Telemetry & Telecommand Data Specification

This OMG document replaces the submission document (space/03-03-01) and the Draft Adopted Specification (dtc/03-05-07). It is an OMG Final Adopted Specification and is currently in the finalization phase. Comments on the content of this document are welcomed, and should be directed to *issues@omg.org* by August 31, 2003.

You may view the pending issues for this specification from the OMG revision issues web page *http://www.omg.org/issues/*; however, at the time of this writing there were no pending issues.

The FTF Recommendation and Report for this specification will be published on November 30, 2003. If you are reading this after that date, please download the available specification from the OMG Specifications Catalog.

OMG Adopted Specification
dtc/03-09-02

Telemetry and Telecommand Data Specification

September 2003
Final Adopted Specification
dtc/03-09-02

An Adopted Specification of the Object Management Group, Inc.

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, or mechanical, 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, 250 First Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

The OMG Object Management Group Logo®, CORBA®, CORBA Academy®, The Information Brokerage®, XMI® and IIOP® are registered trademarks of the Object Management Group. OMG™, Object Management Group™, CORBA logos™, OMG Interface Definition Language (IDL)™, The Architecture of Choice for a Changing World™, CORBAservices™, CORBAfacilities™, CORBAmed™, CORBAnet™, Integrate 2002™, Middleware That's Everywhere™, UML™, Unified Modeling Language™, The UML Cube logo™, MOF™, CWM™, The CWM Logo™, Model Driven Architecture™, Model Driven Architecture Logos™, MDA™, OMG Model Driven Architecture™, OMG MDA™ and the XMI Logo™ 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.

ISSUE REPORTING

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 & Specifications, Report a Bug/Issue.

Contents

1. Overview 1-1

1.1 Background 1-1

1.2 Introduction 1-2

1.3 Scope 1-3

1.4 Applicable and Reference Documents 1-4

1.4.1 Applicable Documents 1-4

1.4.2 Reference Documents 1-5

1.5 Compliance Points 1-5

1.6 Glossary of Acronyms 1-9

2. Telemetry and Telecommand Data Specification 2-1

2.1 Specification 2-1

2.1.1 The Root Object - The SpaceSystem 2-3

2.1.2 The Header Record 2-4

2.1.3 Parameters 2-4

2.1.4 Command Definitions 2-4

2.1.5 Algorithm 2-5

2.1.6 Stream 2-5

2.1.7 Packaging 2-5

2.2 Processing the Telemetry Stream 2-5

2.2.1 Typical Delivery Mechanisms for Telemetry Streams . . . 2-5

2.2.2 Telecommanding 2-11

3. Schema Introduction and Usage 3-1

3.1 Overview 3-1

3.2 Complete Schema Definition 3-2

3.2.1 Schema SpaceSystem.xsd 3-2
3.2.2 element SpaceSystem 3-6
3.2.3 complexType SpaceSystemType 3-7
3.2.4 element SpaceSystemType/Header 3-8
3.2.5 element SpaceSystemType/LongDescription 3-8
3.2.6 element SpaceSystemType/TelemetryAndCommanding . 3-9
3.2.7 element SpaceSystemType/TelemetryAnd Commanding/DefaultBusAttributes 3-10
3.2.8 element SpaceSystemType/TelemetryAndCommanding/ DefaultCommandDefinition 3-10
3.2.9 element SpaceSystemType/TelemetryAndCommanding/ CommandDefinitionList 3-12
3.2.10 element SpaceSystemType/TelemetryAndCommanding/ CommandDefinitionList/ArgumentDefinition 3-13
3.2.11 element SpaceSystemType/TelemetryAndCommanding/ CommandDefinitionList/CommandDefinition 3-14
3.2.12 element SpaceSystemType/TelemetryAndCommanding/ ParameterList 3-15
3.2.13 element SpaceSystemType/TelemetryAndCommanding/ ParameterList/Parameter 3-16
3.2.14 element SpaceSystemType/TelemetryAndCommanding/ PropertyList 3-17
3.2.15 element SpaceSystemType/TelemetryAndCommanding/ PropertyList/Property 3-18
3.2.16 element SpaceSystemType/TelemetryAndCommanding/ ServiceList 3-18
3.2.17 element SpaceSystemType/TelemetryAndCommanding/ ServiceList/Service 3-19
3.2.18 element SpaceSystemType/TelemetryAndCommanding/ AlgorithmList 3-20
3.2.19 element SpaceSystemType/TelemetryAndCommanding/ AlgorithmList/CustomAlgorithm 3-20
3.2.20 element SpaceSystemType/TelemetryAndCommanding/ AlgorithmList/ConcatenationAlgorithm 3-21
3.2.21 element SpaceSystemType/TelemetryAndCommanding/ AlgorithmList/MathAlgorithm 3-22
3.2.22 element SpaceSystemType/TelemetryAndCommanding/ AlgorithmList/ReedSolomonEncoder 3-23
3.2.23 element SpaceSystemType/TelemetryAndCommanding/ AlgorithmList/ReedSolomonDecoder. 3-23
3.2.24 element SpaceSystemType/TelemetryAndCommanding/ StreamList 3-24

3.2.25 element SpaceSystemType/TelemetryAndCommanding/StreamList/Stream 3-24
3.2.26 element SpaceSystemType/TelemetryAndCommanding/NonStandardData 3-25
3.2.27 element SpaceSystemType/NonStandardData 3-25
3.2.28 element tc:FixedFrameSync 3-26
3.2.29 element tc:FixedFrameSync/InputParameterRef. 3-26
3.2.30 element tc:FixedFrameSync/OutputParameterRef 3-27
3.2.31 element tc:FixedFrameSync/FixedFrameStream 3-27
3.2.32 complexType tc:CalibratorType 3-28
3.2.33 element tc:CalibratorType/SplineCalibrator 3-29
3.2.34 element tc:CalibratorType/SplineCalibrator/SplinePoint . 3-29
3.2.35 element tc:CalibratorType/PolynomialCalibrator 3-30
3.2.36 element tc:CalibratorType/ToStringCalibrator 3-30
3.2.37 complexType tc:ConcatenationAlgorithmType 3-31
3.2.38 element tc:ConcatenationAlgorithmType/InputParameterList 3-31
3.2.39 element tc:ConcatenationAlgorithmType/InputParameterList/InputParameterRef 3-32
3.2.40 element tc:ConcatenationAlgorithmType/OutputParameterRef 3-32
3.2.41 complexType tc:CustomAlgorithmType 3-33
3.2.42 element tc:CustomAlgorithmType/LongDescription 3-34
3.2.43 element tc:CustomAlgorithmType/Trigger 3-35
3.2.44 element tc:CustomAlgorithmType/Inputs 3-36
3.2.45 element tc:CustomAlgorithmType/Inputs/ParameterRef . 3-36
3.2.46 element tc:CustomAlgorithmType/Inputs/Constant 3-37
3.2.47 element tc:CustomAlgorithmType/OutputParameterRef . 3-38
3.2.48 element tc:CustomAlgorithmType/AlgorithmText 3-38
3.2.49 element tc:CustomAlgorithmType/ExternalAlgorithm . . 3-39
3.2.50 complexType tc:MathAlgorithmType 3-40
3.2.51 element tc:MathAlgorithmType/OutputParameterRef . . . 3-41
3.2.52 element tc:MathAlgorithmType/Trigger 3-41
3.2.53 complexType tc:ReedSolomonType 3-42
3.2.54 element tc:ReedSolomonType/InputParameterRef 3-43
3.2.55 element tc:ReedSolomonType/OutputParameterRef 3-43
3.2.56 element tc:ReedSolomonType/GeneratorPolynomial 3-44
3.2.57 complexType tc:TriggerType 3-44
3.2.58 element tc:TriggerType/ParameterRef 3-45
3.2.59 element tc:TriggerType/Time 3-45
3.2.60 element tc:DwellSet 3-46

3.2.61 complexType tc:ComplexAlarmsType 3-46
3.2.62 element tc:ComplexAlarmsType/StaticRanges 3-47
3.2.63 element tc:ComplexAlarmsType/RateOfChangeRanges . 3-48
3.2.64 element tc:ComplexAlarmsType/States 3-48
3.2.65 element tc:ComplexAlarmsType/States/UseRangeWhen . 3-49
3.2.66 element tc:ComplexAlarmsType/States/UseRangeWhen/EqualityCheck 3-49
3.2.67 element tc:ComplexAlarmsType/States/UseRangeWhen/CustomCheck 3-50
3.2.68 element tc:ComplexAlarmsType/States/State 3-50
3.2.69 element tc:ComplexAlarmsType/CustomAlarmCheck . . 3-51
3.2.70 complexType tc:OffsetParameterRefType 3-52
3.2.71 element tc:OffsetParameterRefType/OffsetInBits. 3-53
3.2.72 complexType tc:ParameterRefMatchListType 3-53
3.2.73 element tc:ParameterRefMatchListType/ParameterMatch 3-54
3.2.74 complexType tc:ParameterRefType 3-55
3.2.75 element tc:ParameterRefType/Occurs 3-56
3.2.76 element tc:ParameterRefType/DependantOn 3-56
3.2.77 complexType tc:ParameterType 3-57
3.2.78 element tc:ParameterType/Alias 3-58
3.2.79 element tc:ParameterType/SystemName. 3-59
3.2.80 element tc:ParameterType/ToString 3-59
3.2.81 element tc:ParameterType/Validity 3-60
3.2.82 element tc:ParameterType/AlarmConditions 3-61
3.2.83 element tc:ParameterType/Alarms 3-61
3.2.84 element tc:ParameterType/PhysicalAddress 3-62
3.2.85 element tc:ParameterType/NonStandardData 3-62
3.2.86 complexType tc:PhysicalAddressType 3-63
3.2.87 element tc:PhysicalAddressType/SubAddress 3-63
3.2.88 complexType tc:RangeListType 3-64
3.2.89 element tc:RangeListType/UseRangeWhen 3-64
3.2.90 element tc:RangeListType/UseRangeWhen/EqualityCheck 3-65
3.2.91 element tc:RangeListType/UseRangeWhen/CustomCheck 3-65
3.2.92 element tc:RangeListType/AlarmRange 3-66
3.2.93 complexType tc:SimpleAlarmsType 3-66
3.2.94 element tc:SimpleAlarmsType/CriticalRange 3-67
3.2.95 element tc:SimpleAlarmsType/WarningRange 3-67
3.2.96 complexType tc:StopType 3-68
3.2.97 complexType tc:VariableParameterType. 3-68
3.2.98 element tc:VariableParameterType/WidthRef 3-70
3.2.99 element tc:VariableParameterType/stop 3-71

3.2.100 complexType tc:AbsoluteTimeType 3-71
3.2.101 complexType tc:AliasType 3-72
3.2.102 complexType tc:BaseDataType 3-73
3.2.103 element tc:BaseDataType/Any 3-74
3.2.104 element tc:BaseDataType/Any/SourceParameterRef 3-74
3.2.105 element tc:BaseDataType/Float 3-74
3.2.106 element tc:BaseDataType/Float/DefaultValue 3-75
3.2.107 element tc:BaseDataType/Float/ValidRange 3-75
3.2.108 element tc:BaseDataType/Integer 3-75
3.2.109 element tc:BaseDataType/Integer/DefaultValue 3-76
3.2.110 element tc:BaseDataType/Integer/ValidRange 3-76
3.2.111 element tc:BaseDataType/Enumerated 3-76
3.2.112 element tc:BaseDataType/Enumerated/DefaultValue 3-77
3.2.113 element tc:BaseDataType/Enumerated/ValidRange 3-77
3.2.114 element tc:BaseDataType/Binary 3-77
3.2.115 element tc:BaseDataType/Binary/DefaultValue 3-78
3.2.116 element tc:BaseDataType/Boolean 3-78
3.2.117 element tc:BaseDataType/Boolean/DefaultValue 3-78
3.2.118 element tc:BaseDataType/Boolean/ValidRange 3-79
3.2.119 element tc:BaseDataType/String 3-79
3.2.120 element tc:BaseDataType/String/DefaultValue 3-79
3.2.121 element tc:BaseDataType/String/Enumeration 3-80
3.2.122 element tc:BaseDataType/AbsoluteTime 3-80
3.2.123 element tc:BaseDataType/AbsoluteTime/DefaultValue . . 3-81
3.2.124 element tc:BaseDataType/AbsoluteTime/ValidRange . . . 3-82
3.2.125 element tc:BaseDataType/RelativeTime 3-82
3.2.126 element tc:BaseDataType/RelativeTime/DefaultValue . . 3-83
3.2.127 element tc:BaseDataType/RelativeTime/ValidRange 3-84
3.2.128 element tc:BaseDataType/Units 3-84
3.2.129 element tc:BaseDataType/Units/Unit 3-84
3.2.130 element tc:BaseDataType/Unitless 3-85
3.2.131 complexType tc:BaseParameterType 3-85
3.2.132 element tc:BaseParameterType/LongDescription 3-86
3.2.133 element tc:BaseParameterType/BusAttributes 3-87
3.2.134 complexType tc:BitOffsetType 3-87
3.2.135 element tc:BitOffsetType/AbsoluteOffset 3-88
3.2.136 element tc:BitOffsetType/RelativeOffset 3-88
3.2.137 complexType tc:BusAttributesType 3-89
3.2.138 element tc:BusAttributesType/ErrorDetectCorrect 3-90
3.2.139 element tc:BusAttributesType/DefaultCalibrator 3-91
3.2.140 element tc:BusAttributesType/ByteOrder 3-92

3.2.141 element tc:BusAttributesType/ByteOrder/Byte 3-92
3.2.142 element tc:BusAttributesType/ContextCalibrator 3-92
3.2.143 element tc:BusAttributesType/ContextCalibrator/Context 3-93
3.2.144 element tc:BusAttributesType/ContextCalibrator/ Calibrator 3-94
3.2.145 complexType tc:ComparisonCheckType. 3-95
3.2.146 element tc:ComparisonCheckType/ParameterRef 3-96
3.2.147 element tc:ComparisonCheckType/ComparisonOperator 3-96
3.2.148 element tc:ComparisonCheckType/ArgumentRef 3-96
3.2.149 element tc:ComparisonCheckType/ParameterRef 3-97
3.2.150 element tc:ComparisonCheckType/Value 3-97
3.2.151 complexType tc:DefaultValueType 3-98
3.2.152 complexType tc:ErrorDetectCorrectType 3-98
3.2.153 element tc:ErrorDetectCorrectType/Parity 3-99
3.2.154 element tc:ErrorDetectCorrectType/CRC 3-99
3.2.155 element tc:ErrorDetectCorrectType/CRC/ PolynomialType 3-100
3.2.156 complexType tc:HeaderType. 3-100
3.2.157 element tc:HeaderType/Author 3-101
3.2.158 element tc:HeaderType/Note 3-101
3.2.159 element tc:HeaderType/History 3-101
3.2.160 complexType tc:IntegerRangeType. 3-101
3.2.161 complexType tc:MathOperationType 3-102
3.2.162 element tc:MathOperationType/ParameterRef 3-102
3.2.163 element tc:MathOperationType/Value 3-102
3.2.164 element tc:MathOperationType/Operator 3-103
3.2.165 element tc:MathOperationType/ParameterRef 3-103
3.2.166 element tc:MathOperationType/Value 3-103
3.2.167 complexType tc:NonStandardDataType 3-104
3.2.168 complexType tc:OccursType 3-104
3.2.169 element tc:OccursType/CountRef 3-105
3.2.170 element tc:OccursType/Count 3-106
3.2.171 element tc:OccursType/OffsetInBits 3-106
3.2.172 complexType tc:PolynomialType 3-107
3.2.173 element tc:PolynomialType/Term 3-107
3.2.174 complexType tc:PropertyType 3-107
3.2.175 element tc:PropertyType/Property. 3-108
3.2.176 complexType tc:RangeType 3-108
3.2.177 complexType tc:RealRangeType 3-109
3.2.178 complexType tc:SimpleParameterRefType 3-110
3.2.179 complexType tc:SplinePointType 3-110

3.2.180 complexType tc:SystemNameType 3-111
3.2.181 complexType tc:ToStringType 3-111
3.2.182 element tc:ToStringType/ValueEnumeration 3-112
3.2.183 element tc:ToStringType/RangeEnumeration 3-112
3.2.184 element tc:ToStringType/RangeEnumeration/Range 3-112
3.2.185 element tc:ToStringType/NumberFormat 3-113
3.2.186 complexType tc:UnitType 3-113
3.2.187 simpleType tc:ComparisonOperatorsType 3-114
3.2.188 simpleType tc:FormatType 3-114
3.2.189 simpleType tc:LimitCheckType 3-114
3.2.190 simpleType tc:LongDescriptionType 3-115
3.2.191 simpleType tc:MathOperatorsType 3-115
3.2.192 simpleType tc:NameReferenceType 3-116
3.2.193 simpleType tc:NameType 3-116
3.2.194 simpleType tc:RadixType 3-117
3.2.195 simpleType tc:RelativeTimeType 3-117
3.2.196 simpleType tc:ShortDescriptionType 3-118
3.2.197 complexType tc:ArgumentType 3-118
3.2.198 element tc:ArgumentType/Argument 3-121
3.2.199 complexType tc:CommandConstraintType 3-122
3.2.200 element tc:CommandConstraintType/CustomAlgorithm . 3-123
3.2.201 element tc:CommandConstraintType/Comparison 3-124
3.2.202 complexType tc:CommandDefinitionType 3-125
3.2.203 element tc:CommandDefinitionType/LongDescription . . 3-126
3.2.204 element tc:CommandDefinitionType/Alias 3-126
3.2.205 element tc:CommandDefinitionType/SystemName 3-126
3.2.206 element tc:CommandDefinitionType/Interlock 3-127
3.2.207 element tc:CommandDefinitionType/Field 3-127
3.2.208 element tc:CommandDefinitionType/Field/LongDescription 3-128
3.2.209 element tc:CommandDefinitionType/Field/Argument . . . 3-128
3.2.210 element tc:CommandDefinitionType/Field/ArgumentRef 3-129
3.2.211 element tc:CommandDefinitionType/Field/ParameterRef 3-130
3.2.212 element tc:CommandDefinitionType/Field/BinaryConstant 3-130
3.2.213 element tc:CommandDefinitionType/BinaryTransform . . 3-131
3.2.214 element tc:CommandDefinitionType/BinaryTransform/FieldBitPosition 3-131
3.2.215 element tc:CommandDefinitionType/BinaryTransform/FieldBitPosition/BitPositionFromStart 3-131

3.2.216 element tc:CommandDefinitionType/BinaryTransform/FieldBitPosition/BitPositionFromPreviousArgument . . . 3-132
3.2.217 element tc:CommandDefinitionType/BinaryTransform/ContainerRef 3-132
3.2.218 element tc:CommandDefinitionType/CommandProcessor 3-133
3.2.219 element tc:CommandDefinitionType/CommandValidator 3-135
3.2.220 element tc:CommandDefinitionType/CommandConstraint 3-136
3.2.221 element tc:CommandDefinitionType/CommandVerifier . 3-137
3.2.222 element tc:CommandDefinitionType/ParametersToSet . . 3-137
3.2.223 element tc:CommandDefinitionType/ParametersToSet/ParameterRef 3-138
3.2.224 element tc:CommandDefinitionType/ParametersToSet/Value 3-138
3.2.225 element tc:CommandDefinitionType/ParametersToSet/ArgumentRef 3-138
3.2.226 element tc:CommandDefinitionType/NonStandardData . 3-139
3.2.227 complexType tc:CommandVerifierType 3-140
3.2.228 element tc:CommandVerifierType/CustomAlgorithm . . . 3-141
3.2.229 element tc:CommandVerifierType/Comparison 3-142
3.2.230 element tc:CommandVerifierType/ContainerRef 3-143
3.2.231 element tc:CommandVerifierType/Window 3-143
3.2.232 element tc:CommandVerifierType/Window/StartTime . . . 3-144
3.2.233 element tc:CommandVerifierType/Window/Interval 3-144
3.2.234 element tc:CommandVerifierType/Window/ChangeCount 3-144
3.2.235 simpleType tc:InterlockScopeType 3-144
3.2.236 simpleType tc:StageType 3-144
3.2.237 element tc:Packaging 3-145
3.2.238 complexType tc:ChoiceType 3-146
3.2.239 element tc:ChoiceType/Occurs 3-146
3.2.240 element tc:ChoiceType/ContainerRef 3-147
3.2.241 complexType tc:ContainerRefType 3-148
3.2.242 element tc:ContainerRefType/Occurs 3-149
3.2.243 element tc:DynamicContainerRefType/DependantOn . . . 3-150
3.2.244 element tc:DynamicContainerRefType/OffsetInBits 3-151
3.2.245 element tc:DynamicContainerRefType/ParameterRef . . . 3-151
3.2.246 complexType tc:MessageKeyMatchListType 3-152
3.2.247 element tc:MessageKeyMatchListType/KeyMatch 3-152
3.2.248 complexType tc:MessageKeyMatchType 3-153
3.2.249 complexType tc:MessageKeyType 3-154

3.2.250 element tc:MessageKeyType/LongDescription 3-154
3.2.251 element tc:MessageKeyType/OffsetInBits 3-155
3.2.252 complexType tc:MessageType 3-155
3.2.253 element tc:MessageType/LongDescription 3-156
3.2.254 element tc:MessageType/Key 3-156
3.2.255 complexType tc:PackagingDefinitionType 3-157
3.2.256 element tc:PackagingDefinitionType/MessageList 3-157
3.2.257 element tc:PackagingDefinitionType/MessageList/ Message 3-158
3.2.258 element tc:PackagingDefinitionType/ContainerList 3-158
3.2.259 element tc:PackagingDefinitionType/ContainerList/ Container 3-159
3.2.260 element tc:PackagingDefinitionType/NonStandardData . 3-159
3.2.261 complexType tc:SequenceType 3-160
3.2.262 element tc:SequenceType/Occurs 3-161
3.2.263 element tc:SequenceType/ContainerRef 3-162
3.2.264 element tc:SequenceType/DynamicContainerRef 3-163
3.2.265 element tc:SequenceType/ArgumentRef 3-164
3.2.266 element tc:SequenceType/ParameterRef 3-165
3.2.267 element tc:SequenceType/DynamicParameterRef 3-166
3.2.268 element tc:SequenceType/BinaryConstant 3-167
3.2.269 element tc:SequenceType/BinaryConstant/ LongDescription 3-167
3.2.270 element tc:SequenceType/BinaryConstant/OffsetInBits . 3-168
3.2.271 element tc:SequenceType/Choice 3-168
3.2.272 element tc:SequenceType/Sequence 3-169
3.2.273 complexType tc:ServiceType 3-170
3.2.274 element tc:ServiceType/LongDescription 3-170
3.2.275 element tc:ServiceType/MessageList 3-171
3.2.276 element tc:ServiceType/MessageList/NameRef 3-171
3.2.277 element tc:ServiceType/ContainerList 3-171
3.2.278 element tc:ServiceType/ContainerList/Id 3-172
3.2.279 simpleType tc:ContainerNameType 3-172
3.2.280 simpleType tc:MessageKeyIdType 3-172
3.2.281 simpleType tc:MessageNameType 3-173
3.2.282 simpleType tc:ServiceNameType 3-173
3.2.283 complexType tc:ConvolutionalStreamType 3-173
3.2.284 element tc:ConvolutionalStreamType/Polynomial 3-174
3.2.285 complexType tc:EncryptedStreamType 3-175
3.2.286 complexType tc:FixedFrameStreamType 3-175
3.2.287 element tc:FixedFrameStreamType/SyncStrategy 3-176

3.2.288 complexType tc:PCMStreamType 3-177
3.2.289 element tc:PCMStreamType/LongDescription 3-178
3.2.290 complexType tc:StreamType 3-178
3.2.291 element tc:StreamType/FixedFrameStream 3-179
3.2.292 element tc:StreamType/EncryptedStream 3-180
3.2.293 element tc:StreamType/VariableFrameStream 3-180
3.2.294 element tc:StreamType/ConvolutionalStream 3-181
3.2.295 element tc:StreamType/NonStandardData 3-181
3.2.296 complexType tc:SyncStrategyType 3-182
3.2.297 element tc:SyncStrategyType/AutoInvert 3-183
3.2.298 element tc:SyncStrategyType/MessageMatch 3-183
3.2.299 element tc:SyncStrategyType/MessageMatch/NameRef . 3-183
3.2.300 element tc:SyncStrategyType/SyncPattern 3-184
3.2.301 complexType tc:VariableFrameStreamType 3-184
3.2.302 element tc:VariableFrameStreamType/SyncStrategy 3-185
3.2.303 simpleType tc:StreamSourceType 3-186

A - XML Schema A-1

Preface

About This Document

Under the terms of the collaboration between OMG and The Open Group, this document is a candidate for adoption by The Open Group, as an Open Group Technical Standard. The collaboration between OMG and The Open Group ensures joint review and cohesive support for emerging object-based specifications.

Object Management Group

The Object Management Group, Inc. (OMG) is an international organization supported by over 600 members, including information system vendors, software developers and users. Founded in 1989, the OMG promotes the theory and practice of object-oriented technology in software development. The organization's charter includes the establishment of industry guidelines and object management specifications to provide a common framework for application development. Primary goals are the reusability, portability, and interoperability of object-based software in distributed, heterogeneous environments. Conformance to these specifications will make it possible to develop a heterogeneous applications environment across all major hardware platforms and operating systems.

OMG's objectives are to foster the growth of object technology and influence its direction by establishing the Object Management Architecture (OMA). The OMA provides the conceptual infrastructure upon which all OMG specifications are based. More information is available at http://www.omg.org/.

The Open Group

The Open Group, a vendor and technology-neutral consortium, is committed to delivering greater business efficiency by bringing together buyers and suppliers of information technology to lower the time, cost, and risks associated with integrating new technology across the enterprise.

The mission of The Open Group is to drive the creation of boundaryless information flow achieved by:

- Working with customers to capture, understand and address current and emerging requirements, establish policies, and share best practices;
- Working with suppliers, consortia and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies;
- Offering a comprehensive set of services to enhance the operational efficiency of consortia; and
- Developing and operating the industry's premier certification service and encouraging procurement of certified products.

The Open Group has over 15 years experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification. The Open Group portfolio of test suites includes tests for CORBA, the Single UNIX Specification, CDE, Motif, Linux, LDAP, POSIX.1, POSIX.2, POSIX Realtime, Sockets, UNIX, XPG4, XNFS, XTI, and X11. The Open Group test tools are essential for proper development and maintenance of standards-based products, ensuring conformance of products to industry-standard APIs, applications portability, and interoperability. In-depth testing identifies defects at the earliest possible point in the development cycle, saving costs in development and quality assurance.

More information is available at http://www.opengroup.org/ .

OMG Documents

The OMG Specifications Catalog is available from the OMG website at:

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

The OMG documentation is organized as follows:

OMG Modeling Specifications

Includes the UML, MOF, XMI, and CWM specifications.

OMG Middleware Specifications

Includes CORBA/IIOP, IDL/Language Mappings, Specialized CORBA specifications, and CORBA Component Model (CCM).

Platform Specific Model and Interface Specifications

Includes CORBAservices, CORBAfacilities, OMG Domain specifications, OMG Embedded Intelligence specifications, and OMG Security specifications.

Obtaining OMG Documents

The OMG collects information for each book in the documentation set by issuing Requests for Information, Requests for Proposals, and Requests for Comment and, with its membership, evaluating the responses. Specifications are adopted as standards only when representatives of the OMG membership accept them as such by vote. (The policies and procedures of the OMG are described in detail in the *Object Management Architecture Guide*.) OMG formal documents are available from our web site in PostScript and PDF format. Contact the Object Management Group, Inc. at:

OMG Headquarters

250 First Avenue

Needham, MA 02494

USA

Tel: +1-781-444-0404

Fax: +1-781-444-0320

pubs@omg.org

http://www.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.

Helvetica bold - OMG Interface Definition Language (OMG IDL) and syntax elements.

`Courier bold` - Programming language elements.

Helvetica - Exceptions

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

Acknowledgments

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

- Lockheed Martin
- The Boeing Company
- The European Space Agency

Overview 1

Contents

This chapter contains the following sections.

Section Title	Page
“Background”	1-1
“Introduction”	1-2
“Scope”	1-3
“Applicable and Reference Documents”	1-4
“Compliance Points”	1-5
“Glossary of Acronyms”	1-9

The Telemetry & Command data specification is intended as a way to describe telemetry and command “databases” as used in ground telemetry systems, packet and TDM based systems. The intent is to allow the easy interchange of these databases between systems and organizations.

The vision is that this standard will one day be the “native” format for ground systems. Until that time, companies and organizations using this specification can employ converters to go from one system to another, or can convert an existing database into this format for exchange with other parties.

1.1 Background

Space mission implementations face a very dynamic environment with fast-paced information technology advancement and shrinking space budgets. A more focused use of decreasing public investments in space requires a cost reduction over their entire life

cycle, from development up to the end of the useful life of a spacecraft. The use of standards specifications from the early stages of satellite development through mission operation can be utilized to reduce life-cycle cost.

Satellite design and development is performed today through the use of a number of disparate tools and techniques. Interface design to satellite systems and to the payloads the satellites are housing is still a manual and time-consuming effort. Data design, both telemetry and commanding, is still performed multiple times by multiple contractors during the lifecycle of the satellite, well before the satellite is ever deployed for mission operations. The standardization of satellite telemetry and command data for spacecraft health and safety, as well as payload interfaces will reduce the cost of these implementations as well as decrease the schedule of development, integration, and test of the satellite and its component systems. This specification can also be used to support multiple, heterogeneous missions, facilitating interoperability between ground control systems, simulators, testing facilities, etc.

This specification addresses the need for an information model for telemetry and commanding in support of all phases of the satellite, payload, and ground segment lifecycle: system design, development, test, validation, and mission operations.

1.2 Introduction

This is a standard specification to define an information model for spacecraft telemetry and commanding data. This specification is crucial to the general standardization of the telemetry and commanding area.

For a given mission there are a number of lifecycle phases that are supported by a variety of systems and organizations. Additionally, many of these organizations support multiple heterogeneous missions using a common ground segment infrastructure. Telemetry and command definitions must be exchanged among all of these phases, systems, and organizations. This is made difficult and costly because there is no standard method for exchanging this information. The lack of standardization currently requires custom ingestion of the telemetry and commanding information. This customization is inherently error-prone, resulting in the need to revalidate at each step in the lifecycle.

A typical example of this process is between the spacecraft manufacturer and spacecraft-operating agency. The spacecraft manufacturer defines the telemetry and command data in a format that is much different than the one used in the ground segment. This creates the need for database translation, increased testing, software customization, and increased probability of error. Standardization of the command and telemetry data definition format will streamline the process allowing dissimilar systems to communicate without the need for the development of mission specific database import/export tools.

Ideally, a spacecraft operator should be able to transition from one ground system to another by simply moving an already existing command and telemetry database compliant with this command and telemetry database specification.

In addition, standardization will enable space or ground segment simulators to more easily support multiple heterogeneous missions.

As a quick background on spacecraft telemetry and commanding, the following paragraphs will serve to educate those not familiar with spacecraft communications data.

Telemetering is defined (from IEEE Std 1000 [1972]) as “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. Telemetry as used here refers to these measurements whether on-board the spacecraft or transmitted to the ground system. Most telemetry measurements will require engineering unit conversion and measurements will have associated validation ranges or lists of acceptable values.

Commands, as defined for this RFP, are messages originating from the ground or the spacecraft to perform a function on the spacecraft or ground system. 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. Packaging of both telemetry and commands can be performed in a number of ways. The most common way to package data for transmission is to use the CCSDS Telemetry and Commanding Packaging format.

1.3 Scope

The specification addresses the need for a standardized information model capable of supporting TM/TC definitions across the broadest possible range of space domain activities. The goal is to allow TM/TC definitions to be exchanged between different organizations and systems, often at the boundaries of mission phases, without the need for customized import/export, re-validation, or even re-implementation of mission databases.

The scope of this specification is limited to satellite telemetry and commanding data constructs necessary to support satellite and payload data design:

- Telemetry data definitions including support for CCSDS packets as well as TDM frames.
- Data manipulation algorithms to support packaging and unpacking of individual data items.
- Commanding data definitions including command identification, argument specification, and validation criteria.
- Data representation definitions
- Data properties including such things as its default value, validity criteria, and data dependencies.
- The definition of extensible formats such that blocks of information (whether frames of data that are not decommutated or object references or object method calls) can be portrayed in this architecture.

The scope of this specification does not extend to:

- Data distribution mechanisms.
- Command and data protocol specifications.
- RF or analog stream characterization
- Data groupings including aggregation and coherent data sets
- Data Representation (visualization properties)
- Scheduling configuration properties
- Orbital properties

The specification addresses only the definition of TM/TC data, and not the transfer of live or historical TM/TC data.

1.4 Applicable and Reference Documents

All documents published by ESA can be provided on request.

1.4.1 Applicable Documents

A.01	CCSDS 102.0-B-4	CCSDS Packet Telemetry
A.02	CCSDS 203.0-B-1	CCSDS Telecommand
A.03	ECSS-E-70-41	Telemetry and Telecommand Packet Utilization, Draft 5.3, 5 Apr 2001
A.04	S2K-MCS-ICD-0001-TOS-GCI	SCOS-2000 Database Import ICD, Issue 5.0, 19 Jun 2001
A.05	Space/01-04-01	Telemetric and Command Data Specification, Space RFP-1, 20 Aug 2001
A.06	99-11-02, Version 1.11	Policies and Procedures of the OMG Technical Process, 27 Nov 1999

1.4.2 Reference Documents

R.01	ESA PSS-07-101	Packet Utilization Standard, Issue 1, May 1994
R.02	CCSDS 301.0-B-2	CCSDS Time Code Formats, Issue 2, April 1990
R.03	Http://www.w3.org/TR/REC-xml	W3C Recommendation - Extensible Markup Language (XML) 1.0 (Second Edition, 6 October 2000)
R.04	Http://www.w3.org/TR/xmlschema-0/	W3C Recommendation - XML Schema Part 0: Primer (2 May 2001)
R.05	Http://www.w3.org/TR/xmlschema-1/	W3C Recommendation - XML Schema Part 1: Structures (2 May 2001)
R.06	Http://www.w3.org/TR/xmlschema-2/	W3C Recommendation - XML Schema Part 2: Datatypes (2 May 2001)
R.07	S2K-XML-ROMEOSA-1.0	S2K XMIB – Use of XML for MIB Management. (ESA Report of study activity). September 2001.
R.08	S2K-MCS-SUM-0019-TOS-GCI	SCOS-2000 Synthetic Parameters Software User Manual, Issue 3.1, September 2001.

1.5 Compliance Points

The xsd files in Appendix A are normative. A compliant database is an XML file that complies with these xsd files. Fully compliant implementing software will interpret and/or generate any databases compliant with this specification. Compliant implementing software will interpret and/or generate all database elements required by the schema.

A statement of compliance to each of the mandatory requirements is given below. The specification is not compliant to optional requirements

Requirement	Compliance	Comment
Telemetry Properties		
Administrative (e.g., telemetry point identifier, telemetry mnemonics (possibly multiple), units)	Yes	
Categorization (system, subsystem, people, ownership) - multiple values	Part	Specification supports a full system of systems hierarchy; however, telemetry parameters may only “appear” once in the hierarchy.
Ability to set telemetry items	Yes	Internal ‘settable’ parameters supported
Source information (spacecraft address description, "memorized", derived)	Yes	
Position dependence (data source reference)	Yes	
Decommutation support - start byte, start bit, length	Yes	

Parsing support - Information is not absolute position dependent (i.e. text message, or event)	Yes	Support for variable packets
Support for concatenating multiple pieces of source data to form a single telemetry item	Yes	
Data Dependence (Telemetry points it relies on, telemetry points that rely on it)	Yes	
Logical relationships between them	Yes	Validation/Verification/Synthetic expressions
Identity properties (data type)	Yes	
Default value	Yes	
Representation information (see below for list)		See below
Algorithmic properties (see below for list)		See below
Validation checks (red, yellow limits, deltas, possible values - desired and or undesired)	Yes	Alarms
Validation conditions (when the individual validation checks are used)	Yes	
Command Properties		
Administrative (e.g., command identifier, command mnemonic)	Yes	
Categorization (system, subsystem, people, ownership)	Yes	
Intrinsic properties (default value)	Yes	
Fields properties	Yes	
Position Dependence (start byte, start bit, length)	Yes	
Algorithmic Properties (transformation definition)	Yes	
Data Dependence (Telemetry points it relies on)	Yes	
Representation (format, precision)	Yes	
Support for nesting of fields	Yes	Group Repeaters and Conditional structures
Validation criteria (possible values)	Yes	
Authorization (who, what, when)	Part	Command Constraints provide a mechanism for “when”, but the submitters all concluded that “who” and “when” are inappropriate for this specification
Verification information (associated telemetry)	Yes	
Memorized information (associated mnemonics)	Yes	
Timing constraints (command dependencies, timing information)	Yes	
Execution constraints (prompting, not able to execute)	Yes	

Algorithmic Properties		
Masking and Shifting	Yes	Since telemetry streams are defined in this data spec at the bit level rather than word level, fewer mask and shift operations will be necessary. Masking and shifting operations are created as custom algorithms
Concatenation	Yes	
Polynomial Conversion	Yes	A defined algorithm in the AlgorithmDirectory.xsd. An arbitrary polynomial length is supported.
Slope Intercept	Yes	Yes. Slope intercept is a special case of a polynomial conversion (i.e. C_0 is the intercept and C_1 is the slope)
Interpolation	Yes	
Table Lookup (and variations)	Yes	Implemented as enumerated
Enumerated (Discrete Value mapping, i.e. ON/OFF)	Yes	
Generic (Submitters are also encouraged to provide a generic syntax for algorithmic representation)	Yes	Algorithm inputs, outputs, triggers, and names can be defined as Custom algorithms or algorithm text – in an arbitrary language – may be included. Implimentations are not required to support the arbitrary language.
Identification - (i.e. Mnemonic, Name, Function)	Yes	Algorithm inputs, outputs, triggers, and names can be defined as Custom algorithms.
Input data	Yes	Algorithm inputs, outputs, triggers, and names can be defined as Custom algorithms.
Output data	Yes	Algorithm inputs, outputs, triggers, and names can be defined as Custom algorithms.
Triggering information (data, rate)	Yes	Algorithm inputs, outputs, triggers, and names can be defined as Custom algorithms.
Equation definition	Yes	
Combinations of any of the above techniques	Yes	
Data Representation		
IEEE Float	Yes	
1758 Float	Yes	MIL-STD-1750A floats are a core bus data type.
Double	Yes	
Integer	Yes	
String	Yes	
CCSDS Time Formats	Yes	
Ones Complement Integer	Yes	

Twos Complement Integer	Yes	
Sign Magnitude Integer	Yes	
Bit Order	Yes	
Byte Order	Yes	
Object References	Partial	As a Binary data type.
Sequences of the above techniques	Yes	Repeating data may be definied
Time tagging of the above data types	No	The submitters all concure that time tagging is performed by the implimentation.
Combinations of any of the above techniques	Yes	
Validation Techniques		
Equivalency checks	Yes	
Discrete value checks	Yes	
Range checks	Yes	
Delta checks (gradient)	Yes	
Trend checks	Yes	
Status checks	Yes	
List checks	Yes	
Parity checks	Yes	
Checksum techniques (XOR at a minimum)	Yes	
Combinations of any of the above techniques	Yes	
Packaging Information		
Telemetry Framing information	No	A robust frame synchronization strategy can be defined using StreamDefinitions.xsd.
CCSDS support - packets and frames	Yes	
TDM support - major frame, minor frame	Yes	A robust TDM structure can be defined
Telemetry item support (start byte, start bit, length; combinations of individual fields)	Yes	
Command	Yes	
Packaging information - multiple levels of specification	Yes	
Directions for assembling the commands	Yes	
Fields properties	Yes	
Location (start byte, start bit, length)	Yes	
Representation (format, precision)	Yes	

Nested fields	Yes	
Timing constraints	Yes	
Steam Definition	Yes	
Stream Encoding (e.g., Convolution)	Yes	
Encryption	Yes	Would be selected at run-time
Block Encoding (e.g. BCH, CRC)	Yes	Would be selected at run-time
Randomization	Yes	Would be selected at run-time

1.6 Glossary of Acronyms

This section lists the Acronyms used in this document. Note that the acronyms of the tables are defined in the Summary of Tables section.

ASCII	American Standard Code for Information Interchange
ATV	Automated Transfer Vehicle
BSS	Boeing Satellite Systems
CCSDS	Consultative Committee for Space Data Systems
CSA	Canadian Space Agency
CTCU	Central Telemetry and Command Unit
DLR	Deutschen Zentrum für Luft- und Raumfahrt (German Space Agency)
DTF	Domain Task Force
ECSS	European Co-operation for Space Standardization
ESA	European Space Agency
EU	Engineering Units
LEOP	Launch and Early Orbit Phase
MIB	Mission Information Base
MSG	Meteosat Second Generation
OMG	Object Management Group
PSS	Portable Satellite Simulator
PUS	Packet Utilization Standard
RFP	Request for Proposal
RTCU	Remote Telemetry and Command Unit

SCTV	Spacecraft Thermal Vacuum
SCOS	Spacecraft Control and Operations System
TC	Telecommand
TDM	Time Division Multiplexing
TM	Telemetry
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XMM	X-Ray Multi Mirror

Telemetry and Telecommand Data Specification 2

Contents

This chapter contains the following sections.

Section Title	Page
“Specification”	2-1
“Processing the Telemetry Stream”	2-5

2.1 Specification

The specification provides a standard format for defining the Telemetric and Telecommand (TM/TC) data required to perform the processing shown in the figure below. The normative portion of this specification is presented as an XML schema compliant with W3C recommendation of 02/05/2001.

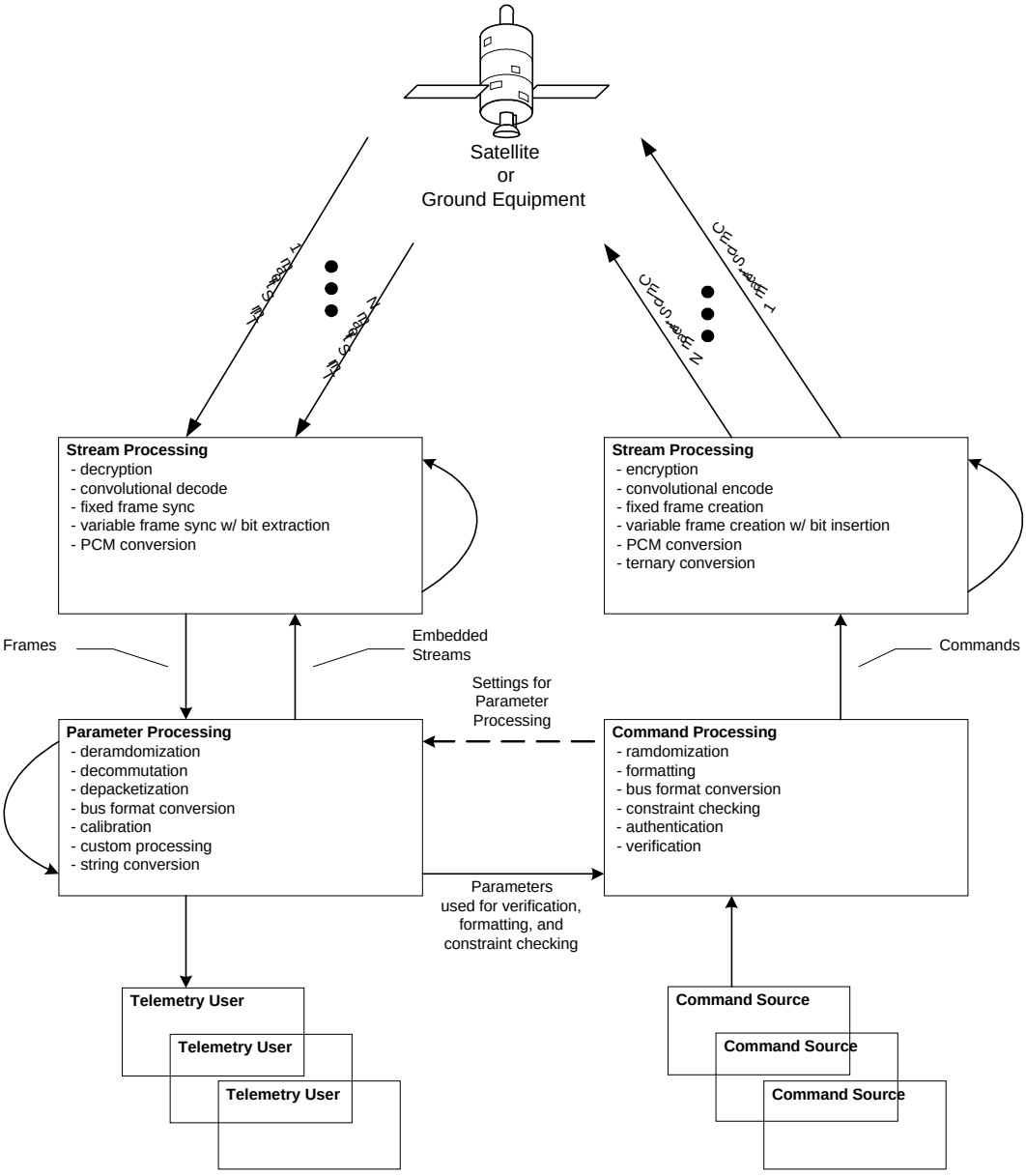


Figure 2-1

This section presents some background on space systems database concepts to facilitate a better understanding of the schema. The XML schema itself, the subject of this specification, is described in Part III.

This specification and the XML schemas that makeup the schema is divided into 5 sections: SpaceSystem, CommonTypes, Parameter, CommandDefinition, and Algorithm.

This T&C specification has an object-oriented structure where all the elements of the specification belong to a single root object – the SpaceSystem.

2.1.1 The Root Object - The SpaceSystem

Recognizing that spacecraft operations involve much more than simply controlling the spacecraft, the top-level object is not ‘Spacecraft’ but a more the more generic term ‘SpaceSystem.’ This name provides deference to the fact that a spacecraft operations center must control antennas, recorders, ground processing equipment, RF hardware and other many other devices that could potentially share a common T&C data specification; each of these objects is a ‘SpaceSystem.’ A SpaceSystem *has a* Header, zero or more sub-SpaceSystems and a CommandAndTelemetry component. The CommandAndTelemetry component contains the bulk of the Telemetric and Command data and the sub-SpaceSystems give the data a hierarchical structure. The CommandAndTelemetry component contains a ParameterList a CommandDefinition List, an AlgorithmList, a StreamList, a Packaging component, a ServiceList and some optional defaults.

Note on the sub-SpaceSystem and the hierarchical structure

Because a SpaceSystem may itself contain other SpaceSystems, the organization of the data may be organized hierarchical structure – similar to the structure of a real space system. The hierarchical organization offers several important advantages over a flat entity list:

- Fewer name space collisions – Almost every spacecraft contains redundant components for reliability or to accomplish the mission. A communications spacecraft may have a dozen transponders each with the same set of telemetry points and commands. In a flat namespace each of those telemetry points needs to be mapped into a unique name. Using a hierarchical namespace, those identical telemetry points can be simply placed into separate sub-SpaceSystems.
- Better organization – modern spacecraft typically have thousands of commands and tens of thousands of telemetry parameters; this number is trending upward. The directory structure provided by this specification provides an improved way to manage this large volume of data. Each subsystem developer can deliver SpaceSystems representing their subsystem without integration issues.
- Defaults at the SpaceSystem level – many of the attributes needed to define spacecraft parameters (e.g., bit order, byte order) are common to most of the parameters in the spacecraft or spacecraft sub-system. This specification allows these attributes to be assigned at the directory level, thereby avoiding their repetition in each parameter.
- Spacecraft which are normally thought of as a SpaceSystem, may actually be sub-SpaceSystems for a constellation of spacecraft SpaceSystems.
- Natural hierarchy – spacecraft designs are increasing in complexity and are normally comprised of systems of systems. The hierarchical organization allowed by a directory structure reflects this.

Note on Names

Parameter, and CommandDefinition and other major entity names within this database may be any length but are prohibited from containing the '/', '.', and ':' characters as these are reserved. The '/' is used as the SpaceSystem separator (Unix and HTTP style). The ':' is reserved for future use as a selector for data from other SpaceSystems. The '.' is reserved as an attribute selector.

2.1.2 The Header Record

A SpaceSystem may contain an optional header record. This record contains some basic context on the data itself (e.g., source, version, revision history, notes, and classification).

2.1.3 Parameters

A Parameter is a description of something that can have a value; it is not the value itself. There are two basic types of Parameters, 'Parameters' and a sub-type of Parameter 'BusParameter'. Parameters are a super class of BusParameters and contain information including data type, description, name, limits, filter information, and string conversion specifications. BusParameters could have also been called Measurands because they are data points *measured* off of the Device. In addition to the data contained in Parameter, BusParameters must contain information about bit order, byte order, and extra data type information (may be one of several integer types or floating point types).

Required attributes for Parameters:

- raw – to refer to the uncalibrated value of a Parameter in its native bit/byte order
- precalibrated – to refer to the uncalibrated value of a Parameter
- string – to refer to the string representation of a Parameter

Optional attributes for Parameters:

- max – the highest seen value of a Parameter
- min – the lowest seen value of a Parameter

Future attributes for Parameters may include a variety of statistical measures (e.g., average, moving average, standard deviation), temporal measures (e.g., rate-of-change (1^{st} derivative), acceleration (2^{nd} derivative)), and other measures (e.g., samples seen, quality, times out of limit, etc.).

2.1.4 Command Definitions

A Command Definition provides a name the bit representation, and the allowable arguments to create a command that may be sent to a SpaceSystem. The CommandDefinition also contains the instructions for how to perform the binary transformation of this command and its arguments. A CommandDefinition also optionally contains CommandValidations, CommandContraints, and CommandVerifiers.

A CommandValidation is a process whereby the command issuer is checked as a valid authority to command a device. This check may be temporal and may be command specific. Note: this definition of command definition does not match the traditional one within the spacecraft domain industry; however, this definition does more closely match the definition of validation within the broader IT industry.

A CommandConstraint is a conditional check of the operating environment that allows the Command to proceed.

A CommandVerifier is a conditional check on the telemetry from a SpaceSystem that provides positive indication on the successful execution of a command.

2.1.5 Algorithm

In spacecraft ground systems, it is necessary to perform some specialized processing to process the telemetry, and preprocess commands. There are a number of predefined algorithms and the algorithm section makes it possible to reference externally defined algorithms for arbitrarily sophisticated data processing.

2.1.6 Stream

Spacecraft uplinks and spacecraft downlinks are digital streams of data are there are a number of processing functions that are done on the stream level. The stream section contains the knowledge for how to assemble, disassemble and process spacecraft uplink and downlink streams.

2.1.7 Packaging

The packaging section contains the information required to assemble an uplink from its component parts and disassemble a downlink from its component parts. The packaging section has been created to be extremely generic so that it may be used to define TDM telemetry streams, packetized streams or any other package format.

2.2 Processing the Telemetry Stream

This section describes the processes that deal with Processing the Telemetry Stream using data in the underlying XML schema.

2.2.1 Typical Delivery Mechanisms for Telemetry Streams

2.2.1.1 Packetized Telemetry

Over the last decade, ESA has been developing and promoting the Packet Utilization Standard (PUS), first through its own PSS standards series [R.01], and latterly through the European Co-operation for Space Standardization (ECSS) [A.03]. The PUS supports the concept of end-to-end services, in which on-board processes communicate directly with peer processes on the ground, through a set of defined message structures. Services

typically comprise both downlink [TM] and uplink [TC] messages. The PUS identifies a standard set of services with associated message structures overlaid on the existing Consultative Committee for Space Data Systems (CCSDS) packet TM/TC standards, and also provides a framework for the definition of mission specific services.

