce:Condition>
< xtce:ParameterInstanceRef parameterRef="SBCTCRECVD"/>
< xtce:ComparisonOperator>!=</ComparisonOperator>
< xtce:ParameterInstanceRef parameterRef="exp_tc_rcv"/>
</ xtce:Condition>
< xtce:Condition>
< xtce:ParameterInstanceRef parameterRef="SBCTCREJECT"/>
< xtce:ComparisonOperator>!=</ComparisonOperator>
< xtce:ParameterInstanceRef parameterRef=" exp_tc_rej"/>
</ xtce:Condition>
</ xtce:ORedConditions>
</ xtce:BooleanExpression>
< xtce:CheckWindow timeToStartChecking="PT1M" timeToStopChecking="PT10M"/>
</ xtce:FailedVerifier>
</ xtce:VerifierSet>
</ xtce:MetaCommand>
```

Formatted ... (×40)

Comment [blk13]: 18371

Formatted ... (×2)

6.2.3.6 Command Packet Body

The following EntryList items are added for command side only and defined in the MetaCommand/CommandContainer element.

- ArgumentRefEntry
- ArrayArgumentRefEntry
  - Only 1D and 2D arrays are supported
- FixedValueEntry

In a few places, the attribute parameterRef appears when the intent might be for an argumentRef. This is true for the Repeat and IncludeCondition elements.

For ArgumentRefEntry and ArrayArgumentRefEntry, any @parameterRef should be interpreted as @argumentRef.

For the Repeat element if DynamicValue it used, assume it refers to an Argument.

Referring to either a parameter or argument is perfectly valid depending on use case. Unfortunately, the schema does not at this time give a clear way to indicate which reference is desired.

6.3 Template

An XTCE document template is provided as a machine-consumable file, http://www.omg.org/spec/XUSP/20140801/XUSPTemplate.xtce. The template is normative and forms the basis for any XUSP document. Creating the mission-specific XTCE definition file involves setting the values for the template elements/attributes described in the following table and inserting new, valid XTCE elements marked as "Supported" in the rules table. Other than the tailoring values described in the table below, all elements, attributes, and values in the normative template must be present in the XTCE document for the document to be XUSP-compliant. New SequenceContainer elements describing a CCSDS telemetry packet must use the CCSDSTelemetryPacket SequenceContainer as a BaseContainer. New CCSDS command packets must be described by creating new CommandContainer elements using the CCSDSCommandPacket as a BaseContainer or by creating new MetaCommand elements using the CCSDSCommand MetaCommand as a BaseMetaCommand.

Comment [blk14]: 19594

XTCE Element or Element@attribute	Mission Unique Tailoring Description
/SpaceSystem@name	Provide mission name up to 64 characters long.
/SpaceSystem@shortDescription	Provide a mission description up to 128 characters long.
/SpaceSystem/AliasSet/Alias@alias	For the Alias SpacecraftID namespace, the numeric ID for the mission must be.specified.
/SpaceSystem/Header@version	Supply the version string identifying the XTCE document version for the mission
/SpaceSystem/Header@validationStatus	Specify the ValidationStatusType enumeration value that best describes the XTCE document status
/SpaceSystem/Header@date	Specify the date the XTCE document was created or last modified.
/SpaceSystem/Header@classification	Specify any required special handling or sensitivity of the XTCE document.

~~The following template is normative and forms the basis for any GovSat file. Each user based on their mission must change some items. This includes items such as: the SpaceSystem elements name, the SpacecraftID and the items in the Header element.~~

```
<?xml version="1.0" encoding="UTF-8"?>
<xtce:SpaceSystem xmlns:xtce="http://www.omg.org/space/xtce" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
name="GovSat" shortDescription="The GovSat template" xsi:schemaLocation="http://www.omg.org/space/xtce
SpaceSystemV1.1.xsd">
    <xtce:AliasSet>
        <xtce:Alias nameSpace="SpacecraftID" alias="123"/>
    </xtce:AliasSet>
    <xtce:Header version="1.1" validationStatus="Draft" date="2011-02-28" classification="GovSat1.1.0"/>
    <xtce:TelemetryMetaData>
        <xtce:ParameterTypeSet>
            <xtce:EnumeratedParameterType name="MissionType" shortDescription="Reserved Mission Phase
and Operating Modes">
                <xtce:UnitSet/>
                <xtce:EnumerationList>
                    <xtce:Enumeration value="0" label="TestBed"/>
                    <xtce:Enumeration value="1" label="IntegrationTest"/>
                    <xtce:Enumeration value="2" label="Vibration"/>
                    <xtce:Enumeration value="3" label="ThermalVac"/>
                    <xtce:Enumeration value="4" label="EMC"/>
                    <xtce:Enumeration value="5" label="Platform"/>
                    <xtce:Enumeration value="6" label="Launch"/>
                    <xtce:Enumeration value="7" label="Orbit"/>
                    <xtce:Enumeration value="8" label="Checkout"/>
                    <xtce:Enumeration value="9" label="Mission"/>
                    <xtce:Enumeration value="10" label="Safe"/>
                </xtce:EnumerationList>
            </xtce:EnumeratedParameterType>
            <xtce:IntegerParameterType signed="false" name="CCSDSSCIDType">
                <xtce:UnitSet/>
```

```
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSVCIDType">
                        <xtce:UnitSet/>
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSVersionType">
                        <xtce:UnitSet/>
                        <xtce:IntegerDataEncoding sizeInBits="3"/>
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSTypeType">
                        <xtce:UnitSet/>
                        <xtce:IntegerDataEncoding sizeInBits="1"/>
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSSecHType">
                        <xtce:UnitSet/>
                        <xtce:IntegerDataEncoding sizeInBits="1"/>
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSAPIDType">
                        <xtce:UnitSet/>
                        <xtce:IntegerDataEncoding sizeInBits="11"/>
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSGroupFlagsType">
                        <xtce:UnitSet/>
                        <xtce:IntegerDataEncoding sizeInBits="2"/>
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSSourceSequenceCountType">
                        <xtce:UnitSet/>
                        <xtce:IntegerDataEncoding sizeInBits="14"/>
                    </xtce:IntegerParameterType>
                    <xtce:IntegerParameterType signed="false" name="CCSDSPacketLengthType">
                        <xtce:UnitSet/>
                        <xtce:IntegerDataEncoding sizeInBits="16"/>
                    </xtce:IntegerParameterType>
                </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="MissionType" name="Mission">
                        <xtce:ParameterProperties dataSource="local"/>
                    </xtce:Parameter>
                </xtce:ParameterSet>
                <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:ContainerSet>
        </xtce:TelemetryMetaData>
        <xtce:CommandMetaData>
            <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:MetaCommandSet>
        </xtce:CommandMetaData>
    </xtce:SpaceSystem>
```

Comment [blk15]: 18372