Most TM/TC formats support basic housekeeping telemetry and simple device commanding, and while PUS does offer benefits here, it is with other classes of data uplink/downlink that the service concept comes into its own. The identification of standard command verification, on-board memory management and on-board schedule management services, has enabled standardized capabilities to be implemented to support these functions across a range of missions, when previously this had required mission-specific development.

Future ESA spacecraft will be compatible with the PUS, and as a consequence the concepts it espouses are being built into the product lines of most European spacecraft manufacturers. The following space missions have used or plan to use the PUS: XMM, Meteosat Second Generation (MSG), INTEGRAL, GOMOS (Envisat Instrument), ATV (Automated Transfer Vehicle), Orsted (Danish microsatellite), PROBA, ROSETTA, MARS EXPRESS, FIRST/PLANK, CRYOSAT, GOCE, GALILEO.

From a European perspective, it is critical that the Telemetric and Command Data Specification adopted by OMG should provide for the representation of PUS-compliant telemetry and telecommand definitions. A key objective of the specification should be that it fosters interoperability and compatibility of systems within the Space domain, through convergent TM/TC definitions. This will only be achieved by an inclusive specification that takes into account compatibility with existing standards.

ESA's own satellite control system kernel, SCOS-2000, has been designed to support both packet TM/TC and the PUS service concepts. This approach does not, however, limit its applicability to missions using these standards: it has generally proven to be much easier to map other [simpler] TM/TC structures, such as fixed format frame-based telemetry, on to the service/packet model than to do this the other way around. SCOS-2000, although designed as a mission operations support tool, has also been successfully deployed in support of spacecraft integration and test activities.

SCOS-2000 won two awards at the OMG's Object World conference in Berlin, 1999:

- 1st prize for “Best Implementation of a Distributed Application using Object Technology”
- Finalist for “Best use of Object Technology within an Enterprise or Large System Environment.”

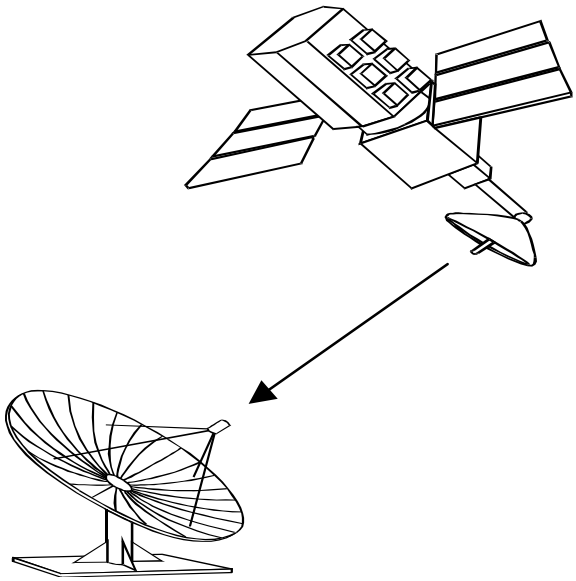
SCOS-2000 is being used for all recent and future ESA spacecraft missions, including: Huygens, MTP LEOP, Proba, MSG LEOP, Integral, Rosetta, Mars Express, SMART-1, Cryosat and GOCE. It has also been provided to other space agencies, which are using, or plan to use, SCOS in the control of their own missions: these include the German (DLR) and Canadian Space Agencies (CSA) and EUTELSAT.

2.2.1.2 Time Division Multiplexing

A *telemetry stream* can be defined as a long stream of binary data originating on the spacecraft that contains health, status and other information generated by the spacecraft bus and payload. *Decommutation* is defined as the process by which the telemetry stream is broken into component pieces for use by the ground software. This section is devoted to explaining how to use the decommutation records to correctly break apart the frame-based telemetry stream.

2.2.1.3 Telemetry Decommutation

TDM streams are uniquely characterized by the presence of a predictable telemetry format. This is because spacecraft generating a TDM stream stores a table of sequencing code consisting of ordered sets of telemetry requests. Since these sequencing code tables are known on the ground, the order in which telemetry requests are generated on the spacecraft and assembled into the telemetry stream can also be predicted. It is only that telemetry that we can predict which can be decommutated using the method contained in this section. The spacecraft generates a repeating stream of bits and sends it to the ground. Each stream of bits is generated using a set of directions that direct the spacecraft to include specific pieces of data in specific locations within the stream. Each set of directions is called a telemetry *format*. Since multiple formats usually exist, the spacecraft is typically restricted from generating more than one format at any given time.



A spacecraft generates one format at a time.

Figure 2-2

The database differentiates between distinct sets of decommutation information using a *format number.* Each format number contains one complete set of directions for the decommutation of a designated telemetry stream. In order to determine which format the spacecraft is producing at any given time, a fixed location common to all formats is found in the telemetry stream that is devoted to the identification of the format number.

A *minor frame* is a subset of a format. Each format is typically broken into minor frames. Minor frames are characterized by their repetition of specific telemetry measurands in specific locations. For example, every minor frame begins with information that identifies the format number as well as a minor frame number. Once a minor frame has been identified within a format, it may be broken into its component *words*. Words are composed of some predetermined numbers of individual bits; typical words are eight bits long. The *start bit* represents the first bit within the 8-bit word of the designated minor frame and format. The identification of the format, minor frame, word and start bit uniquely references the starting location for a Parameter. Once this information is declared, the length of the Parameter can be obtained and the binary string from the telemetry stream can be assigned to the Parameter for additional processing. This additional processing typically includes assigning a name, performing EU conversion and limit checking.

The following illustration has been provided to help visualize how a *major frame* is broken into its component parts. A major frame is the repeating section of a format and consists of minor frames, words and start-bits. Major frames will repeat as long as the spacecraft is generating telemetry; major frame numbers have no meaning.

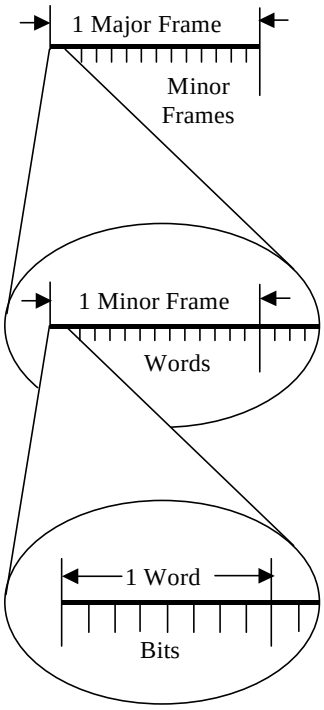


Figure 2-3

Each format generates a repeating stream of telemetry called a major frame. Each format is broken into some predetermined number of minor frames, usually 32. These minor frames are displayed in a stacked form as shown at the bottom of this page in horizontal rows. Minor frames always have a domain that starts at zero and increments in integer steps. (i.e., 0, 1, 2, 3, …)

Minor frames are additionally broken into some predetermined number of words. This means that a word is a subset of a minor frame. This is because all telemetry requests from the spacecraft consist of single word requests. Words are displayed as vertical columns in the “grid” at the bottom of this page. Words always have a domain that starts at zero and increments in integer steps. (i.e., 0, 1, 2, 3, …)

Finally, each word is broken into some predetermined number of bits, usually 8. Once the word is broken into bits, the reference for Parameter locations can begin. All Parameters will be referenced by format, minor frame, word and start bit. The number of bits to assign to the Parameter is located within the Parameter definition itself. The total number of bits in a major frame can be calculated using the following equation:

$$\frac{\textbf{\# of bits}}{\textbf{major frame}} = \frac{\textbf{\# of minor frames}}{\textbf{major frame}} \times \frac{\textbf{\# of words}}{\textbf{minor frame}} \times \frac{\textbf{\# of bits}}{\textbf{word}}$$

The illustration below has been provided to show how a major frame is typically presented. Notice that minor frame and word numbers start at zero at the origin. This illustration will be useful when the concepts of dwell and compression are introduced later.

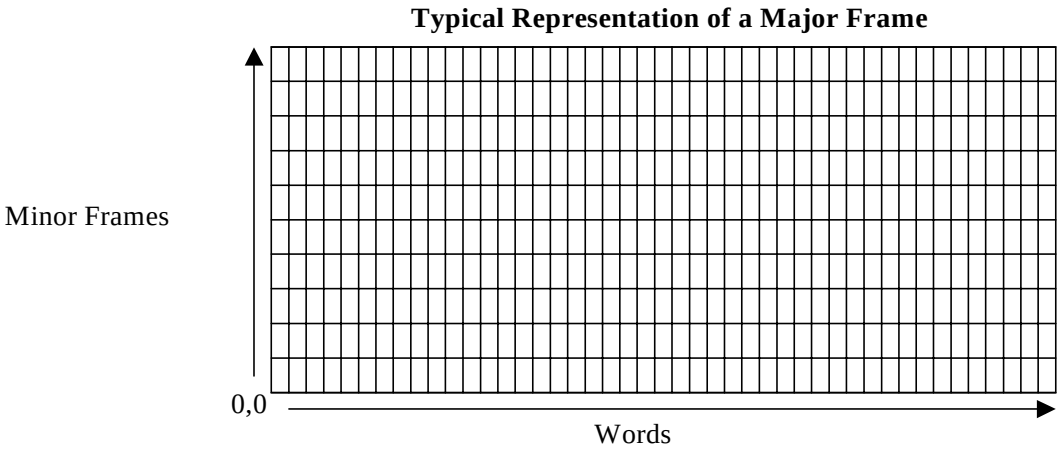


Figure 2-4

Table 2-1 Sample Normal Telemetry Frame File

Parameter	format	minor_frame	word	start_bit	major_frame_rate
30050	1	0	0	0	32
30051	1	0	1	0	32
30052	1	0	2	0	32
30053	1	0	3	0	32
30054	1	0	4	0	32
30055	1	0	4	1	32
30056	1	0	4	2	32
1	1	0	4	5	32
30057	1	0	4	6	32
30058	1	0	4	7	32
2	1	0	5	0	32
30059	1	0	5	2	32
30060	1	0	6	0	1
30076	1	0	6	3	1
31180	1	0	7	0	1
31220	1	0	8	0	1
31060	1	0	9	0	32
31061	1	0	10	0	32
32010	1	0	11	0	32
30130	1	0	12	0	32

Table 2-1 has been filtered for format = 1 records and shows the first twenty records correctly sorted by minor frame, word then start bit. Each column contains information that may not be discarded by any real-time system if it is to correctly interpret the information being received by the spacecraft. Columns 1 through 5 have been discussed already, so the remainder of this section will be on the remaining columns: major frame rate and non-preemptable.

2.2.1.4 Dwell

Dwell is a special TDM mode in which portions of the telemetry stream is *preempted* by customized telemetry requests in specified word locations. This document will not detail all of the unique requirements associated with the dwell function, instead it will explain the interfaces between the dwell function and the data tables.

Configuring for Dwell

Before the telemetry stream will be preempted with dwell telemetry, the operator will configure for dwell using commands to identify which specific telemetry requests will occur in which words of the telemetry stream. The spacecraft must then be commanded to switch from *normal mode* into *dwell mode* before the telemetry stream will actually change. Note that telemetry modes are still dependent upon the format number. The commands to configure for dwell and to switch modes are contained within the database.

Preemptability

Decommutation records contain all of the directions for decommutating the normal-mode (or normal) telemetry stream. When the spacecraft switches into dwell mode, portions of the normal telemetry frame get over-written with dynamic records wherever the record is labeled as *preemptable*.

2.2.2 *Telecommanding*

Command Construction

Directions for the assembling all bits of the common command are stored in the database. The database uses an argument driven structure and to avoid storing repeating values default arguments may be defined at any point in the hierarchal spacecraft level. Specifying this option each time for the thousands of commands is unnecessary and introduces the possibility of inconsistencies within the data.

Command Assembly

Variable (v)

A field type of variable shall be used when the data type of the field depends upon the address of the parameter being loaded. For example, the SCP Memory Upload Command (ID AE) allows the upload of up to 50 16-bit words that start at an operator specified address. The data type of each of these 50 words depends upon each of the 50 addresses. The data type for these variable type fields shall be limited to the following: integer, long integer, 1750A float, 1750A long float, IEEE float, and IEEE long float. These fields will have an associated *sign flag* for data encoding. They will require operator input and the operator's value must be within the associated *minimum value* and *maximum value*. Variable fields are associated with multiword commands since they require four 16-bit fields to transmit the value to the spacecraft.

Odd Parity (p)

A field type of odd parity shall be used for fields whose value must be calculated as "0" or "1" as the command is constructed in order to yield a command with odd parity. This field type requires calculation by the ground software since it typically cannot be calculated in advance and should not be assigned to the operator for determination.

Even Parity (q)

A field type of even parity shall be used for fields whose value must be calculated as “0” or “1” as the command is constructed in order to yield a command with even parity. This field type requires calculation by the ground software since it typically cannot be calculated in advance and should not be assigned to the operator for determination.

Schema Introduction and Usage 3

Contents

This chapter contains the following sections.

Section Title	Page
“Overview”	3-1
“Complete Schema Definition”	3-2

3.1 Overview

The Schema is organized into seven separate W3C schema files: SpaceSystem.xsd, Parameter.xsd, CommonTypes.xsd, Packaging.xsd, Algorithm.xsd, StreamDefinitions.xsd, and CommandDefinitions.xsd.

SpaceSystem.xsd contains the schema for a SpaceSystem. A SpaceSystem is a collection of SpaceSystem(s) including space assets, ground assets, multi-satellite systems and sub-systems. A SpaceSystem is the root element for the set of data necessary to monitor and command an arbitrary space device - this includes the binary decomposition the data streams going into and out of a device. The major data component of a Space System is the TelemetryAndCommanding section where all Telemetric and Command data is stored. Additionally, a Space System has a Header record, containing general information on the data, and 'Algorithms' containing certain well known algorithms and references to external algorithms for mission unique processing.

Parameter.xsd contains the schema for a Parameter. A Parameter is a description of something (usually from telemetry) that can have a value; it is not the value itself. Parameters have a data type, and may have native data type attributes (e.g. integer format). Parameters may also have alarms, validity checks, descriptions and data on how to represent the value of the parameter as a string.

- CommonTypes.xsd – Is a collection of W3C schema data types that are re-used throughout this schema specification.
- Packaging.xsd – Contains the dictionary for messages and containers, which in turn describe the physical composition of data in a communication system.
- Algorithm.xsd – Contains the structure for an Algorithm. An Algorithm may be one of a growing set of pre-defined algorithms or a named escape into a user defined algorithm where (depending on the system) the name of the algorithm may be a java class, a function in a shared library, an external program or some other reference to an outside algorithm.
- StreamDefinitions.xsd - provides a language for defining binary stream data.
- CommandDefinition.xsd - defines the structure for a CommandDefinition. A CommandDefinition provides a description of the command, a name for the command, the allowable arguments for the command, and (optionally) for binary commands destined for an off-platform bus) the opcode for the command and the binary transformation of the command.

3.2 Complete Schema Definition

This section contains the Telemetry and Commanding schema. The Schemas have been validated and checked for well-formedness using the XML Spy 4.4.

The Schemas are presented as in a graphical form as output by XMLSpy. The XML Schema itself is in Appendix A.

3.2.1 Schema SpaceSystem.xsd

schema location:	SpaceSystem.xsd
targetNamespace:	http://www.omg.org/space/tcspec

Elements	**Complex types**
SpaceSystem	SpaceSystemType

schema location:	Algorithm.xsd
targetNamespace:	http://www.omg.org/space/tcspec

Elements	Complex types
FixedFrameSync	CalibratorType
	ConcatenationAlgorithmType
	CustomAlgorithmType
	MathAlgorithmType
	ReedSolomonType
	TriggerType

schema location:	Parameter.xsd
targetNamespace:	http://www.omg.org/space/tcspec

Elements	Complex types
DwellSet	ComplexAlarmsType
	OffsetParameterRefType
	ParameterRefMatchListType
	ParameterRefType
	ParameterType
	PhysicalAddressType
	RangeListType
	SimpleAlarmsType
	StopType
	VariableParameterType

schema location:	CommonTypes.xsd
targetNamespace:	http://www.omg.org/space/tcspec

Complex types	Simple types
AbsoluteTimeType	ComparisonOperatorsType
AliasType	FormatType
BaseDataType	IdentificationNumberType
BaseParameterType	LimitCheckType

BitOffsetType	LongDescriptionType
BusAttributesType	MathOperatorsType
ComparisonCheckType	NameReferenceType
DefaultValueType	NameType
ErrorDetectCorrectType	RadixType
HeaderType	RelativeTimeType
IntegerRangeType	ShortDescriptionType
MathOperationType	
NonStandardDataType	
OccursType	
PolynomialType	
PropertyType	
RangeType	
RealRangeType	
SimpleParameterRefType	
SplinePointType	
SystemNameType	
ToStringType	
UnitType	

schema location:	CommandDefinition.xsd
targetNamespace:	http://www.omg.org/space/tcspec

Complex types	**Simple types**
ArgumentType	InterlockScopeType
CommandConstraintType	StageType
CommandDefinitionType	
CommandVerifierType	

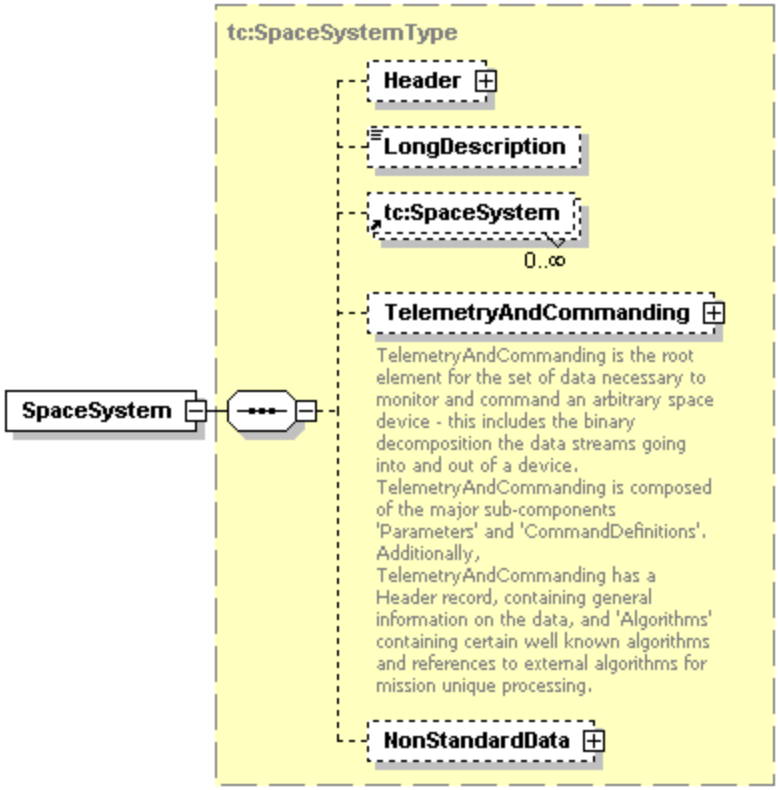
schema location:	Packaging.xsd
targetNamespace:	http://www.omg.org/space/tcspec

Elements	Complex types	Simple types
Packaging	ChoiceType	ContainerNameType
	ContainerRefType	MessageKeyIdType
	ContainerType	MessageNameType
	DynamicContainerRefType	ServiceNameType
	MessageKeyMatchListType	
	MessageKeyMatchType	
	MessageKeyType	
	MessageType	
	PackagingDefinitionType	
	SequenceType	
	ServiceType	

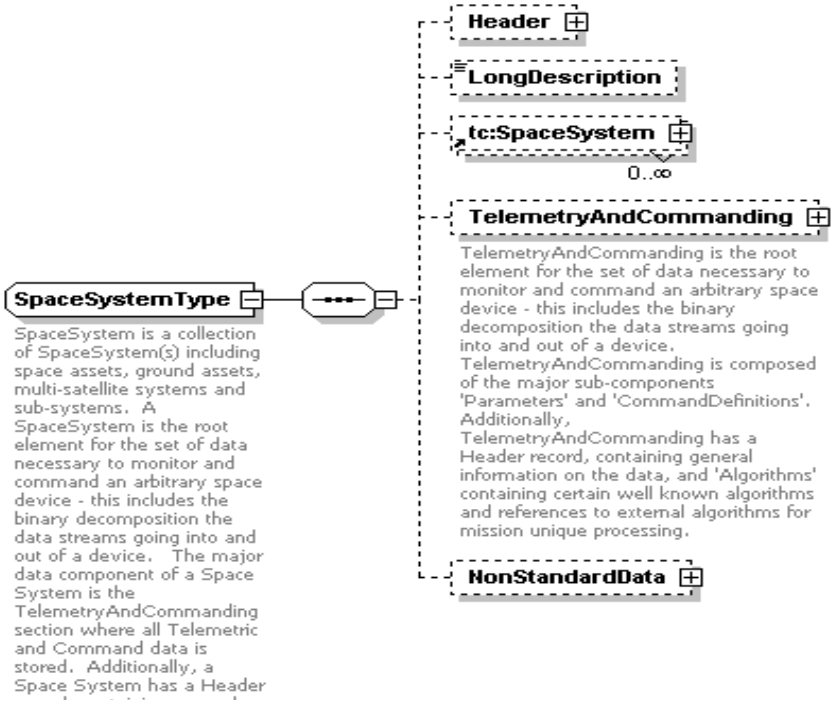
schema location:	StreamDefinitions.xsd
targetNamespace:	http://www.omg.org/space/tcspec

Complex types	Simple types
ConvolutionalStreamType	StreamSourceType
EncryptedStreamType	
FixedFrameStreamType	
PCMStreamType	
StreamType	
SyncStrategyType	
VariableFrameStreamType	

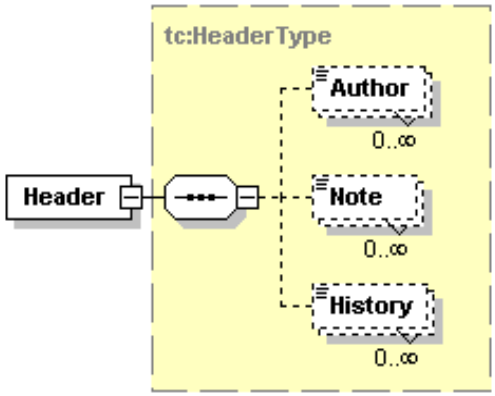
3.2.2 element SpaceSystem

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:SpaceSystemType					
children	Header LongDescription tc:SpaceSystem TelemetryAndCommanding NonStandardData					
used by	complexType SpaceSystemType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string				
	short description	tc:ShortDescriptionType				

3.2.3 complexType SpaceSystemType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	Header LongDescription tc:SpaceSystem TelemetryAndCommanding NonStandardData					
used by	element SpaceSystem					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string				
	short description	tc:ShortDescriptionType				
annotation	documentation	SpaceSystem is a collection of SpaceSystem(s) including space assets, ground assets, multi-satellite systems and sub-systems. A SpaceSystem is the root element for the set of data necessary to monitor and command an arbitrary space device - this includes the binary decomposition the data streams going into and out of a device. The major data component of a Space System is the TelemetryAndCommanding section where all Telemetric and Command data is stored. Additionally, a Space System has a Header record, containing general information on the data, and 'Algorithms' containing certain well known algorithms and references to external algorithms for mission unique processing.				

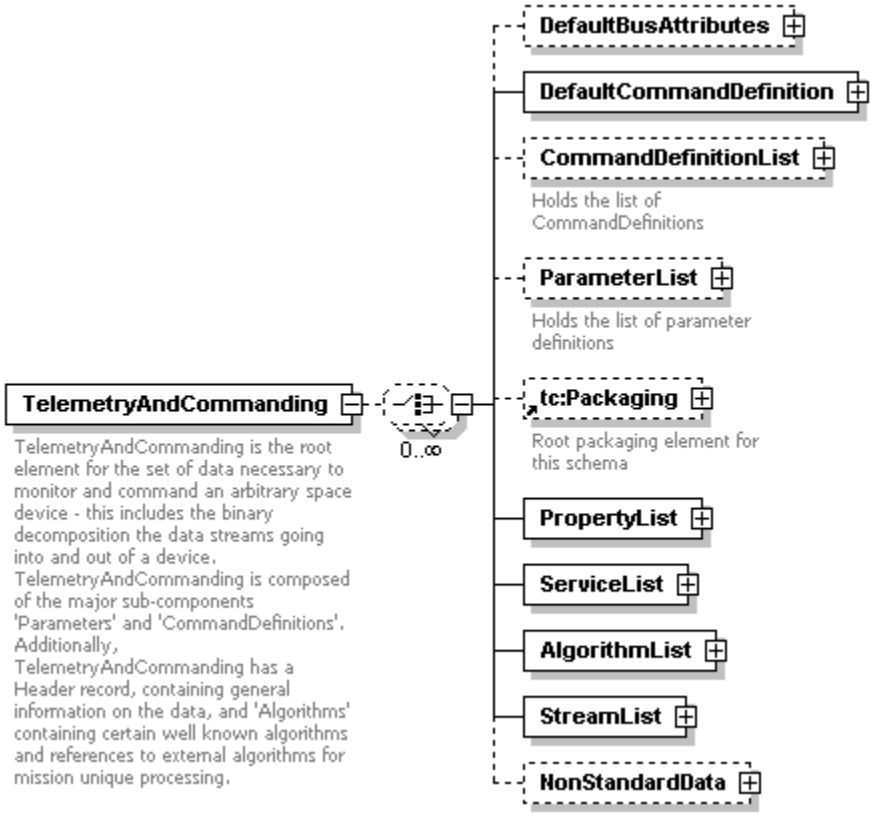
3.2.4 element SpaceSystemType/Header

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:HeaderType					
children	Author Note History					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	version	string				
	date	string				
	classification	string		Unclassified		

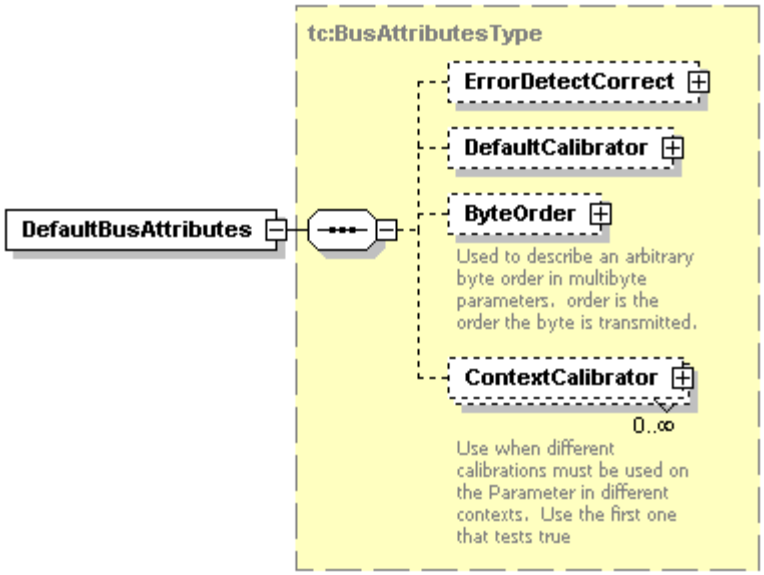
3.2.5 element SpaceSystemType/LongDescription

diagram	
namespace	http://www.omg.org/space/tcspec
type	tc:LongDescriptionType

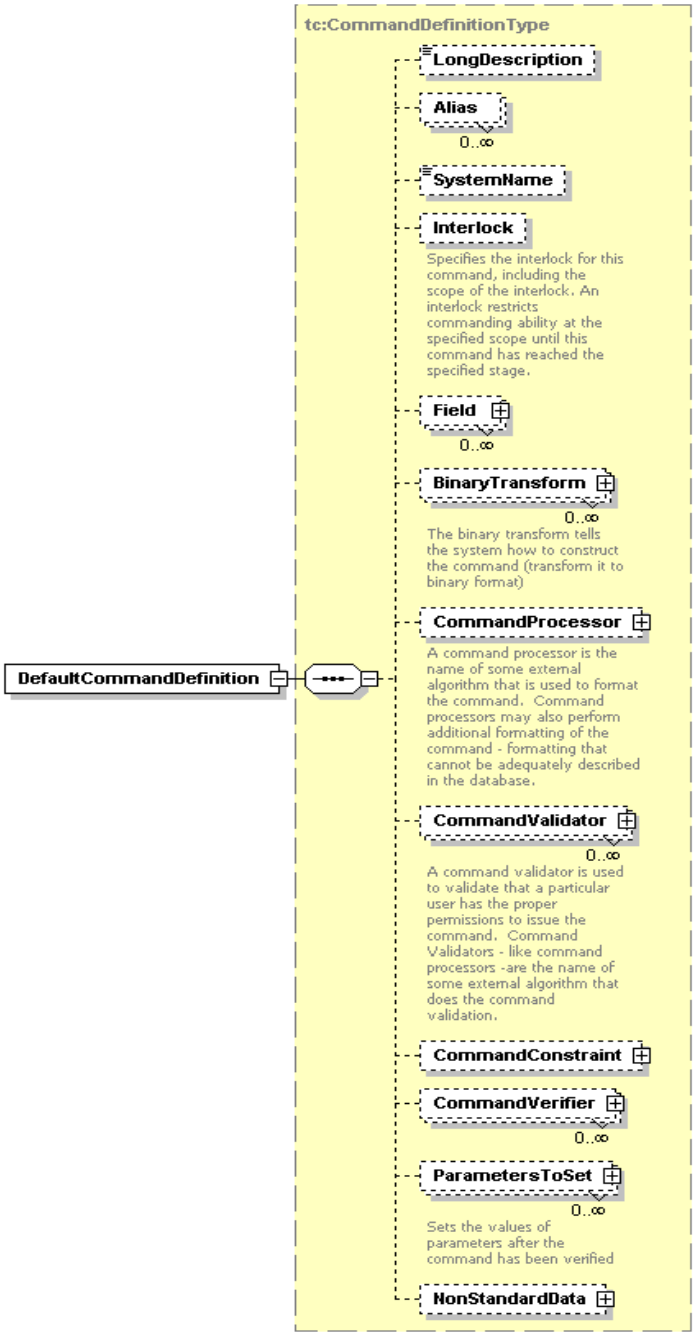
3.2.6 element SpaceSystemType/TelemetryAndCommanding

diagram		
namespace	http://www.omg.org/space/tcspec	
children	DefaultBusAttributes DefaultCommandDefinition CommandDefinitionList ParameterList tc:Packaging PropertyList ServiceList AlgorithmList StreamList NonStandardData	
annotation	documentation	TelemetryAndCommanding is the root element for the set of data necessary to monitor and command an arbitrary space device - this includes the binary decomposition the data streams going into and out of a device. TelemetryAndCommanding is composed of the major sub-components 'Parameters' and 'CommandDefinitions'. Additionally, TelemetryAndCommanding has a Header record, containing general information on the data, and 'Algorithms' containing certain well known algorithms and references to external algorithms for mission unique processing.

3.2.7 element SpaceSystemType/TelemetryAndCommanding/DefaultBusAttributes

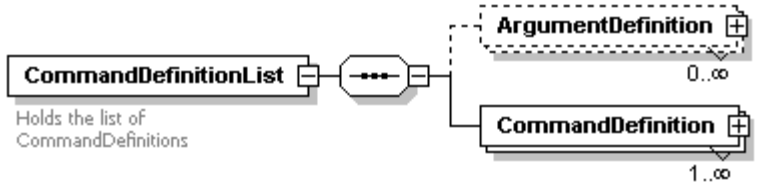
diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:BusAttributesType					
children	ErrorDetectCorrect DefaultCalibrator ByteOrder ContextCalibrator					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	bitLength	unsignedInt	required			
	bitOrder	string		MostSignificantBitFirst		
	busType	string	required			

3.2.8 element SpaceSystemType/TelemetryAndCommanding/ DefaultCommandDefinition

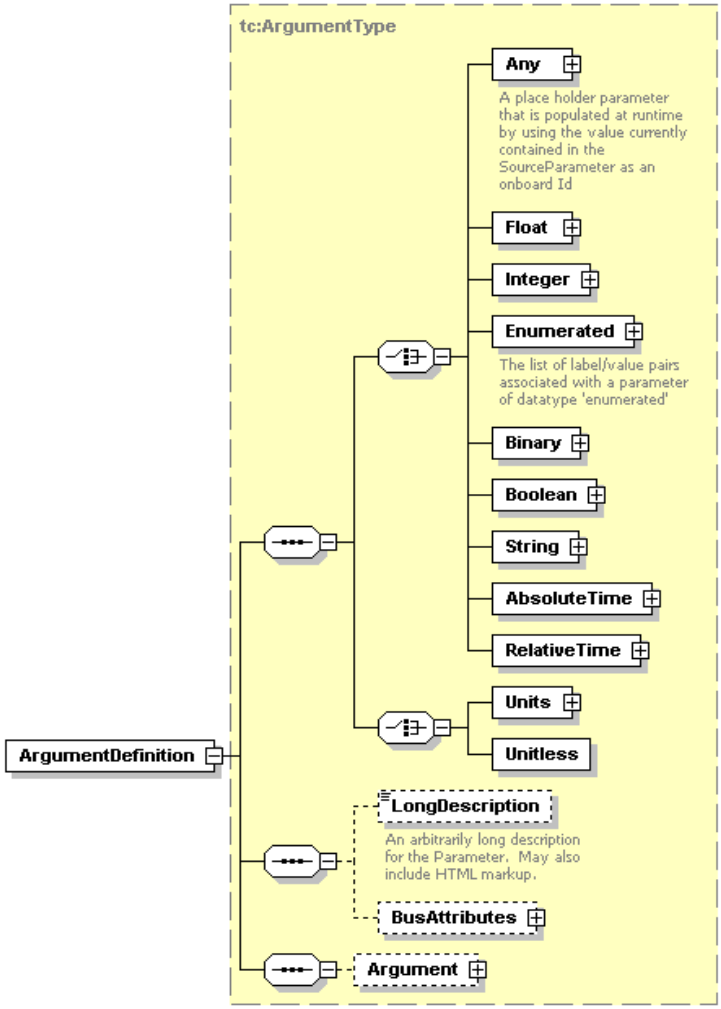
diagram	
namespace	http://www.omg.org/space/tcspec
type	tc:CommandDefinitionType

children	LongDescription Alias SystemName Interlock Field BinaryTransform CommandProcessor CommandValidator CommandConstraint CommandVerifier ParametersToSet NonStandardData					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescription-Type	optional			
	critical	boolean	optional	false		
	highPriority	boolean	optional	false		

3.2.9 element SpaceSystemType/TelemetryAndCommanding/ CommandDefinitionList

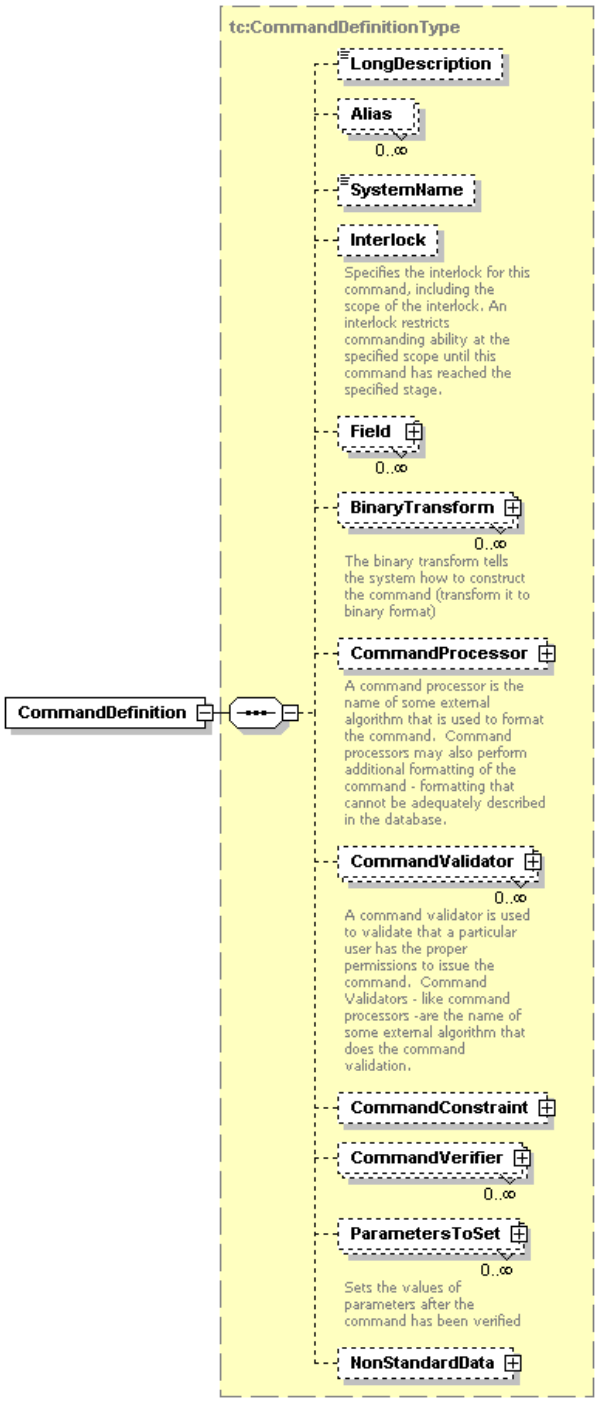
diagram					
namespace	http://www.omg.org/space/tcspec				
children	ArgumentDefinition CommandDefinition				
identity constraints		**Name**	**Refer**	**Selector**	**Field(s)**
	key	CommandDefinitionListKey		./CommandDefinition	@name
annotation	documentation	Holds the list of CommandDefinitions.			

3.2.10 element SpaceSystemType/TelemetryAndCommanding/ CommandDefinitionList/ArgumentDefinition

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:ArgumentType					
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes Argument					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	ShortDescription	tc:ShortDescriptionType	optional			

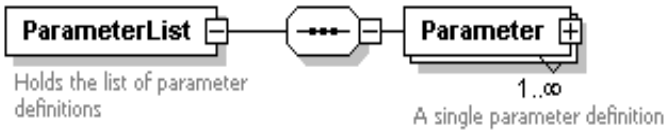
3.2.11 element SpaceSystemType/TelemetryAndCommanding/CommandDefinitionList /CommandDefinition

diagram



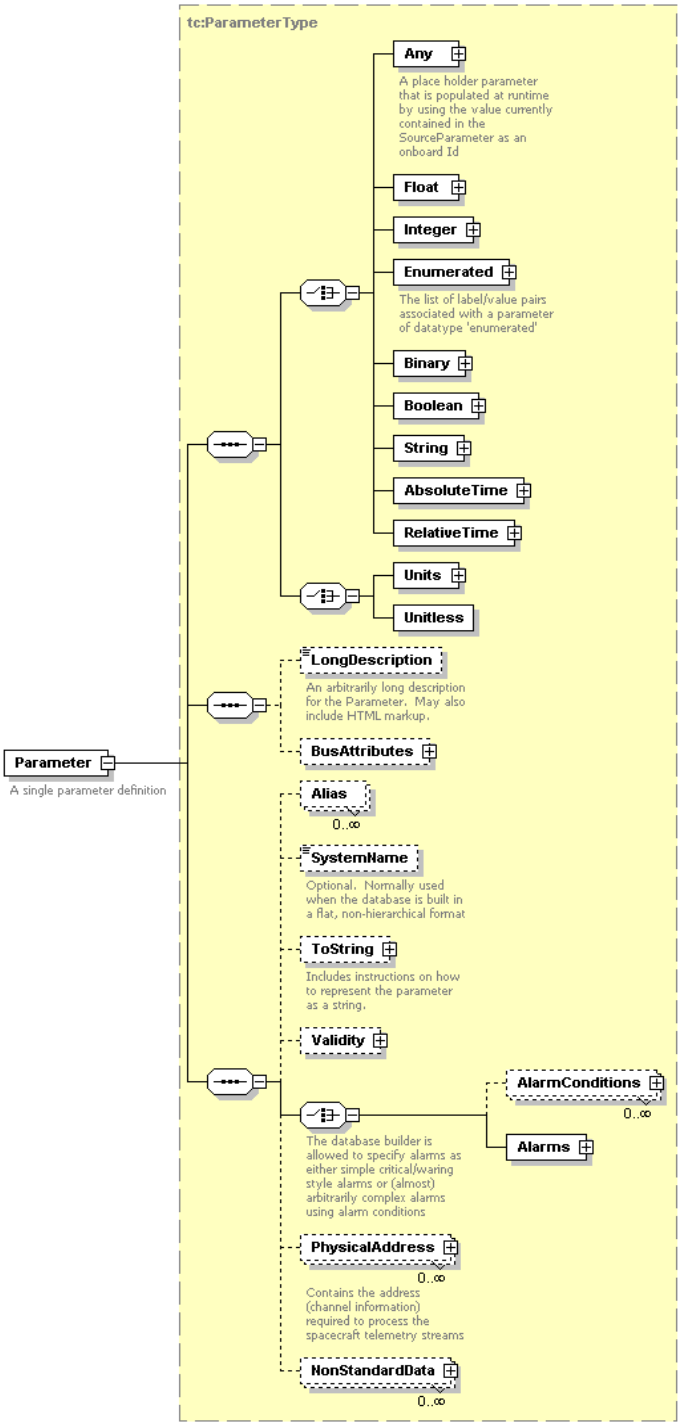
namespace	http://www.omg.org/space/tcspec					
type	tc:CommandDefinitionType					
children	LongDescription Alias SystemName Interlock Field BinaryTransform CommandProcessor CommandValidator CommandConstraint CommandVerifier ParametersToSet NonStandardData					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescription-Type	optional			
	critical	boolean	optional	false		
	highPriority	boolean	optional	false		

3.2.12 element SpaceSystemType/TelemetryAndCommanding/ParameterList

diagram					
namespace	http://www.omg.org/space/tcspec				
children	Parameter				
identity constraints		**Name**	**Refer**	**Selector**	**Field(s)**
	key	ParameterListKey		./Parameter	@name
annotation	documentation	Holds the list of parameter definitions.			

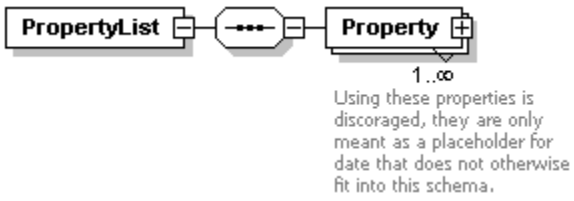
3.2.13 element SpaceSystemType/TelemetryAndCommanding/ ParameterList/Parameter

diagram

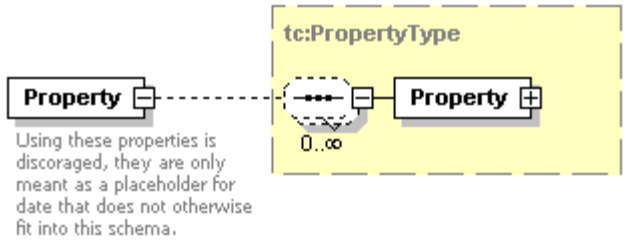


namespace	http://www.omg.org/space/tcspec					
type	tc:ParameterType					
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes Alias SystemName ToString Validity AlarmConditions Alarms PhysicalAddress NonStandardData					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	ShortDescription	tc:ShortDescriptionType	optional			
	dataSource	string	optional			
	readOnly	boolean	optional	false		
	validRangeAppliesToCalibrated	boolean	optional	true		
	onboardID	string	optional			
annotation	documentation	A single parameter definition.				

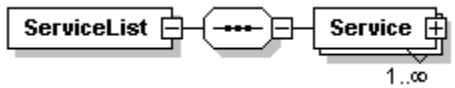
3.2.14 element SpaceSystemType/TelemetryAndCommanding/PropertyList

diagram	
namespace	http://www.omg.org/space/tcspec
children	Property

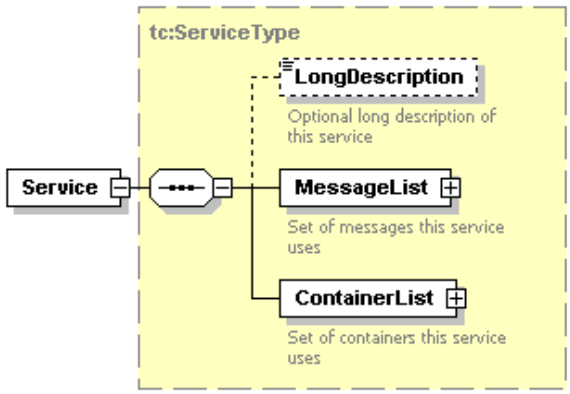
3.2.15 element SpaceSystemType/TelemetryAndCommanding/PropertyList/Property

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:PropertyType					
children	Property					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameReferenceType	required			
	shortDescription	string				
	longDescription	string				
	value	string	required			
annotation	documentation	Using these properties is discoraged, they are only meant as a placeholder for date that does not otherwise fit into this schema.				

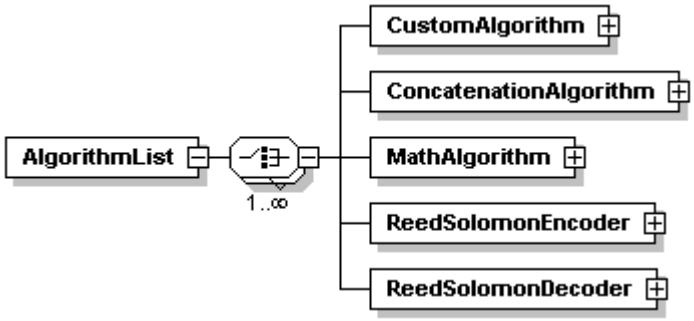
3.2.16 element SpaceSystemType/TelemetryAndCommanding/ServiceList

diagram					
namespace	http://www.omg.org/space/tcspec				
children	Service				
identity constraints		**Name**	**Refer**	**Selector**	**Field(s)**
	key	ServiceListKey		./Service	@name

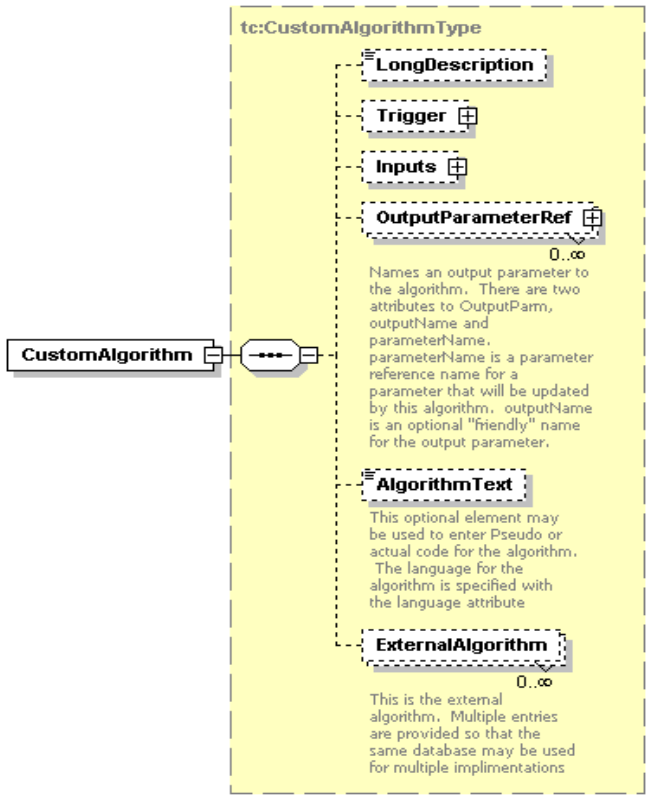
3.2.17 element SpaceSystemType/TelemetryAndCommanding/ServiceList/Service

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:ServiceType					
children	LongDescription MessageList ContainerList					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:ServiceNameType	required			
	shortDescription	tc:ShortDescriptionType	optional			
annotation	documentation	Unique identifier for this service set.				
	documentation	Optional short description of this service.				

3.2.18 element SpaceSystemType/TelemetryAndCommanding/AlgorithmList

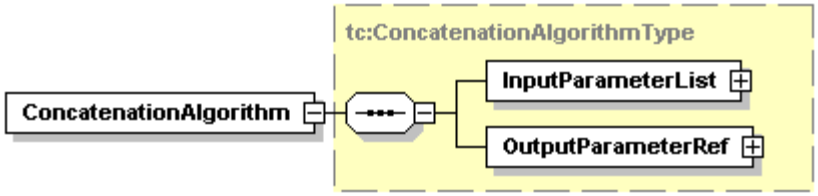
diagram	
namespace	http://www.omg.org/space/tcspec
children	CustomAlgorithm ConcatenationAlgorithm MathAlgorithm ReedSolomonEncoder ReedSolomonDecoder

3.2.19 element SpaceSystemType/TelemetryAndCommanding/AlgorithmList/ CustomAlgorithm

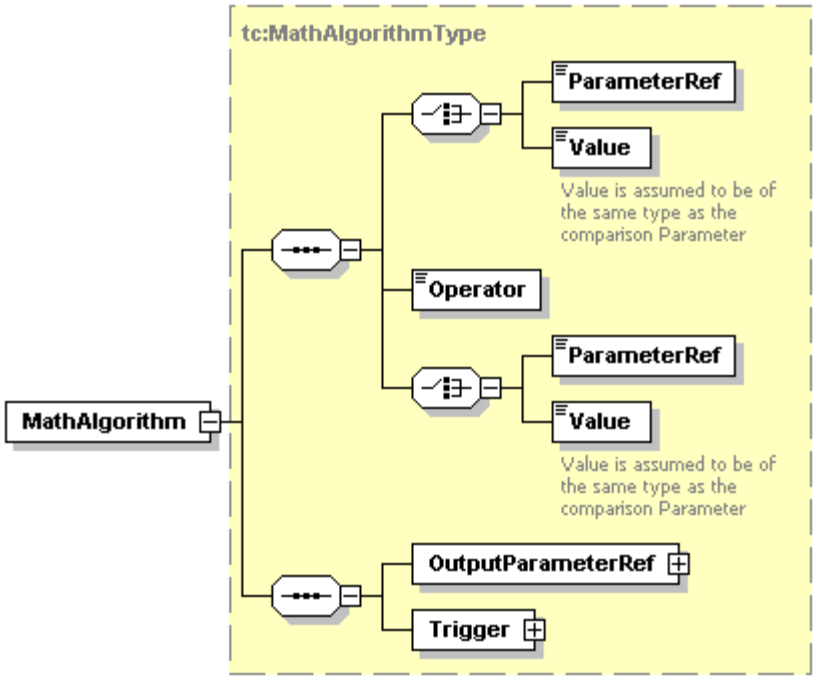
diagram	

namespace	http://www.omg.org/space/tcspec					
type	tc:CustomAlgorithmType					
children	LongDescription Trigger Inputs OutputParameterRef AlgorithmText ExternalAlgorithm					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	required			
	shortDescription	tc:ShortDescriptionType	optional			
	thread	boolean	optional			
	triggerContainer	string	optional			
	priority	integer	optional			
annotation	documentation	First telemetry container from which the output parameter should be calculated.				
	documentation	Algorithm processing priority.				

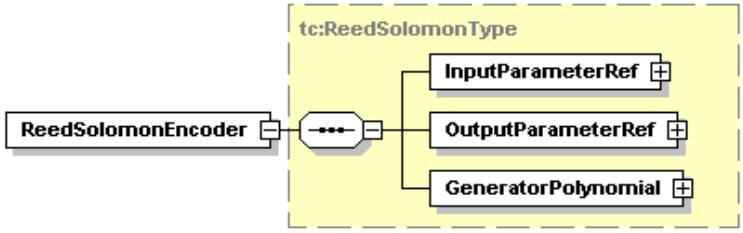
3.2.20 element SpaceSystemType/TelemetryAndCommanding/AlgorithmList/ ConcatenationAlgorithm

diagram	
namespace	http://www.omg.org/space/tcspec
type	tc:ConcatenationAlgorithmType
children	InputParameterList OutputParameterRef

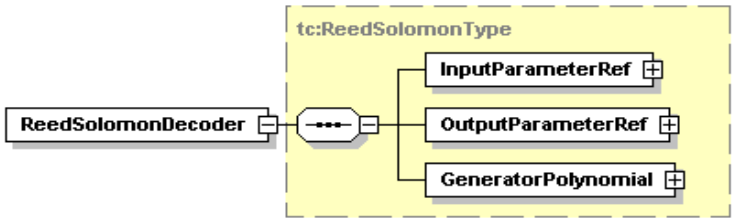
3.2.21 element SpaceSystemType/TelemetryAndCommanding/AlgorithmList/MathAlgorithm

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:MathAlgorithmType					
children	ParameterRef Value Operator ParameterRef Value OutputParameterRef Trigger					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	trigger	tc:NameReferenceType	required			

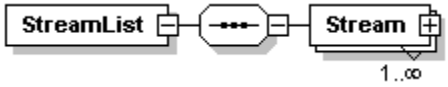
3.2.22 element SpaceSystemType/TelemetryAndCommanding/AlgorithmList/ReedSolomonEncoder

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:ReedSolomonType					
children	InputParameterRef OutputParameterRef GeneratorPolynomial					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	inputParameterRef	tc:NameReferenceType	required			
	outputParameterRef	tc:NameReferenceType	required			

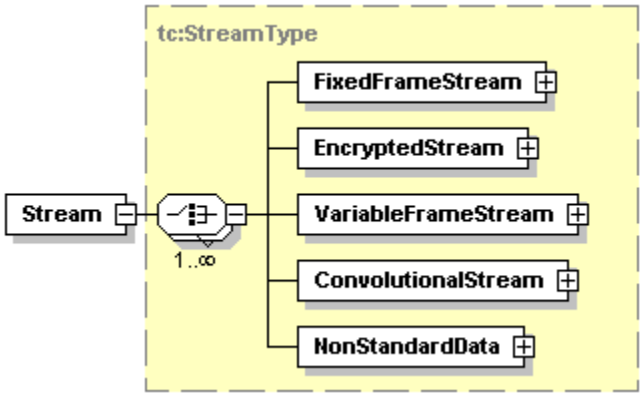
3.2.23 element SpaceSystemType/TelemetryAndCommanding/AlgorithmList/ReedSolomonDecoder

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:ReedSolomonType					
children	InputParameterRef OutputParameterRef GeneratorPolynomial					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	inputParameterRef	tc:NameReferenceType	required			
	outputParameterRef	tc:NameReferenceType	required			

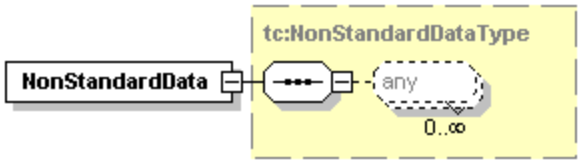
3.2.24 element SpaceSystemType/TelemetryAndCommanding/StreamList

diagram	
namespace	http://www.omg.org/space/tcspec
children	Stream

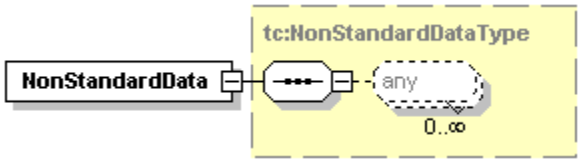
3.2.25 element SpaceSystemType/TelemetryAndCommanding/StreamList/Stream

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:StreamType					
children	FixedFrameStream EncryptedStream VariableFrameStream ConvolutionalStream NonStandardData					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	streamSource	tc:StreamSourceType	required			

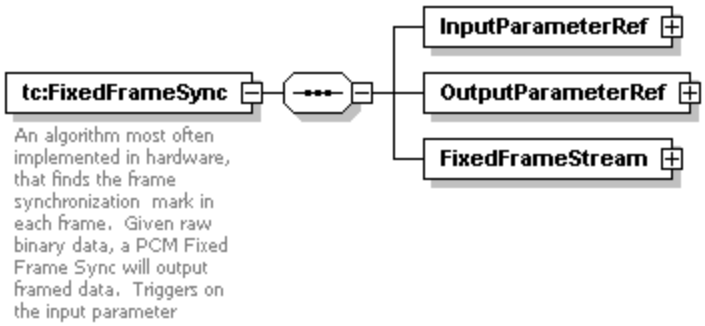
3.2.26 element SpaceSystemType/TelemetryAndCommanding/NonStandardData

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:NonStandardDataType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	usage	string	required			

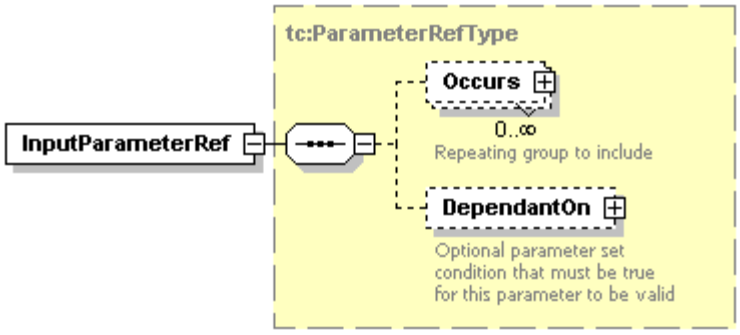
3.2.27 element SpaceSystemType/NonStandardData

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:NonStandardDataType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	usage	string	required			

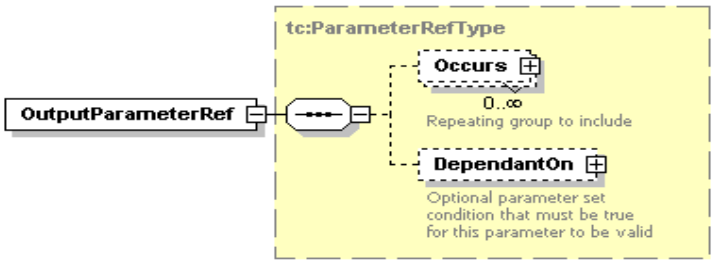
3.2.28 element tc:FixedFrameSync

diagram		
namespace	http://www.omg.org/space/tcspec	
children	InputParameterRef OutputParameterRef FixedFrameStream	
annotation	documentation	An algorithm most often implemented in hardware, that finds the frame synchronization mark in each frame. Given raw binary data, a PCM Fixed Frame Sync will output framed data. Triggers on the input parameter.

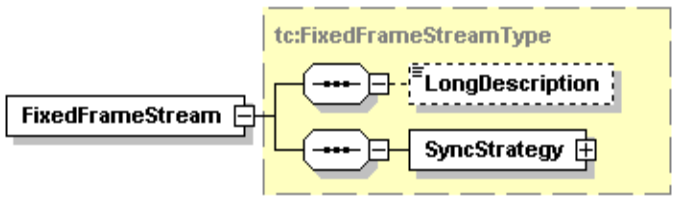
3.2.29 element tc:FixedFrameSync/InputParameterRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

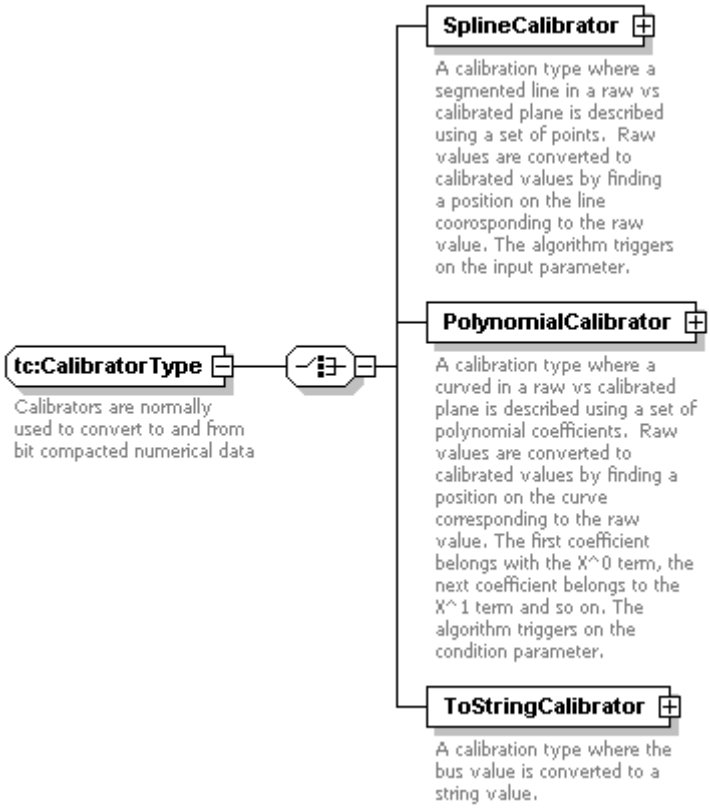
3.2.30 element tc:FixedFrameSync/OutputParameterRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

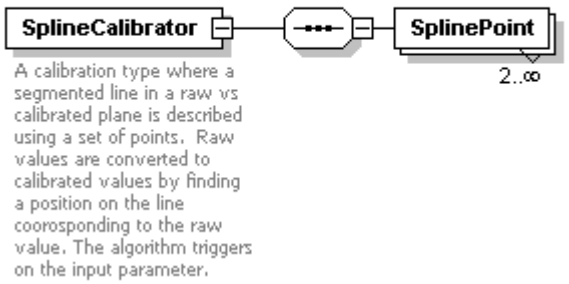
3.2.31 element tc:FixedFrameSync/FixedFrameStream

diagram						
type	tc:FixedFrameStreamType					
children	LongDescription SyncStrategy					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType				
	shortDescription	tc:ShortDescriptionType				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
	frameLengthInBits	long	required			

3.2.32 complexType tc:CalibratorType

<table>
<tr><td>diagram</td><td colspan="6"> 
</td></tr>
<tr><td>namespace</td><td colspan="6">http://www.omg.org/space/tcspec</td></tr>
<tr><td>children</td><td colspan="6">SplineCalibrator PolynomialCalibrator ToStringCalibrator</td></tr>
<tr><td>used by</td><td colspan="6">elements: tc:BusAttributesType/ContextCalibrator/Calibrator tc:BusAttributesType/DefaultCalibrator</td></tr>
<tr><td rowspan="3">attributes</td><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>name</td><td>string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>shortDescription</td><td>tc:ShortDescriptionType</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>annotation</td><td>documentation</td><td colspan="5">Calibrators are normally used to convert to and from bit compacted numerical data.</td></tr>
</table>

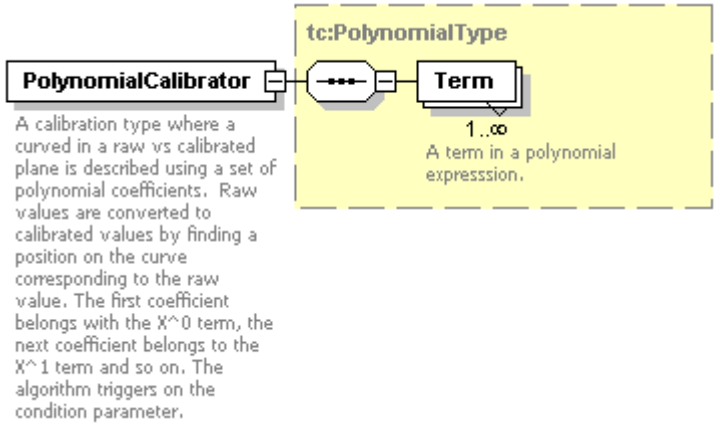
3.2.33 element tc:CalibratorType/SplineCalibrator

diagram						
children	SplinePoint					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	order	positiveInteger		1		
	extrapolate	boolean		false		
annotation	documentation	A calibration type where a segmented line in a raw vs. calibrated plane is described using a set of points. Raw values are converted to calibrated values by finding a position on the line corresponding to the raw value. The algorithm triggers on the input parameter.				

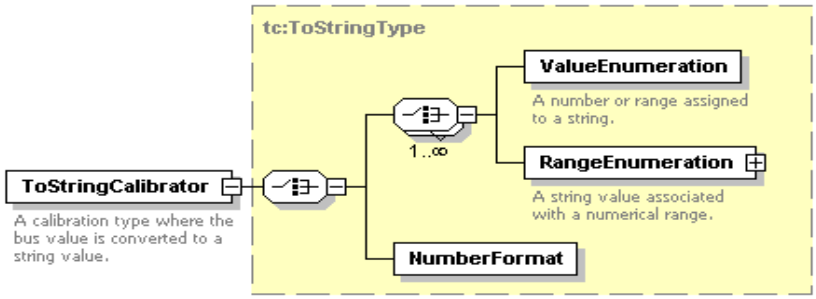
3.2.34 element tc:CalibratorType/SplineCalibrator/SplinePoint

diagram						
type	tc:SplinePointType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	order	positiveInteger				
	raw	double	required			
	calibrated	double	required			

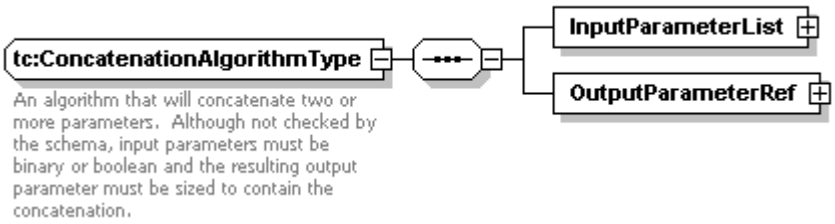
3.2.35 element tc:CalibratorType/PolynomialCalibrator

diagram		
type	tc:PolynomialType	
children	Term	
annotation	documentation	A calibration type where a curved in a raw vs calibrated plane is described using a set of polynomial coefficients. Raw values are converted to calibrated values by finding a position on the curve corresponding to the raw value. The first coefficient belongs with the X^0 term, the next coefficient belongs to the X^1 term and so on. The algorithm triggers on the condition parameter.

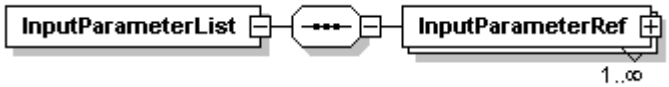
3.2.36 element tc:CalibratorType/ToStringCalibrator

diagram		
type	tc:ToStringType	
children	ValueEnumeration RangeEnumeration NumberFormat	
annotation	documentation	A calibration type where the bus value is converted to a string value.

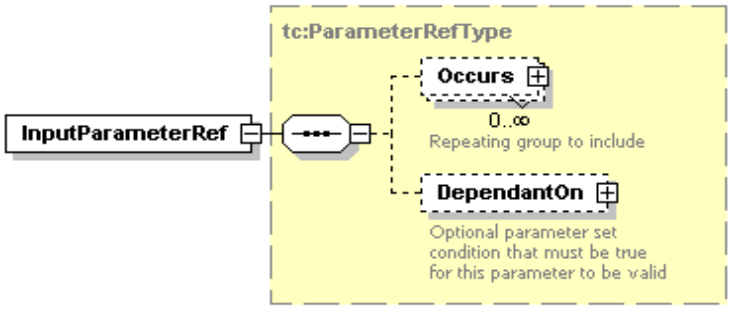
3.2.37 complexType tc:ConcatenationAlgorithmType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	InputParameterList OutputParameterRef	
used by	element: SpaceSystemType/TelemetryAndCommanding/AlgorithmList/ConcatenationAlgorithm	
annotation	documentation	An algorithm that will concatenate two or more parameters. Although not checked by the schema, input parameters must be binary or boolean and the resulting output parameter must be sized to contain the concatenation.

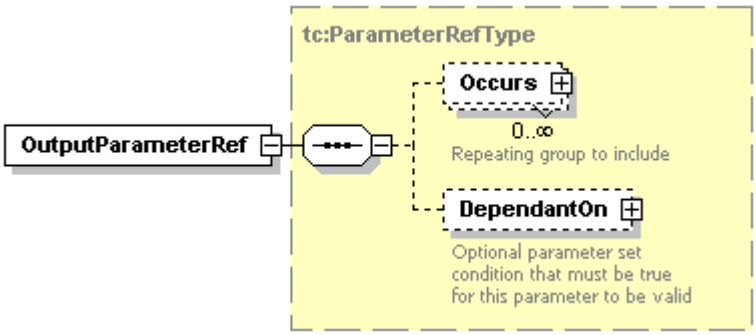
3.2.38 element tc:ConcatenationAlgorithmType/InputParameterList

diagram	
children	InputParameterRef

3.2.39 element tc:ConcatenationAlgorithmType/InputParameterList/InputParameterRef

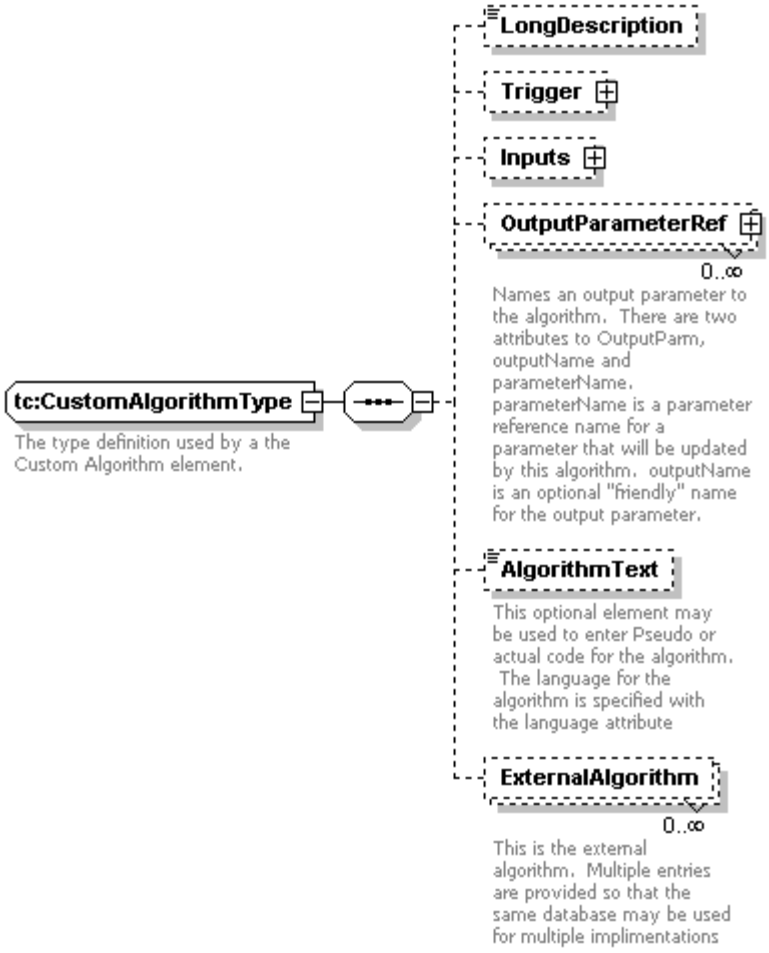
diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

3.2.40 element tc:ConcatenationAlgorithmType/OutputParameterRef

diagram	
type	tc:ParameterRefType
children	Occurs DependantOn

attributes	Name	Type	Use	Default	Fixed	Annotation
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

3.2.41 complexType tc:CustomAlgorithmType

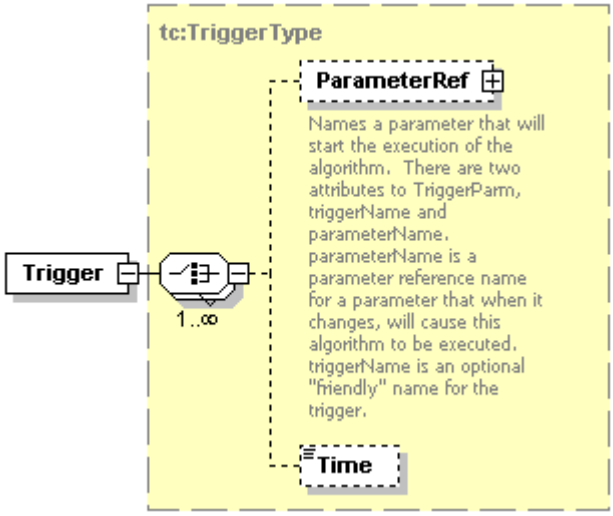
diagram	
namespace	http://www.omg.org/space/tcspec
children	LongDescription Trigger Inputs OutputParameterRef AlgorithmText ExternalAlgorithm

used by	elements: tc:CommandDefinitionType/CommandProcessor tc:CommandDefinitionType/CommandValidator tc:ComplexAlarmsType/CustomAlarmCheck SpaceSystemType/TelemetryAndCommanding/AlgorithmList/CustomAlgorithm tc:CommandConstraintType/CustomAlgorithm tc:CommandVerifierType/CustomAlgorithm					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	required			
	shortDescription	tc:ShortDescriptionType	optional			
	thread	boolean	optional			
	triggerContainer	string	optional			documentation: First telemetry container from which the output parameter should be calculated.
	priority	integer	optional			documentation: Algorithm processing priority
annotation	documentation	The type definition used by a the Custom Algorithm element.				
	documentation	Complete algorithm written in the algorithm language. Embedded new line characters are legal within this attribute. All algorithms should contain a change log that is modified with each update.				
	documentation	A hint to the ground system to tell it to calculate the algorithm in a new processing thread.				

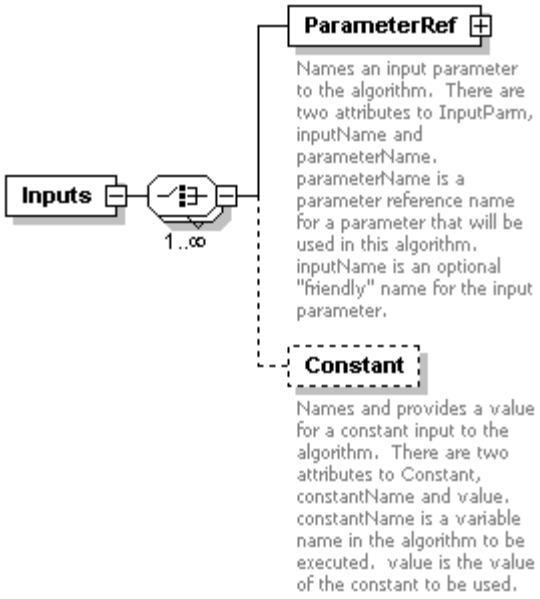
3.2.42 element tc:CustomAlgorithmType/LongDescription

diagram	
type	tc:LongDescriptionType

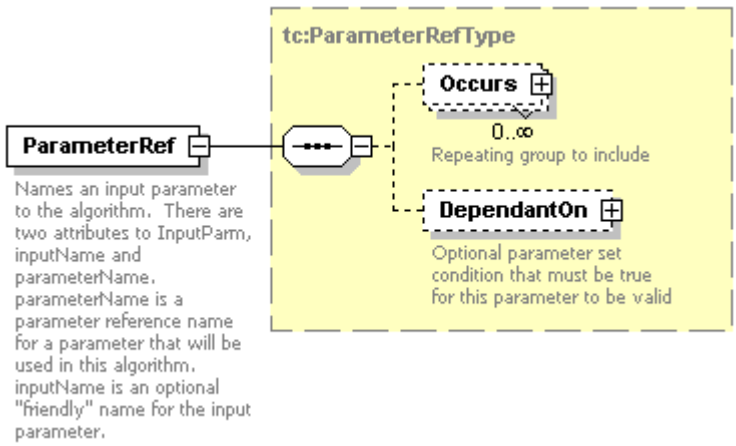
3.2.43 element tc:CustomAlgorithmType/Trigger

diagram						
type	tc:TriggerType					
children	ParameterRef Time					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	optional			
	triggerRate	nonNegativeInteger	optional	1		

3.2.44 element tc:CustomAlgorithmType/Inputs

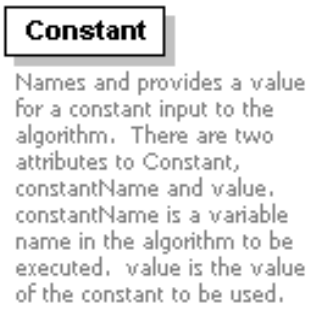
diagram	
children	ParameterRef Constant

3.2.45 element tc:CustomAlgorithmType/Inputs/ParameterRef

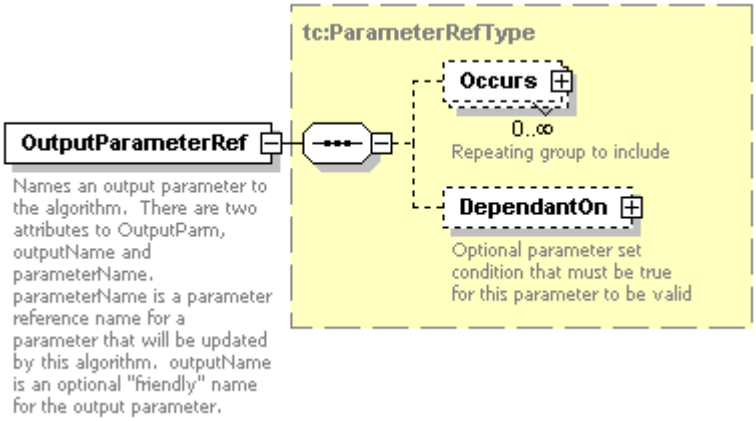
diagram	
type	extension of tc:ParameterRefType
children	Occurs DependantOn

attributes	Name	Type	Use	Default	Fixed	Annotation
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
	inputName	string				
annotation	documentation	Names an input parameter to the algorithm. There are two attributes to InputParm, inputName and parameterName. parameterName is a parameter reference name for a parameter that will be used in this algorithm. inputName is an optional "friendly" name for the input parameter.				

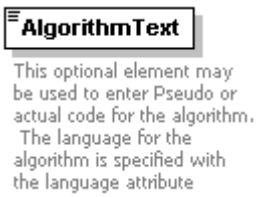
3.2.46 element tc:CustomAlgorithmType/Inputs/Constant

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	constantName	string				
	value	string	required			
annotation	documentation	Names and provides a value for a constant input to the algorithm. There are two attributes to Constant, constantName and value. constantName is a variable name in the algorithm to be executed. value is the value of the constant to be used.				

3.2.47 element tc:CustomAlgorithmType/OutputParameterRef

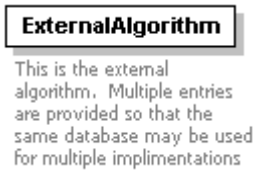
diagram						
type	extension of tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
	outputName	string				
annotation	documentation	Names an output parameter to the algorithm. There are two attributes to OutputParm, outputName and parameterName. parameterName is a parameter reference name for a parameter that will be updated by this algorithm. outputName is an optional "friendly" name for the output parameter.				

3.2.48 element tc:CustomAlgorithmType/AlgorithmText

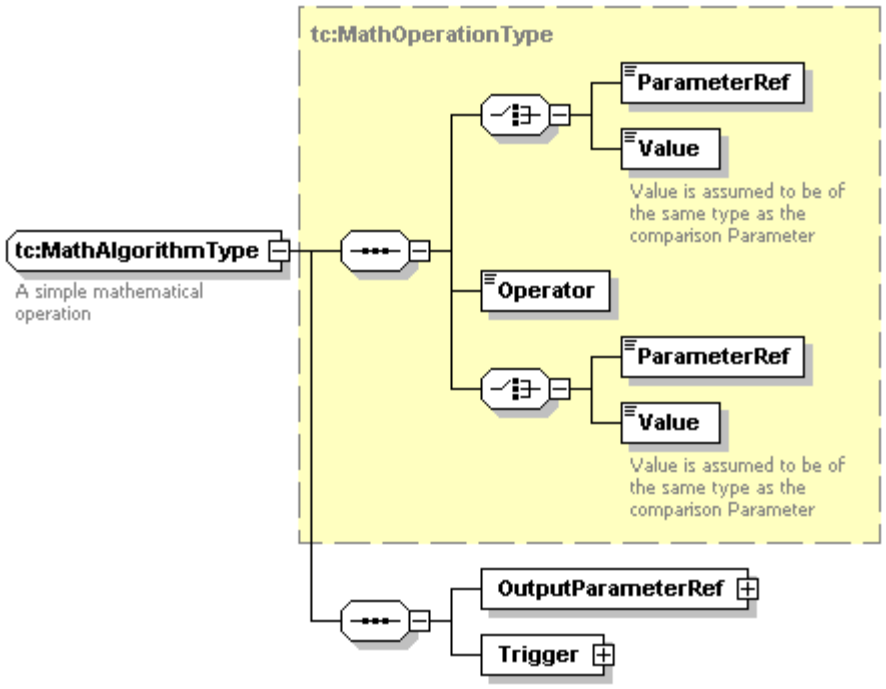
diagram	
type	extension of string

attributes	Name	Type	Use	Default	Fixed	Annotation
	language	string		pseudo		
annotation	documentation	This optional element may be used to enter Pseudo or actual code for the algorithm. The language for the algorithm is specified with the language attribute				

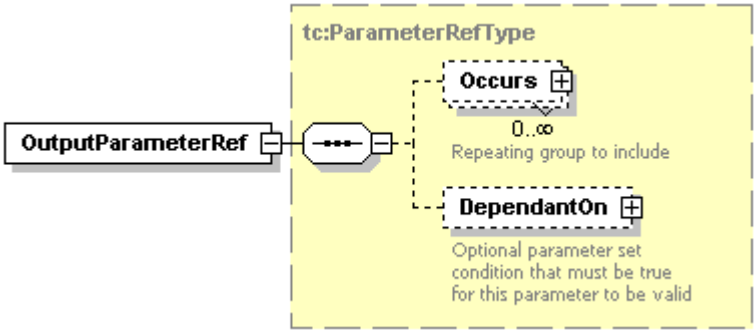
3.2.49 element tc:CustomAlgorithmType/ExternalAlgorithm

diagram						
						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	implementationName	string				
	algorithmLocation	string	required			
annotation	documentation	This is the external algorithm. Multiple entries are provided so that the same database may be used for multiple implementations.				

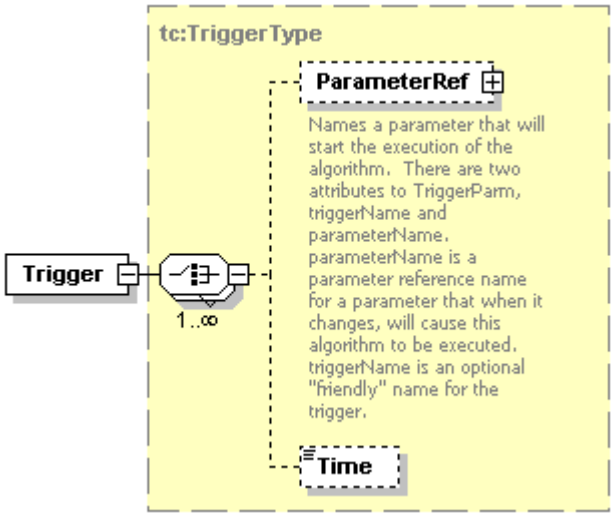
3.2.50 complexType tc:MathAlgorithmType

diagram						
namespace	http://www.omg.org/space/tcspec					
type	extension of tc:MathOperationType					
children	ParameterRef Value Operator ParameterRef Value OutputParameterRef Trigger					
used by	element: SpaceSystemType/TelemetryAndCommanding/AlgorithmList/MathAlgorithm					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	trigger	tc:NameReferenceType	required			
annotation	documentation	A simple mathematical operation.				

3.2.51 element tc:MathAlgorithmType/OutputParameterRef

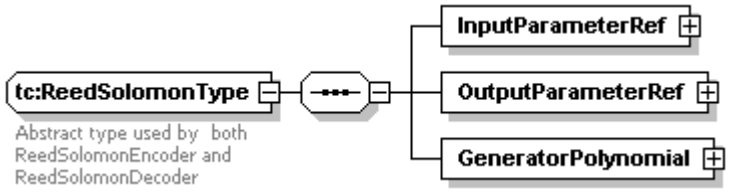
diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

3.2.52 element tc:MathAlgorithmType/Trigger

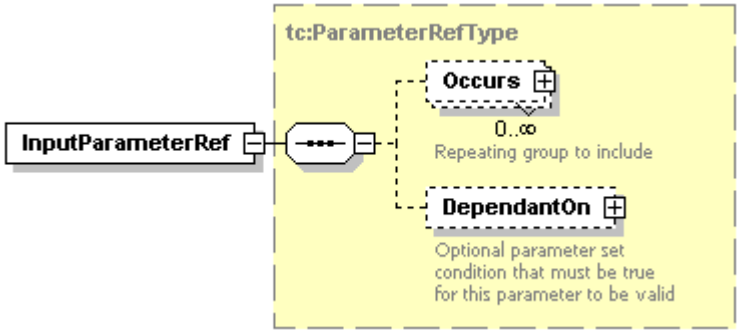
diagram	
type	tc:TriggerType

children	ParameterRef Time					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	optional			
	triggerRate	nonNegativeInteger	optional	1		

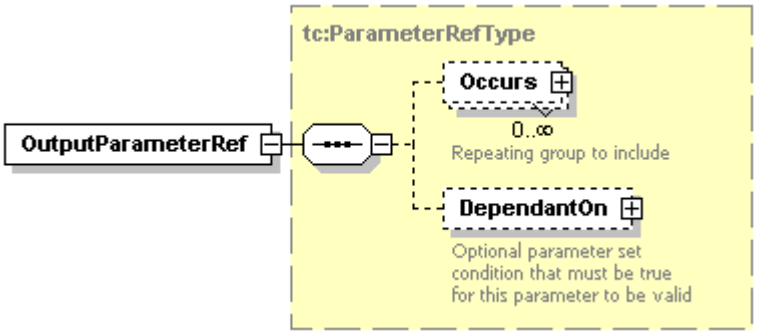
3.2.53 complexType tc:ReedSolomonType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	InputParameterRef OutputParameterRef GeneratorPolynomial					
used by	elements: SpaceSystemType/TelemetryAndCommanding/AlgorithmList/ReedSolomonDecoder SpaceSystemType/TelemetryAndCommanding/AlgorithmList/ReedSolomonEncoder					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	inputParameterRef	tc:NameReferenceType	required			
	outputParameterRef	tc:NameReferenceType	required			
annotation	documentation	Abstract type used by both ReedSolomonEncoder and ReedSolomonDecoder.				

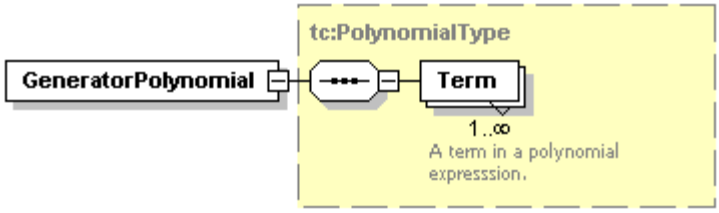
3.2.54 element tc:ReedSolomonType/InputParameterRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

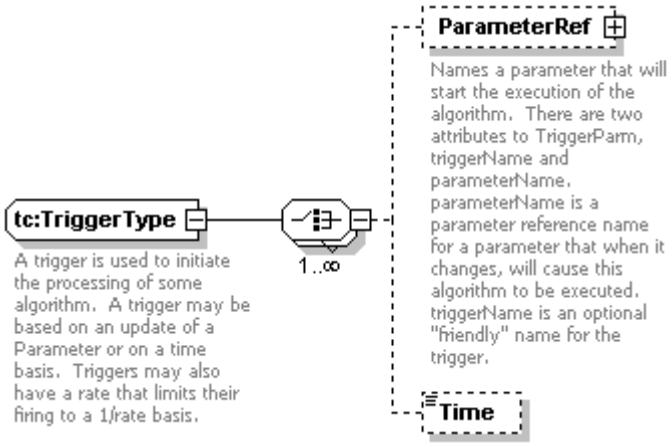
3.2.55 element tc:ReedSolomonType/OutputParameterRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

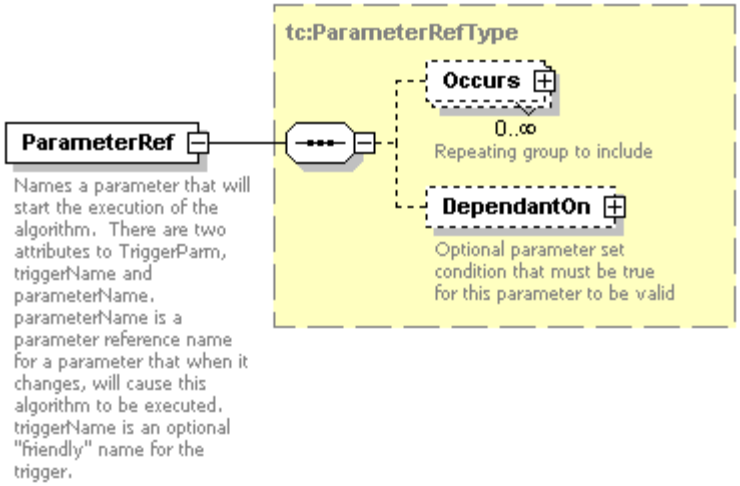
3.2.56 element tc:ReedSolomonType/GeneratorPolynomial

diagram	
type	tc:PolynomialType
children	Term

3.2.57 complexType tc:TriggerType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	ParameterRef Time					
used by	element: tc:CustomAlgorithmType/Trigger tc:MathAlgorithmType/Trigger					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	optional			
	triggerRate	nonNegativeInteger	optional	1		
annotation	documentation	A trigger is used to initiate the processing of some algorithm. A trigger may be based on an update of a Parameter or on a time basis. Triggers may also have a rate that limits their firing to a 1/rate basis.				

3.2.58 element tc:TriggerType/ParameterRef

diagram						
type	extension of tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
annotation	documentation	Names a parameter that will start the execution of the algorithm. There are two attributes to TriggerParm, triggerName and parameterName. parameterName is a parameter reference name for a parameter that when it changes, will cause this algorithm to be executed. triggerName is an optional "friendly" name for the trigger.				

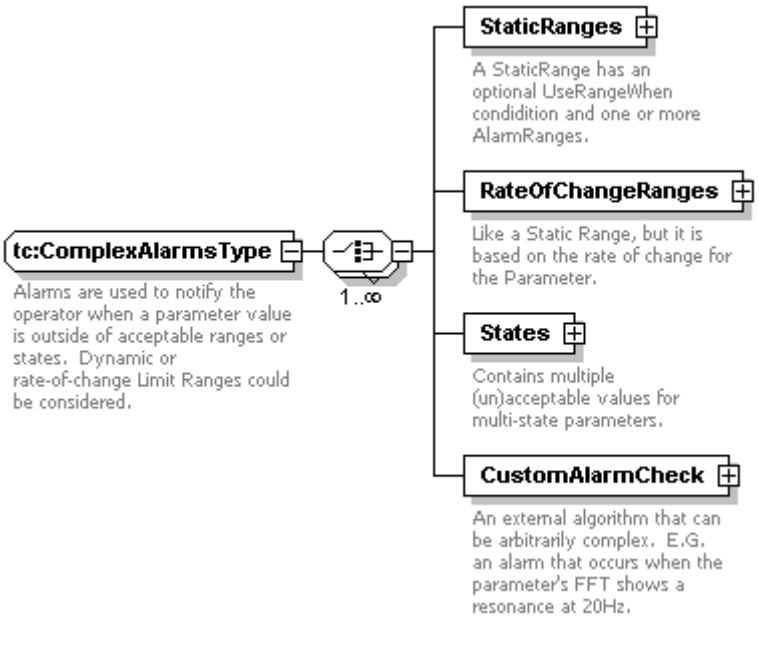
3.2.59 element tc:TriggerType/Time

diagram	
type	tc:RelativeTimeType

3.2.60 element tc:DwellSet

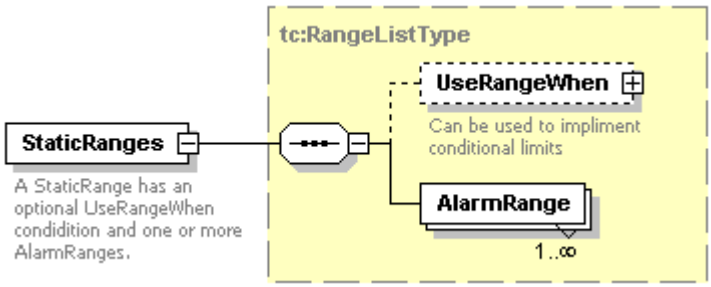
diagram						
namespace	http://www.omg.org/space/tcspec					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	onboardID	string	required			
	address	string	required			
	startBit	nonNegativeInteger	required			
	length	nonNegativeInteger	required			
	parameterName	tc:NameReferenceType	required			
	alternateParameterName	tc:NameReferenceType	optional			
annotation	documentation	Dwelling parameters				

3.2.61 complexType tc:ComplexAlarmsType

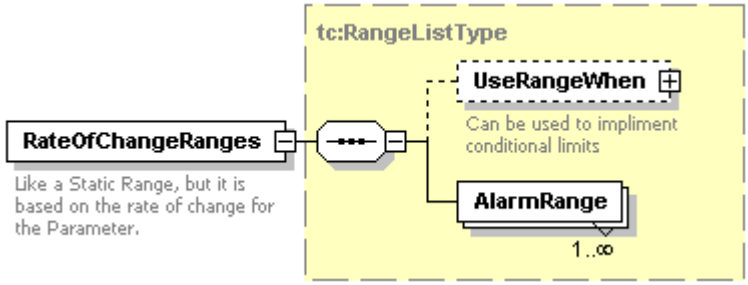
diagram	

namespace	http://www.omg.org/space/tcspec					
children	StaticRanges RateOfChangeRanges States CustomAlarmCheck					
used by	element: tc:ParameterType/AlarmConditions					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minViolations	integer	required			Documentation: Doesn't get reported until it has been out of range for this value times.
	limitsApplyTo-Calibrated	boolean		true		
annotation	documentation	Alarms are used to notify the operator when a parameter value is outside of acceptable ranges or states. Dynamic or rate-of-change Limit Ranges could be considered.				

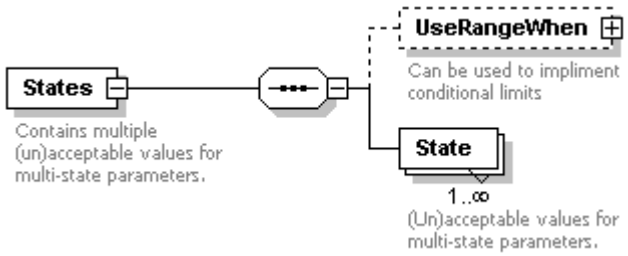
3.2.62 element tc:ComplexAlarmsType/StaticRanges

diagram						
type	tc:RangeListType					
children	UseRangeWhen AlarmRange					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	limitSetPriority	nonNegativeInteger	optional			
annotation	documentation	A StaticRange has an optional UseRangeWhen condition and one or more AlarmRanges.				

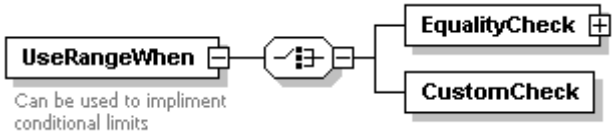
3.2.63 element tc:ComplexAlarmsType/RateOfChangeRanges

diagram						
type	tc:RangeListType					
children	UseRangeWhen AlarmRange					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	limitSetPriority	nonNegativeInteger	optional			
	perTimeBase	string	optional	second		documentation: Rate of change alarms must be reference to some relative time base e.g. 3.4 volts per second.
annotation	documentation	Like a Static Range, but it is based on the rate of change for the Parameter.				

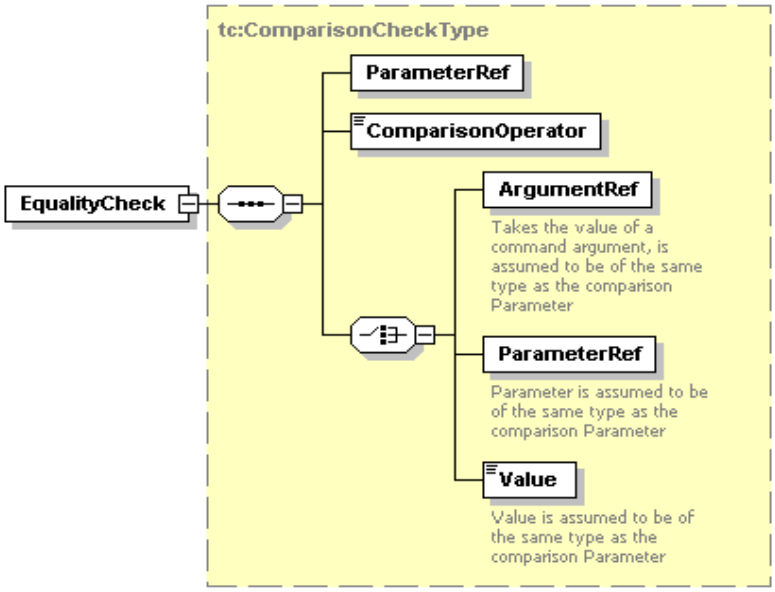
3.2.64 element tc:ComplexAlarmsType/States

diagram						
children	UseRangeWhen State					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	limitSetPriority	nonNegativeInteger	optional			
annotation	documentation	Contains multiple (un)acceptable values for multi-state parameters.				

3.2.65 element tc:ComplexAlarmsType/States/UseRangeWhen

diagram		
children	EqualityCheck CustomCheck	
annotation	documentation	Can be used to implement conditional limits.

3.2.66 element tc:ComplexAlarmsType/States/UseRangeWhen/EqualityCheck

diagram						
type	tc:ComparisonCheckType					
children	ParameterRef ComparisonOperator ArgumentRef ParameterRef Value					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibratedComparison	boolean		false		documentation: Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.

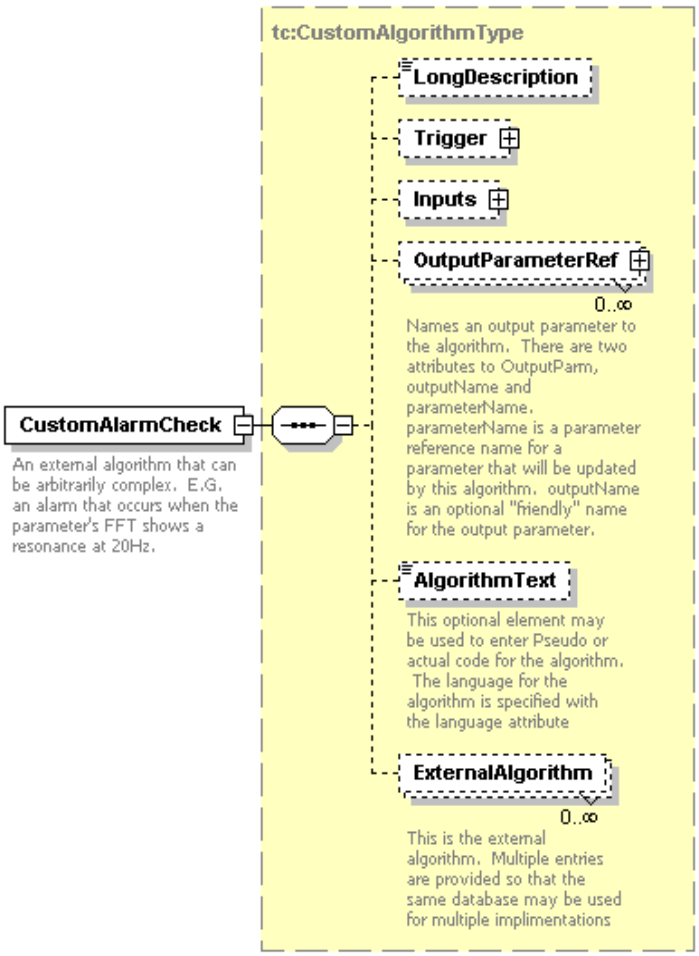
3.2.67 element tc:ComplexAlarmsType/States/UseRangeWhen/CustomCheck

diagram	

3.2.68 element tc:ComplexAlarmsType/States/State

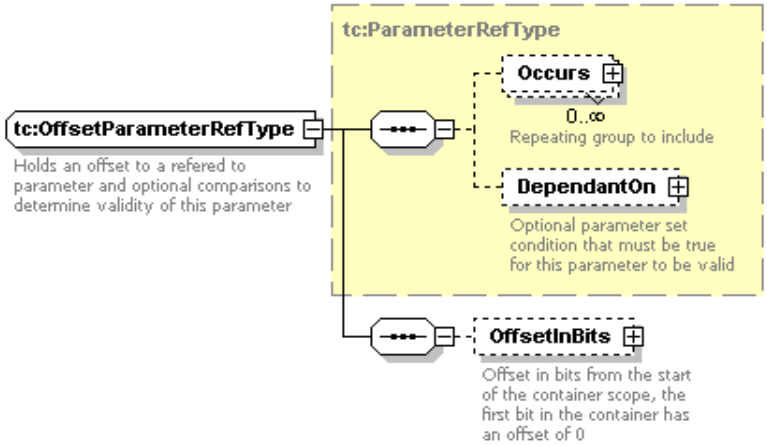
diagram						
children	UseRangeWhen State					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	statePriority	nonNegativeInteger	optional			
	type	tc:LimitCheckType	required			
	state	nonNegativeInteger	required			
annotation	documentation	(Un)acceptable values for multi-state parameters.				

3.2.69 element tc:ComplexAlarmsType/CustomAlarmCheck

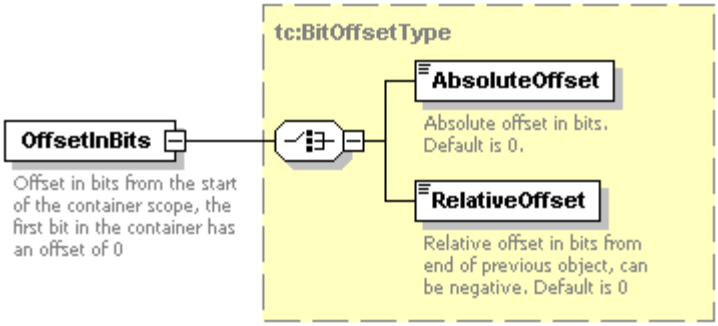
<table>
<tr><td>diagram</td><td colspan="6"> 
</td></tr>
<tr><td>type</td><td colspan="6">tc:CustomAlgorithmType</td></tr>
<tr><td>children</td><td colspan="6">LongDescription Trigger Inputs OutputParameterRef AlgorithmText ExternalAlgorithm</td></tr>
<tr><td>attributes</td><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td></td><td>name</td><td>string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td></td><td>shortDescription</td><td>tc:ShortDescription-Type</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td></td><td>thread</td><td>boolean</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td></td><td>triggerContainer</td><td>string</td><td>optional</td><td></td><td></td><td>documentation: First telemetry container from which the output parame-ter should be calculated.</td></tr>
</table>

	priority	integer	optional			documentation: Algorithm processing priority.
annotation	documentation	An external algorithm that can be arbitrarily complex. E.G. an alarm that occurs when the parameter's FFT shows a resonance at 20Hz.				

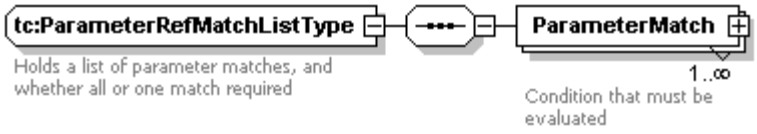
3.2.70 complexType tc:OffsetParameterRefType

diagram						
namespace	http://www.omg.org/space/tcspec					
type	extension of tc:ParameterRefType					
children	Occurs DependantOn OffsetInBits					
used by	elements: tc:SequenceType/ArgumentRef tc:OccursType/CountRef tc:SequenceType/DynamicParameterRef tc:DynamicContainerRefType/ParameterRef tc:SequenceType/ParameterRef					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystem-Name	string				
	TimeOffset	integer	optional	0		documentation: Time delay in milliseconds.
annotation	documentation	Holds an offset to a referred to parameter and optional comparisons to determine validity of this parameter.				

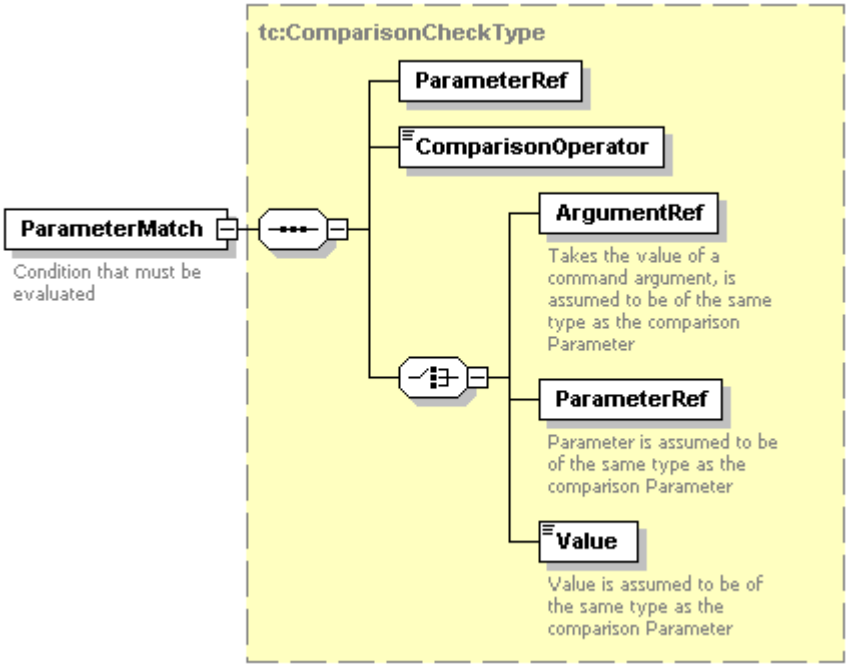
3.2.71 element tc:OffsetParameterRefType/OffsetInBits

diagram		
type	tc:BitOffsetType	
children	AbsoluteOffset RelativeOffset	
annotation	documentation	Offset in bits from the start of the container scope, the first bit in the container has an offset of 0.

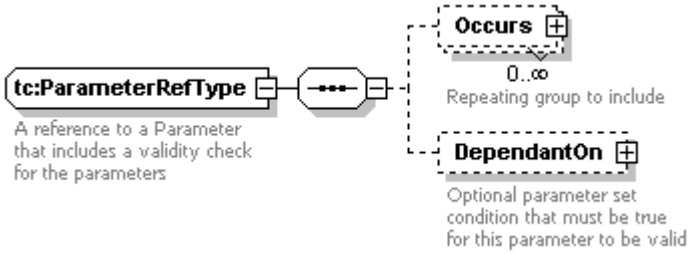
3.2.72 complexType tc:ParameterRefMatchListType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	ParameterMatch					
used by	element: tc:ParameterRefType/DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	AllMustBeTrue	boolean				documentation: Must all matched be true? If not present then all must be true.
annotation	documentation	Holds a list of parameter matches, and whether all or one match required.				

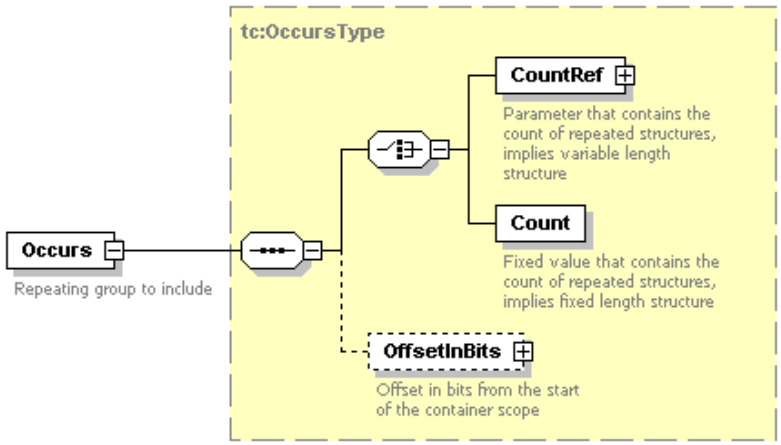
3.2.73 element tc:ParameterRefMatchListType/ParameterMatch

<table>
<tr><td>diagram</td><td colspan="6"> 
</td></tr>
<tr><td>type</td><td colspan="6">tc:ComparisonCheckType</td></tr>
<tr><td>children</td><td colspan="6">ParameterRef ComparisonOperator ArgumentRef ParameterRef Value</td></tr>
<tr><td>attributes</td><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td></td><td>calibratedComparison</td><td>boolean</td><td></td><td>false</td><td></td><td>documentation:
Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.</td></tr>
<tr><td>annotation</td><td>documentation</td><td colspan="5">Condition that must be evaluated.</td></tr>
</table>

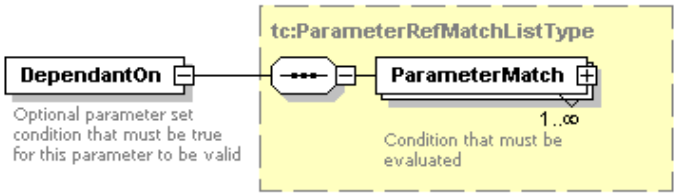
3.2.74 complexType tc:ParameterRefType

diagram						
namespace	http://www.omg.org/space/tcspec					
type	extension of tc:SimpleParameterRefType					
children	Occurs DependantOn					
used by	elements: tc:CommandDefinitionType/Field/ArgumentRef tc:FixedFrameSync/InputParameterRef tc:ConcatenationAlgorithmType/InputParameterList/InputParameterRef tc:ReedSolomonType/InputParameterRef tc:FixedFrameSync/OutputParameterRef tc:CustomAlgorithmType/OutputParameterRef tc:ConcatenationAlgorithmType/OutputParameterRef tc:MathAlgorithmType/OutputParameterRef tc:ReedSolomonType/OutputParameterRef tc:CustomAlgorithmType/Inputs/ParameterRef tc:TriggerType/ParameterRef tc:CommandDefinitionType/Field/ParameterRef tc:CommandDefinitionType/ParametersToSet/ParameterRef tc:VariableParameterType/WidthRef complexType tc:OffsetParameterRefType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
annotation	documentation	A reference to a Parameter that includes a validity check for the parameters.				

3.2.75 element tc:ParameterRefType/Occurs

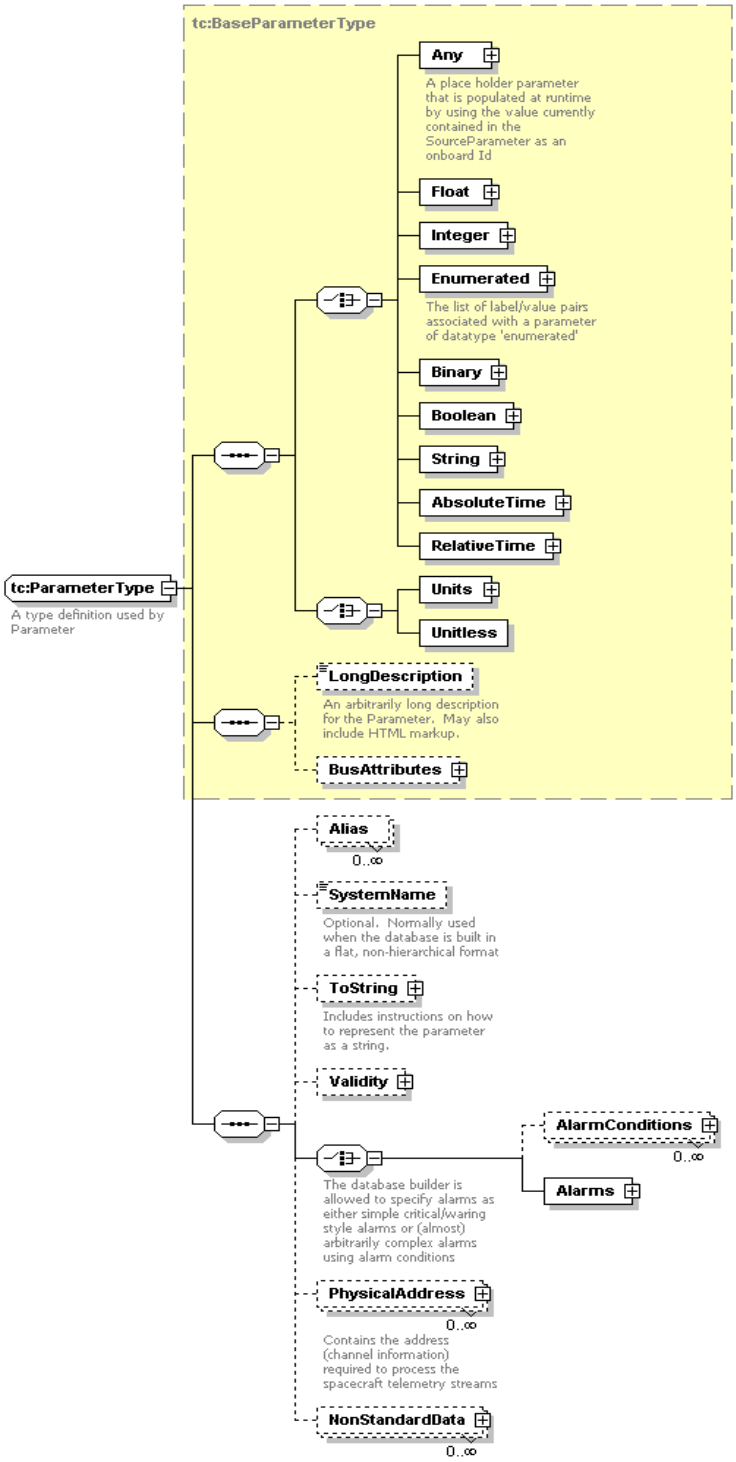
diagram						
type	tc:OccursType					
children	CountRef Count OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	type	NMTOKENS		Occurs		
annotation	documentation	Repeating group to include.				

3.2.76 element tc:ParameterRefType/DependantOn

diagram						
type	tc:ParameterRefMatchListType					
children	ParameterMatch					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	AllMustBeTrue	boolean				documentation: Must all matched be true? If not present then all must be true.
annotation	documentation	Optional parameter set condition that must be true for this parameter to be valid.				

3.2.77 complexType tc:ParameterType

diagram

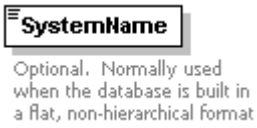


namespace	http://www.omg.org/space/tcspec					
type	extension of tc:BaseParameterType					
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes Alias SystemName ToString Validity AlarmConditions Alarms PhysicalAddress NonStandardData					
used by	element: SpaceSystemType/TelemetryAndCommanding/ParameterList/Parameter complexType: tc:VariableParameterType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescriptionType	optional			
	dataSource	string	optional			
	readOnly	boolean	optional	false		
	validRangeAppliesTo-Calibrated	boolean	optional	true		
	onboardID	string	optional			
annotation	documentation	A type definition used by Parameter.				

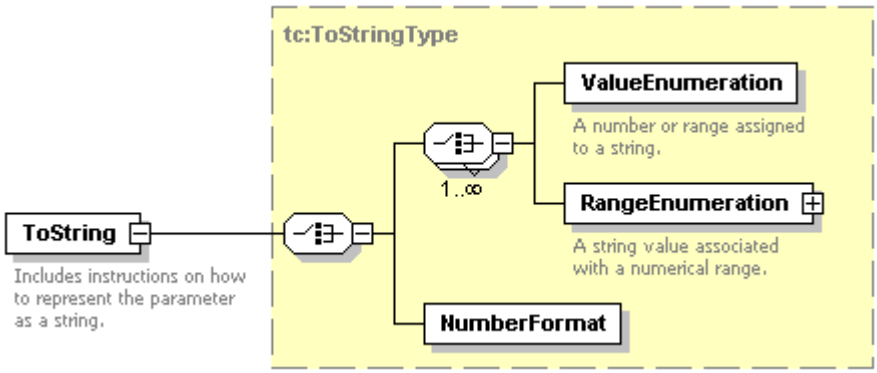
3.2.78 element tc:ParameterType/Alias

diagram						
type	tc:AliasType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameSet	string	required			
	alias	string	required			

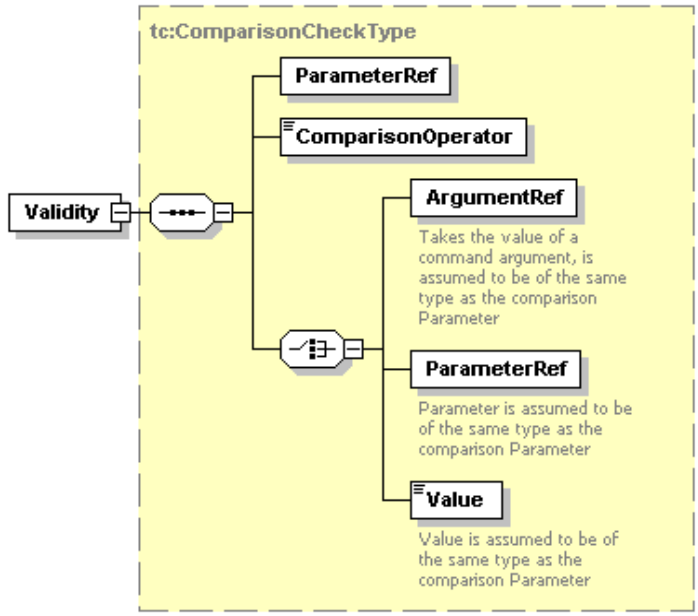
3.2.79 element tc:ParameterType/SystemName

diagram		
type	tc:SystemNameType	
annotation	document	Optional. Normally used when the database is built in a flat, non-hierarchical format.

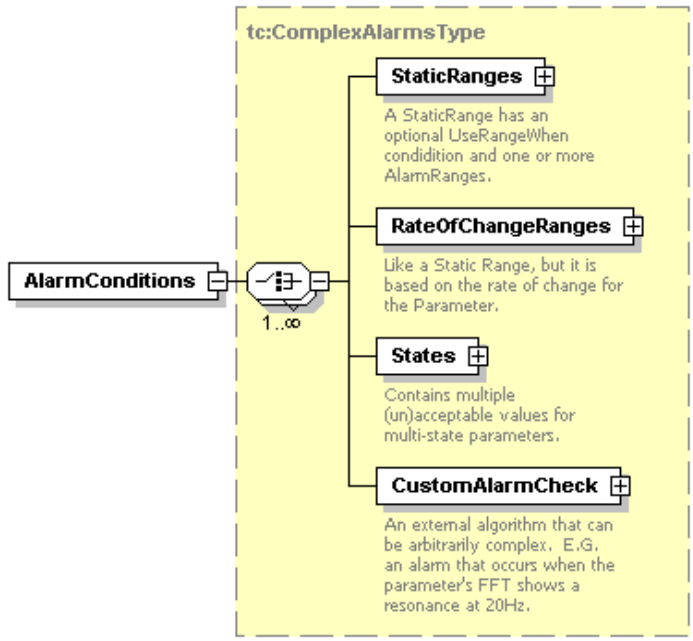
3.2.80 element tc:ParameterType/ToString

diagram		
type	tc:ToStringType	
children	ValueEnumeration RangeEnumeration NumberFormat	
annotation	document	Includes instructions on how to represent the parameter as a string.

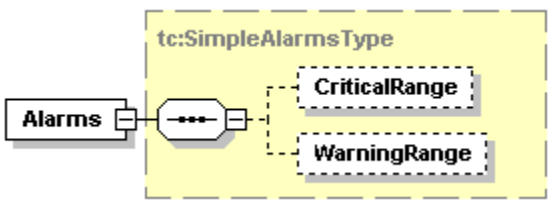
3.2.81 element tc:ParameterType/Validity

diagram						
type	tc:ComparisonCheckType					
children	ParameterRef ComparisonOperator ArgumentRef ParameterRef Value					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibratedComparison	boolean		false		documentation: Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too..

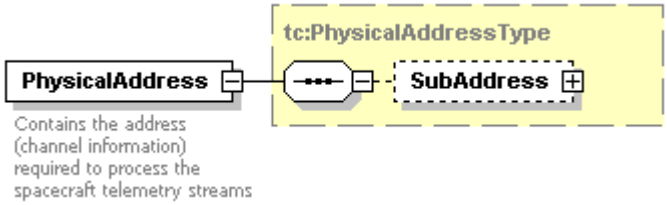
3.2.82 element tc:ParameterType/AlarmConditions

diagram						
type	tc:ComplexAlarmsType					
children	StaticRanges RateOfChangeRanges States CustomAlarmCheck					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minViolations	integer	required			documentation: Doesn't get reported until it has been out of range for this value times.
	limitsApplyToCalibrated	boolean		true		

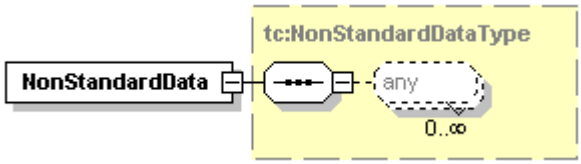
3.2.83 element tc:ParameterType/Alarms

diagram	
type	tc:SimpleAlarmsType
children	CriticalRange WarningRange

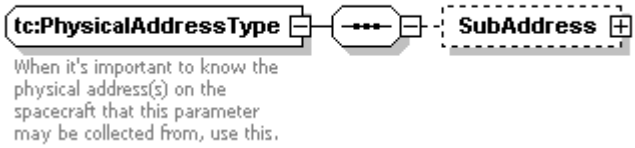
3.2.84 element tc:ParameterType/PhysicalAddress

diagram						
type	extension of tc:PhysicalAddressType					
children	SubAddress					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	sourceName	string				
	sourceAddress	string				
annotation	documentation	Contains the address (channel information) required to process the spacecraft telemetry streams.				
	documentation	Contains the address (channel information) required to process the spacecraft telemetry streams.				

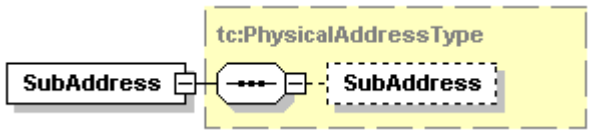
3.2.85 element tc:ParameterType/NonStandardData

diagram						
type	tc:NonStandardDataType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	usage	string	required			

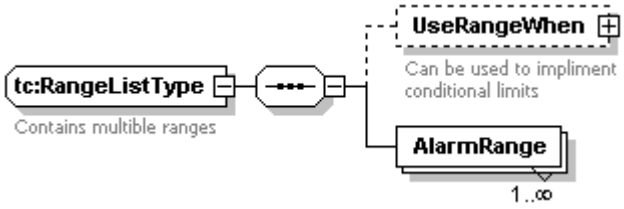
3.2.86 complexType tc:PhysicalAddressType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	SubAddress					
used by	elements: tc:ParameterType/PhysicalAddress tc:PhysicalAddressType/SubAddress					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	sourceName	string				
	sourceAddress	string				
annotation	documentation	When it's important to know the physical address(s) on the spacecraft that this parameter may be collected from, use this.				

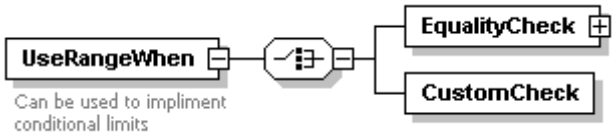
3.2.87 element tc:PhysicalAddressType/SubAddress

diagram						
type	tc:PhysicalAddressType					
children	SubAddress					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	sourceName	string				
	sourceAddress	string				

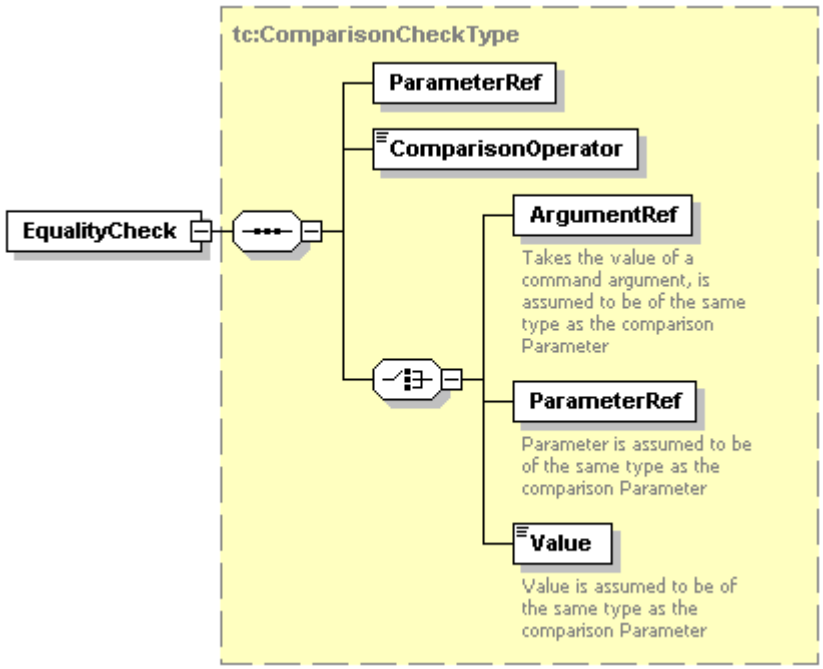
3.2.88 complexType tc:RangeListType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	UseRangeWhen AlarmRange					
used by	elements: tc:ComplexAlarmsType/RateOfChangeRanges tc:ComplexAlarmsType/StaticRanges					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	limitSetPriority	nonNegativeInteger	optional			
annotation	documentation	Contains multiple ranges.				

3.2.89 element tc:RangeListType/UseRangeWhen

diagram		
children	EqualityCheck CustomCheck	
annotation	documentation	Can be used to implement conditional limits.

3.2.90 element tc:RangeListType/UseRangeWhen/EqualityCheck

diagram						
type	tc:ComparisonCheckType					
children	ParameterRef ComparisonOperator ArgumentRef ParameterRef Value					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibratedComparison	boolean		false		documentation: Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too..

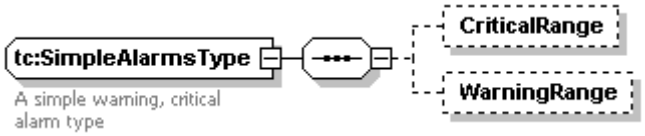
3.2.91 element tc:RangeListType/UseRangeWhen/CustomCheck

diagram	CustomCheck

3.2.92 element tc:RangeListType/AlarmRange

diagram						
type	extension of tc:RangeType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		
	severity	nonNegativeInteger	required			
	type	tc:LimitCheckType	required			
	rangeName	string	optional			

3.2.93 complexType tc:SimpleAlarmsType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	CriticalRange WarningRange	
used by	elements: tc:ParameterType/Alarms	
annotation	documentation	A simple warning, critical alarm type.

3.2.94 element tc:SimpleAlarmsType/CriticalRange

diagram						
type	tc:RangeType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.95 element tc:SimpleAlarmsType/WarningRange

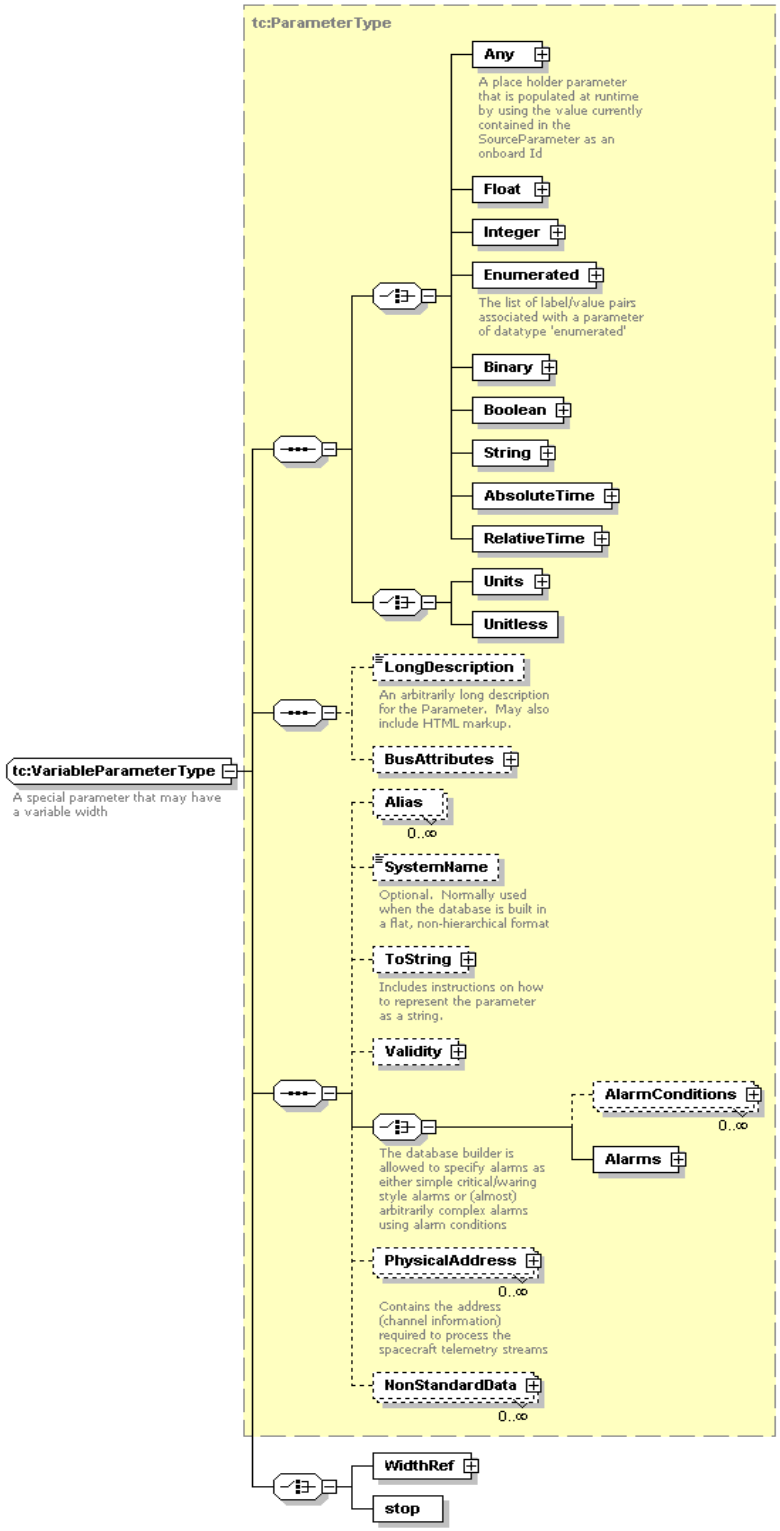
diagram						
type	tc:RangeType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.96 complexType tc:StopType

diagram						
namespace	http://www.omg.org/space/tcspec					
used by	element: tc:VariableParameterType/stop					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	representation	hexBinary				
	widthInBits	positiveInteger				
annotation	document	Specifies either a width in bits or a termination for variable width parameters.				

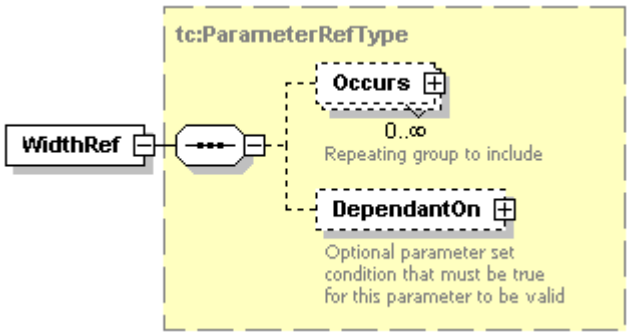
3.2.97 complexType tc:VariableParameterType

diagram


tc:ParameterType
Any
A place holder parameter that is populated at runtime by using the value currently contained in the SourceParameter as an onboard Id
Float
Integer
Enumerated
The list of label/value pairs associated with a parameter of datatype 'enumerated'
Binary
Boolean
String
AbsoluteTime
RelativeTime
Units
Unitless
LongDescription
An arbitrarily long description for the Parameter. May also include HTML markup.
BusAttributes
Alias
0..∞
SystemName
Optional. Normally used when the database is built in a flat, non-hierarchical format
ToString
Includes instructions on how to represent the parameter as a string.
Validity
AlarmConditions
0..∞
Alarms
The database builder is allowed to specify alarms as either simple critical/waring style alarms or (almost) arbitrarily complex alarms using alarm conditions
PhysicalAddress
0..∞
Contains the address (channel information) required to process the spacecraft telemetry streams
NonStandardData
0..∞
tc:VariableParameterType
A special parameter that may have a variable width
WidthRef
stop

namespace	http://www.omg.org/space/tcspec					
type	extension of tc:ParameterType					
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes Alias SystemName ToString Validity AlarmConditions Alarms PhysicalAddress NonStandardData WidthRef stop					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescriptionType	optional			
	dataSource	string	optional			
	readOnly	boolean	optional	false		
	validRangeAppliesToCalibrated	boolean	optional	true		
	onboardID	string	optional			
annotation	document	A special parameter that may have a variable width.				

3.2.98 element tc:VariableParameterType/WidthRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType				
	spaceSystemName	string				

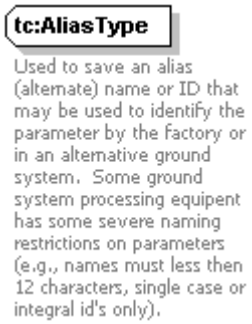
3.2.99 element tc:VariableParameterType/stop

diagram						
type	tc:StopType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	representation	hexBinary				
	widthInBits	positiveInteger				

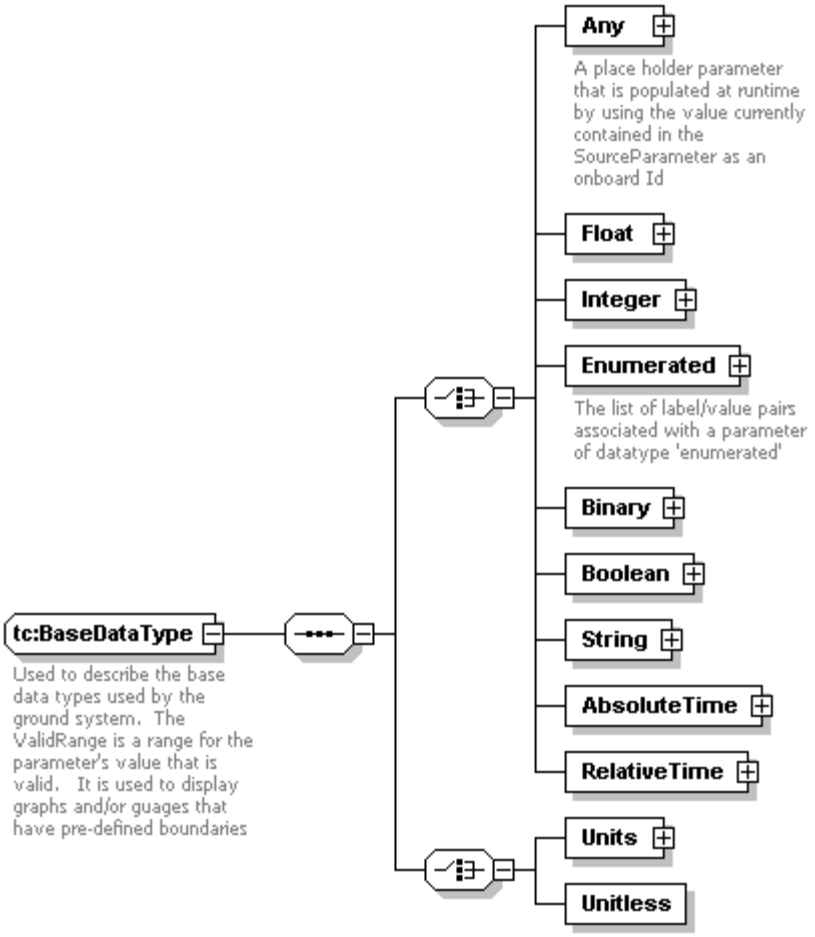
3.2.100 complexType tc:AbsoluteTimeType

diagram		
namespace	http://www.omg.org/space/tcspec	
type	extension of **dateTime**	
annotation	documentation	Contains an absolute (to a known epoch) time.

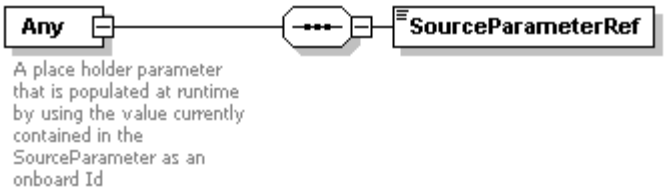
3.2.101 complexType tc:AliasType

diagram						
namespace	http://www.omg.org/space/tcspec					
used by	elements: tc:ParameterType/Alias tc:CommandDefinitionType/Alias					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameSet	string	required			
	alias	string	required			
annotation	document	Used to save an alias (alternate) name or ID that may be used to identify the parameter by the factory or in an alternative ground system. Some ground system processing equipment has some severe naming restrictions on parameters (e.g., names must less then 12 characters, single case or integral id's only).				

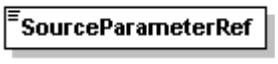
3.2.102 complexType tc:BaseDataType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless	
used by	complexType: tc:BaseParameterType	
annotation	document	Used to describe the base data types used by the ground system. The ValidRange is a range for the parameter's value that is valid. It is used to display graphs and/or gauges that have pre-defined boundaries.

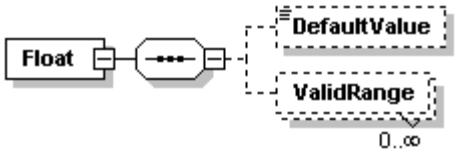
3.2.103 element tc:BaseDataType/Any

diagram		
children	SourceParameterRef	
annotation	document	A place holder parameter that is populated at runtime by using the value currently contained in the SourceParameter as an onboard Id.

3.2.104 element tc:BaseDataType/Any/SourceParameterRef

diagram		
type	tc:NameReferenceType	
facets	pattern	[a-zA-Z0-9_\- /]*

3.2.105 element tc:BaseDataType/Float

diagram		
children	DefaultValue ValidRange	

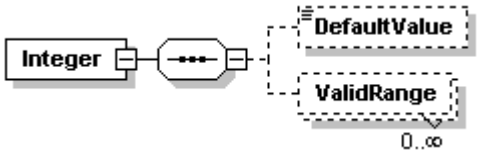
3.2.106 element tc:BaseDataType/Float/DefaultValue

diagram						
type	tc:DefaultValueType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	calibrated	boolean		false		

3.2.107 element tc:BaseDataType/Float/ValidRange

diagram						
type	tc:RangeType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.108 element tc:BaseDataType/Integer

diagram	
children	DefaultValue ValidRange

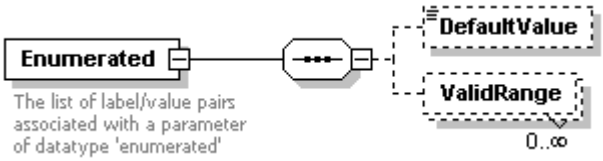
3.2.109 element tc:BaseDataType/Integer/DefaultValue

diagram						
type	tc:DefaultValueType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	calibrated	boolean		false		

3.2.110 element tc:BaseDataType/Integer/ValidRange

diagram						
type	tc:RangeType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.111 element tc:BaseDataType/Enumerated

diagram		
children	DefaultValue ValidRange	
annotation	document	The list of label/value pairs associated with a parameter of datatype 'enumerated'.

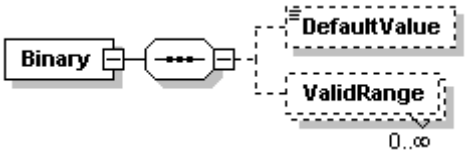
3.2.112 element tc:BaseDataType/Enumerated/DefaultValue

diagram						
children	tc:DefaultValueType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibrated	boolean		false		

3.2.113 element tc:BaseDataType/Enumerated/ValidRange

diagram						
type	tc:RangeType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

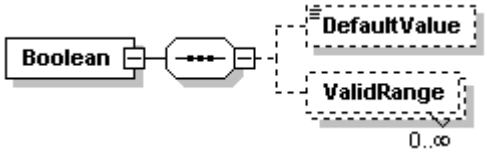
3.2.114 element tc:BaseDataType/Binary

diagram						
children	DefaultValue ValidRange					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	lengthInBits	nonNegativeInteger				

3.2.115 element tc:BaseDataType/Binary/DefaultValue

diagram						
children	tc:RangeType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.116 element tc:BaseDataType/Boolean

diagram	
children	DefaultValue ValidRange

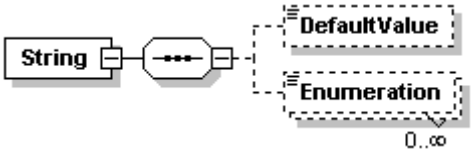
3.2.117 element tc:BaseDataType/Boolean/DefaultValue

diagram						
type	tc:DefaultValueType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	calibrated	boolean		false		

3.2.118 element tc:BaseDataType/Boolean/ValidRange

diagram						
children	tc:RangeType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.119 element tc:BaseDataType/String

diagram						
children	DefaultValue Enumeration					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	maxLength	nonNegativeInteger	optional			

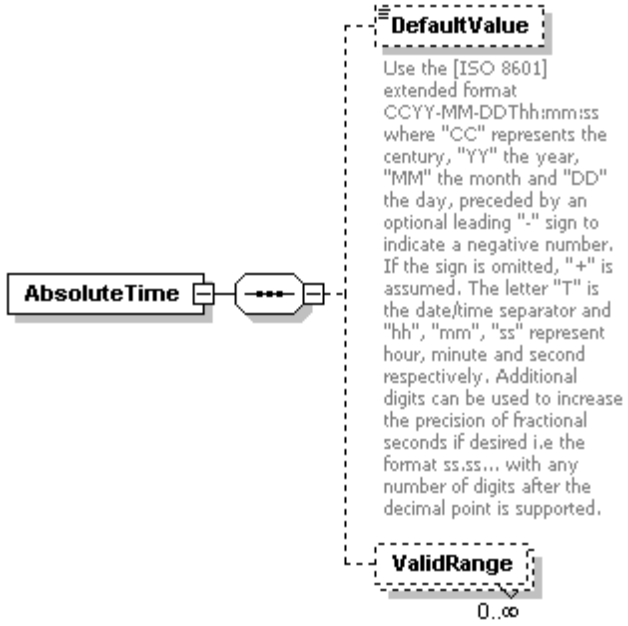
3.2.120 element tc:BaseDataType/String/DefaultValue

diagram						
type	tc:DefaultValueType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibrated	boolean		false		

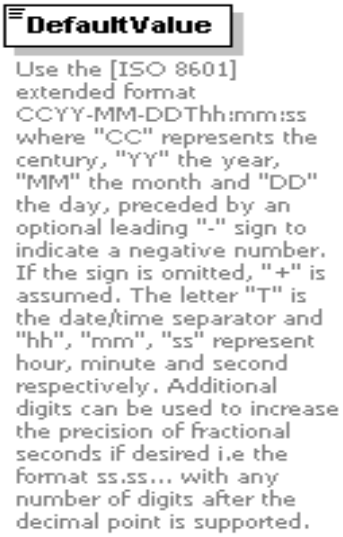
3.2.121 element tc:BaseDataType/String/Enumeration

diagram	
type	**string**

3.2.122 element tc:BaseDataType/AbsoluteTime

diagram	
children	DefaultValue ValidRange

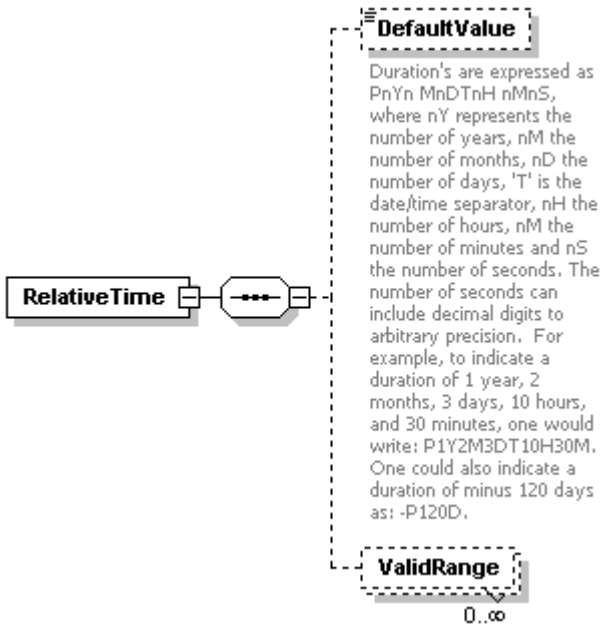
3.2.123 element tc:BaseDataType/AbsoluteTime/DefaultValue

diagram						
type	tc:DefaultValueType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibrated	boolean		false		
annotation	document	Use the [ISO 8601] extended format CCYY-MM-DDThh:mm:ss where "CC" represents the century, "YY" the year, "MM" the month and "DD" the day, preceded by an optional leading "-" sign to indicate a negative number. If the sign is omitted, "+" is assumed. The letter "T" is the date/time separator and "hh", "mm", "ss" represent hour, minute and second respectively. Additional digits can be used to increase the precision of fractional seconds if desired i.e the format ss.ss... with any number of digits after the decimal point is supported.				

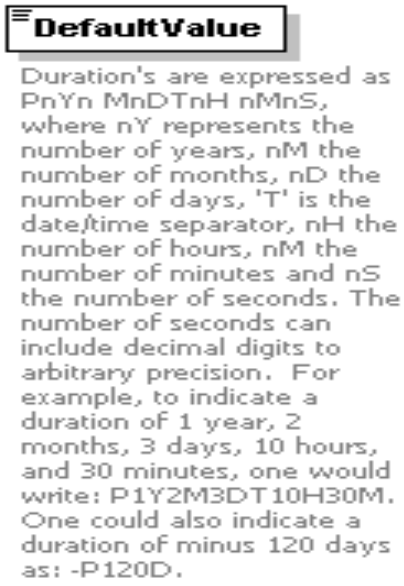
3.2.124 element tc:BaseDataType/AbsoluteTime/ValidRange

diagram						
children	tc:RangeType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.125 element tc:BaseDataType/RelativeTime

diagram	
children	DefaultValue ValidRange

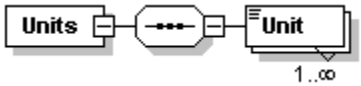
3.2.126 element tc:BaseDataType/RelativeTime/DefaultValue

diagram						
type	tc:DefaultValueType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibrated	boolean		false		
annotation	document	Duration's are expressed as PnYn MnDTnH nMnS, where nY represents the number of years, nM the number of months, nD the number of days, 'T' is the date/time separator, nH the number of hours, nM the number of minutes and nS the number of seconds. The number of seconds can include decimal digits to arbitrary precision. For example, to indicate a duration of 1 year, 2 months, 3 days, 10 hours, and 30 minutes, one would write: P1Y2M3DT10H30M. One could also indicate a duration of minus 120 days as: -P120D.				

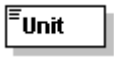
3.2.127 element tc:BaseDataType/RelativeTime/ValidRange

diagram						
type	tc:RangeType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.128 element tc:BaseDataType/Units

diagram	
children	Unit

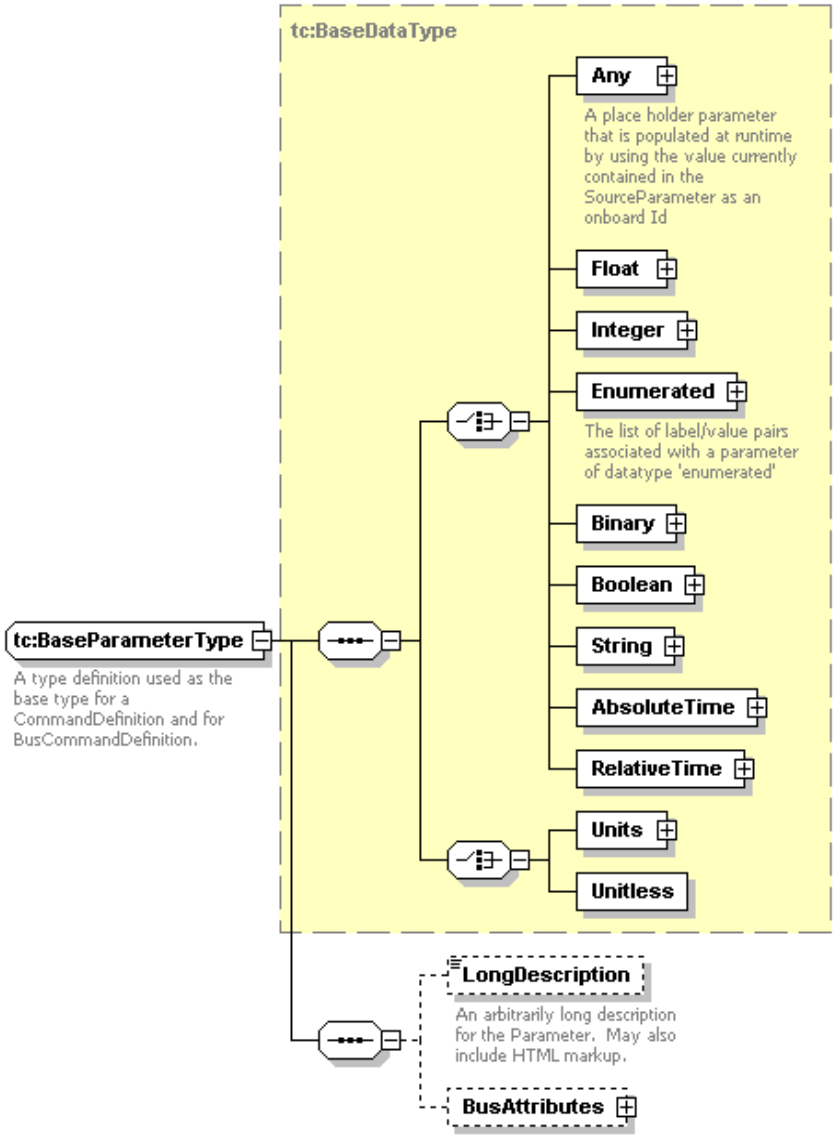
3.2.129 element tc:BaseDataType/Units/Unit

diagram						
type	tc:UnitType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	power	decimal	optional	1		
	description	string				

3.2.130 element tc:BaseDataType/Unitless

diagram	

3.2.131 complexType tc:BaseParameterType

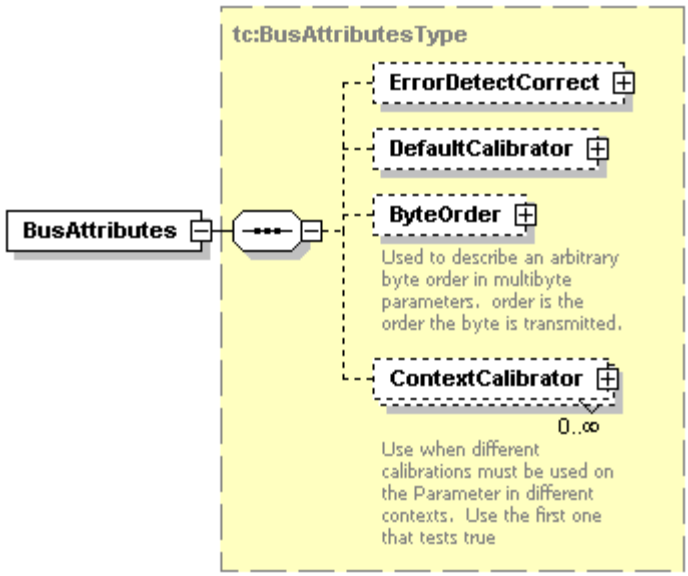
diagram	
namespace	http://www.omg.org/space/tcspec

children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes					
used by	complexType: tc:ArgumentType tc:ParameterType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescriptionType	optional			
annotation	document	A type definition used as the base type for a CommandDefinition and for BusCommandDefinition.				

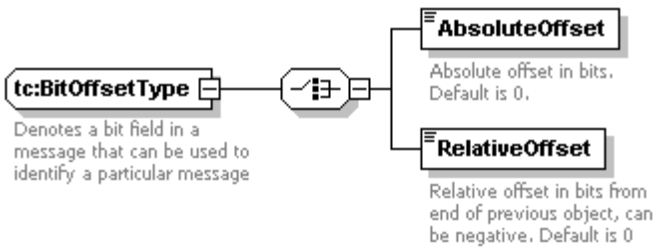
3.2.132 element tc:BaseParameterType/LongDescription

diagram		
type	tc:LongDescriptionType	
annotation	document	An arbitrarily long description for the Parameter. May also include HTML markup.

3.2.133 element tc:BaseParameterType/BusAttributes

diagram						
type	tc:BusAttributesType					
children	ErrorDetectCorrect DefaultCalibrator ByteOrder ContextCalibrator					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	bitLength	unsignedInt	required			
	bitOrder	string		MostSignificantBitFirst		
	busType	string	required			

3.2.134 complexType tc:BitOffsetType

diagram	
namespace	http://www.omg.org/space/tcspec

children	AbsoluteOffset RelativeOffset	
used by	elements: tc:OffsetParameterRefType/OffsetInBits tc:OccursType/OffsetInBits tc:MessageKeyType/OffsetInBits tc:ContainerRefType/OffsetInBits tc:DynamicContainerRefType/OffsetInBits tc:SequenceType/BinaryConstant/OffsetInBits	
annotation	document	Denotes a bit field in a message that can be used to identify a particular message.

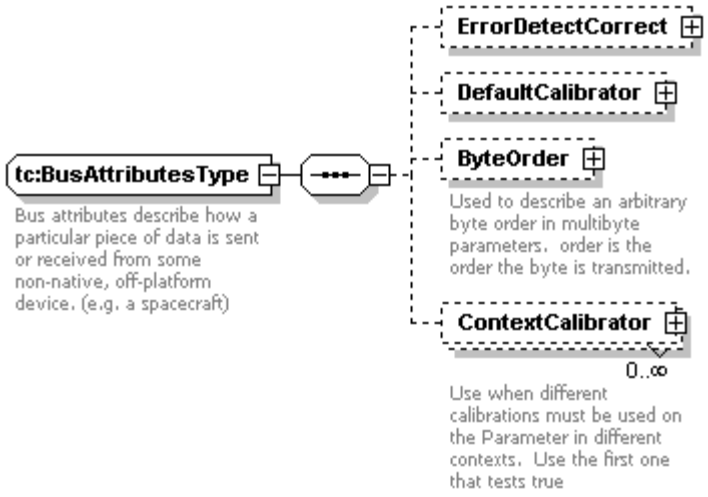
3.2.135 element tc:BitOffsetType/AbsoluteOffset

diagram		
type	nonNegativeInteger	
annotation	document	Absolute offset in bits. Default is 0.

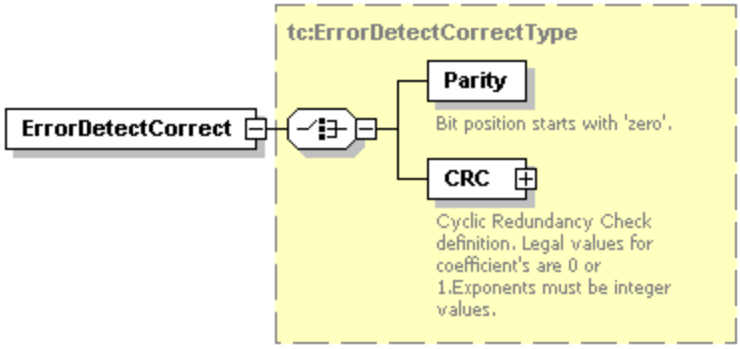
3.2.136 element tc:BitOffsetType/RelativeOffset

diagram		
type	integer	
annotation	document	Relative offset in bits from end of previous object, can be negative. Default is 0.

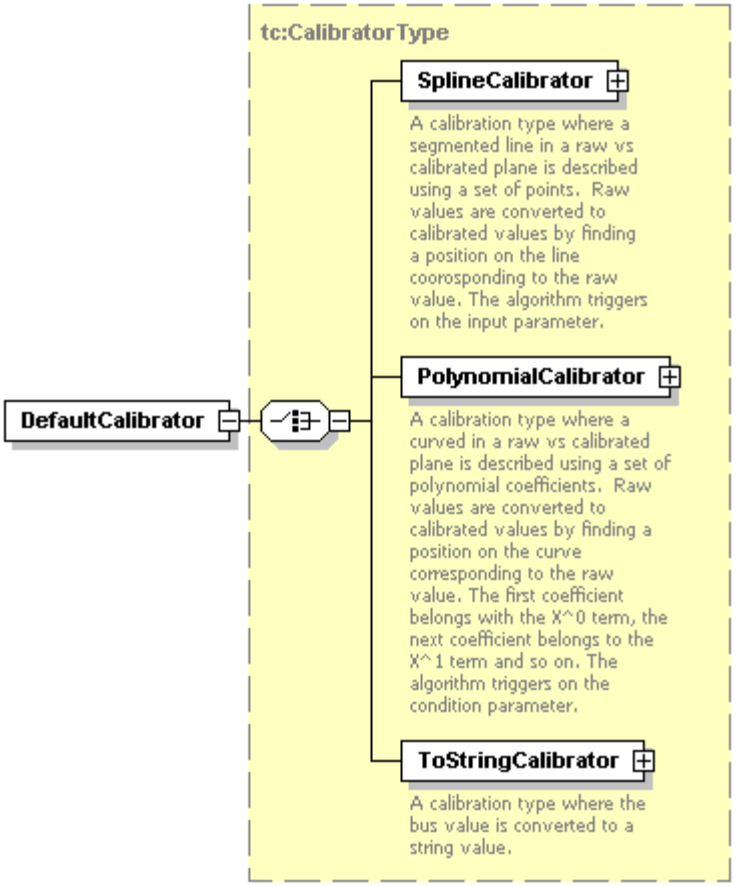
3.2.137 complexType tc:BusAttributesType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	ErrorDetectCorrect DefaultCalibrator ByteOrder ContextCalibrator					
used by	elements: tc:BaseParameterType/BusAttributes SpaceSystemType/TelemetryAndCommanding/DefaultBusAttributes					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	bitLength	unsignedInt	required			
	bitOrder	string		MostSignificantBitFirst		
	busType	string	required			
annotation	document	Bus attributes describe how a particular piece of data is sent or received from some non-native, off-platform device. (e.g. a spacecraft).				

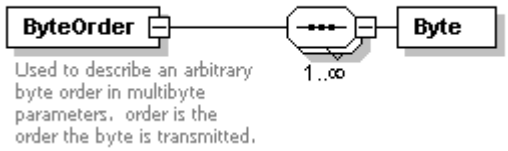
3.2.138 element *tc:BusAttributesType/ErrorDetectCorrect*

diagram	
type	tc:ErrorDetectCorrectType
children	Parity CRC

3.2.139 element tc:BusAttributesType/DefaultCalibrator

diagram						
type	tc:CalibratorType					
children	SplineCalibrator PolynomialCalibrator ToStringCalibrator					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	optional			
	shortDescription	tc:ShortDescriptionType	optional			

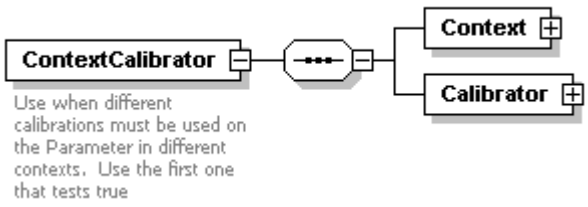
3.2.140 element tc:BusAttributesType/ByteOrder

diagram		
children	byte	
annotation	document	Used to describe an arbitrary byte order in multibyte parameters. order is the order the byte is transmitted.

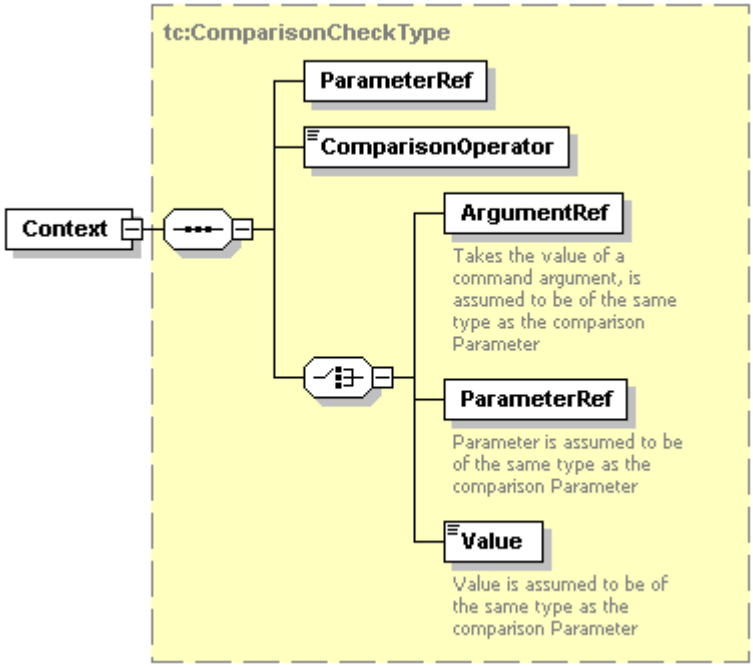
3.2.141 element tc:BusAttributesType/ByteOrder/Byte

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	appearanceInStream	nonNegativeInteger	required			
	byteSignificance	nonNegativeInteger	required			

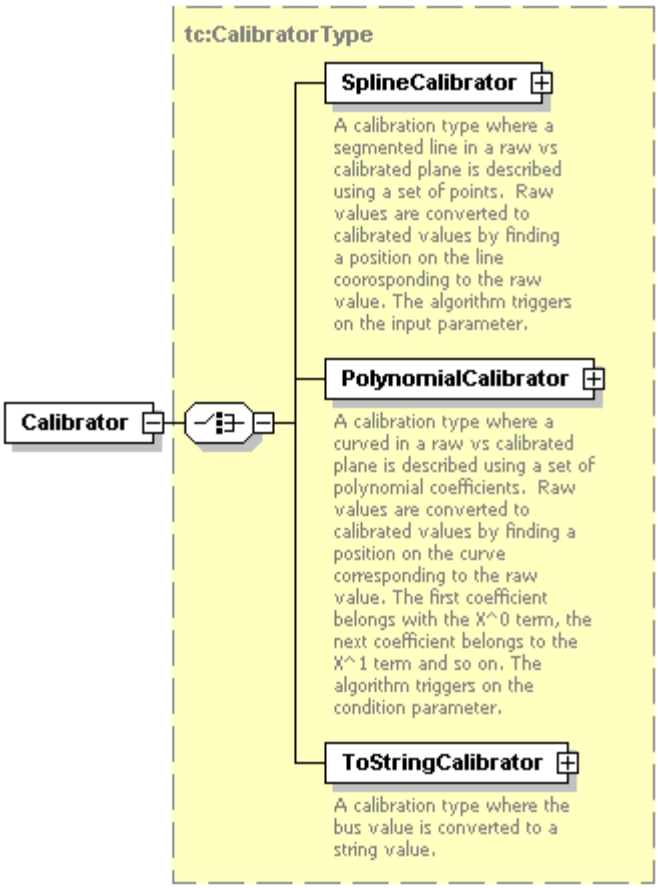
3.2.142 element tc:BusAttributesType/ContextCalibrator

diagram		
children	Context Calibrator	
annotation	document	Use when different calibrations must be used on the Parameter in different contexts. Use the first one that tests true.

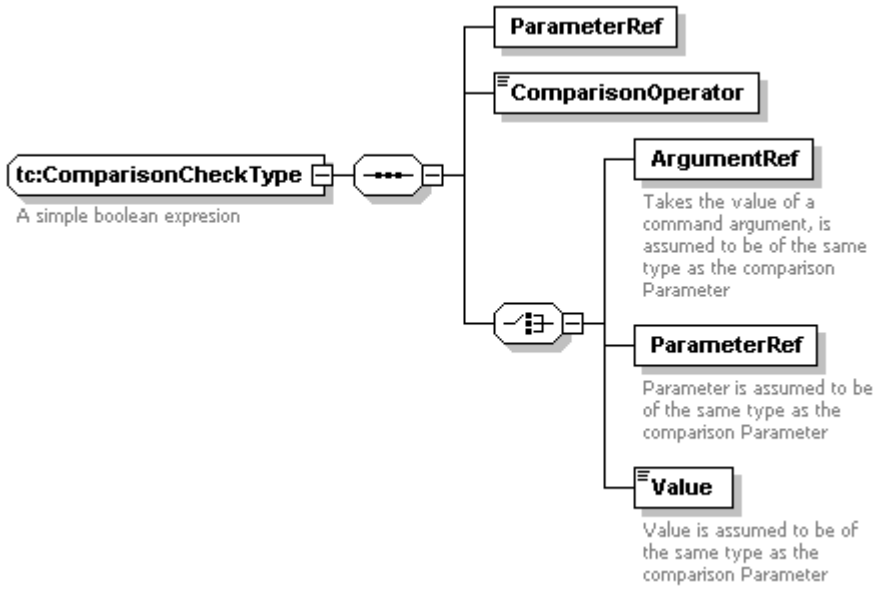
3.2.143 element tc:BusAttributesType/ContextCalibrator/Context

diagram						
type	tc:ComparisonCheckType					
children	ParameterRef ComparisonOperator ArgumentRef ParameterRef Value					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibratedComparison	boolean		false		documentation: Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.

3.2.144 element tc:BusAttributesType/ContextCalibrator/Calibrator

diagram						
type	tc:CalibratorType					
children	SplineCalibrator PolynomialCalibrator ToStringCalibrator					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	optional			
	shortDescription	tc:ShortDescriptionType	optional			

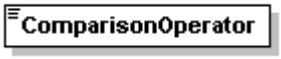
3.2.145 complexType tc:ComparisonCheckType

<table>
<tr><td>diagram</td><td colspan="6"> 
</td></tr>
<tr><td>namespace</td><td colspan="6">http://www.omg.org/space/tcspec</td></tr>
<tr><td>children</td><td colspan="6">ParameterRef ComparisonOperator ArgumentRef ParameterRef Value</td></tr>
<tr><td>used by</td><td colspan="6">elements:
tc:CommandConstraintType/Comparison
tc:CommandVerifierType/Comparison
tc:BusAttributesType/ContextCalibrator/Context
tc:ContainerRefType/DependantOn
tc:DynamicContainerRefType/DependantOn
tc:ComplexAlarmsType/States/UseRangeWhen/EqualityCheck
tc:RangeListType/UseRangeWhen/EqualityCheck
tc:ParameterRefMatchListType/ParameterMatch
tc:ParameterType/Validity</td></tr>
<tr><td>attributes</td><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td></td><td>calibratedComparison</td><td>boolean</td><td></td><td>false</td><td></td><td>documentation:
Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.</td></tr>
<tr><td>annotation</td><td>document</td><td colspan="5">A simple boolean expression.</td></tr>
</table>

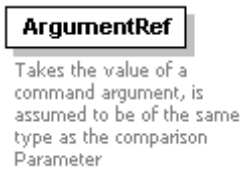
3.2.146 element tc:ComparisonCheckType/ParameterRef

diagram						
type	tc:SimpleParameterRefType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

3.2.147 element tc:ComparisonCheckType/ComparisonOperator

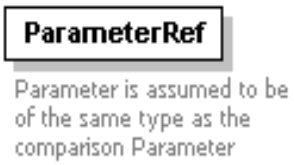
diagram		
type	tc:ComparisonOperatorsType	
facets	enumeration	==
	enumeration	!=
	enumeration	<
	enumeration	<=
	enumeration	>

3.2.148 element tc:ComparisonCheckType/ArgumentRef

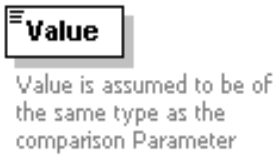
diagram						
type	tc:SimpleParameterRefType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	parameterName	tc:NameReferenceType	required			

	spaceSystemName	string				
annotation	document	Takes the value of a command argument, is assumed to be of the same type as the comparison Parameter.				

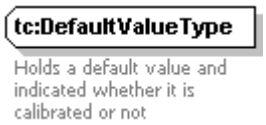
3.2.149 element tc:ComparisonCheckType/ParameterRef

diagram						
type	tc:SimpleParameterRefType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
annotation	document	Parameter is assumed to be of the same type as the comparison Parameter.				

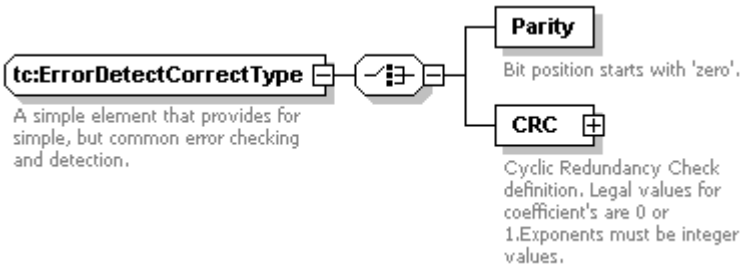
3.2.150 element tc:ComparisonCheckType/Value

diagram		
type	string	
annotation	document	Value is assumed to be of the same type as the comparison Parameter.

3.2.151 complexType tc:DefaultValueType

diagram						
namespace	http://www.omg.org/space/tcspec					
used by	elements: tc:BaseDataType/Float/DefaultValue tc:BaseDataType/Integer/DefaultValue tc:BaseDataType/Enumerated/DefaultValue tc:BaseDataType/Binary/DefaultValue tc:BaseDataType/Boolean/DefaultValue tc:BaseDataType/String/DefaultValue tc:BaseDataType/AbsoluteTime/DefaultValue tc:BaseDataType/RelativeTime/DefaultValue					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibrated	boolean		false		
annotation	document	Holds a default value and indicated whether it is calibrated or not.				

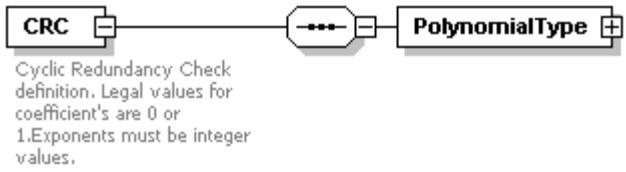
3.2.152 complexType tc:ErrorDetectCorrectType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	Parity CRC	
used by	elements: tc:BusAttributesType/ErrorDetectCorrect	
annotation	document	A simple element that provides for simple, but common error checking and detection.

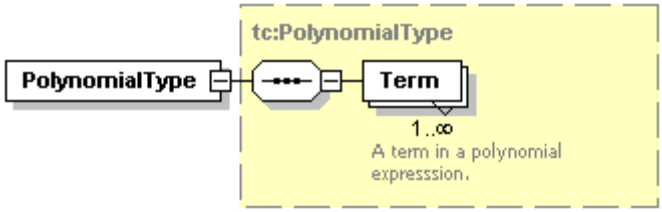
3.2.153 element tc:ErrorDetectCorrectType/Parity

diagram						
						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	type	string	required			
	bitsFromStart	nonNegativeInteger	required			
annotation	document	Bit position starts with 'zero'.				

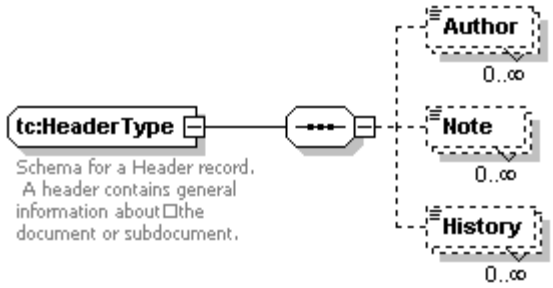
3.2.154 element tc:ErrorDetectCorrectType/CRC

diagram						
						
children	PolynomialType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	bitsFromStart	nonNegativeInteger				
annotation	document	Cyclic Redundancy Check definition. Legal values for coefficient's are 0 or 1. Exponents must be integer values.				

3.2.155 element tc:ErrorDetectCorrectType/CRC/PolynomialType

diagram	
type	tc:PolynomialType
children	Term

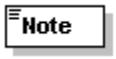
3.2.156 complexType tc:HeaderType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	Author Note History					
used by	element: SpaceSystemType/Header					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	version	string				
	date	string				
	classification	string		Unclassified		
annotation	document	Schema for a Header record. A header contains general information about the document or subdocument.				

3.2.157 element tc:HeaderType/Author

diagram	
type	string

3.2.158 element tc:HeaderType/Note

diagram	
type	string

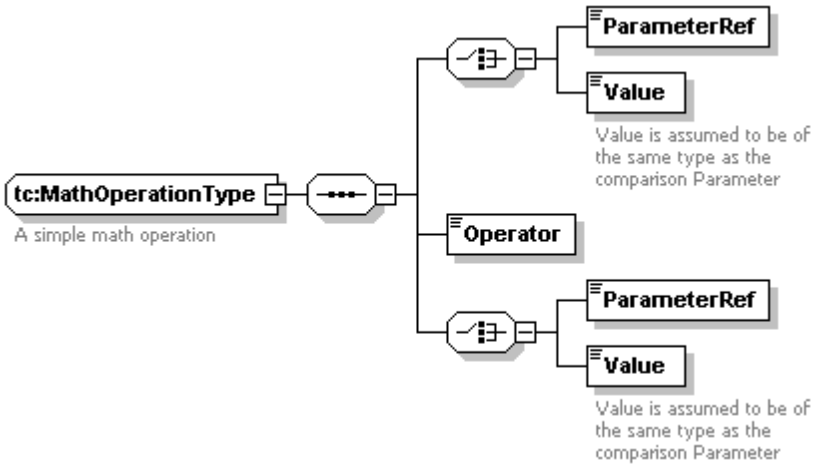
3.2.159 element tc:HeaderType/History

diagram	
type	string

3.2.160 complexType tc:IntegerRangeType

diagram						
namespace	http://www.omg.org/space/tcspec					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	min	long				
	max	long				
annotation	document	A range for integers.				

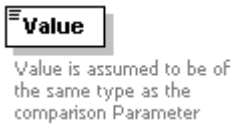
3.2.161 complexType tc:MathOperationType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	ParameterRef Value Operator ParameterRef Value	
used by	complexType: tc:MathAlgorithmType	
annotation	document	A simple math operation.

3.2.162 element tc:MathOperationType/ParameterRef

diagram		
type	tc:NameReferenceType	
facets	pattern	a-zA-Z0-9_\- /]*

3.2.163 element tc:MathOperationType/Value

diagram		
type	string	
annotation	documentation	Value is assumed to be of the same type as the comparison Parameter.

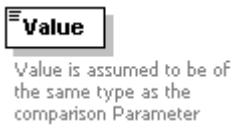
3.2.164 element tc:MathOperationType/Operator

diagram		
type	tc:MathOperatorsType	
facets	enumeration	+
	enumeration	-
	enumeration	mult
	enumeration	div
	enumeration	mod
	enumeration	exp
	enumeration	bitor
	enumeration	bitand
	enumeration	bitxor

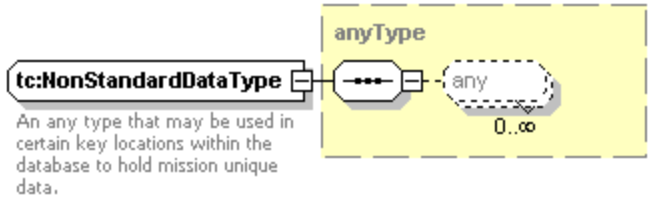
3.2.165 element tc:MathOperationType/ParameterRef

diagram		
type	tc:NameReferenceType	
facets	pattern	a-zA-Z0-9_\- /]*

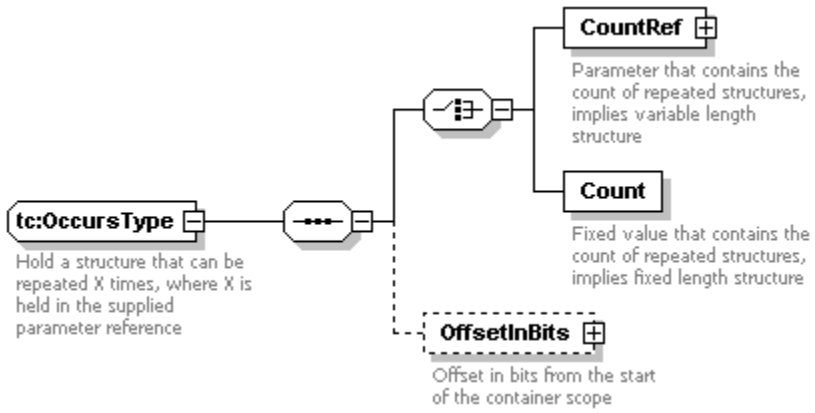
3.2.166 element tc:MathOperationType/Value

diagram		
type	string	
annotation	document	Value is assumed to be of the same type as the comparison Parameter.

3.2.167 complexType tc:NonStandardDataType

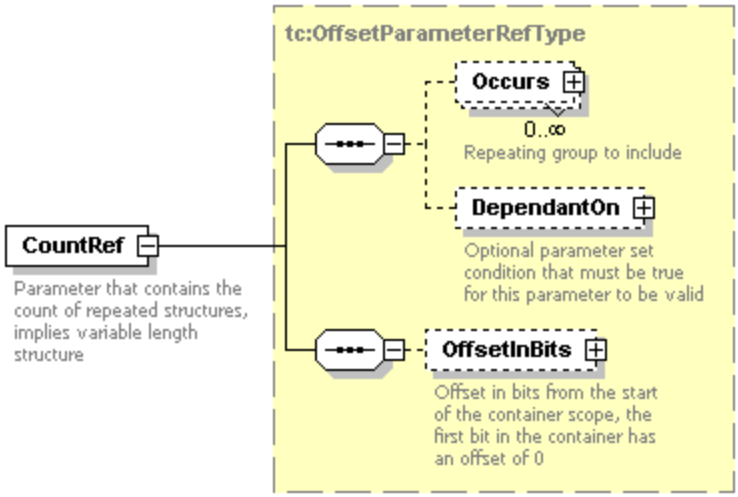
diagram						
namespace	http://www.omg.org/space/tcspec					
type	extension of anyType					
used by	elements: SpaceSystemType/TelemetryAndCommanding/NonStandardData SpaceSystemType/NonStandardData tc:ParameterType/NonStandardData tc:CommandDefinitionType/NonStandardData tc:PackagingDefinitionType/NonStandardData tc:StreamType/NonStandardData					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	usage	string	required			
annotation	document	An any type that may be used in certain key locations within the database to hold mission unique data.				

3.2.168 complexType tc:OccursType

diagram	
namespace	http://www.omg.org/space/tcspec
children	CountRef Count OffsetInBits

used by	elements: tc:ParameterRefType/Occurs tc:ContainerRefType/Occurs tc:DynamicContainerRefType/Occurs tc:SequenceType/Occurs tc:ChoiceType/Occurs					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	type	NMTOKENS		Occurs		
annotation	document	Hold a structure that can be repeated X times, where X is held in the supplied parameter reference.				

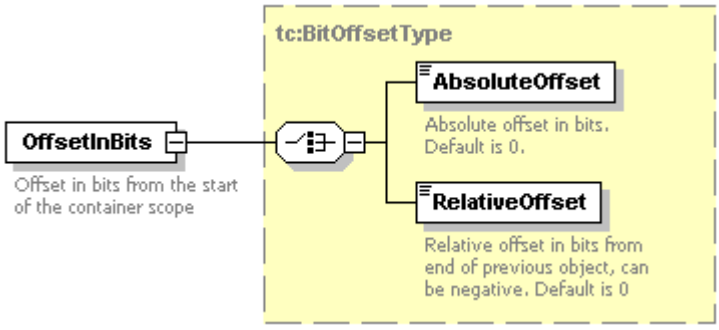
3.2.169 element tc:OccursType/CountRef

diagram						
type	tc:OffsetParameterRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
	TimeOffset	integer	optional	0		documentation: Time delay in milliseconds.
annotation	document	Parameter that contains the count of repeated structures, implies variable length structure.				

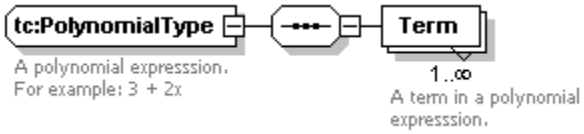
3.2.170 element tc:OccursType/Count

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	NumOfOcc	positiveInteger	required			documentation: Number of occurrences of structure in container.
	RelativeOffsetInBits	nonNegativeInteger	optional			documentation: Number of bits between the start of two consecutive structures.
	TimeOffset	integer	optional	0		documentation:
annotation	document	Fixed value that contains the count of repeated structures, implies fixed length structure.				

3.2.171 element tc:OccursType/OffsetInBits

diagram		
type	tc:BitOffsetType	
children	AbsoluteOffset RelativeOffset	
annotation	document	Offset in bits from the start of the container scope.

3.2.172 complexType tc:PolynomialType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	Term	
used by	elements: tc:ReedSolomonType/GeneratorPolynomial tc:ConvolutionalStreamType/Polynomial tc:CalibratorType/PolynomialCalibrator tc:ErrorDetectCorrectType/CRC/PolynomialType	
annotation	document	A polynomial expression. For example: 3 + 2x

3.2.173 element tc:PolynomialType/Term

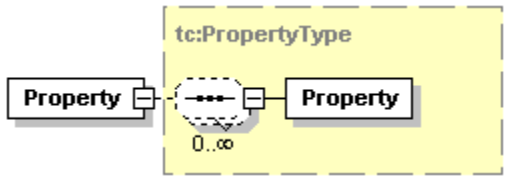
diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	coefficient	double	required			
	exponent	double	required			
annotation	document	A term in a polynomial expression.				

3.2.174 complexType tc:PropertyType

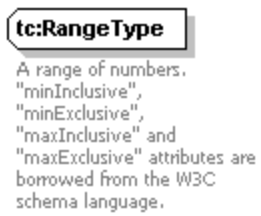
diagram	
namespace	http://www.omg.org/space/tcspec
children	Property
used by	elements: SpaceSystemType/TelemetryAndCommanding/PropertyList/Property tc:PropertyType/Property

attributes	Name	Type	Use	Default	Fixed	Annotation
	name	tc:NameReferenceType	required			
	shortDescription	string				
	longDescription	string				
	value	string	required			
annotation	document	Used for custom user properties.				

3.2.175 element tc:PropertyType/Property

diagram						
type	tc:PropertyType					
children	Property					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameReferenceType	required			
	shortDescription	string				
	longDescription	string				
	value	string	required			

3.2.176 complexType tc:RangeType

diagram	
namespace	http://www.omg.org/space/tcspec

used by	elements: tc:RangeListType/AlarmRange tc:SimpleAlarmsType/CriticalRange tc:ToStringType/RangeEnumeration/Range tc:BaseDataType/Float/ValidRange tc:BaseDataType/Integer/ValidRange tc:BaseDataType/Enumerated/ValidRange tc:BaseDataType/Binary/ValidRange tc:BaseDataType/Boolean/ValidRange tc:BaseDataType/AbsoluteTime/ValidRange tc:BaseDataType/RelativeTime/ValidRange tc:SimpleAlarmsType/WarningRange					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		
annotation	document	A range of numbers. "minInclusive", "minExclusive", "maxInclusive" and "maxExclusive" attributes are borrowed from the W3C schema language.				

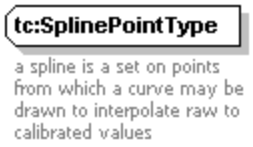
3.2.177 complexType tc:RealRangeType

diagram						
namespace	http://www.omg.org/space/tcspec					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minInclusive	float				
	minExclusive	float				
	maxInclusive	float				
	maxExclusive	float				
annotation	document	A range for real numbers. "minInclusive", "minExclusive", "maxInclusive" and "maxExclusive" attributes are borrowed from the W3C schema language.				

3.2.178 complexType tc:SimpleParameterRefType

diagram						
namespace	http://www.omg.org/space/tcspec					
used by	elements: tc:ComparisonCheckType/ArgumentRef tc:ComparisonCheckType/ParameterRef tc:ComparisonCheckType/ParameterRef complexType tc:ParameterRefType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
annotation	document	A simple reference to a Parameter.				

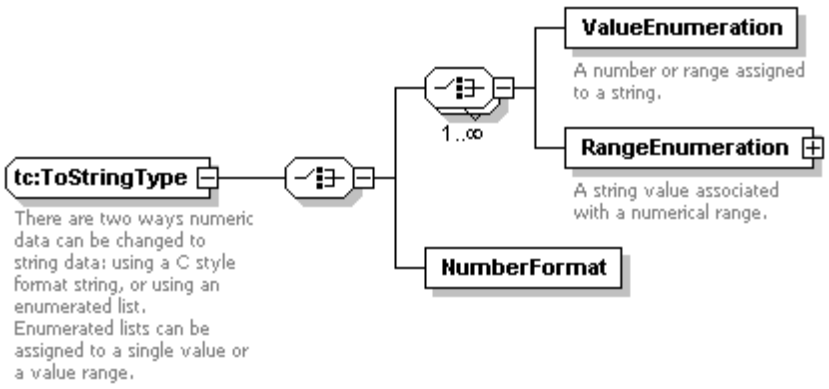
3.2.179 complexType tc:SplinePointType

diagram						
namespace	http://www.omg.org/space/tcspec					
used by	elements: tc:CalibratorType/SplineCalibrator/SplinePoint					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	order	positiveInteger		1		
	raw	double	required			
	calibrated	double	required			
annotation	document	A spline is a set on points from which a curve may be drawn to interpolate raw to calibrated values.				

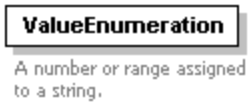
3.2.180 complexType tc:SystemNameType

diagram		
namespace	http://www.omg.org/space/tcspec	
type	extension of string	
used by	elements: tc:ParameterType/SystemName tc:CommandDefinitionType/SystemName	
annotation	document	Sub-system name.

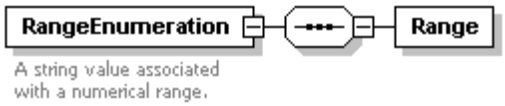
3.2.181 complexType tc:ToStringType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	ValueEnumeration RangeEnumeration NumberFormat	
used by	elements: tc:ParameterType/ToString tc:CalibratorType/ToStringCalibrator	
annotation	document	There are two ways numeric data can be changed to string data: using a C style format string, or using an enumerated list. Enumerated lists can be assigned to a single value or a value range.

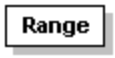
3.2.182 element tc:ToStringType/ValueEnumeration

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	integer	required			
	stringValue	string	required			
annotation	document	A number or range assigned to a string.				

3.2.183 element tc:ToStringType/RangeEnumeration

diagram						
children	Range					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	stringValue	string	required			
annotation	document	A string value associated with a numerical range.				

3.2.184 element tc:ToStringType/RangeEnumeration/Range

diagram						
type	tc:RangeType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	minInclusive	string				
	minExclusive	string				
	maxInclusive	string				
	maxExclusive	string				
	calibrated	boolean		false		

3.2.185 element tc:ToStringType/NumberFormat

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	numberBase	tc:RadixType	optional			
	minimumFractionDigits	nonNegativeInteger	optional			
	maximumFractionDigits	nonNegativeInteger	optional			
	minimumIntegerDigits	nonNegativeInteger	optional			
	maximumIntegerDigits	nonNegativeInteger	optional			
	negativeSuffix	string	optional			
	positiveSuffix	string	optional			
	negativePrefix	string	optional			
	positivePrefix	string	optional			
	showThousandsGrouping	boolean	optional	true		
	notation	string	optional	normal		

3.2.186 complexType tc:UnitType

diagram						
namespace	http://www.omg.org/space/tcspec					
used by	element: tc:BaseDataType/Units/Unit					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	power	decimal	optional	1		
	description	string				
annotation	document	Used to hold the unit(s) plus possibly the exponents for the units.				

3.2.187 simpleType tc:ComparisonOperatorsType

namespace	http://www.omg.org/space/tcspec	
type	restriction of **string**	
used by	element: tc:ComparisonCheckType/ComparisonOperator attribute: tc:MessageKeyMatchType/@Operator	
facets	enumeration	==
	enumeration	!=
	enumeration	<
	enumeration	<=
	enumeration	>
	enumeration	>=
annotation	document	Operators to use when testing a boolean condition for a validity check.

3.2.188 simpleType tc:FormatType

namespace	http://www.omg.org/space/tcspec	
type	restriction of string	
facets	enumeration	SignedInteger
	enumeration	UnsignedInteger
	enumeration	Real
annotation	document	Add

3.2.189 simpleType tc:LimitCheckType

namespace	http://www.omg.org/space/tcspec	
type	restriction of string	
used by	attributes: tc:ComplexAlarmsType/States/State/@type tc:RangeListType/AlarmRange/@type	
facets	enumeration	warning
	enumeration	critical
	enumeration	statusConsistency

	enumeration	eventGeneration				
annotation	document	Identifies the type of limit set check.				

3.2.190 simpleType tc:LongDescriptionType

namespace	http://www.omg.org/space/tcspec	
type	**string**	
used by	elements: SpaceSystemType/LongDescription tc:CustomAlgorithmType/LongDescription tc:BaseParameterType/LongDescription tc:CommandDefinitionType/LongDescription tc:CommandDefinitionType/Field/LongDescription tc:MessageKeyType/LongDescription tc:MessageType/LongDescription tc:ContainerType/LongDescription tc:ServiceType/LongDescription tc:SequenceType/BinaryConstant/LongDescription tc:PCMStreamType/LongDescription	
annotation	document	An unounded description.

3.2.191 simpleType tc:MathOperatorsType

namespace	http://www.omg.org/space/tcspec	
type	restriction of string	
used by	element: tc:MathOperationType/Operator	
facets	enumeration	+
	enumeration	-
	enumeration	mult
	enumeration	div
	enumeration	mod
	enumeration	exp
	enumeration	bitor
	enumeration	bitand
	enumeration	bitxor
annotation	document	Mathematical operators

3.2.192 simpleType tc:NameReferenceType

namespace	http://www.omg.org/space/tcspec	
type	restriction of string	
used by	elements: tc:ServiceType/MessageList/NameRef tc:SyncStrategyType/MessageMatch/NameRef tc:MathOperationType/ParameterRef tc:MathOperationType/ParameterRef tc:BaseDataType/Any/SourceParameterRef attributes: tc:DwellSet/@alternateParameterName tc:ReedSolomonType/@inputParameterRef tc:PropertyType/@name tc:ReedSolomonType/@outputParameterRef tc:DwellSet/@ParameterName tc:SimpleParameterRefType/@parameterName tc:MathAlgorithmType/@trigger	
facets	pattern	[a-zA-Z0-9_\- /]*
annotation	document	Used when referencing a directory style "NameType"..

3.2.193 simpleType tc:NameType

namespace	http://www.omg.org/space/tcspec	
type	restriction of string	
used by	simpleTypes: tc:ContainerNameType tc:MessageKeyIdType tc:MessageNameType tc:ServiceNameType attributes: tc:BaseParameterType/@name tc:CommandDefinitionType/@name tc:PCMStreamType/@name	
facets	pattern	[a-zA-Z0-9_\- /]*
annotation	document	Used for "directory" style unique names. Only letters, digits, '_', ' ' and "-" are allowed.

3.2.194 simpleType tc:RadixType

namespace	http://www.omg.org/space/tcspec

type	restriction of string	
used by	attribute: tc:ToStringType/NumberFormat/@numberBase	
facets	enumeration	Decimal
	enumeration	Hexadecimal
	enumeration	Octal
	enumeration	Binary
annotation	document	Specifies the number base.

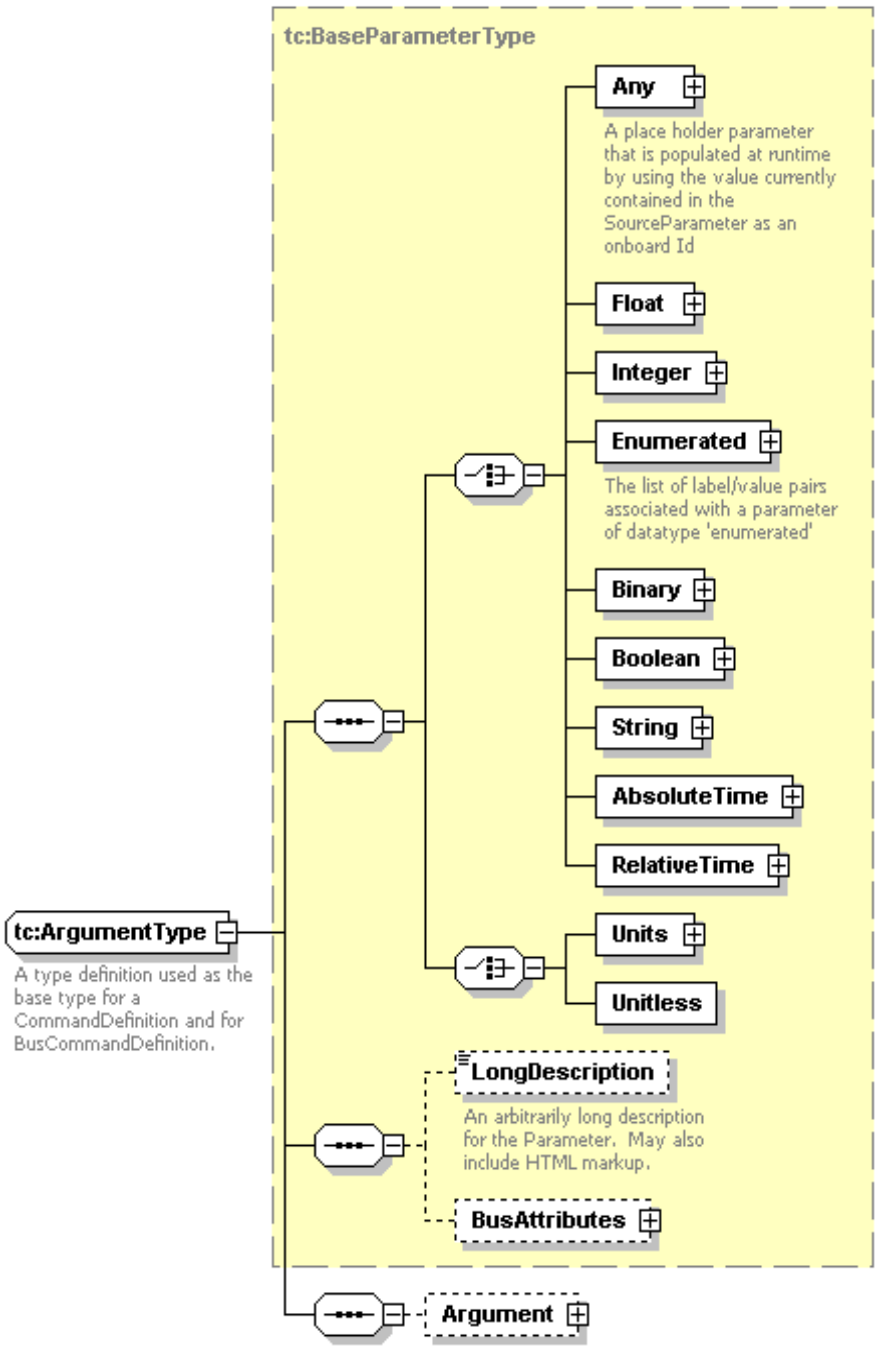
3.2.195 simpleType tc:RelativeTimeType

namespace	http://www.omg.org/space/tcspec	
type	duration	
used by	element: tc:TriggerType/Time	
annotation	document	Used to describe a relative time. Normally used for time offsets. A Relative time is expressed as PnYn MnDTnH nMnS, where nY represents the number of years, nM the number of months, nD the number of days, 'T' is the date/time separator, nH the number of hours, nM the number of minutes and nS the number of seconds. The number of seconds can include decimal digits to arbitrary precision. For example, to indicate a duration of 1 year, 2 months, 3 days, 10 hours, and 30 minutes, one would write: P1Y2M3DT10H30M. One could also indicate a duration of minus 120 days as: -P120D.

3.2.196 simpleType tc:ShortDescriptionType

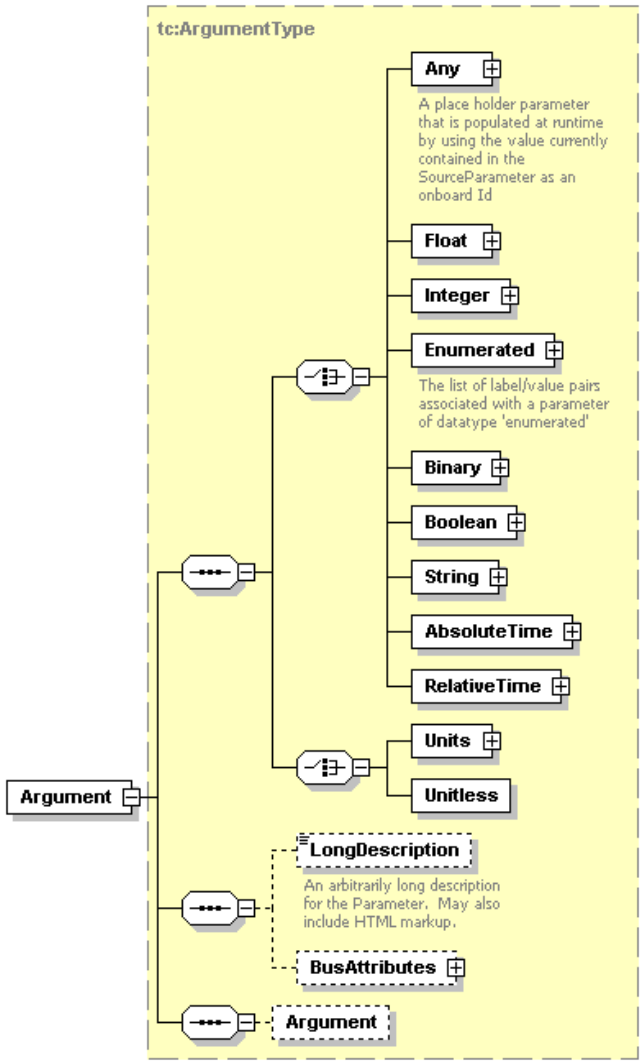
namespace	http://www.omg.org/space/tcspec	
type	restriction of string	
used by	attributes: SpaceSystemType/@shortDescription tc:CustomAlgorithmType/@shortDescription tc:CalibratorType/@shortDescription tc:BaseParameterType/@shortDescription tc:CommandDefinitionType/@shortDescription tc:MessageKeyType/@ShortDescription tc:MessageType/@shortDescription tc:ContainerType/@shortDescription tc:ServiceType/@shortDescription tc:PCMStreamType/@shortDescription	
facets	maxLength	32
annotation	document	A bounded description.

3.2.197 complexType tc:ArgumentType

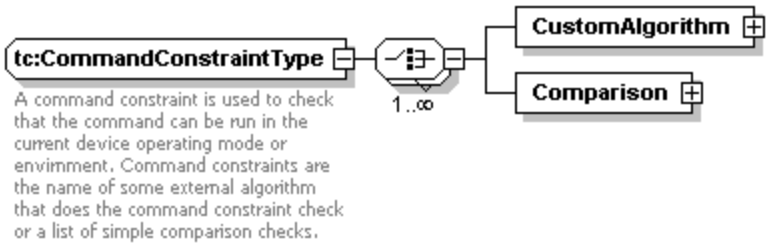
diagram	
namespace	http://www.omg.org/space/tcspec
type	extension of tc:BaseParameterType
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes Argument

used by	elements: tc:ArgumentType/Argument tc:CommandDefinitionType/Field/ArgumentSpaceSystemType/TelemetryAndCommanding/CommandDefinitionList/ArgumentDefinition					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescriptionType	optional			
annotation	document	A type definition used as the base type for a CommandDefinition and for BusCommandDefinition.				

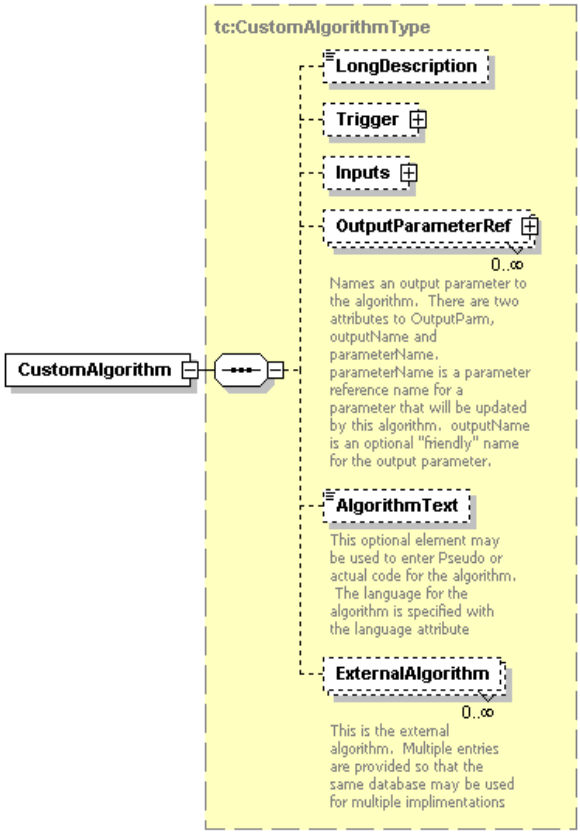
3.2.198 element tc:ArgumentType/Argument

diagram						
type	tc:ArgumentType					
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes Argument					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescriptionType	optional			

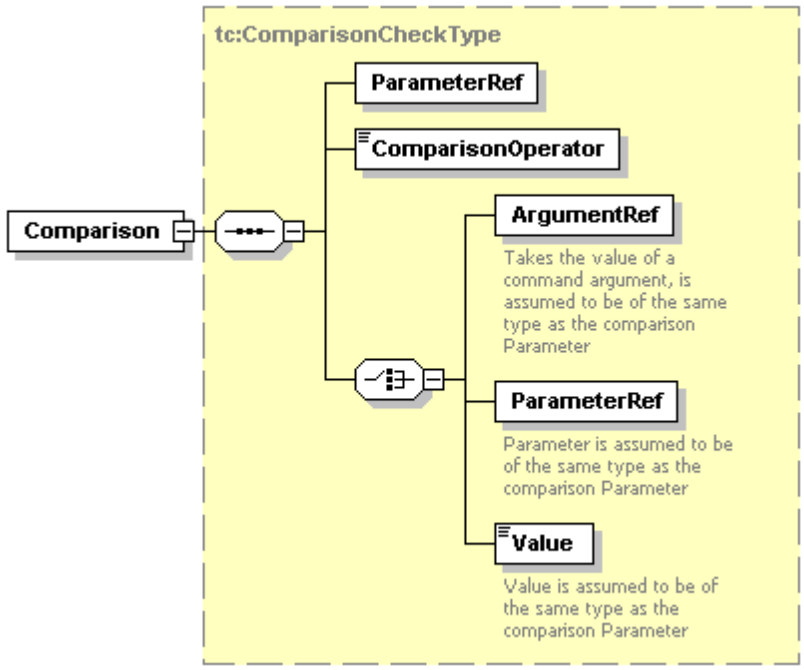
3.2.199 complexType tc:CommandConstraintType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	CustomAlgorithm Comparison	
used by	element: tc:CommandDefinitionType/CommandConstraint	
annotation	document	A command constraint is used to check that the command can be run in the current device operating mode or environment. Command constraints are the name of some external algorithm that does the command constraint check or a list of simple comparison checks.

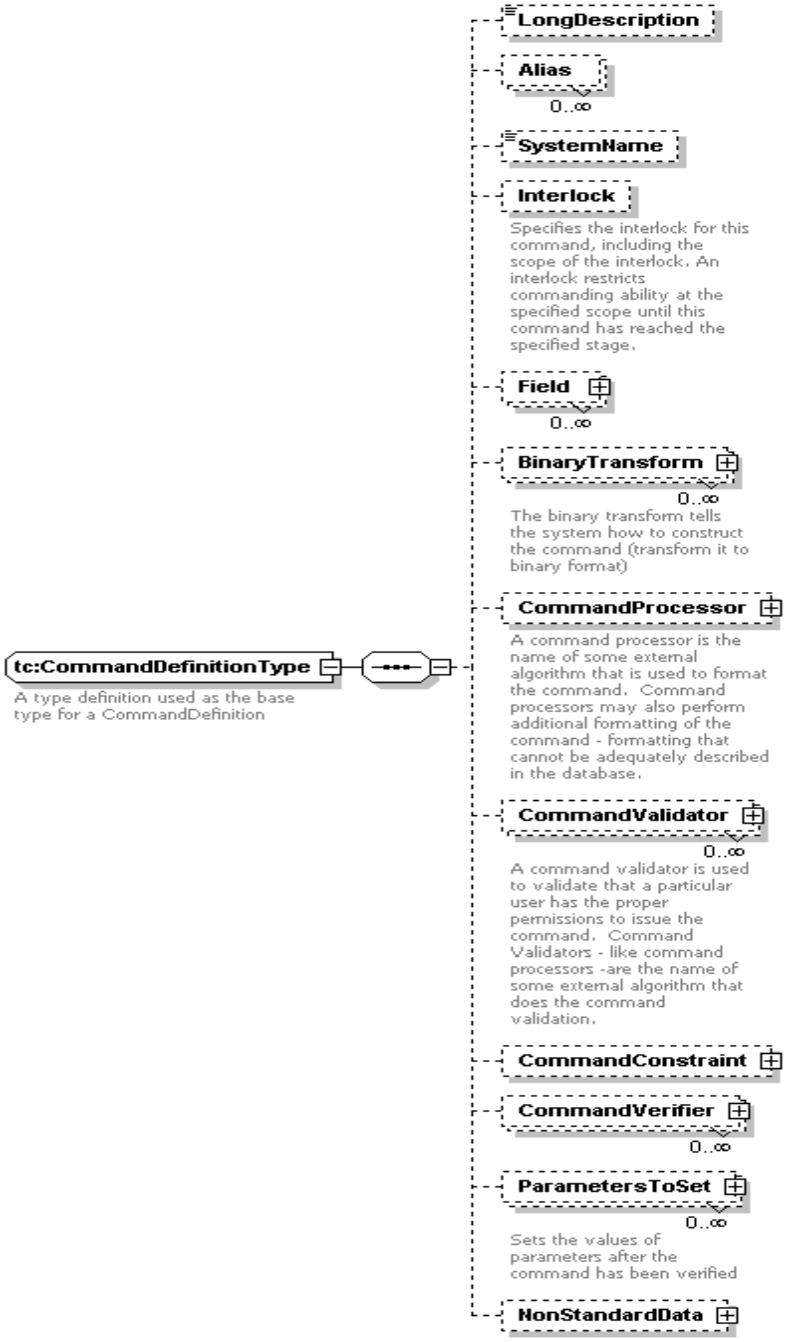
3.2.200 element tc:CommandConstraintType/CustomAlgorithm

diagram						
type	tc:CustomAlgorithmType					
children	LongDescription Trigger Inputs OutputParameterRef AlgorithmText ExternalAlgorithm					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	required			
	shortDescription	tc:ShortDescription-Type	optional			
	thread	boolean	optional			
	triggerContainer	string	optional			documentation: First telemetry container from which the output parameter should be calculated.
	priority	integer	optional			documentation: Algorithm processing priority..

3.2.201 element tc:CommandConstraintType/Comparison

diagram						
type	tc:ComparisonCheckType					
children	ParameterRef ComparisonOperator ArgumentRef ParameterRef Value					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibratedComparison	boolean		false		documentation: Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.

3.2.202 complexType tc:CommandDefinitionType

diagram	
namespace	http://www.omg.org/space/tcspec

children	LongDescription Alias SystemName Interlock Field BinaryTransform CommandProcessor CommandValidator CommandConstraint CommandVerifier ParametersToSet NonStandardData					
used by	elements SpaceSystemType/TelemetryAndCommanding/CommandDefinitionList/CommandDefinition SpaceSystemType/TelemetryAndCommanding/DefaultCommandDefinition					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType	required			
	shortDescription	tc:ShortDescriptionType	optional			
	critical	boolean	optional	false		
	highPriority	boolean	optional	false		
annotation	document	A type definition used as the base type for a CommandDefinition.				

3.2.203 element tc:CommandDefinitionType/LongDescription

diagram	
type	tc:LongDescriptionType

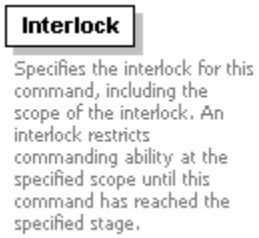
3.2.204 element tc:CommandDefinitionType/Alias

diagram						
type	tc:AliasType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameSet	string	required			
	alias	string	required			

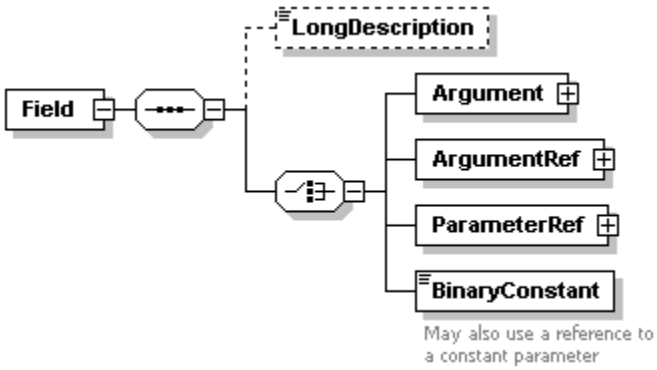
3.2.205 element tc:CommandDefinitionType/SystemName

diagram	
type	tc:SystemNameType

3.2.206 element tc:CommandDefinitionType/Interlock

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	scope	tc:InterlockScopeType	required			
	stage	tc:StageType	required			
annotation	document	Specifies the interlock for this command, including the scope of the interlock. An interlock restricts commanding ability at the specified scope until this command has reached the specified stage.				

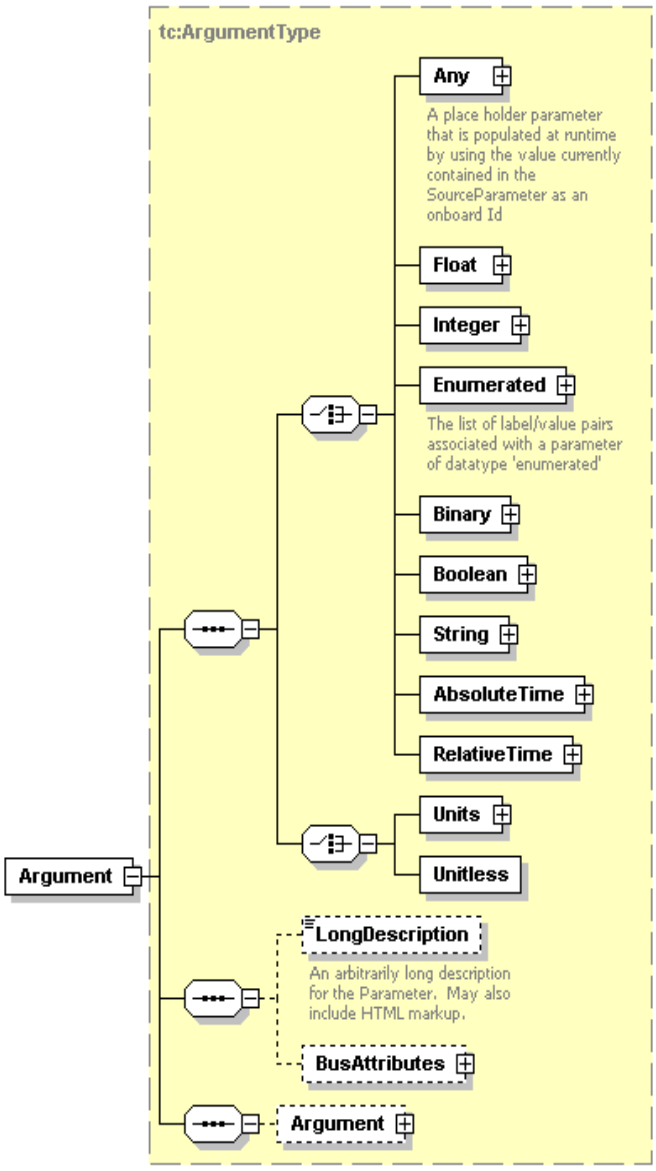
3.2.207 element tc:CommandDefinitionType/Field

diagram						
children	LongDescription Argument ArgumentRef ParameterRef BinaryConstant					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string				
	editable	boolean		true		

3.2.208 element tc:CommandDefinitionType/Field/LongDescription

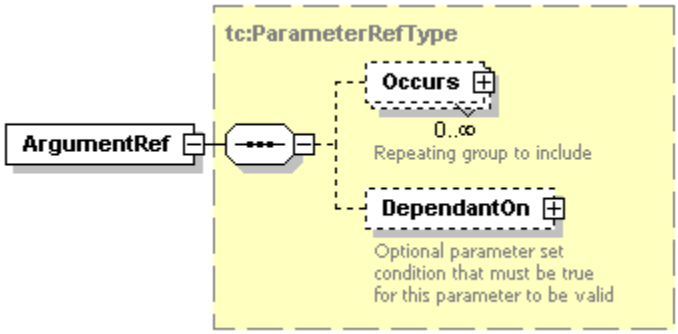
diagram	
type	tc:LongDescriptionType

3.2.209 element tc:CommandDefinitionType/Field/Argument

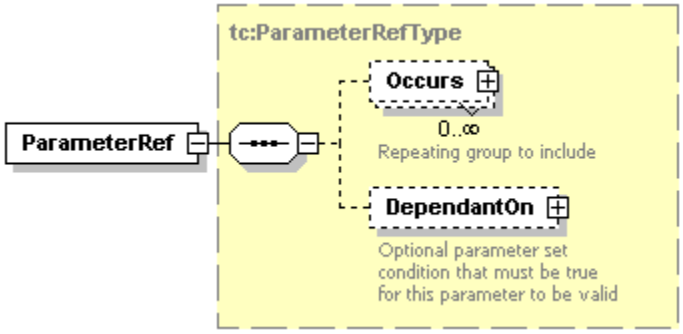
diagram	

type	tc:ArgumentType					
children	Any Float Integer Enumerated Binary Boolean String AbsoluteTime RelativeTime Units Unitless LongDescription BusAttributes Argument					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	tc:NameType	required			
	shortDescription	tc:ShortDescriptionType	optional			

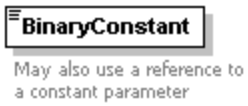
3.2.210 element tc:CommandDefinitionType/Field/ArgumentRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	Name	Type	Use	Default	Fixed	Annotation
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

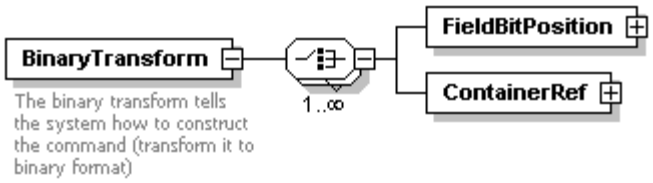
3.2.211 element tc:CommandDefinitionType/Field/ParameterRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

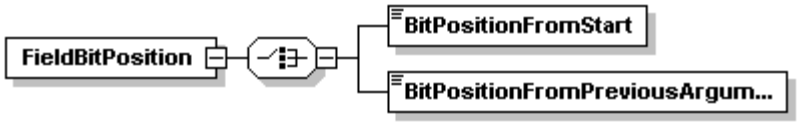
3.2.212 element tc:CommandDefinitionType/Field/BinaryConstant

diagram						
type	extension of **hexBinary**					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	lengthInBits	nonNegativeInteger				
annotation	document	May also use a reference to a constant parameter.				

3.2.213 element tc:CommandDefinitionType/BinaryTransform

diagram		
children	FieldBitPosition ContainerRef	
annotation	document	The binary transform tells the system how to construct the command (transform it to binary format).

3.2.214 element tc:CommandDefinitionType/BinaryTransform/FieldBitPosition

diagram						
children	BitPositionFromStart BitPositionFromPreviousArgument					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	argumentName	string				

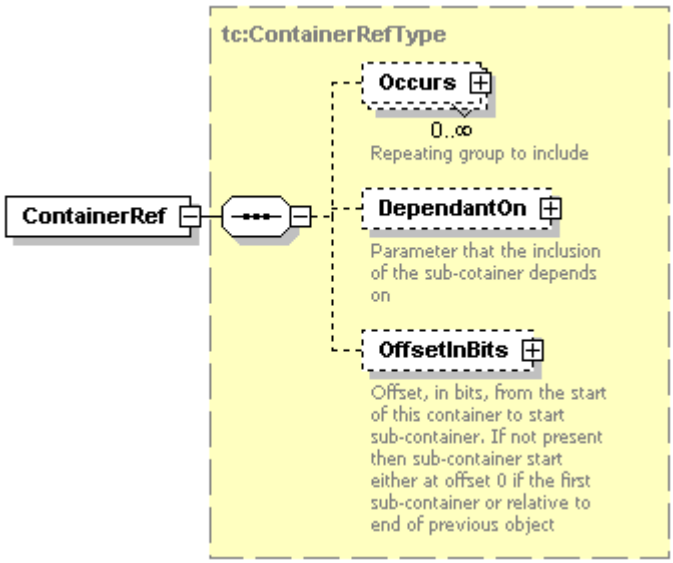
3.2.215 element tc:CommandDefinitionType/BinaryTransform/FieldBitPosition/BitPositionFromStart

diagram	
type	**nonNegativeInteger**

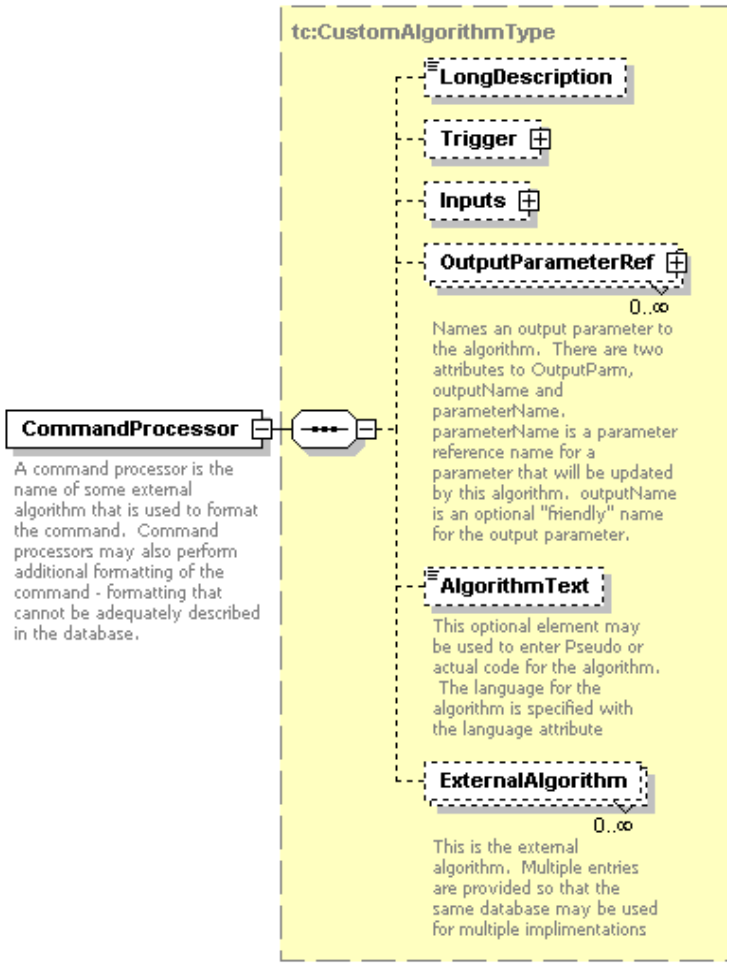
3.2.216 element tc:CommandDefinitionType/BinaryTransform/FieldBitPosition/BitPositionFromPreviousArgument

diagram	
type	**integer**

3.2.217 element tc:CommandDefinitionType/BinaryTransform/ContainerRef

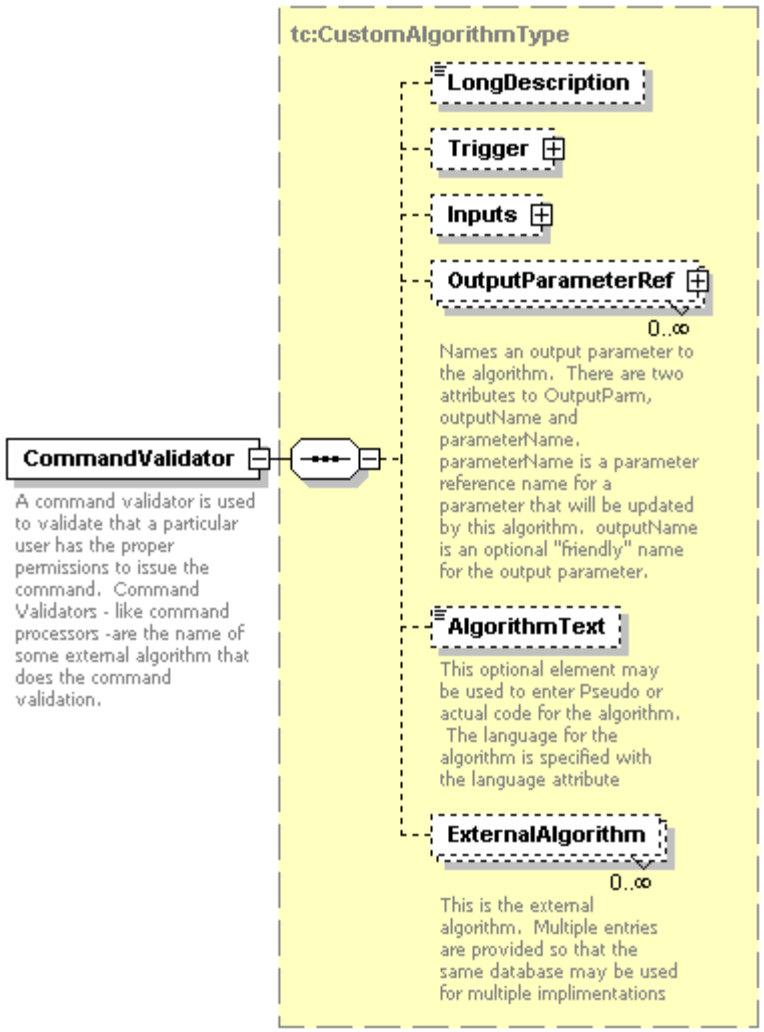
diagram						
type	tc:ContainerRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameRef	tc:ContainerNameType	required			documentation: Id of container to contain.

3.2.218 element tc:CommandDefinitionType/CommandProcessor

diagram							
type	tc:CustomAlgorithmType						
children	LongDescription Trigger Inputs OutputParameterRef AlgorithmText ExternalAlgorithm						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	required			
	shortDescription	tc:ShortDescriptionType	optional			

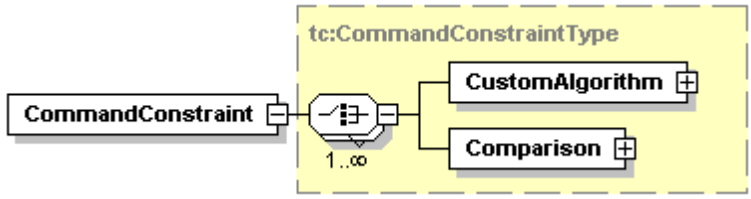
attributes	Name	Type	Use	Default	Fixed	Annotation
	thread	boolean	optional			
	triggerContainer	string	optional			documentation: First telemetry container from which the output parameter should be calculated.
	priority	integer	optional			documentation: Algorithm processing priority.
annotation	document	A command processor is the name of some external algorithm that is used to format the command. Command processors may also perform additional formatting of the command - formatting that cannot be adequately described in the database.				

3.2.219 element tc:CommandDefinitionType/CommandValidator

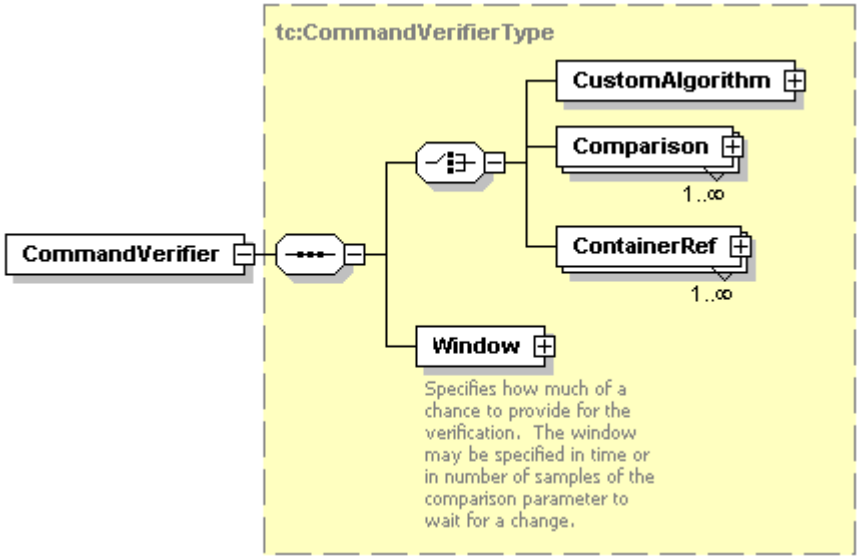
diagram	
type	tc:CustomAlgorithmType

children	LongDescription Trigger Inputs OutputParameterRef AlgorithmText ExternalAlgorithm					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	string	required			
	shortDescription	tc:ShortDescriptionType	optional			
	thread	boolean	optional			
	triggerContainer	string	optional			documentation: First telemetry container from which the output parameter should be calculated.
	priority	integer	optional			documentation: Algorithm processing priority.
annotation	document	A command validator is used to validate that a particular user has the proper permissions to issue the command. Command Validators - like command processors -are the name of some external algorithm that does the command validation.				

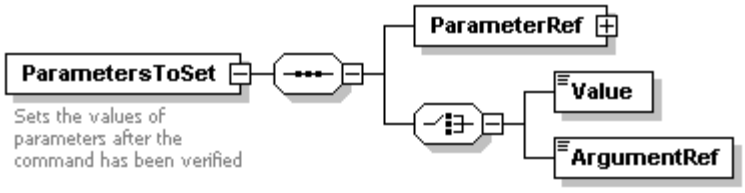
3.2.220 element tc:CommandDefinitionType/CommandConstraint

diagram	
type	tc:CommandConstraintType
children	CustomAlgorithm Comparison

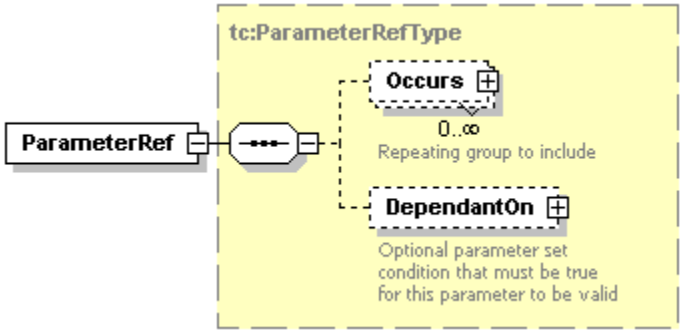
3.2.221 element tc:CommandDefinitionType/CommandVerifier

diagram						
type	tc:CommandVerifierType					
children	CustomAlgorithm Comparison ContainerRef Window					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	verificationStage	tc:StageType		completion		

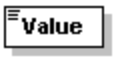
3.2.222 element tc:CommandDefinitionType/ParametersToSet

diagram		
children	ParameterRef Value ArgumentRef	
annotation	document	Sets the values of parameters after the command has been verified.

3.2.223 element tc:CommandDefinitionType/ParametersToSet/ParameterRef

diagram						
type	tc:ParameterRefType					
children	Occurs DependantOn					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				

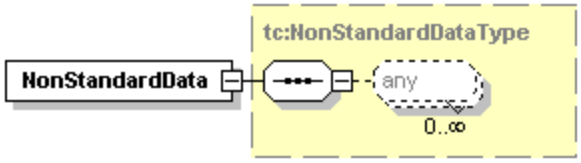
3.2.224 element tc:CommandDefinitionType/ParametersToSet/Value

diagram	
type	**string**

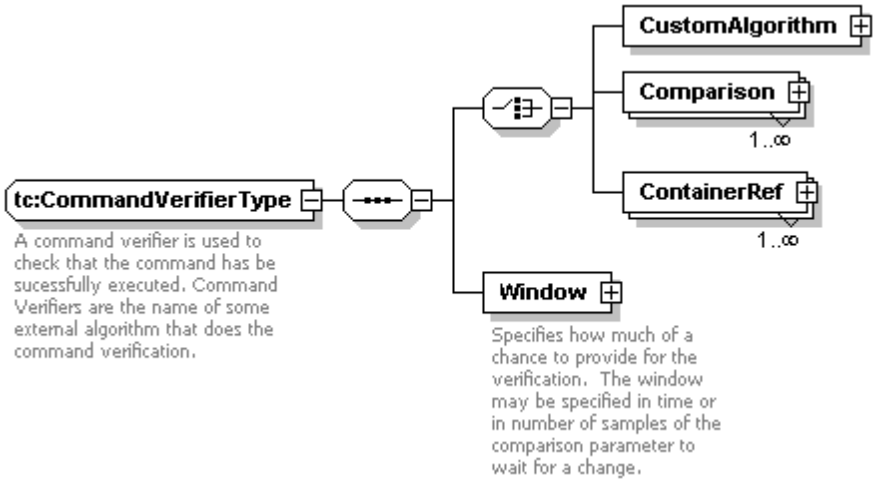
3.2.225 element tc:CommandDefinitionType/ParametersToSet/ArgumentRef

diagram	
type	**string**

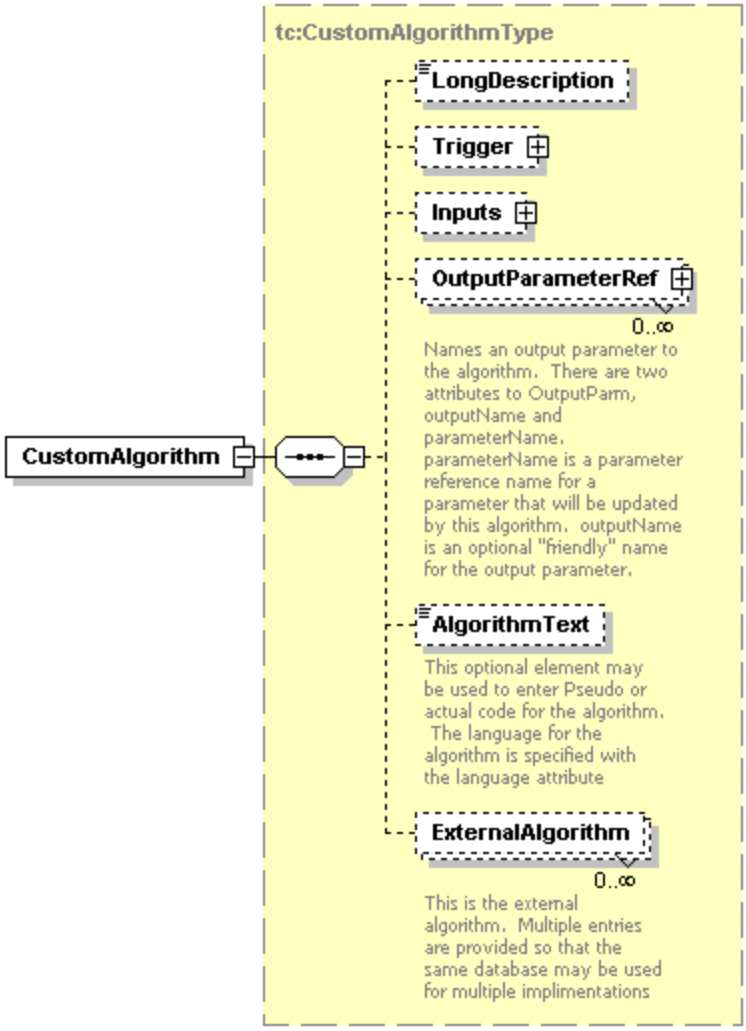
3.2.226 element tc:CommandDefinitionType/NonStandardData

diagram						
type	tc:NonStandardDataType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	usage	string	required			

3.2.227 complexType tc:CommandVerifierType

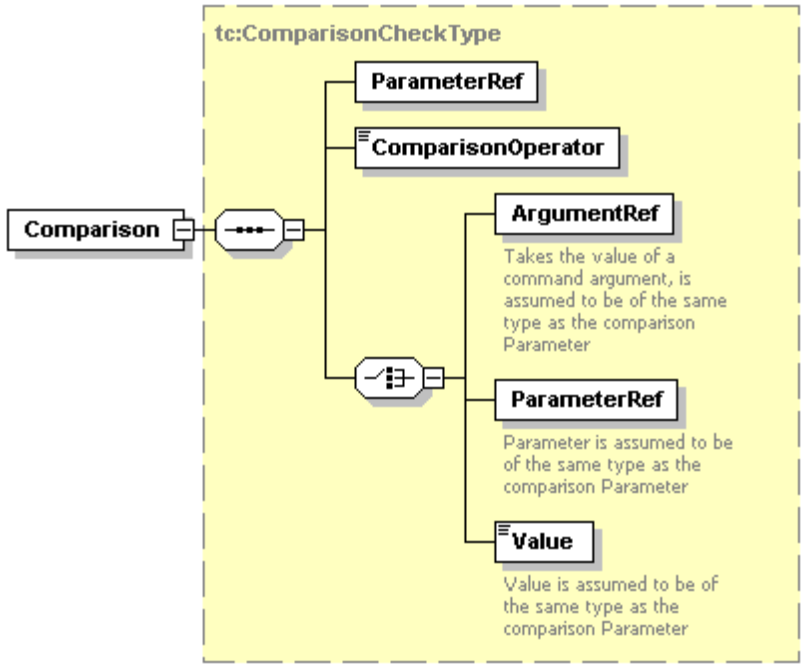
diagram						
namespace	http://www.omg.org/space/tcspec					
children	CustomAlgorithm Comparison ContainerRef Window					
used by	element: tc:CommandDefinitionType/CommandVerifier					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	verificationStage	tc:StageType		Completion		
annotation	document	A command verifier is used to check that the command has be sucessfully executed. Command Verifiers are the name of some external algorithm that does the command verification.				

3.2.228 element tc:CommandVerifierType/CustomAlgorithm

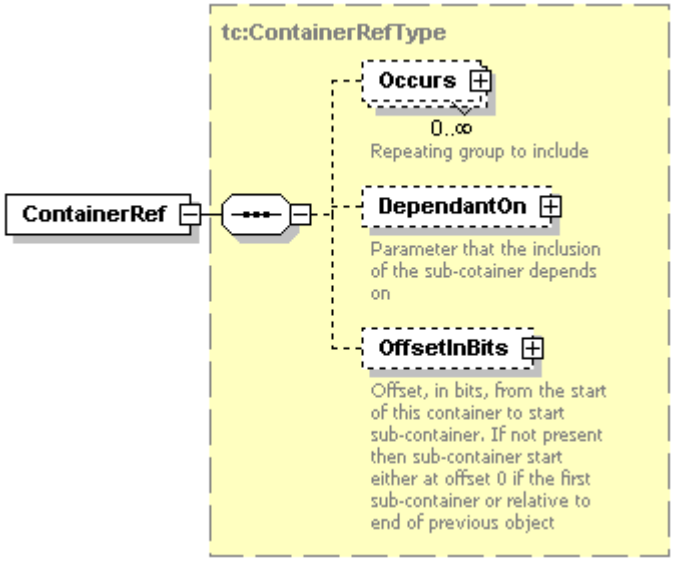
diagram							
type	tc:CustomAlgorithmType						
children	LongDescription Trigger Inputs OutputParameterRef AlgorithmText ExternalAlgorithm						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**	
	name	string	required				
	shortDescription	tc:ShortDescription-Type	optional				
	thread	boolean	optional				

	triggerContainer	string	optional			documentation: First telemetry container from which the output parameter should be calculated.
	priority	integer	optional			documentation: Algorithm processing priority..

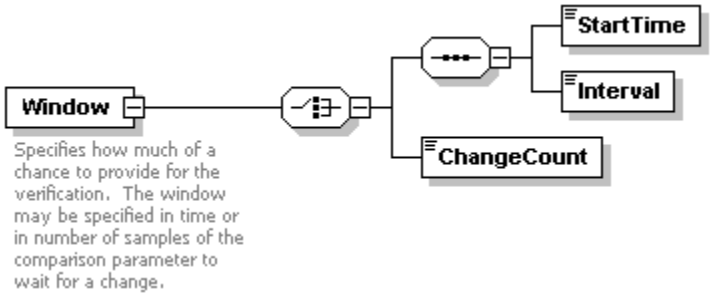
3.2.229 element tc:CommandVerifierType/Comparison

diagram						
type	tc:ComparisonCheckType					
children	ParameterRef ComparisonOperator ArgumentRef ParameterRef Value					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	calibratedComparison	boolean		false		documentation: Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.

3.2.230 element tc:CommandVerifierType/ContainerRef

diagram						
type	tc:ContainerRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameRef	tc:ContainerNameType	required			documentation: Id of container to contain.

3.2.231 element tc:CommandVerifierType/Window

diagram		
children	StartTime Interval ChangeCount	
annotation	document	Specifies how much of a chance to provide for the verification. The window may be specified in time or in number of samples of the comparison parameter to wait for a change.

3.2.232 element tc:CommandVerifierType/Window/StartTime

diagram	
type	nonNegativeInteger

3.2.233 element **tc:CommandVerifierType/Window/Interval**

diagram	
type	positiveInteger

3.2.234 element **tc:CommandVerifierType/Window/ChangeCount**

diagram	
type	positiveInteger

3.2.235 simpleType tc:InterlockScopeType

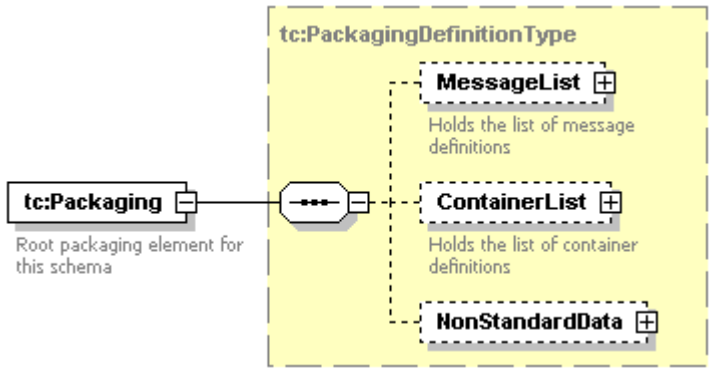
namespace	http://www.omg.org/space/tcspec	
type	restriction of **string**	
used by	element: tc:CommandDefinitionType/Interlock/@scope	
facets	enumeration	SystemGlobal
	enumeration	SystemLocal
	enumeration	SubsystemGlobal
	enumeration	SubsystemLocal
annotation	document	Specifies the scope of the interlock.

3.2.236 simpleType tc:StageType

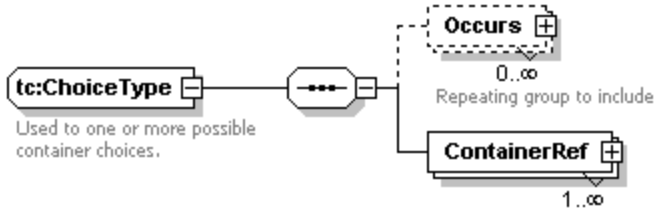
namespace	http://www.omg.org/space/tcspec
type	restriction of string

used by	element: tc:CommandDefinitionType/Interlock/@stage tc:CommandVerifierType/@verificationStage	
facets	enumeration	Reception
	enumeration	Uplink
	enumeration	Transfer
	enumeration	Acceptance
	enumeration	Execution
	enumeration	Completion
annotation	document	Specifies the stage of command uplink.

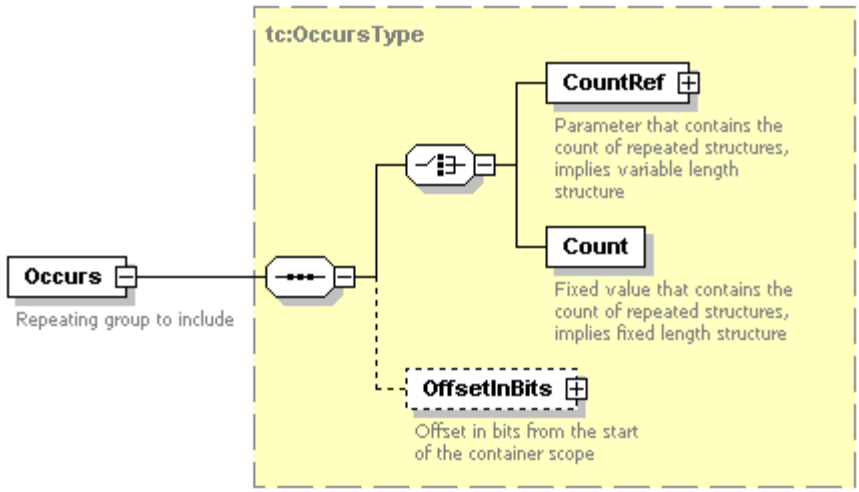
3.2.237 element tc:Packaging

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:PackagingDefinitionType	
children	MessageList ContainerList NonStandardData	
used by	element: SpaceSystemType/TelemetryAndCommanding	
annotation	document	Root packaging element for this schema.

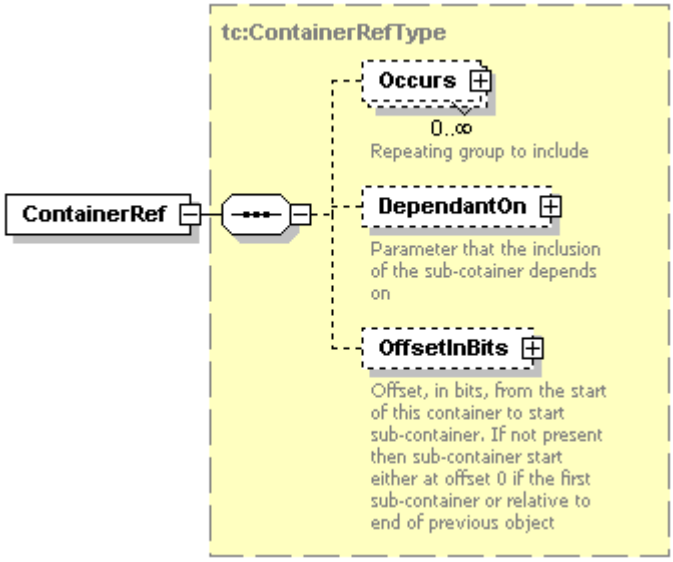
3.2.238 complexType tc:ChoiceType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	Occurs ContainerRef	
used by	element: tc:SequenceType/Choice	
annotation	document	Used to one or more possible container choices.

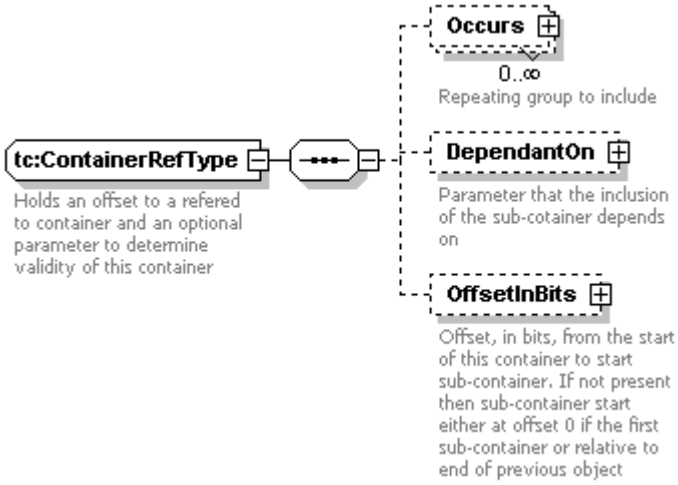
3.2.239 element tc:ChoiceType/Occurs

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:OccursType					
children	CountRef Count OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	type	NMTOKENS		Occurs		
annotation	documentation	Repeating group to include.				

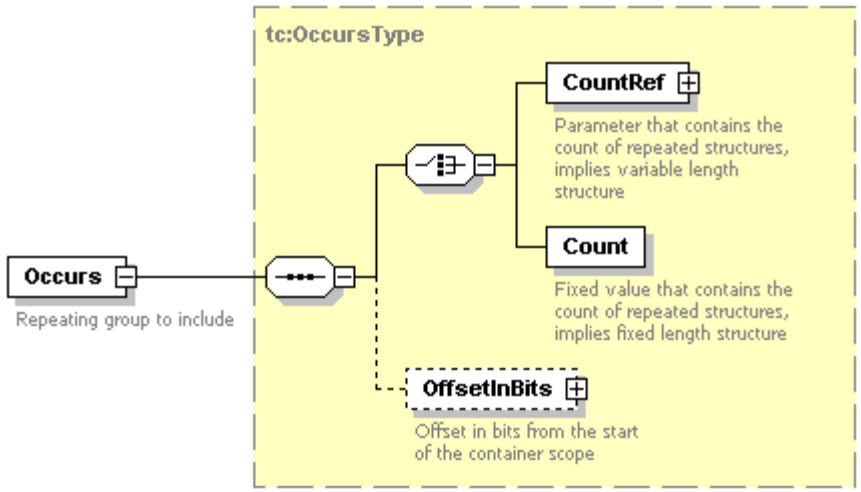
3.2.240 element tc:ChoiceType/ContainerRef

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:ContainerRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameRef	tc:ContainerNameType	required			documentation: Id of container to contain.

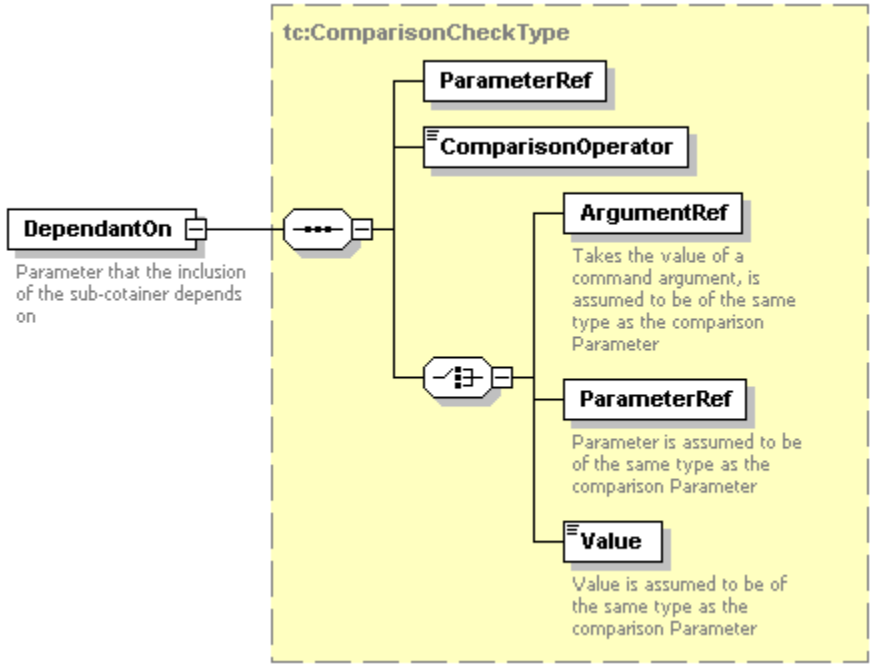
3.2.241 complexType tc:ContainerRefType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	Occurs DependantOn OffsetInBits					
used by	elements: tc:CommandDefinitionType/BinaryTransform/ContainerRef tc:CommandVerifierType/ContainerRef tc:SequenceType/ContainerRef tc:ChoiceType/ContainerRef					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameRef	tc:ContainerNameType	required			documentation: Id of container to contain.
annotation	documentation	Holds an offset to a referred to container and an optional parameter to determine validity of this container.				

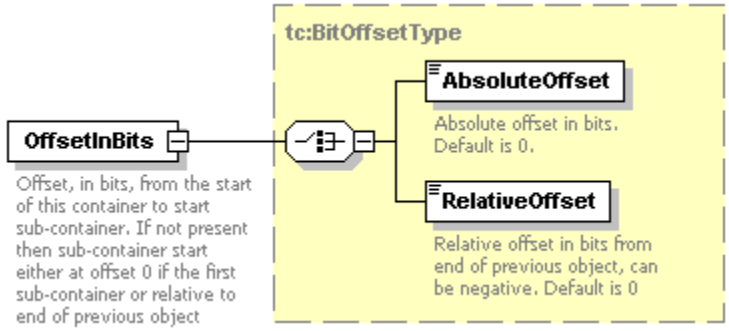
3.2.242 element tc:ContainerRefType/Occurs

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:OccursType					
children	CountRef Count OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	type	NMTOKENS		Occurs		
annotation	documentation	Repeating group to include.				

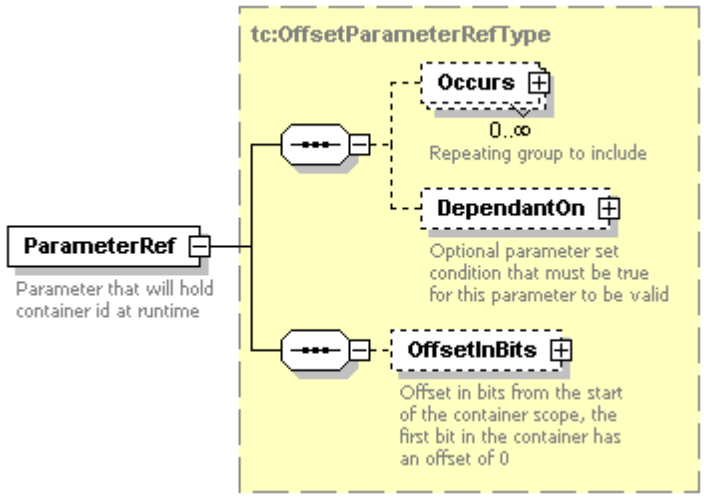
3.2.243 element tc:DynamicContainerRefType/DependantOn

<table>
<tr><td>diagram</td><td colspan="6"> 
</td></tr>
<tr><td>namespace</td><td colspan="6">http://www.omg.org/space/tcspec</td></tr>
<tr><td>type</td><td colspan="6">tc:ComparisonCheckType</td></tr>
<tr><td>children</td><td colspan="6">ParameterRef ComparisonOperator ArgumentRef ParameterRef Value</td></tr>
<tr><td>attributes</td><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td></td><td>calibratedComparison</td><td>boolean</td><td></td><td>false</td><td></td><td>documentation:
Should the comparison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.</td></tr>
<tr><td>annotation</td><td>documentation</td><td colspan="5">Parameter that the inclusion of the sub-container depends on.</td></tr>
</table>

3.2.244 element tc:DynamicContainerRefType/OffsetInBits

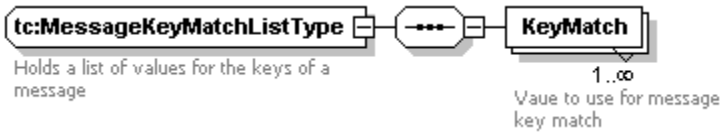
diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:BitOffsetType	
children	AbsoluteOffset RelativeOffset	
annotation	documentation	Offset, in bits, from the start of this container to start sub-container. If not present, then sub-container start either at offset 0 if the first sub-container or relative to end of previous object.

3.2.245 element tc:DynamicContainerRefType/ParameterRef

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:OffsetParameterRefType	
children	Occurs DependantOn OffsetInBits	

attributes	Name	Type	Use	Default	Fixed	Annotation
	parameterName	tc:NameReference-Type	required			
	spaceSystem-Name	string				
	TimeOffset	integer	optional	0		documentation: Time delay in milliseconds.
annotation	documentation	Parameter that will hold container id at runtime.				

3.2.246 complexType tc:MessageKeyMatchListType

diagram						
namespace	http://www.omg.org/space/tcspec					
type	KeyMatch					
used by	element: tc:ContainerType/ImplementsMessage					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:Message-NameType	required			documentation: Unique identifier for the message
	AllMustBeTrue	boolean				documentation: Must all matched be true? If not present, then all must be true.
annotation	documentation	Holds a list of values for the keys of a message.				

3.2.247 element tc:MessageKeyMatchListType/KeyMatch

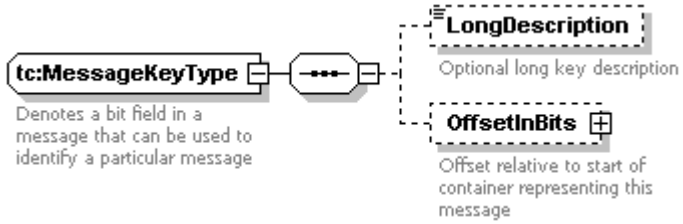
diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:MessageKeyMatchType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**

	Id	tc:MessageKeyIdType	required			documentation: Id of message key to match to.
	Operator	tc:ComparisonOperatorsType				documentation: Operator to compare with, optional, if not present equivalence assumed.
	Value	hexBinary	required			documentation: Value of key to match with.
annotation	documentation	Value to use for message key match.				

3.2.248 complexType tc:MessageKeyMatchType

diagram						
namespace	http://www.omg.org/space/tcspec					
type	KeyMatch					
used by	element: tc:MessageKeyMatchListType/KeyMatch					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	Id	tc:MessageKeyIdType	required			documentation: Id of message key to match to.
	Operator	tc:ComparisonOperatorsType				documentation: Operator to compare with, optional, if not present equivalence assumed.
	Value	hexBinary	required			documentation: Value of key to match with.
annotation	documentation	Holds a value to use in a match to a referred to key.				

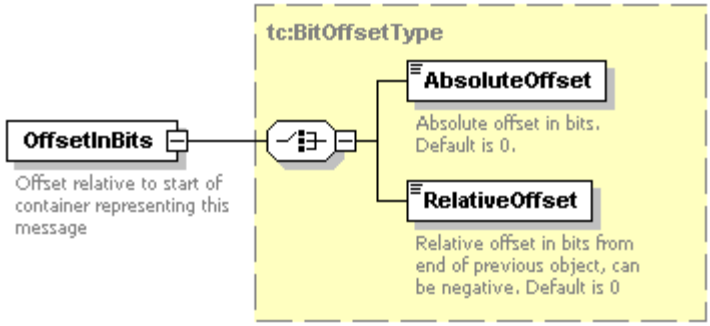
3.2.249 complexType tc:MessageKeyType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	LongDescription OffsetInBits					
used by	element: tc:MessageType/Key					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	Id	tc:MessageKeyIdType	required			documentation: Id of message key field.
	ShortDescription	tc:ShortDescription-Type	optional			documentation: Optional short key description.
	WidthInBits	positiveInteger	required			documentation: Width of key field in bits.
annotation	documentation	Denotes a bit field in a message that can be used to identify a particular message.				

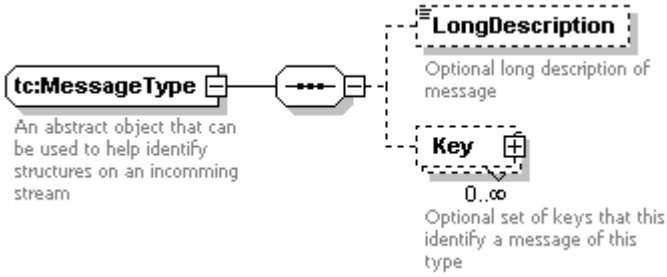
3.2.250 element tc:MessageKeyType/LongDescription

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:LongDescriptionType	
annotation	documentation	Optional long key description.

3.2.251 element tc:MessageKeyType/OffsetInBits

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:BitOffsetType	
children	AbsoluteOffset RelativeOffset	
annotation	documentation	Offset relative to start of container representing this message.

3.2.252 complexType tc:MessageType

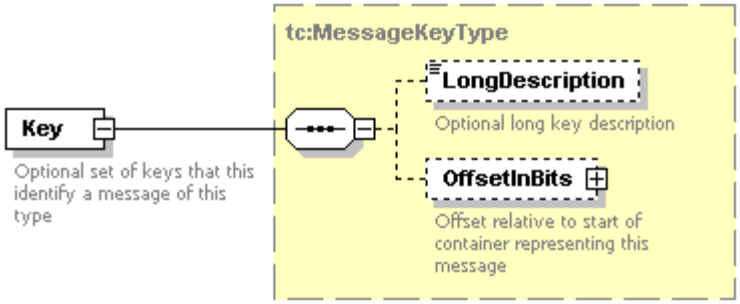
diagram						
namespace	http://www.omg.org/space/tcspec					
children	LongDescription Key					
used by	element: tc:PackagingDefinitionType/MessageList/Message					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:MessageNameType	required			documentation: Unique identifier for the message.

	shortDescription	tc:ShortDescription-Type	optional			documentation: Optional short name of the message.
annotation	documentation	An abstract object that can be used to help identify structures on an incoming stream.				

3.2.253 element tc:MessageType/LongDescription

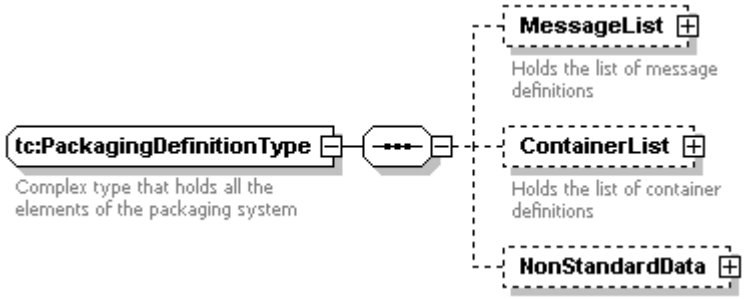
diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:LongDescriptionType	
annotation	documentation	Optional long description of message.

3.2.254 element tc:MessageType/Key

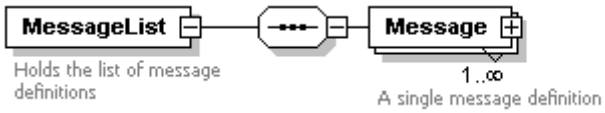
diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:MessageKeyType					
children	LongDescription OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	Id	tc:MessageKeyIdType	required			documentation: Id of message key field.
	ShortDescription	tc:ShortDescription-Type	optional			documentation: Optional short key description.

	WidthInBits	positiveInteger	required			documentation: Width of key field in bits.
annotation	documentation	Optional set of keys that this identify a message of this type.				

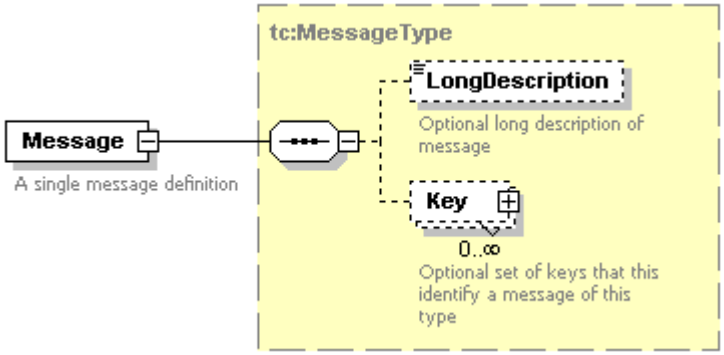
3.2.255 complexType tc:PackagingDefinitionType

diagram		
namespace	http://www.omg.org/space/tcspec	
children	MessageList ContainerList NonStandardData	
used by	element: tc:Packaging	
annotation	documentation	Complex type that holds all the elements of the packaging system.

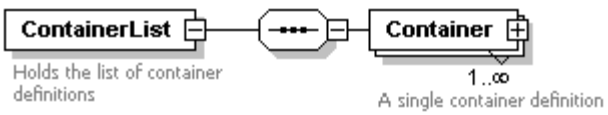
3.2.256 element tc:PackagingDefinitionType/MessageList

diagram					
namespace	http://www.omg.org/space/tcspec				
children	Message				
identity constraints		**Name**	**Refer**	**Selector**	**Field(s)**
	key	MessageKey		Message	Id
annotation	documentation	Holds the list of message definitions.			

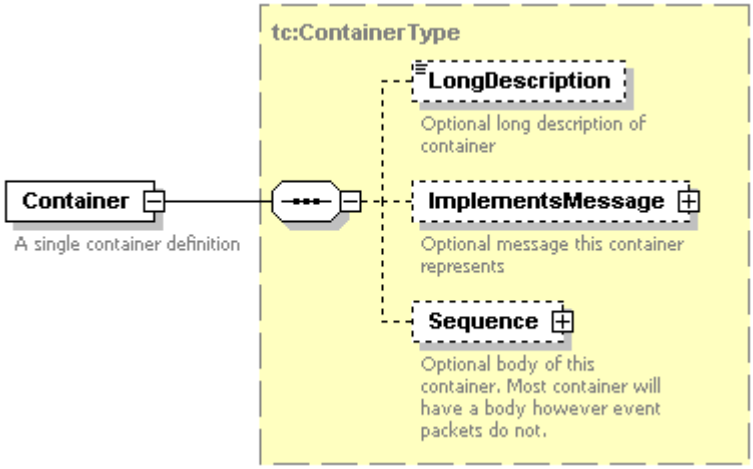
3.2.257 element tc:PackagingDefinitionType/MessageList/Message

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:MessageType					
children	LongDescription Key					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:MessageNameType	required			documentation: Unique identifier for the message.
	shortDescription	tc:ShortDescription-Type	optional			documentation: Optional short name of the message.
annotation	documentation	A single message definition.				

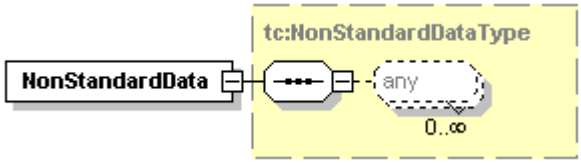
3.2.258 element tc:PackagingDefinitionType/ContainerList

diagram					
namespace	http://www.omg.org/space/tcspec				
children	Container				
identity constraints		**Name**	**Refer**	**Selector**	**Field(s)**
	key	ContainerKey		Container	Id
annotation	documentation	Holds the list of container definitions.			

3.2.259 element tc:PackagingDefinitionType/ContainerList/Container

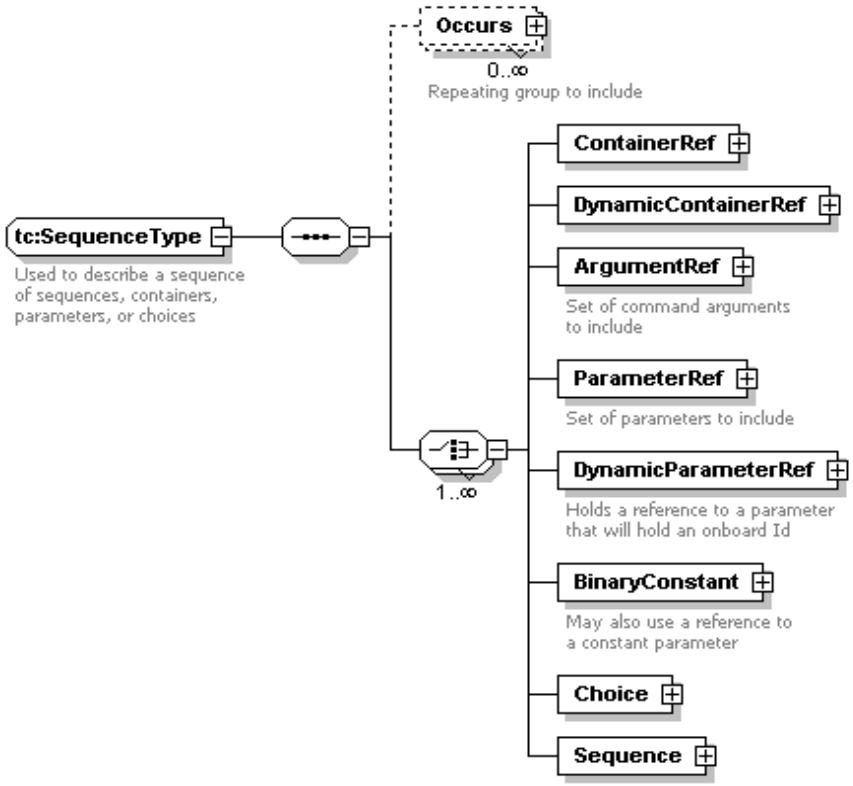
diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:ContainerType					
children	LongDescription ImplementsMessage Sequence					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:ContainerNameType	required			documentation: Id of container.
	shortDescription	tc:ShortDescription-Type	optional			documentation: Optional short description of container.
annotation	documentation	A single container definition.				

3.2.260 element tc:PackagingDefinitionType/NonStandardData

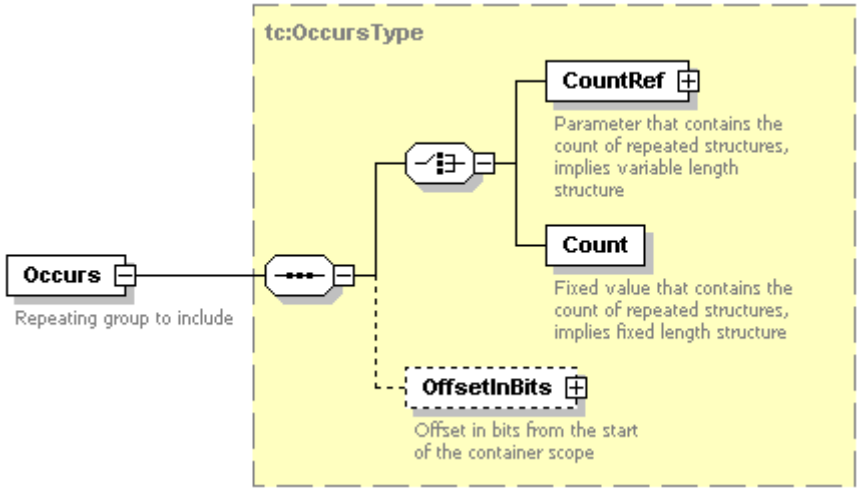
diagram	
namespace	http://www.omg.org/space/tcspec

type	tc:NonStandardDataType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	usage	string	required			documentation: Unique identifier for the message.

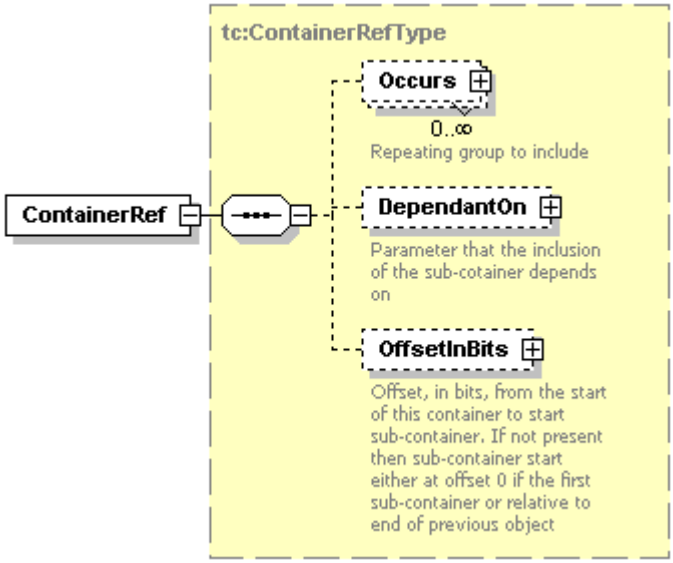
3.2.261 complexType tc:SequenceType

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:ContainerType	
children	Occurs ContainerRef DynamicContainerRef ArgumentRef ParameterRef DynamicParameterRef BinaryConstant Choice Sequence	
used by	elements: tc:ContainerType/Sequence tc:SequenceType/Sequence	
annotation	documentation	Used to describe a sequence of sequences, containers, parameters, or choices.

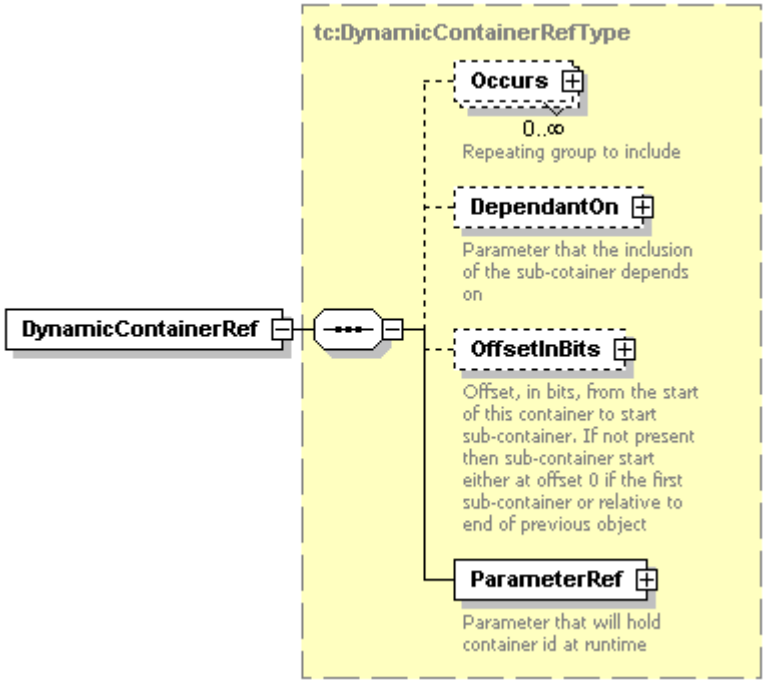
3.2.262 element tc:SequenceType/Occurs

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:OccursType					
children	CountRef Count OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	type	NMTOKENS		Occurs		
annotation	documentation	Repeating group to include.				

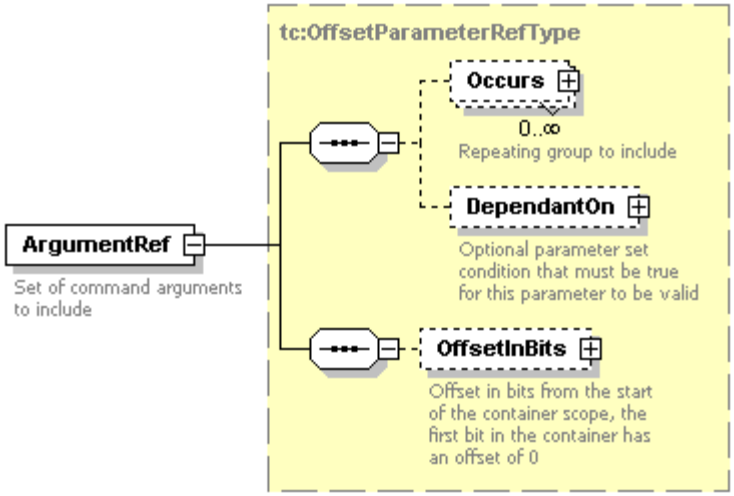
3.2.263 element tc:SequenceType/ContainerRef

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:ContainerRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	nameRef	tc:ContainerNameType	required			documentation: Id of container to contain.

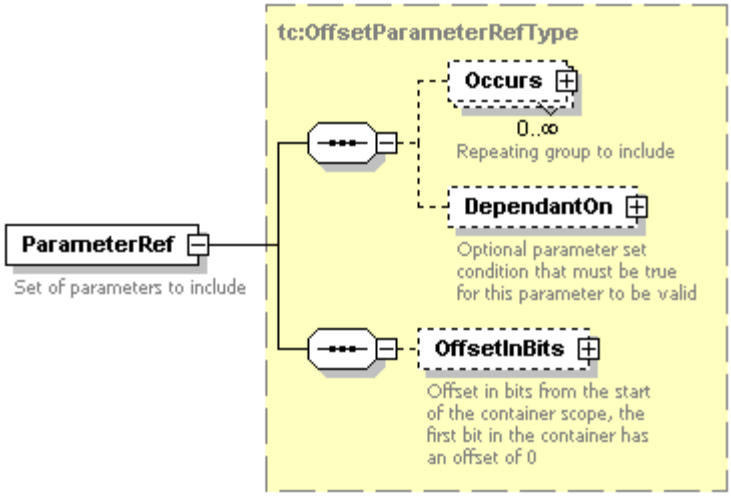
3.2.264 element tc:SequenceType/DynamicContainerRef

diagram	
namespace	http://www.omg.org/space/tcspec
type	tc:DynamicContainerRefType
children	Occurs DependantOn OffsetInBits ParameterRef

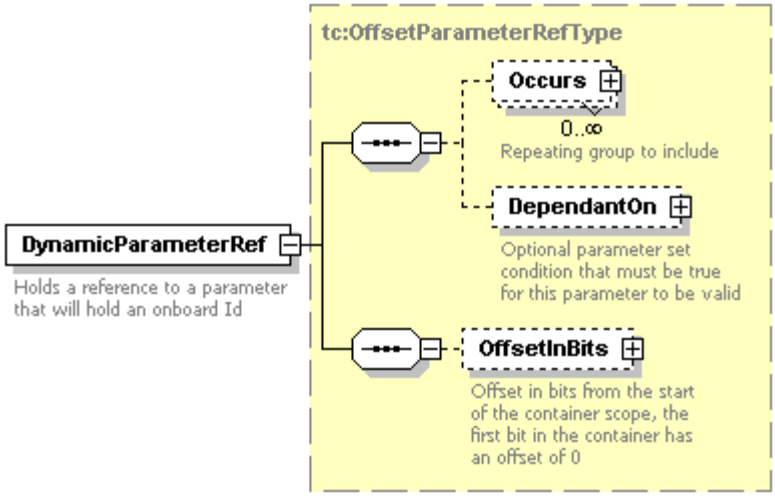
3.2.265 element tc:SequenceType/ArgumentRef

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:OffsetParameterRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
	TimeOffset	integer	optional	0		documentation: Time delay in millisec-onds.
annotation	documentation	Set of command arguments to include.				

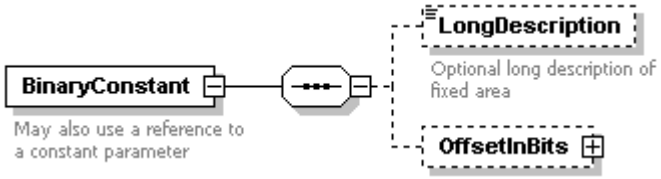
3.2.266 element tc:SequenceType/ParameterRef

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:OffsetParameterRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
	TimeOffset	integer	optional	0		documentation: Time delay in milliseconds.
annotation	documentation	Set of parameters to include.				

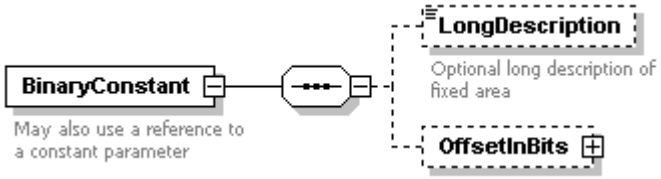
3.2.267 element tc:SequenceType/DynamicParameterRef

diagram						
namespace	http://www.omg.org/space/tcspec					
type	tc:OffsetParameterRefType					
children	Occurs DependantOn OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	parameterName	tc:NameReferenceType	required			
	spaceSystemName	string				
	TimeOffset	integer	optional	0		documentation: Time delay in millisec-onds.
annotation	documentation	Holds a reference to a parameter that will hold an onboard Id.				

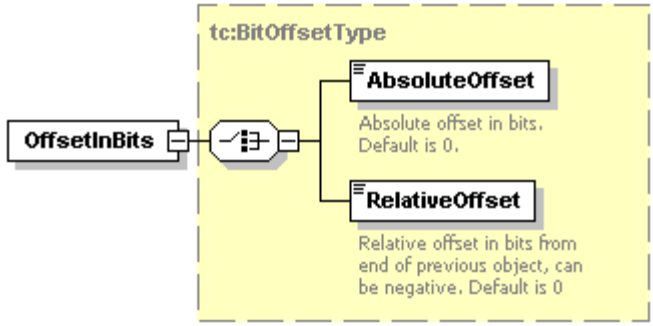
3.2.268 element tc:SequenceType/BinaryConstant

diagram						
namespace	http://www.omg.org/space/tcspec					
children	LongDescription OffsetInBits					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	lengthInBits	nonNegativeInteger	required			
	value	hexBinary	required			
annotation	documentation	May also use a reference to a constant parameter.				

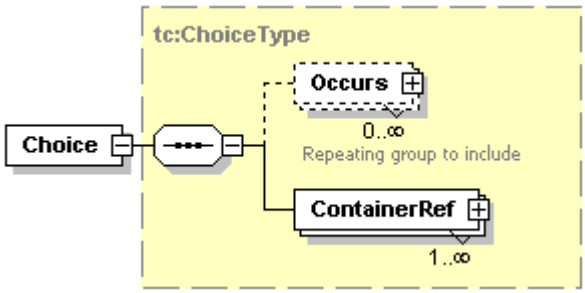
3.2.269 element tc:SequenceType/BinaryConstant/LongDescription

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:LongDescriptionType	
annotation	documentation	Optional long description of fixed area.

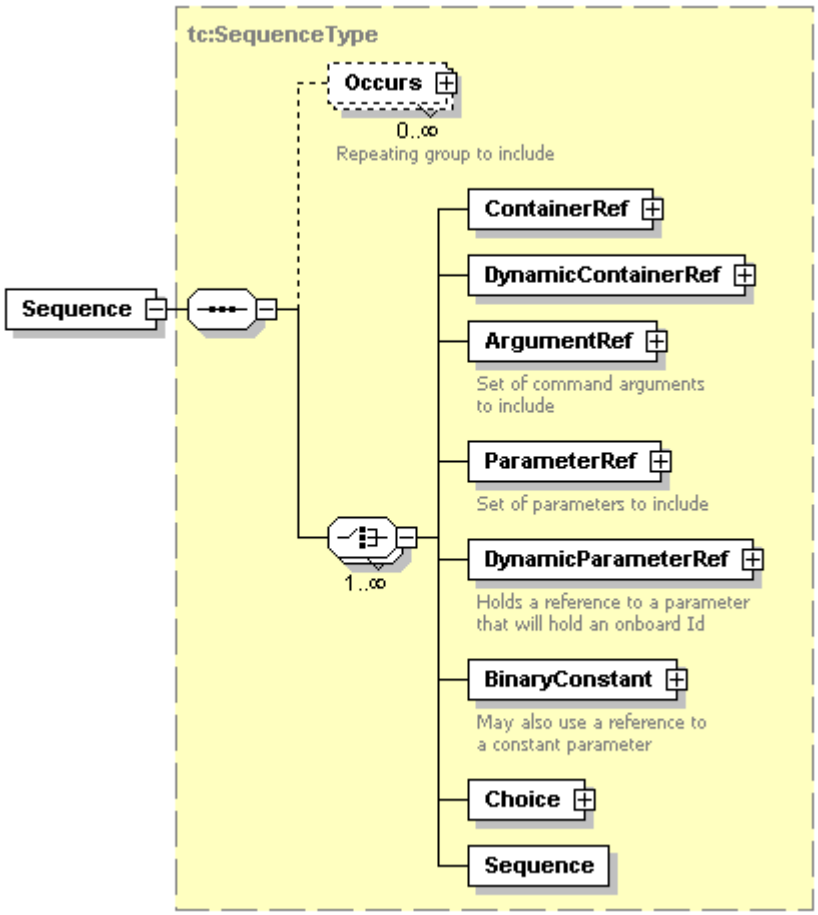
3.2.270 element tc:SequenceType/BinaryConstant/OffsetInBits

diagram	
namespace	http://www.omg.org/space/tcspec
type	tc:BitOffsetType
children	AbsoluteOffset RelativeOffset

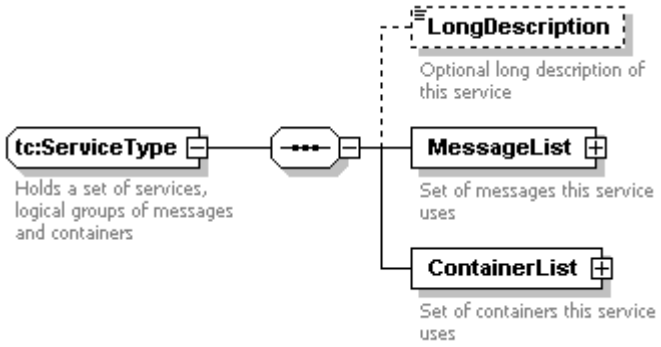
3.2.271 element ***tc:SequenceType/Choice***

diagram	
namespace	http://www.omg.org/space/tcspec
type	tc:ChoiceType
children	Occurs ContainerRef

3.2.272 element tc:SequenceType/Sequence

diagram	
namespace	http://www.omg.org/space/tcspec
type	tc:SequenceType
children	Occurs ContainerRef DynamicContainerRef ArgumentRef ParameterRef DynamicParameterRef BinaryConstant Choice Sequence

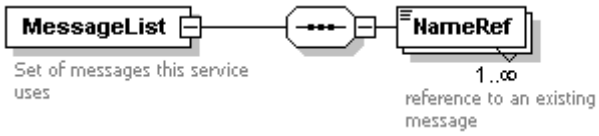
3.2.273 complexType tc:ServiceType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	LongDescription MessageList ContainerList					
used by	element: SpaceSystemType/TelemetryAndCommanding/ServiceList/Service					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:ServiceNameType	required			documentation: Unique identifier for this service set.
	shortDescription	tc:ShortDescriptionType	optional			documentation: Optional short description of this service.
annotation	documentation	Holds a set of services, logical groups of messages and containers.				

3.2.274 element tc:ServiceType/LongDescription

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:LongDescriptionType	
annotation	documentation	Optional long description of this service.

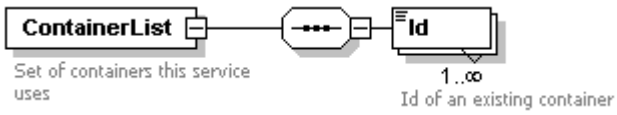
3.2.275 element tc:ServiceType/MessageList

diagram		
namespace	http://www.omg.org/space/tcspec	
children	NameRef	
annotation	documentation	Set of messages this service uses.

3.2.276 element tc:ServiceType/MessageList/NameRef

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:NameReferenceType	
facets	pattern	a-zA-Z0-9_\- /]*
annotation	documentation	Reference to an existing message.

3.2.277 element tc:ServiceType/ContainerList

diagram		
namespace	http://www.omg.org/space/tcspec	
children	Id	
annotation	documentation	Set of containers this service uses.

3.2.278 element tc:ServiceType/ContainerList/Id

diagram		
namespace	http://www.omg.org/space/tcspec	
type	tc:ContainerNameType	
facets	pattern	a-zA-Z0-9_\- /]*
annotation	documentation	Id of an existing container.

3.2.279 simpleType tc:ContainerNameType

namespace	http://www.omg.org/space/tcspec	
type	tc:NameType	
used by	element tc:ServiceType/ContainerList/Id attributes tc:ContainerType/@name tc:ContainerRefType/@nameRef	
facets	pattern	a-zA-Z0-9_\- /]*
annotation	documentation	Identifier for a container.

3.2.280 simpleType tc:MessageKeyIdType

namespace	http://www.omg.org/space/tcspec	
type	tc:NameType	
used by	attributes tc:MessageKeyType/@Id tc:MessageKeyMatchType/@Id	
facets	pattern	a-zA-Z0-9_\- /]*
annotation	documentation	Identifier for a message key.

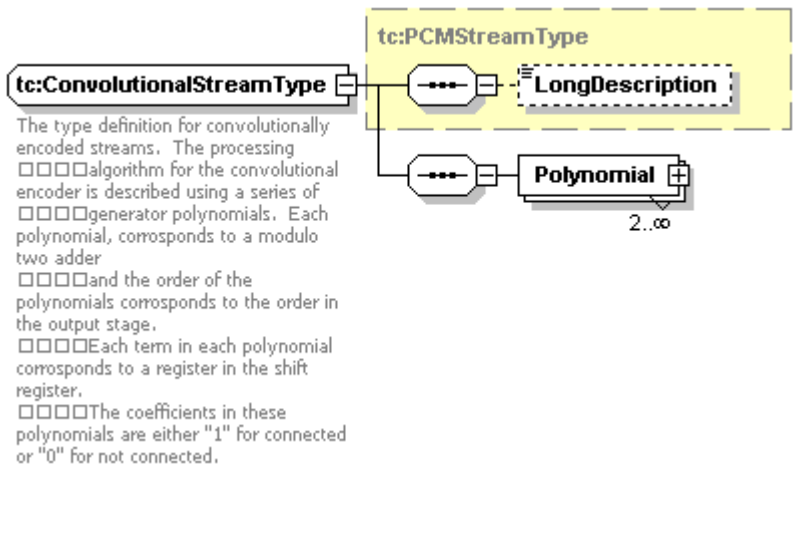
3.2.281 simpleType tc:MessageNameType

namespace	http://www.omg.org/space/tcspec	
type	tc:NameType	
used by	attributes tc:MessageType/@name tc:MessageKeyMatchListType/@name	
facets	pattern	a-zA-Z0-9_\- /]*
annotation	documentation	Identifier for a message.

3.2.282 simpleType tc:ServiceNameType

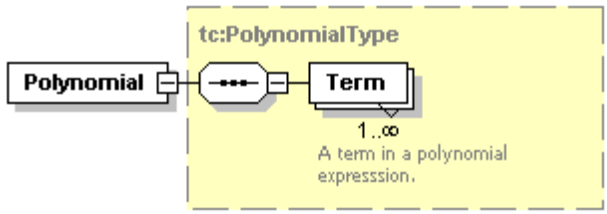
namespace	http://www.omg.org/space/tcspec	
type	tc:NameType	
used by	attribute: tc:ServiceType/@name	
facets	pattern	a-zA-Z0-9_\- /]*
annotation	documentation	Identifier for a service.

3.2.283 complexType tc:ConvolutionalStreamType

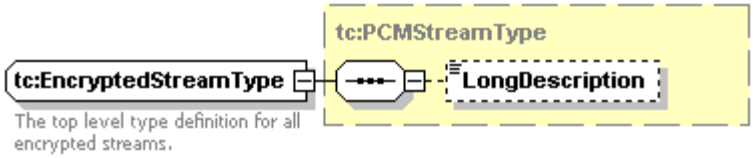
diagram	

namespace	http://www.omg.org/space/tcspec					
type	extension of tc:PCMStreamType					
children	LongDescription Polynomial					
used by	element: tc:StreamType/ConvolutionalStream					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType				
	shortDescription	tc:ShortDescription-Type				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
annotation	documentation	The type definition for convolutionally encoded streams. The processing algorithm for the convolutional encoder is described using a series of generator polynomials. Each polynomial, corresponds to a modulo two adder and the order of the polynomials corresponds to the order in the output stage. Each term in each polynomial corresponds to a register in the shift register. The coefficients in these polynomials are either "1" for connected or "0" for not connected.				

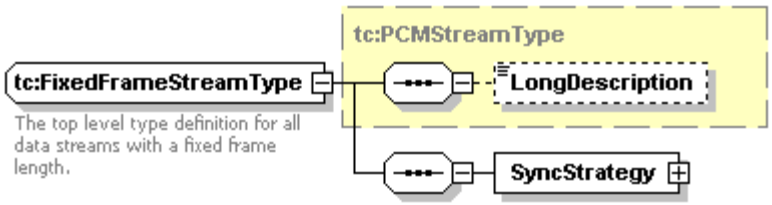
3.2.284 element tc:ConvolutionalStreamType/Polynomial

diagram	
type	tc:PolynomialType
children	Term

3.2.285 complexType tc:EncryptedStreamType

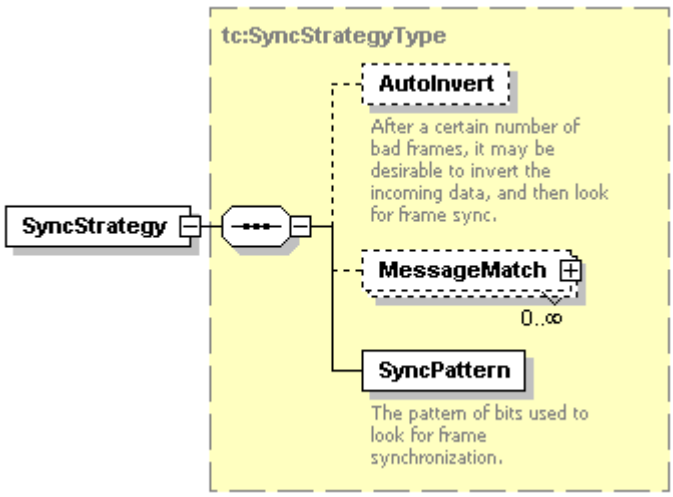
diagram						
namespace	http://www.omg.org/space/tcspec					
type	extension of tc:PCMStreamType					
children	LongDescription					
used by	element: tc:StreamType/EncryptedStream					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType				
	shortDescription	tc:ShortDescription-Type				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
	algorithmName	string	required			
	key	string				
annotation	documentation	The top level type definition for all encrypted streams.				

3.2.286 complexType tc:FixedFrameStreamType

diagram	

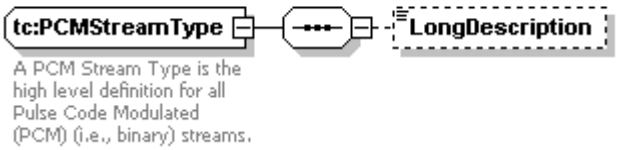
namespace	http://www.omg.org/space/tcspec					
type	extension of tc:PCMStreamType					
children	LongDescription SyncStrategy					
used by	element: tc:FixedFrameSync/FixedFrameStream tc:StreamType/FixedFrameStream					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType				
	shortDescription	tc:ShortDescription-Type				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
	frameLengthInBits	long	required			
annotation	documentation	The top level type definition for all data streams with a fixed frame length.				

3.2.287 element tc:FixedFrameStreamType/SyncStrategy

diagram	
type	tc:SyncStrategyType
children	AutoInvert MessageMatch SyncPattern

attributes	Name	Type	Use	Default	Fixed	Annotation
	verifyToLockGoodFrames	integer		4		
	checkToLockGoodFrames	integer		1		
	maxErrorsInSyncWord	integer		0		
	syncAperature	integer		0		

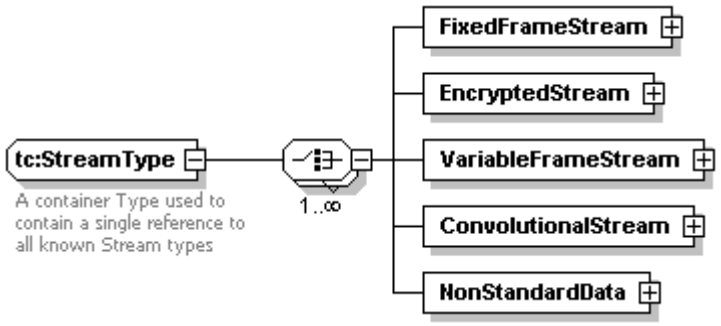
3.2.288 complexType tc:PCMStreamType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	LongDescription					
used by	complexTypes tc:ConvolutionalStreamType tc:EncryptedStreamType tc:FixedFrameStreamType tc:VariableFrameStreamType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType				
	shortDescription	tc:ShortDescription-Type				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
annotation	documentation	A PCM Stream Type is the high level definition for all Pulse Code Modulated (PCM) (i.e., binary) streams.				

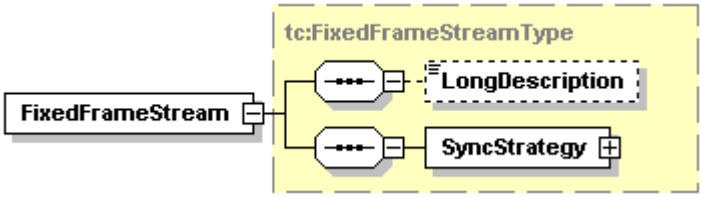
3.2.289 element tc:PCMStreamType/LongDescription

diagram	
type	tc:LongDescriptionType

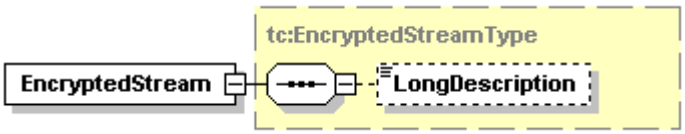
3.2.290 complexType tc:StreamType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	FixedFrameStream EncryptedStream VariableFrameStream ConvolutionalStream NonStandardData					
used by	element: SpaceSystemType/TelemetryAndCommanding/StreamList/Stream					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	streamSource	tc:StreamSourceType	required			
annotation	documentation	A container Type used to contain a single reference to all known Stream types.				

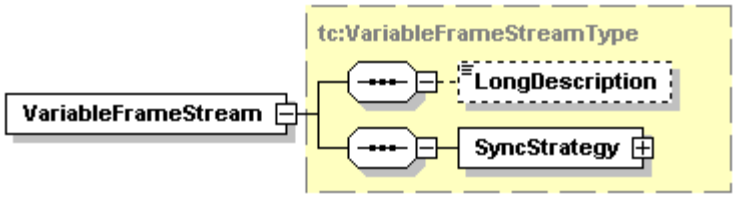
3.2.291 element tc:StreamType/FixedFrameStream

diagram						
type	tc:FixedFrameStreamType					
children	LongDescription SyncStrategy					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType				
	shortDescription	tc:ShortDescription-Type				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
	frameLengthInBits	long	required			

3.2.292 element tc:StreamType/EncryptedStream

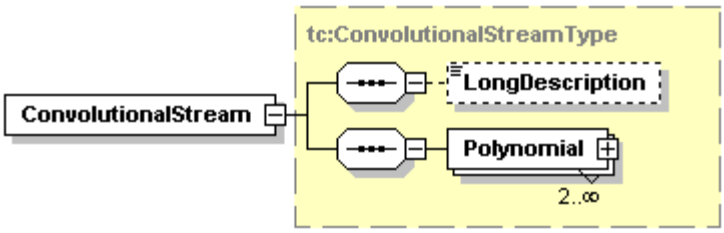
diagram						
type	tc:EncryptedStreamType					
children	LongDescription					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	name	tc:NameType				
	shortDescription	tc:ShortDescription-Type				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
	algorithmName	string	required			
	key	string				

3.2.293 element tc:StreamType/VariableFrameStream

diagram	
type	tc:VariableFrameStreamType
children	LongDescription SyncStrategy

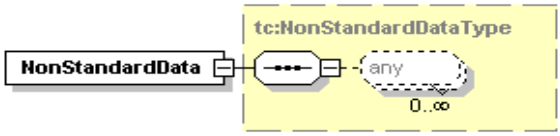
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	tc:NameType				
	shortDescription	tc:ShortDescriptionType				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
	zeroBitInsertion	boolean	optional	true		

3.2.294 element tc:StreamType/ConvolutionalStream

diagram	
	
type	tc:ConvolutionalStreamType
children	LongDescription Polynomial

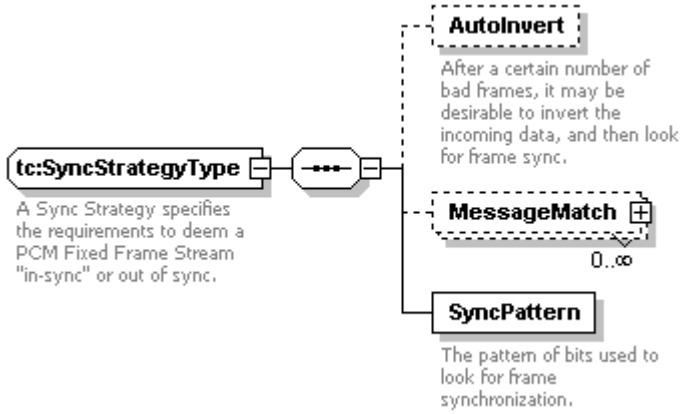
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	tc:NameType				
	shortDescription	tc:ShortDescription-Type				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		

3.2.295 element tc:StreamType/NonStandardData

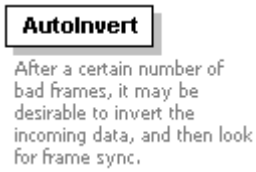
diagram	
	

type	tc:NonStandardDataType					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	usage	string	required			

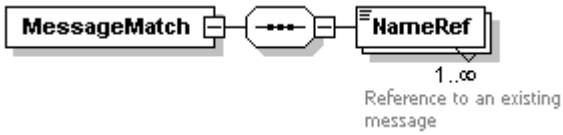
3.2.296 complexType tc:SyncStrategyType

diagram						
namespace	http://www.omg.org/space/tcspec					
children	AutoInvert MessageMatch SyncPattern					
used by	elements: tc:FixedFrameStreamType/SyncStrategy tc:VariableFrameStreamType/SyncStrategy					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	verifyToLockGoodFrames	integer		4		
	checkToLockGoodFrames	integer		1		
	maxErrorsInSyncWord	integer		0		
	syncAperature	integer		0		
annotation	documentation	A Sync Strategy specifies the requirements to deem a PCM Fixed Frame Stream "in-sync" or out of sync.				

3.2.297 element tc:SyncStrategyType/AutoInvert

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	badFramesToAutoInvert	positiveInteger		2		
	algorithmName	string				
annotation	documentation	After a certain number of bad frames, it may be desirable to invert the incoming data, and then look for frame sync.				

3.2.298 element tc:SyncStrategyType/MessageMatch

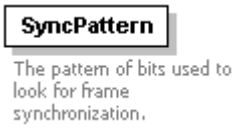
diagram						
children	NameRef					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	bitLocation	integer		0		

3.2.299 element tc:SyncStrategyType/MessageMatch/NameRef

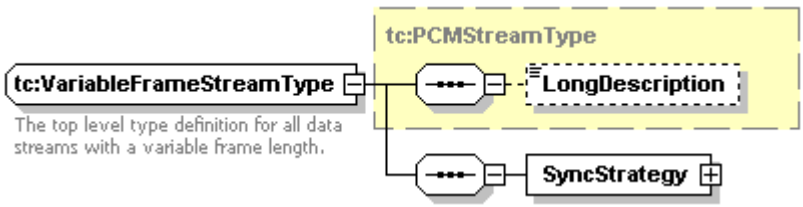
diagram	
type	tc:NameReferenceType

facets	pattern	a-zA-Z0-9_\- /]*
annotation	documentation	Reference to an existing message.

3.2.300 element tc:SyncStrategyType/SyncPattern

diagram						
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	pattern	hexBinary	required			
	bitLocation	integer		0		
	mask	hexBinary				
	maskLengthInBits	positiveInteger				
	patternLengthInBits	positiveInteger	required			
annotation	documentation	The pattern of bits used to look for frame synchronization..				

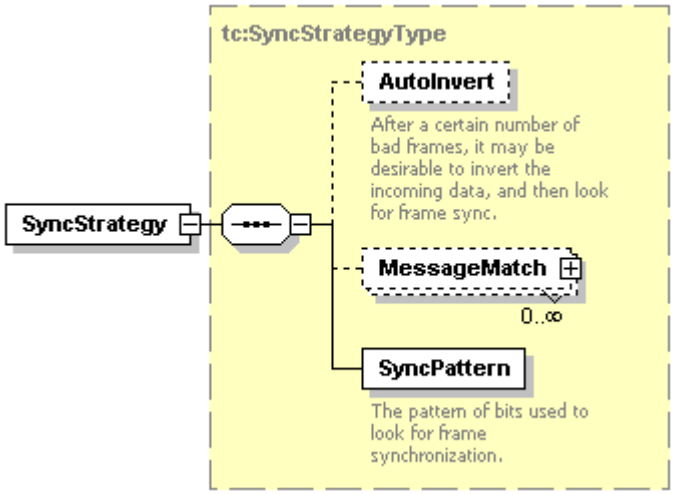
3.2.301 complexType tc:VariableFrameStreamType

diagram	
namespace	http://www.omg.org/space/tcspec
type	extension of tc:PCMStreamType
children	LongDescription SyncStrategy
used by	element: tc:StreamType/VariableFrameStream

attributes	Name	Type	Use	Default	Fixed	Annotation
	name	tc:NameType				
	shortDescription	tc:ShortDescriptionType				
	bitRateInBPS	double				
	pcmType	string		NRZL		
	inverted	boolean		false		
	zeroBitInsertion	boolean	optional	true		
annotation	documentation	The top level type definition for all data streams with a variable frame length.				

3.2.302 element tc:VariableFrameStreamType/SyncStrategy

diagram	



type	tc:SyncStrategyType
children	AutoInvert MessageMatch SyncPattern

attributes	Name	Type	Use	Default	Fixed	Annotation
	verifyToLockGoodFrames	integer		4		
	checkToLockGoodFrames	integer		1		
	maxErrorsInSyncWord	integer		0		
	syncAperature	integer		0		

3.2.303 simpleType tc:StreamSourceType

namespace	http://www.omg.org/space/tcspec	
type	restriction of **string**	
used by	attribute: tc:StreamType/@streamSource	
facets	enumeration	TC
	enumeration	TM

XML Schema A

Note – XML Schema documentation generated with **XML Spy** Schema Editor **www.xmlspy.com**

A.1 SpaceSystem.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Nori Jungmann (Engineering) -->
<!--

Style Notes, used througout all schemas:

    - Element and Type names begin with a capitol letter.
    - Type names end with the word "Type".
    - Attribute names begin with a lowercase letter.
    - Usually, when the UML class diagram references classes, W3C Elements are used, and whenever
      the UML references simple types (strings, ints), W3C Attributes are used.  In general,
      attributes are preferred over elements because they're easier to deal with
      in SAX and DOM, but whenever the Element/Attribute may one day carry metadata,
      elements should be used.  One exception, is enumerated classes, because enumerations
      may be defined for attributes but not for elements.
    - Bias toward self-describing names over short, bandwidth conserving ones.
    - Use mixed case in names rather than underscores to combine multiple words (camelCase).
    - A documentation annotation is included in every element and type definition.
    - Hints on units (for values with units) are provided in the names of attributes and
      elements (e.g. "dataRateInBPS" is preferred over "dataRate" OR "frameLengthInBits" is
      preferred over "frameLength").
    - Major elements or any elements used multiple times are first defined with a complexType definition
-->
<schema targetNamespace="http://www.omg.org/space/tcspec" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:tc="http://www.omg.org/space/tcspec" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <annotation>
        <documentation>$Id: SpaceSystem.xsd,v 1.12 2003/03/03 14:34:40 gerry Exp $</documentation>
        <documentation xml:lang="en">This is the master schema for the OMG Space Domain Task Force Telemetric and Command data format.</documentation>
    </annotation>
    <include schemaLocation="Algorithm.xsd"/>
    <include schemaLocation="Parameter.xsd"/>
    <include schemaLocation="CommonTypes.xsd"/>
```

```
<include schemaLocation="CommandDefinition.xsd"/>
<include schemaLocation="Packaging.xsd"/>
<include schemaLocation="StreamDefinitions.xsd"/>
<complexType name="SpaceSystemType">
    <annotation>
        <documentation>SpaceSystem is a collection of SpaceSystem(s) including space assets, ground assets, multi-satellite systems and sub-systems.  A SpaceSystem is the root element for the set of data necessary to monitor and command an arbitrary space device - this includes the binary decomposition the data streams going into and out of a device.  The major data component of a Space System is the TelemetryAndCommanding section where all Telemetric and Command data is stored.  Additionally, a Space System has a Header record, containing general information on the data, and 'Algorithms' containing certain well known algorithms and references to external algorithms for mission unique processing.</documentation>
    </annotation>
    <sequence>
        <element name="Header" type="tc:HeaderType" minOccurs="0"/>
        <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0"/>
        <element ref="tc:SpaceSystem" minOccurs="0" maxOccurs="unbounded"/>
        <element name="TelemetryAndCommanding" minOccurs="0">
            <annotation>
                <documentation xml:lang="en">TelemetryAndCommanding is the root element for the set of data necessary to monitor and command an arbitrary space device - this includes the binary decomposition the data streams going into and out of a device.  TelemetryAndCommanding is composed of the major sub-components 'Parameters' and 'CommandDefinitions'. Additionally, TelemetryAndCommanding has a Header record, containing general information on the data, and 'Algorithms' containing certain well known algorithms and references to external algorithms for mission unique processing.</documentation>
            </annotation>
            <complexType>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="DefaultBusAttributes" type="tc:BusAttributesType" minOccurs="0"/>
                    <element name="DefaultCommandDefinition" type="tc:CommandDefinitionType"/>
                    <element name="CommandDefinitionList" minOccurs="0">
                        <annotation>
                            <documentation xml:lang="en">Holds the list of CommandDefinitions</documentation>
                        </annotation>
                        <complexType>
                            <sequence>
                                <element name="ArgumentDefinition" type="tc:ArgumentType" minOccurs="0" maxOccurs="unbounded"/>
                                <element name="CommandDefinition" type="tc:CommandDefinitionType" maxOccurs="unbounded"/>
                            </sequence>
                        </complexType>
                        <key name="CommandDefinitionListKey">
                            <selector xpath="./CommandDefinition"/>
                            <field xpath="@name"/>
                        </key>
                    </element>
                    <element name="ParameterList" minOccurs="0">
                        <annotation>
                            <documentation xml:lang="en">Holds the list of parameter definitions</documentation>
                        </annotation>
                        <complexType>
                            <sequence>
                                <element name="Parameter" type="tc:ParameterType" maxOccurs="unbounded">
                                    <annotation>
                                        <documentation xml:lang="en">A single parameter definition</documentation>
                                    </annotation>
                                </element>
                            </sequence>
                        </complexType>
                        <key name="ParameterListKey">
                            <selector xpath="./Parameter"/>
                            <field xpath="@name"/>
                        </key>
                    </element>
                    <element ref="tc:Packaging" minOccurs="0">
                        <annotation>
                            <documentation>Optional description of message</documentation>
```

```
                    </annotation>
                </element>
                <element name="PropertyList">
                    <complexType>
                        <sequence>
                            <element name="Property" type="tc:PropertyType" maxOccurs="unbounded">
                                <annotation>
                                    <documentation xml:lang="en">Using these properties is discoraged, they are only meant as a placeholder for
date that does not otherwise fit into this schema.</documentation>
                                </annotation>
                            </element>
                        </sequence>
                    </complexType>
                </element>
                <element name="ServiceList">
                    <complexType>
                        <sequence>
                            <element name="Service" type="tc:ServiceType" maxOccurs="unbounded"/>
                        </sequence>
                    </complexType>
                    <key name="ServiceListKey">
                        <selector xpath="./Service"/>
                        <field xpath="@name"/>
                    </key>
                </element>
                <element name="AlgorithmList">
                    <complexType>
                        <choice maxOccurs="unbounded">
                            <element name="CustomAlgorithm" type="tc:CustomAlgorithmType"/>
                            <element name="ConcatenationAlgorithm" type="tc:ConcatenationAlgorithmType"/>
                            <element name="MathAlgorithm" type="tc:MathAlgorithmType"/>
                            <element name="ReedSolomonEncoder" type="tc:ReedSolomonType"/>
                            <element name="ReedSolomonDecoder" type="tc:ReedSolomonType"/>
                        </choice>
                    </complexType>
                </element>
                <element name="StreamList">
                    <complexType>
                        <sequence>
                            <element name="Stream" type="tc:StreamType" maxOccurs="unbounded"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="NonStandardData" type="tc:NonStandardDataType" minOccurs="0"/>
            </choice>
        </complexType>
    </element>
    <element name="NonStandardData" type="tc:NonStandardDataType" minOccurs="0"/>
</sequence>
<attribute name="name" type="string"/>
<attribute name="shortDescription" type="tc:ShortDescriptionType"/>
</complexType>
<!-- SpaceSystem -->
<element name="SpaceSystem" type="tc:SpaceSystemType" nillable="true"/>
</schema>
```

A.2 Parameter.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 2 U (http://www.xmlspy.com) by Ed Shaya (NASA) -->
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Nori Jungmann (Engineering) -->
```

```
<schema targetNamespace="http://www.omg.org/space/tcspec" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:tc="http://www.omg.org/space/tcspec">
    <annotation>
        <documentation>$Id: Parameter.xsd,v 1.18 2003/03/03 14:34:40 gerry Exp $</documentation>
        <documentation xml:lang="en">This schema defines the structure for a Parameter.  A Parameter is an object which can assume a value (parameter instance).</documentation>
    </annotation>
    <include schemaLocation="CommonTypes.xsd"/>
    <complexType name="ParameterType">
        <annotation>
            <documentation xml:lang="en">A type definition used by Parameter</documentation>
        </annotation>
        <complexContent>
            <extension base="tc:BaseParameterType">
                <sequence>
                    <element name="Alias" type="tc:AliasType" minOccurs="0" maxOccurs="unbounded"/>
                    <element name="SystemName" type="tc:SystemNameType" minOccurs="0">
                        <annotation>
                            <documentation>Optional.  Normally used when the database is built in a flat, non-hierarchical format</documentation>
                        </annotation>
                    </element>
                    <element name="ToString" type="tc:ToStringType" minOccurs="0">
                        <annotation>
                            <documentation>Includes instructions on how to represent the parameter as a string.</documentation>
                        </annotation>
                    </element>
                    <element name="Validity" type="tc:ComparisonCheckType" minOccurs="0"/>
                    <choice>
                        <annotation>
                            <documentation>The database builder is allowed to specify alarms as either simple critical/waring style alarms or (almost) arbitrarily complex alarms using alarm conditions</documentation>
                        </annotation>
                        <element name="AlarmConditions" type="tc:ComplexAlarmsType" minOccurs="0" maxOccurs="unbounded"/>
                        <element name="Alarms" type="tc:SimpleAlarmsType"/>
                    </choice>
                    <element name="PhysicalAddress" minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <documentation  xml:lang="en">Contains the address (channel information) required to process the spacecraft telemetry streams</documentation>
                            <documentation  xml:lang="en">Contains the address (channel information) required to process the spacecraft telemetry streams</documentation>
                        </annotation>
                        <complexType>
                            <complexContent>
                                <extension base="tc:PhysicalAddressType"/>
                            </complexContent>
                        </complexType>
                    </element>
                    <element name="NonStandardData" type="tc:NonStandardDataType" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
                <attribute name="dataSource" use="optional">
                    <simpleType>
                        <restriction base="string">
                            <enumeration value="telemetered"/>
                            <enumeration value="derived"/>
                            <enumeration value="constant"/>
                            <enumeration value="local"/>
                        </restriction>
                    </simpleType>
                </attribute>
                <attribute name="readOnly" type="boolean" use="optional" default="false"/>
                <attribute name="validRangeAppliesToCalibrated" type="boolean" use="optional" default="true"/>
                <attribute name="onboardID" type="string" use="optional"/>
```

```
            </extension>
        </complexContent>
    </complexType>
    <!-- ParameterType -->
    <complexType name="ParameterRefMatchListType">
        <annotation>
            <documentation xml:lang="en">Holds a list of parameter matches, and whether all or one match required</documentation>
        </annotation>
        <sequence>
            <element name="ParameterMatch" type="tc:ComparisonCheckType" maxOccurs="unbounded">
                <annotation>
                    <documentation xml:lang="en">Condition that must be evaluated</documentation>
                </annotation>
            </element>
        </sequence>
        <attribute name="AllMustBeTrue" type="boolean">
            <annotation>
                <documentation xml:lang="en">Must all matched be true? If not present then all must be true</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="ParameterRefType">
        <annotation>
            <documentation xml:lang="en">A reference to a Parameter that includes a validity check for the parameters</documentation>
        </annotation>
        <complexContent>
            <extension base="tc:SimpleParameterRefType">
                <sequence>
                    <element name="Occurs" type="tc:OccursType" minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <documentation xml:lang="en">Repeating group to include</documentation>
                        </annotation>
                    </element>
                    <element name="DependantOn" type="tc:ParameterRefMatchListType" minOccurs="0">
                        <annotation>
                            <documentation xml:lang="en">Optional parameter set condition that must be true for this parameter to be valid</documentation>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="OffsetParameterRefType">
        <annotation>
            <documentation xml:lang="en">Holds an offset to a refered to parameter and optional comparisons to determine validity of this parameter</documentation>
        </annotation>
        <complexContent>
            <extension base="tc:ParameterRefType">
                <sequence>
                    <element name="OffsetInBits" type="tc:BitOffsetType" minOccurs="0">
                        <annotation>
                            <documentation xml:lang="en">Offset in bits from the start of the container scope, the first bit in the container has an offset of 0</documentation>
                        </annotation>
                    </element>
                </sequence>
                <attribute name="TimeOffset" type="integer" use="optional" default="0">
                    <annotation>
                        <documentation xml:lang="en">Time delay in milliseconds</documentation>
                    </annotation>
                </attribute>
```

```
            </extension>
        </complexContent>
    </complexType>
    <complexType name="ComplexAlarmsType">
        <annotation>
            <documentation>Alarms are used to notify the operator when a parameter value is outside of acceptable ranges or states.  Dynamic or rate-of-change Limit Ranges could be considered.</documentation>
        </annotation>
        <choice maxOccurs="unbounded">
            <element name="StaticRanges" type="tc:RangeListType">
                <annotation>
                    <documentation xml:lang="en">A StaticRange has an optional UseRangeWhen condidition and one or more AlarmRanges.</documentation>
                </annotation>
            </element>
            <element name="RateOfChangeRanges">
                <annotation>
                    <documentation>Like a Static Range, but it is based on the rate of change for the Parameter.</documentation>
                </annotation>
                <complexType>
                    <complexContent>
                        <extension base="tc:RangeListType">
                            <attribute name="perTimeBase" use="optional" default="second">
                                <annotation>
                                    <documentation xml:lang="en">Rate of change alarms must be reference to some relative time base e.g. 3.4 volts per second</documentation>
                                </annotation>
                                <simpleType>
                                    <restriction base="string">
                                        <enumeration value="microsecond"/>
                                        <enumeration value="millisecond"/>
                                        <enumeration value="second"/>
                                    </restriction>
                                </simpleType>
                            </attribute>
                        </extension>
                    </complexContent>
                </complexType>
            </element>
            <element name="States">
                <annotation>
                    <documentation>Contains multiple (un)acceptable values for multi-state parameters.</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="UseRangeWhen" minOccurs="0">
                            <annotation>
                                <documentation>Can be used to impliment conditional limits</documentation>
                            </annotation>
                            <complexType>
                                <choice>
                                    <element name="EqualityCheck" type="tc:ComparisonCheckType"/>
                                    <element name="CustomCheck"/>
                                </choice>
                            </complexType>
                        </element>
                        <element name="State" maxOccurs="unbounded">
                            <annotation>
                                <documentation xml:lang="en">(Un)acceptable values for multi-state parameters.</documentation>
                            </annotation>
                            <complexType>
                                <attribute name="statePriority" type="nonNegativeInteger" use="optional"/>
```

```
                                    <attribute name="type" type="tc:LimitCheckType" use="required"/>
                                    <attribute name="state" type="nonNegativeInteger" use="required"/>
                                </complexType>
                            </element>
                        </sequence>
                        <attribute name="limitSetPriority" type="nonNegativeInteger" use="optional"/>
                    </complexType>
                </element>
                <element name="CustomAlarmCheck" type="tc:CustomAlgorithmType">
                    <annotation>
                        <documentation>An external algorithm that can be arbitrarily complex.  E.G. an alarm that occurs when the parameter's FFT shows a reso-
nance at 20Hz.  </documentation>
                    </annotation>
                </element>
            </choice>
            <attribute name="minViolations" use="required">
                <annotation>
                    <documentation xml:lang="en">Doesn't get reported until it has been out of range for this value times</documentation>
                </annotation>
                <simpleType>
                    <restriction base="integer">
                        <minInclusive value="0"/>
                        <maxInclusive value="99"/>
                    </restriction>
                </simpleType>
            </attribute>
            <attribute name="limitsApplyToCalibrated" type="boolean" default="true"/>
        </complexType>
        <!-- LimitSet -->
        <element name="DwellSet">
            <annotation>
                <documentation xml:lang="en">Dwelling parameters</documentation>
            </annotation>
            <complexType>
                <attribute name="onboardID" type="string" use="required"/>
                <attribute name="address" type="string" use="required"/>
                <attribute name="startBit" type="nonNegativeInteger" use="required"/>
                <attribute name="length" type="nonNegativeInteger" use="required"/>
                <attribute name="ParameterName" type="tc:NameReferenceType" use="required"/>
                <attribute name="alternateParameterName" type="tc:NameReferenceType" use="optional"/>
            </complexType>
        </element>
        <complexType name="VariableParameterType">
            <annotation>
                <documentation>A special parameter that may have a variable width</documentation>
            </annotation>
            <complexContent>
                <extension base="tc:ParameterType">
                    <choice>
                        <element name="WidthRef" type="tc:ParameterRefType"/>
                        <element name="stop" type="tc:StopType"/>
                    </choice>
                </extension>
            </complexContent>
        </complexType>
        <complexType name="StopType">
            <annotation>
                <documentation>Specifies either a width in bits or a termination for variable width parameters</documentation>
            </annotation>
            <attribute name="representation" type="hexBinary"/>
            <attribute name="widthInBits" type="positiveInteger"/>
        </complexType>
```

```
    <complexType name="SimpleAlarmsType">
        <annotation>
            <documentation>A simple warning, critical alarm type</documentation>
        </annotation>
        <sequence>
            <element name="CriticalRange" type="tc:RangeType" minOccurs="0"/>
            <element name="WarningRange" type="tc:RangeType" minOccurs="0"/>
        </sequence>
    </complexType>
    <complexType name="RangeListType">
        <annotation>
            <documentation>Contains multible ranges</documentation>
        </annotation>
        <sequence>
            <element name="UseRangeWhen" minOccurs="0">
                <annotation>
                    <documentation>Can be used to impliment conditional limits</documentation>
                </annotation>
                <complexType>
                    <choice>
                        <element name="EqualityCheck" type="tc:ComparisonCheckType"/>
                        <element name="CustomCheck"/>
                    </choice>
                </complexType>
            </element>
            <element name="AlarmRange" maxOccurs="unbounded">
                <complexType>
                    <complexContent>
                        <extension base="tc:RangeType">
                            <attribute name="severity" type="nonNegativeInteger" use="required"/>
                            <attribute name="type" type="tc:LimitCheckType" use="required"/>
                            <attribute name="rangeName" type="string" use="optional"/>
                        </extension>
                    </complexContent>
                </complexType>
            </element>
        </sequence>
        <attribute name="limitSetPriority" type="nonNegativeInteger" use="optional"/>
    </complexType>
    <complexType name="PhysicalAddressType">
        <annotation>
            <documentation>When it's important to know the physical address(s) on the spacecraft that this parameter may be collected from, use this.  </documentation>
        </annotation>
        <sequence>
            <element name="SubAddress" type="tc:PhysicalAddressType" minOccurs="0"/>
        </sequence>
        <attribute name="sourceName" type="string"/>
        <attribute name="sourceAddress" type="string"/>
    </complexType>
</schema>
```

A.3 CommandDefinition.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Nori Jungmann (Engineering) -->
<!--

$Id: CommandDefinition.xsd,v 1.12 2003/03/03 14:34:40 gerry Exp $
```

```
-->
<schema targetNamespace="http://www.omg.org/space/tcspec" xmlns:tc="http://www.omg.org/space/tcspec" xmlns="http://www.w3.org/2001/XMLSchema">
    <include schemaLocation="CommonTypes.xsd"/>
    <include schemaLocation="Algorithm.xsd"/>
    <include schemaLocation="Parameter.xsd"/>
    <include schemaLocation="Packaging.xsd"/>
    <annotation>
        <documentation xml:lang="en">This schema defines the structure for a CommandDefinition.  A CommandDefinition provides a description of the command, a name for the command, the allowable arguments for the command, and (optionally) for BusCommands (commands destined for an off-platform bus) the opcode for the command and argument codes.</documentation>
    </annotation>
    <complexType name="ArgumentType">
        <annotation>
            <documentation>A type definition used as the base type for a CommandDefinition and for BusCommandDefinition.</documentation>
        </annotation>
        <complexContent>
            <extension base="tc:BaseParameterType">
                <sequence>
                    <element name="Argument" type="tc:ArgumentType" minOccurs="0"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <!--CommandDefinitionType -->
    <complexType name="CommandDefinitionType">
        <annotation>
            <documentation xml:lang="en">A type definition used as the base type for a CommandDefinition</documentation>
        </annotation>
        <sequence>
            <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0"/>
            <element name="Alias" type="tc:AliasType" minOccurs="0" maxOccurs="unbounded"/>
            <element name="SystemName" type="tc:SystemNameType" minOccurs="0"/>
            <element name="Interlock" minOccurs="0">
                <annotation>
                    <documentation>Specifies the interlock for this command, including the scope of the interlock. An interlock restricts commanding ability at the specified scope until this command has reached the specified stage.</documentation>
                </annotation>
                <complexType>
                    <attribute name="scope" type="tc:InterlockScopeType" use="required"/>
                    <attribute name="stage" type="tc:StageType" use="required"/>
                </complexType>
            </element>
            <element name="Field" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                    <sequence>
                        <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0"/>
                        <choice>
                            <element name="Argument" type="tc:ArgumentType"/>
                            <element name="ArgumentRef" type="tc:ParameterRefType"/>
                            <element name="ParameterRef" type="tc:ParameterRefType"/>
                            <element name="BinaryConstant">
                                <annotation>
                                    <documentation>May also use a reference to a constant parameter</documentation>
                                </annotation>
                                <complexType>
                                    <simpleContent>
                                        <extension base="hexBinary">
                                            <attribute name="lengthInBits" type="nonNegativeInteger"/>
                                        </extension>
                                    </simpleContent>
                                </complexType>
```

```
                                </element>
                            </choice>
                        </sequence>
                        <attribute name="name" type="string"/>
                        <attribute name="editable" type="boolean" default="true"/>
                    </complexType>
                </element>
                <element name="BinaryTransform" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation xml:lang="en">The binary transform tells the system how to construct the command (transform it to binary format)</documentation>
                    </annotation>
                    <complexType>
                        <choice maxOccurs="unbounded">
                            <element name="FieldBitPosition">
                                <complexType>
                                    <choice>
                                        <element name="BitPositionFromStart" type="nonNegativeInteger"/>
                                        <element name="BitPositionFromPreviousArgument" type="integer"/>
                                    </choice>
                                    <attribute name="argumentName" type="string"/>
                                </complexType>
                            </element>
                            <element name="ContainerRef" type="tc:ContainerRefType"/>
                        </choice>
                    </complexType>
                </element>
                <element name="CommandProcessor" type="tc:CustomAlgorithmType" minOccurs="0">
                    <annotation>
                        <documentation xml:lang="en">A command processor is the name of some external algorithm that is used to format the command.  Command processors may also perform additional formatting of the command - formatting that cannot be adequately described in the database.</documentation>
                    </annotation>
                </element>
                <element name="CommandValidator" type="tc:CustomAlgorithmType" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation xml:lang="en">A command validator is used to validate that a particular user has the proper permissions to issue the command.  Command Validators - like command processors -are the name of some external algorithm that does the command validation.</documentation>
                    </annotation>
                </element>
                <element name="CommandConstraint" type="tc:CommandConstraintType" minOccurs="0"/>
                <element name="CommandVerifier" type="tc:CommandVerifierType" minOccurs="0" maxOccurs="unbounded"/>
                <element name="ParametersToSet" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation>Sets the values of parameters after the command has been verified</documentation>
                    </annotation>
                    <complexType>
                        <sequence>
                            <element name="ParameterRef" type="tc:ParameterRefType"/>
                            <choice>
                                <element name="Value" type="string"/>
                                <element name="ArgumentRef" type="string"/>
                            </choice>
                        </sequence>
                    </complexType>
                </element>
                <element name="NonStandardData" type="tc:NonStandardDataType" minOccurs="0"/>
            </sequence>
            <attribute name="name" type="tc:NameType" use="required"/>
            <attribute name="shortDescription" type="tc:ShortDescriptionType" use="optional"/>
            <attribute name="critical" type="boolean" use="optional" default="false"/>
            <attribute name="highPriority" type="boolean" use="optional" default="false"/>
        </complexType>
```

```
<complexType name="CommandConstraintType">
    <annotation>
        <documentation>A command constraint is used to check that the command can be run in the current device operating mode or envirnment. Command constraints are the name of some external algorithm that does the command constraint check or a list of simple comparison checks.</documentation>
    </annotation>
    <choice maxOccurs="unbounded">
        <element name="CustomAlgorithm" type="tc:CustomAlgorithmType"/>
        <element name="Comparison" type="tc:ComparisonCheckType"/>
    </choice>
</complexType>
<complexType name="CommandVerifierType">
    <annotation>
        <documentation>A command verifier is used to check that the command has be sucessfully executed. Command Verifiers are the name of some external algorithm that does the command verification. </documentation>
    </annotation>
    <sequence>
        <choice>
            <element name="CustomAlgorithm" type="tc:CustomAlgorithmType"/>
            <element name="Comparison" type="tc:ComparisonCheckType" maxOccurs="unbounded"/>
            <element name="ContainerRef" type="tc:ContainerRefType" maxOccurs="unbounded"/>
        </choice>
        <element name="Window">
            <annotation>
                <documentation>Specifies how much of a chance to provide for the verification.  The window may be specified in time or in number of samples of the comparison parameter to wait for a change.</documentation>
            </annotation>
            <complexType>
                <choice>
                    <sequence>
                        <element name="StartTime" type="nonNegativeInteger"/>
                        <element name="Interval" type="positiveInteger"/>
                    </sequence>
                    <element name="ChangeCount" type="positiveInteger"/>
                </choice>
            </complexType>
        </element>
    </sequence>
    <attribute name="verificationStage" type="tc:StageType" default="Completion"/>
</complexType>
<simpleType name="InterlockScopeType">
    <annotation>
        <documentation xml:lang="en">Specifies the scope of the interlock</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="SystemGlobal"/>
        <enumeration value="SystemLocal"/>
        <enumeration value="SubsystemGlobal"/>
        <enumeration value="SubsystemLocal"/>
    </restriction>
</simpleType>
<simpleType name="StageType">
    <annotation>
        <documentation xml:lang="en">Specifies the stage of command uplink</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="Reception"/>
        <enumeration value="Uplink"/>
        <enumeration value="Transfer"/>
        <enumeration value="Acceptance"/>
        <enumeration value="Execution"/>
        <enumeration value="Completion"/>
    </restriction>
```

```
    </simpleType>
</schema>
```

A.4 Algorithm.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 2 U (http://www.xmlspy.com) by Ed Shaya (NASA) -->
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Nori Jungmann (Engineering) -->
<schema targetNamespace="http://www.omg.org/space/tcspec" xmlns:tc="http://www.omg.org/space/tcspec" xmlns="http://www.w3.org/2001/XMLSchema">
    <include schemaLocation="CommonTypes.xsd"/>
    <include schemaLocation="StreamDefinitions.xsd"/>
    <include schemaLocation="Parameter.xsd"/>
    <annotation>
        <documentation>$Id: Algorithm.xsd,v 1.13 2003/03/03 07:51:22 cooper Exp $</documentation>
        <documentation xml:lang="en">This schema defines the structure for an Algorithm.  An Algorithm may be one of a growing set of pre-defined algorithms or a named escape into a user defined algorithm where (depending on the system) the name of the algorithm may be a java class, a function in a shared library, an external program or some other reference to an outside algorithm.

At some later date, this schema may also allow the logic of the user defined algorithm to be defined within the instance document itself (perhaps using MathML?).</documentation>
    </annotation>
    <complexType name="CustomAlgorithmType">
        <annotation>
            <documentation xml:lang="en">The type definition used by a the Custom Algorithm element.</documentation>
            <documentation xml:lang="en">Complete algorithm written in the algorithm language.  Embedded new line characters are legal within this attribute.

All algorithms should contain a change log that is modified with each update.</documentation>
            <documentation xml:lang="en">A hint to the ground system to tell it to calculate the algorithm in a new processing thread.</documentation>
        </annotation>
        <sequence>
            <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0"/>
            <element name="Trigger" type="tc:TriggerType" minOccurs="0"/>
            <element name="Inputs" minOccurs="0">
                <complexType>
                    <choice maxOccurs="unbounded">
                        <element name="ParameterRef">
                            <annotation>
                                <documentation>Names an input parameter to the algorithm.  There are two attributes to InputParm, inputName and parameterName. parameterName is a parameter reference name for a parameter that will be used in this algorithm.  inputName is an optional "friendly" name for the input parameter.</documentation>
                            </annotation>
                            <complexType>
                                <complexContent>
                                    <extension base="tc:ParameterRefType">
                                        <attribute name="inputName" type="string"/>
                                    </extension>
                                </complexContent>
                            </complexType>
                        </element>
                        <element name="Constant" minOccurs="0">
                            <annotation>
                                <documentation xml:lang="en">Names and provides a value for a constant input to the algorithm.  There are two attributes to Constant, constantName and value.  constantName is a variable name in the algorithm to be executed.  value is the value of the constant to be used.</documentation>
                            </annotation>
                            <complexType>
                                <attribute name="constantName" type="string"/>
                                <attribute name="value" type="string" use="required"/>
```

```
                                </complexType>
                            </element>
                        </choice>
                    </complexType>
                </element>
                <element name="OutputParameterRef" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation>Names an output parameter to the algorithm.  There are two attributes to OutputParm, outputName and parameterName. parameterName is a parameter reference name for a parameter that will be updated by this algorithm.  outputName is an optional "friendly" name for the output parameter.</documentation>
                    </annotation>
                    <complexType>
                        <complexContent>
                            <extension base="tc:ParameterRefType">
                                <attribute name="outputName" type="string"/>
                            </extension>
                        </complexContent>
                    </complexType>
                </element>
                <element name="AlgorithmText" minOccurs="0">
                    <annotation>
                        <documentation xml:lang="en">This optional element may be used to enter Pseudo or actual code for the algorithm.  The language for the algorithm is specified with the language attribute</documentation>
                    </annotation>
                    <complexType>
                        <simpleContent>
                            <extension base="string">
                                <attribute name="language" type="string" default="pseudo"/>
                            </extension>
                        </simpleContent>
                    </complexType>
                </element>
                <element name="ExternalAlgorithm" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation>This is the external algorithm.  Multiple entries are provided so that the same database may be used for multiple implementations</documentation>
                    </annotation>
                    <complexType>
                        <attribute name="implimentationName" type="string"/>
                        <attribute name="algorithmLocation" type="string" use="required"/>
                    </complexType>
                </element>
            </sequence>
            <attribute name="name" type="string" use="required"/>
            <attribute name="shortDescription" type="tc:ShortDescriptionType" use="optional"/>
            <attribute name="thread" type="boolean" use="optional"/>
            <attribute name="triggerContainer" type="string" use="optional">
                <annotation>
                    <documentation xml:lang="en">First telemetry container from which the output parameter should be calculated.</documentation>
                </annotation>
            </attribute>
            <attribute name="priority" type="integer" use="optional">
                <annotation>
                    <documentation xml:lang="en">Algorithm processing priority.</documentation>
                </annotation>
            </attribute>
        </complexType>
        <complexType name="CalibratorType">
            <annotation>
                <documentation xml:lang="en">Calibrators are normally used to convert to and from bit compacted numerical data</documentation>
            </annotation>
            <choice>
```

```
            <element name="SplineCalibrator">
                <annotation>
                    <documentation xml:lang="en">A calibration type where a segmented line in a raw vs calibrated plane is described using a set of points. Raw values are converted to calibrated values by finding a position on the line coorosponding to the raw value. The algorithm triggers on the input parameter.</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="SplinePoint" type="tc:SplinePointType" minOccurs="2" maxOccurs="unbounded"/>
                    </sequence>
                    <attribute name="order" type="positiveInteger" default="1"/>
                    <attribute name="extrapolate" type="boolean" default="false"/>
                </complexType>
            </element>
            <element name="PolynomialCalibrator" type="tc:PolynomialType">
                <annotation>
                    <documentation>A calibration type where a curved in a raw vs calibrated plane is described using a set of polynomial coefficients.  Raw values are converted to calibrated values by finding a position on the curve corresponding to the raw value. The first coefficient belongs with the X^0 term, the next coefficient belongs to the X^1 term and so on. The algorithm triggers on the condition parameter.  </documentation>
                </annotation>
            </element>
            <element name="ToStringCalibrator" type="tc:ToStringType">
                <annotation>
                    <documentation>A calibration type where the bus value is converted to a string value.</documentation>
                </annotation>
            </element>
        </choice>
        <attribute name="name" type="string" use="optional"/>
        <attribute name="shortDescription" type="tc:ShortDescriptionType" use="optional"/>
    </complexType>
    <element name="FixedFrameSync">
        <annotation>
            <documentation xml:lang="en">An algorithm most often implemented in hardware, that finds the frame synchronization  mark in each frame.  Given raw binary data, a PCM Fixed Frame Sync will output framed data.  Triggers on the input parameter</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="InputParameterRef" type="tc:ParameterRefType"/>
                <element name="OutputParameterRef" type="tc:ParameterRefType"/>
                <element name="FixedFrameStream" type="tc:FixedFrameStreamType"/>
            </sequence>
        </complexType>
    </element>
    <complexType name="ConcatenationAlgorithmType">
        <annotation>
            <documentation>An algorithm that will concatenate two or more parameters.  Although not checked by the schema, input parameters must be binary or boolean and the resulting output parameter must be sized to contain the concatenation.</documentation>
        </annotation>
        <sequence>
            <element name="InputParameterList">
                <complexType>
                    <sequence>
                        <element name="InputParameterRef" type="tc:ParameterRefType" maxOccurs="unbounded"/>
                    </sequence>
                </complexType>
            </element>
            <element name="OutputParameterRef" type="tc:ParameterRefType"/>
        </sequence>
    </complexType>
    <complexType name="MathAlgorithmType">
        <annotation>
            <documentation>A simple mathematical operation</documentation>
```

```
        </annotation>
        <complexContent>
            <extension base="tc:MathOperationType">
                <sequence>
                    <element name="OutputParameterRef" type="tc:ParameterRefType"/>
                    <element name="Trigger" type="tc:TriggerType"/>
                </sequence>
                <attribute name="trigger" type="tc:NameReferenceType" use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="ReedSolomonType">
        <annotation>
            <documentation>Abstract type used by  both ReedSolomonEncoder and ReedSolomonDecoder</documentation>
        </annotation>
        <sequence>
            <element name="InputParameterRef" type="tc:ParameterRefType"/>
            <element name="OutputParameterRef" type="tc:ParameterRefType"/>
            <element name="GeneratorPolynomial" type="tc:PolynomialType"/>
        </sequence>
        <attribute name="inputParameterRef" type="tc:NameReferenceType" use="required"/>
        <attribute name="outputParameterRef" type="tc:NameReferenceType" use="required"/>
    </complexType>
    <complexType name="TriggerType">
        <annotation>
            <documentation>A trigger is used to initiate the processing of some algorithm.  A trigger may be based on an update of a Parameter or on a time basis.  Triggers may also have a rate that limits their firing to a 1/rate basis.</documentation>
        </annotation>
        <choice maxOccurs="unbounded">
            <element name="ParameterRef" minOccurs="0">
                <annotation>
                    <documentation>Names a parameter that will start the execution of the algorithm.  There are two attributes to TriggerParm, triggerName and parameterName.  parameterName is a parameter reference name for a parameter that when it changes, will cause this algorithm to be executed.  triggerName is an optional "friendly" name for the trigger.</documentation>
                </annotation>
                <complexType>
                    <complexContent>
                        <extension base="tc:ParameterRefType"/>
                    </complexContent>
                </complexType>
            </element>
            <element name="Time" type="tc:RelativeTimeType" minOccurs="0"/>
        </choice>
        <attribute name="name" type="string" use="optional"/>
        <attribute name="triggerRate" type="nonNegativeInteger" use="optional" default="1"/>
    </complexType>
</schema>
```

A.5 StreamDefinitions.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Nori Jungmann (Engineering) -->
<schema targetNamespace="http://www.omg.org/space/tcspec" xmlns:tc="http://www.omg.org/space/tcspec" xmlns="http://www.w3.org/2001/XMLSchema">
    <!-- unqualified element and attribute names -->
    <annotation>
        <documentation>$Id: StreamDefinitions.xsd,v 1.11 2003/03/03 14:34:40 gerry Exp $</documentation>
        <documentation xml:lang="en">
            This schema provides a language for defining binary stream data.
        </documentation>
```

```
    </annotation>
    <include schemaLocation="CommonTypes.xsd"/>
    <simpleType name="StreamSourceType">
        <restriction base="string">
            <enumeration value="TC"/>
            <enumeration value="TM"/>
        </restriction>
    </simpleType>
    <complexType name="PCMStreamType">
        <annotation>
            <documentation xml:lang="en">A PCM Stream Type is the high level definition for all Pulse Code Modulated (PCM) (i.e., binary) streams.</documen-
tation>
        </annotation>
        <sequence>
            <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0"/>
        </sequence>
        <attribute name="name" type="tc:NameType"/>
        <attribute name="shortDescription" type="tc:ShortDescriptionType"/>
        <attribute name="bitRateInBPS" type="double"/>
        <attribute name="pcmType" default="NRZL">
            <simpleType>
                <restriction base="string">
                    <enumeration value="NRZL"/>
                    <enumeration value="NRZM"/>
                    <enumeration value="NRZS"/>
                    <enumeration value="BiPhaseL"/>
                    <enumeration value="BiPhaseM"/>
                    <enumeration value="BiPhaseS"/>
                </restriction>
            </simpleType>
        </attribute>
        <attribute name="inverted" type="boolean" default="false"/>
    </complexType>
    <complexType name="FixedFrameStreamType">
        <annotation>
            <documentation xml:lang="en">The top level type definition for all data streams with a fixed frame length.</documentation>
        </annotation>
        <complexContent>
            <extension base="tc:PCMStreamType">
                <sequence>
                    <element name="SyncStrategy" type="tc:SyncStrategyType"/>
                </sequence>
                <attribute name="frameLengthInBits" type="long" use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="VariableFrameStreamType">
        <annotation>
            <documentation xml:lang="en">The top level type definition for all data streams with a variable frame length. </documentation>
        </annotation>
        <complexContent>
            <extension base="tc:PCMStreamType">
                <sequence>
                    <element name="SyncStrategy" type="tc:SyncStrategyType"/>
                </sequence>
                <attribute name="zeroBitInsertion" type="boolean" use="optional" default="true"/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="EncryptedStreamType">
        <annotation>
            <documentation xml:lang="en">The top level type definition for all encrypted streams.</documentation>
```

```
        </annotation>
        <complexContent>
            <extension base="tc:PCMStreamType">
                <attribute name="algorithmName" type="string" use="required"/>
                <attribute name="key" type="string"/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="ConvolutionalStreamType">
        <annotation>
            <documentation xml:lang="en">The type definition for convolutionally encoded streams.  The processing
                algorithm for the convolutional encoder is described using a series of
                generator polynomials.  Each polynomial, corrosponds to a modulo two adder
                and the order of the polynomials corrosponds to the order in the output stage.
                Each term in each polynomial corrosponds to a register in the shift register.
                The coefficients in these polynomials are either "1" for connected or "0" for not connected.</documentation>
        </annotation>
        <complexContent>
            <extension base="tc:PCMStreamType">
                <sequence>
                    <element name="Polynomial" type="tc:PolynomialType" minOccurs="2" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="StreamType">
        <annotation>
            <documentation xml:lang="en">A container Type used to contain a single reference to all known Stream types</documentation>
        </annotation>
        <choice maxOccurs="unbounded">
            <element name="FixedFrameStream" type="tc:FixedFrameStreamType"/>
            <element name="EncryptedStream" type="tc:EncryptedStreamType"/>
            <element name="VariableFrameStream" type="tc:VariableFrameStreamType"/>
            <element name="ConvolutionalStream" type="tc:ConvolutionalStreamType"/>
            <element name="NonStandardData" type="tc:NonStandardDataType"/>
        </choice>
        <attribute name="streamSource" type="tc:StreamSourceType" use="required"/>
    </complexType>
    <complexType name="SyncStrategyType">
        <annotation>
            <documentation>A Sync Strategy specifies the requirements to deem a PCM Fixed Frame Stream "in-sync" or out of sync.</documentation>
        </annotation>
        <sequence>
            <element name="AutoInvert" minOccurs="0">
                <annotation>
                    <documentation xml:lang="en">After a certain number of bad frames, it may be desirable to invert the incoming data, and then look for
frame sync.</documentation>
                </annotation>
                <complexType>
                    <attribute name="badFramesToAutoInvert" type="positiveInteger" default="2"/>
                    <attribute name="algorithmName" type="string"/>
                </complexType>
            </element>
            <element name="MessageMatch" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                    <sequence>
                        <element name="NameRef" type="tc:NameReferenceType" maxOccurs="unbounded">
                            <annotation>
                                <documentation xml:lang="en">Reference to an existing message</documentation>
                            </annotation>
                        </element>
                    </sequence>
```

```
                    <attribute name="bitLocation" type="integer" default="0"/>
                </complexType>
            </element>
            <element name="SyncPattern">
                <annotation>
                    <documentation xml:lang="en">The pattern of bits used to look for frame synchronization.</documentation>
                </annotation>
                <complexType>
                    <attribute name="pattern" type="hexBinary" use="required"/>
                    <attribute name="bitLocation" type="integer" default="0"/>
                    <attribute name="mask" type="hexBinary"/>
                    <attribute name="maskLengthInBits" type="positiveInteger"/>
                    <attribute name="patternLengthInBits" type="positiveInteger" use="required"/>
                </complexType>
            </element>
        </sequence>
        <attribute name="verifyToLockGoodFrames" type="integer" default="4"/>
        <attribute name="checkToLockGoodFrames" type="integer" default="1"/>
        <attribute name="maxErrorsInSyncWord" type="integer" default="0"/>
        <attribute name="syncAperature" type="integer" default="0"/>
    </complexType>
</schema>
```

A.6 Packaging.xsd

```
<?xml version="1.0"?>
<!-- edited with XMLSPY v5 rel. 2 U (http://www.xmlspy.com) by Ed Shaya (NASA) -->
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Nori Jungmann (Engineering) -->
<!--

$Id: Packaging.xsd,v 1.17 2003/01/24 11:42:43 cooper Exp $

-->
<schema targetNamespace="http://www.omg.org/space/tcspec" xmlns:tc="http://www.omg.org/space/tcspec" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <annotation>
        <documentation>$Id: Packaging.xsd,v 1.17 2003/01/24 11:42:43 cooper Exp $</documentation>
        <documentation xml:lang="en">This schema definies the dictionary for messages and containers, which in turn describe the physical composition of data in a communication system</documentation>
    </annotation>
    <include schemaLocation="Parameter.xsd"/>
    <simpleType name="ServiceNameType">
        <annotation>
            <documentation xml:lang="en">Identifier for a sevice</documentation>
        </annotation>
        <restriction base="tc:NameType"/>
    </simpleType>
    <simpleType name="MessageKeyIdType">
        <annotation>
            <documentation xml:lang="en">Identifier for a message key</documentation>
        </annotation>
        <restriction base="tc:NameType"/>
    </simpleType>
    <simpleType name="MessageNameType">
        <annotation>
            <documentation xml:lang="en">Identifier for a message</documentation>
        </annotation>
        <restriction base="tc:NameType"/>
    </simpleType>
```

```
<simpleType name="ContainerNameType">
    <annotation>
        <documentation xml:lang="en">Identifier for a container</documentation>
    </annotation>
    <restriction base="tc:NameType"/>
</simpleType>
<complexType name="MessageKeyType">
    <annotation>
        <documentation xml:lang="en">Denotes a bit field in a message that can be used to identify a particular message</documentation>
    </annotation>
    <sequence>
        <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0">
            <annotation>
                <documentation xml:lang="en">Optional long key description</documentation>
            </annotation>
        </element>
        <element name="OffsetInBits" type="tc:BitOffsetType" minOccurs="0">
            <annotation>
                <documentation xml:lang="en">Offset relative to start of container representing this message</documentation>
            </annotation>
        </element>
    </sequence>
    <attribute name="Id" type="tc:MessageKeyIdType" use="required">
        <annotation>
            <documentation xml:lang="en">Id of message key field</documentation>
        </annotation>
    </attribute>
    <attribute name="ShortDescription" type="tc:ShortDescriptionType" use="optional">
        <annotation>
            <documentation xml:lang="en">Optional short key description</documentation>
        </annotation>
    </attribute>
    <attribute name="WidthInBits" type="positiveInteger" use="required">
        <annotation>
            <documentation xml:lang="en">Width of key field in bits</documentation>
        </annotation>
    </attribute>
</complexType>
<complexType name="MessageType">
    <annotation>
        <documentation xml:lang="en">An abstract object that can be used to help identify structures on an incomming stream</documentation>
    </annotation>
    <sequence>
        <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0">
            <annotation>
                <documentation xml:lang="en">Optional long description of message</documentation>
            </annotation>
        </element>
        <element name="Key" type="tc:MessageKeyType" minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation xml:lang="en">Optional set of keys that this identify a message of this type</documentation>
            </annotation>
        </element>
    </sequence>
    <attribute name="name" type="tc:MessageNameType" use="required">
        <annotation>
            <documentation xml:lang="en">Unique identifier for the message</documentation>
        </annotation>
    </attribute>
    <attribute name="shortDescription" type="tc:ShortDescriptionType" use="optional">
        <annotation>
            <documentation xml:lang="en">Optional short name of the message</documentation>
```

```
            </annotation>
        </attribute>
    </complexType>
    <complexType name="MessageKeyMatchType">
        <annotation>
            <documentation xml:lang="en">Holds a value to use in a match to a refered to key</documentation>
        </annotation>
        <attribute name="Id" type="tc:MessageKeyIdType" use="required">
            <annotation>
                <documentation xml:lang="en">Id of message key to match to</documentation>
            </annotation>
        </attribute>
        <attribute name="Operator" type="tc:ComparisonOperatorsType">
            <annotation>
                <documentation xml:lang="en">Operator to compare with, optional, if not present equivalence assumed</documentation>
            </annotation>
        </attribute>
        <attribute name="Value" type="hexBinary" use="required">
            <annotation>
                <documentation xml:lang="en">Value of key to match with</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="MessageKeyMatchListType">
        <annotation>
            <documentation xml:lang="en">Holds a list of values for the keys of a message</documentation>
        </annotation>
        <sequence>
            <element name="KeyMatch" type="tc:MessageKeyMatchType" maxOccurs="unbounded">
                <annotation>
                    <documentation xml:lang="en">Vaue to use for message key match</documentation>
                </annotation>
            </element>
        </sequence>
        <attribute name="name" type="tc:MessageNameType" use="required">
            <annotation>
                <documentation xml:lang="en">Unique identifier for the message</documentation>
            </annotation>
        </attribute>
        <attribute name="AllMustBeTrue" type="boolean">
            <annotation>
                <documentation xml:lang="en">Must all matched be true? If not present then all must be true</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="ContainerType">
        <annotation>
            <documentation xml:lang="en">Represents a packet/frame/sub-frame etc.</documentation>
        </annotation>
        <sequence>
            <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0">
                <annotation>
                    <documentation xml:lang="en">Optional long description of container</documentation>
                </annotation>
            </element>
            <element name="ImplementsMessage" type="tc:MessageKeyMatchListType" minOccurs="0">
                <annotation>
                    <documentation xml:lang="en">Optional message this container represents</documentation>
                </annotation>
            </element>
            <element name="Sequence" type="tc:SequenceType" minOccurs="0">
                <annotation>
```

```
                        <documentation xml:lang="en">Optional body of this container. Most container will have a body however event packets do not.</documentation>
                    </annotation>
                </element>
            </sequence>
            <attribute name="name" type="tc:ContainerNameType" use="required">
                <annotation>
                    <documentation xml:lang="en">Id of container</documentation>
                </annotation>
            </attribute>
            <attribute name="shortDescription" type="tc:ShortDescriptionType" use="optional">
                <annotation>
                    <documentation xml:lang="en">Optional short description of container</documentation>
                </annotation>
            </attribute>
        </complexType>
        <complexType name="ContainerRefType">
            <annotation>
                <documentation xml:lang="en">Holds an offset to a refered to container and an optional parameter to determine validity of this container</documentation>
            </annotation>
            <sequence>
                <element name="Occurs" type="tc:OccursType" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation xml:lang="en">Repeating group to include</documentation>
                    </annotation>
                </element>
                <element name="DependantOn" type="tc:ComparisonCheckType" minOccurs="0">
                    <annotation>
                        <documentation xml:lang="en">Parameter that the inclusion of the sub-cotainer depends on</documentation>
                    </annotation>
                </element>
                <element name="OffsetInBits" type="tc:BitOffsetType" minOccurs="0">
                    <annotation>
                        <documentation xml:lang="en">Offset, in bits, from the start of this container to start sub-container. If not present then sub-container start either at offset 0 if the first sub-container or relative to end of previous object</documentation>
                    </annotation>
                </element>
            </sequence>
            <attribute name="nameRef" type="tc:ContainerNameType" use="required">
                <annotation>
                    <documentation xml:lang="en">Id of container to contain</documentation>
                </annotation>
            </attribute>
        </complexType>
        <complexType name="DynamicContainerRefType">
            <annotation>
                <documentation xml:lang="en">Holds a reference to a parameter that will hold a container id at runtime to expand</documentation>
            </annotation>
            <sequence>
                <element name="Occurs" type="tc:OccursType" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation xml:lang="en">Repeating group to include</documentation>
                    </annotation>
                </element>
                <element name="DependantOn" type="tc:ComparisonCheckType" minOccurs="0">
                    <annotation>
                        <documentation xml:lang="en">Parameter that the inclusion of the sub-cotainer depends on</documentation>
                    </annotation>
                </element>
                <element name="OffsetInBits" type="tc:BitOffsetType" minOccurs="0">
                    <annotation>
```

```
                    <documentation xml:lang="en">Offset, in bits, from the start of this container to start sub-container. If not present then sub-container start either at offset 0 if the first sub-container or relative to end of previous object</documentation>
                </annotation>
            </element>
            <element name="ParameterRef" type="tc:OffsetParameterRefType">
                <annotation>
                    <documentation xml:lang="en">Parameter that will hold container id at runtime</documentation>
                </annotation>
            </element>
        </sequence>
    </complexType>
    <complexType name="ServiceType">
        <annotation>
            <documentation xml:lang="en">Holds a set of services, logical groups of messages and containers</documentation>
        </annotation>
        <sequence>
            <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0">
                <annotation>
                    <documentation xml:lang="en">Optional long description of this service</documentation>
                </annotation>
            </element>
            <element name="MessageList">
                <annotation>
                    <documentation xml:lang="en">Set of messages this service uses</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="NameRef" type="tc:NameReferenceType" maxOccurs="unbounded">
                            <annotation>
                                <documentation xml:lang="en">reference to an existing message</documentation>
                            </annotation>
                        </element>
                    </sequence>
                </complexType>
            </element>
            <element name="ContainerList">
                <annotation>
                    <documentation xml:lang="en">Set of containers this service uses</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="Id" type="tc:ContainerNameType" maxOccurs="unbounded">
                            <annotation>
                                <documentation xml:lang="en">Id of an existing container</documentation>
                            </annotation>
                        </element>
                    </sequence>
                </complexType>
            </element>
        </sequence>
        <attribute name="name" type="tc:ServiceNameType" use="required">
            <annotation>
                <documentation xml:lang="en">Unique identifier for this service set</documentation>
            </annotation>
        </attribute>
        <attribute name="shortDescription" type="tc:ShortDescriptionType" use="optional">
            <annotation>
                <documentation xml:lang="en">Optional short description of this service</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="PackagingDefinitionType">
```

```
        <annotation>
            <documentation xml:lang="en">Complex type that holds all the elements of the packaging system</documentation>
        </annotation>
        <sequence>
            <element name="MessageList" minOccurs="0">
                <annotation>
                    <documentation xml:lang="en">Holds the list of message definitions</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="Message" type="tc:MessageType" maxOccurs="unbounded">
                            <annotation>
                                <documentation xml:lang="en">A single message definition</documentation>
                            </annotation>
                        </element>
                    </sequence>
                </complexType>
                <key name="MessageKey">
                    <selector xpath="Message"/>
                    <field xpath="Id"/>
                </key>
            </element>
            <element name="ContainerList" minOccurs="0">
                <annotation>
                    <documentation xml:lang="en">Holds the list of container definitions</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="Container" type="tc:ContainerType" maxOccurs="unbounded">
                            <annotation>
                                <documentation xml:lang="en">A single container definition</documentation>
                            </annotation>
                        </element>
                    </sequence>
                </complexType>
                <key name="ContainerKey">
                    <selector xpath="Container"/>
                    <field xpath="Id"/>
                </key>
            </element>
            <element name="NonStandardData" type="tc:NonStandardDataType" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="Packaging" type="tc:PackagingDefinitionType">
        <annotation>
            <documentation xml:lang="en">Root packaging element for this schema</documentation>
        </annotation>
    </element>
    <complexType name="SequenceType">
        <annotation>
            <documentation>Used to describe a sequence of sequences, containers, parameters, or choices</documentation>
        </annotation>
        <sequence>
            <element name="Occurs" type="tc:OccursType" minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <documentation xml:lang="en">Repeating group to include</documentation>
                </annotation>
            </element>
            <choice maxOccurs="unbounded">
                <element name="ContainerRef" type="tc:ContainerRefType"/>
                <element name="DynamicContainerRef" type="tc:DynamicContainerRefType"/>
                <element name="ArgumentRef" type="tc:OffsetParameterRefType">
```

```
                    <annotation>
                        <documentation xml:lang="en">Set of command arguments to include</documentation>
                    </annotation>
                </element>
                <element name="ParameterRef" type="tc:OffsetParameterRefType">
                    <annotation>
                        <documentation xml:lang="en">Set of parameters to include</documentation>
                    </annotation>
                </element>
                <element name="DynamicParameterRef" type="tc:OffsetParameterRefType">
                    <annotation>
                        <documentation xml:lang="en">Holds a reference to a parameter that will hold an onboard Id</documentation>
                    </annotation>
                </element>
                <element name="BinaryConstant">
                    <annotation>
                        <documentation>May also use a reference to a constant parameter</documentation>
                    </annotation>
                    <complexType>
                        <sequence>
                            <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0">
                                <annotation>
                                    <documentation xml:lang="en">Optional long description of fixed area</documentation>
                                </annotation>
                            </element>
                            <element name="OffsetInBits" type="tc:BitOffsetType" minOccurs="0"/>
                        </sequence>
                        <attribute name="lengthInBits" type="nonNegativeInteger" use="required"/>
                        <attribute name="value" type="hexBinary" use="required"/>
                    </complexType>
                </element>
                <element name="Choice" type="tc:ChoiceType"/>
                <element name="Sequence" type="tc:SequenceType"/>
            </choice>
        </sequence>
    </complexType>
    <complexType name="ChoiceType">
        <annotation>
            <documentation>Used to one or more possible container choices.</documentation>
        </annotation>
        <sequence>
            <element name="Occurs" type="tc:OccursType" minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <documentation xml:lang="en">Repeating group to include</documentation>
                </annotation>
            </element>
            <element name="ContainerRef" type="tc:ContainerRefType" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
</schema>
```

A.7 CommonTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 2 U (http://www.xmlspy.com) by Ed Shaya (NASA) -->
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Nori Jungmann (Engineering) -->
<schema targetNamespace="http://www.omg.org/space/tcspec" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:tc="http://www.omg.org/space/tcspec">
    <annotation>
        <documentation>$Id: CommonTypes.xsd,v 1.16 2003/03/03 07:51:22 cooper Exp $</documentation>
```

```
        <documentation xml:lang="en">Schema for global type definitions used in the Space TnC specification</documentation>
    </annotation>
    <include schemaLocation="Algorithm.xsd"/>
    <!-- Basic elements used for in all dictionaries -->
    <complexType name="AbsoluteTimeType">
        <annotation>
            <documentation>Contains an absolute (to a known epoch) time</documentation>
        </annotation>
        <simpleContent>
            <extension base="dateTime"/>
        </simpleContent>
    </complexType>
    <complexType name="AliasType">
        <annotation>
            <documentation xml:lang="en">Used to save an alias (alternate) name or ID that may be used to identify the parameter by the factory or in an alternative ground system.  Some ground system processing equipent has some severe naming restrictions on parameters (e.g., names must less then 12 characters, single case or integral id's only).</documentation>
        </annotation>
        <attribute name="nameSet" type="string" use="required"/>
        <attribute name="alias" type="string" use="required"/>
    </complexType>
    <complexType name="BaseDataType">
        <annotation>
            <documentation>Used to describe the base data types used by the ground system.  The ValidRange is a range for the parameter's value that is valid. It is used to display graphs and/or guages that have pre-defined boundaries</documentation>
        </annotation>
        <sequence>
            <choice>
                <element name="Any">
                    <annotation>
                        <documentation>A place holder parameter that is populated at runtime by using the value currently contained in the SourceParameter as an onboard Id</documentation>
                    </annotation>
                    <complexType>
                        <sequence>
                            <element name="SourceParameterRef" type="tc:NameReferenceType"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="Float">
                    <complexType>
                        <sequence>
                            <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0"/>
                            <element name="ValidRange" type="tc:RangeType" minOccurs="0" maxOccurs="unbounded"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="Integer">
                    <complexType>
                        <sequence>
                            <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0"/>
                            <element name="ValidRange" type="tc:RangeType" minOccurs="0" maxOccurs="unbounded"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="Enumerated">
                    <annotation>
                        <documentation>The list of label/value pairs associated with a parameter of datatype 'enumerated'</documentation>
                    </annotation>
                    <complexType>
                        <sequence>
                            <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0"/>
```

```
                    <element name="ValidRange" type="tc:RangeType" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </complexType>
        </element>
        <element name="Binary">
            <complexType>
                <sequence>
                    <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0"/>
                    <element name="ValidRange" type="tc:RangeType" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
                <attribute name="lengthInBits" type="nonNegativeInteger"/>
            </complexType>
        </element>
        <element name="Boolean">
            <complexType>
                <sequence>
                    <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0"/>
                    <element name="ValidRange" type="tc:RangeType" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </complexType>
        </element>
        <element name="String">
            <complexType>
                <sequence>
                    <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0"/>
                    <element name="Enumeration" type="string" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
                <attribute name="maxLength" type="nonNegativeInteger" use="optional"/>
            </complexType>
        </element>
        <element name="AbsoluteTime">
            <complexType>
                <sequence>
                    <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0">
                        <annotation>
                            <documentation>Use the [ISO 8601] extended format CCYY-MM-DDThh:mm:ss where "CC" represents the century, "YY" the year, "MM" the month and "DD" the day, preceded by an optional leading "-" sign to indicate a negative number. If the sign is omitted, "+" is assumed. The letter "T" is the date/time separator and "hh", "mm", "ss" represent hour, minute and second respectively. Additional digits can be used to increase the precision of fractional seconds if desired i.e the format ss.ss... with any number of digits after the decimal point is supported. </documentation>
                        </annotation>
                    </element>
                    <element name="ValidRange" type="tc:RangeType" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </complexType>
        </element>
        <element name="RelativeTime">
            <complexType>
                <sequence>
                    <element name="DefaultValue" type="tc:DefaultValueType" minOccurs="0">
                        <annotation>
                            <documentation>Duration's are expressed as PnYn MnDTnH nMnS, where nY represents the number of years, nM the number of months, nD the number of days, 'T' is the date/time separator, nH the number of hours, nM the number of minutes and nS the number of seconds. The number of seconds can include decimal digits to arbitrary precision.  For example, to indicate a duration of 1 year, 2 months, 3 days, 10 hours, and 30 minutes, one would write: P1Y2M3DT10H30M. One could also indicate a duration of minus 120 days as: -P120D.</documentation>
                        </annotation>
                    </element>
                    <element name="ValidRange" type="tc:RangeType" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </complexType>
        </element>
    </choice>
    <choice>
```

```
                <element name="Units">
                    <complexType>
                        <sequence>
                            <element name="Unit" type="tc:UnitType" maxOccurs="unbounded"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="Unitless"/>
            </choice>
        </sequence>
    </complexType>
    <complexType name="BaseParameterType">
        <annotation>
            <documentation>A type definition used as the base type for a CommandDefinition and for BusCommandDefinition.</documentation>
        </annotation>
        <complexContent>
            <extension base="tc:BaseDataType">
                <sequence>
                    <element name="LongDescription" type="tc:LongDescriptionType" minOccurs="0">
                        <annotation>
                            <documentation>An arbitrarily long description for the Parameter.  May also include HTML markup.</documentation>
                        </annotation>
                    </element>
                    <element name="BusAttributes" type="tc:BusAttributesType" minOccurs="0"/>
                </sequence>
                <attribute name="name" type="tc:NameType" use="required"/>
                <attribute name="shortDescription" type="tc:ShortDescriptionType" use="optional"/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="DefaultValueType" mixed="true">
        <annotation>
            <documentation xml:lang="en">Holds a default value and indicated whether it is calibrated or not</documentation>
        </annotation>
        <attribute name="calibrated" type="boolean" default="false"/>
    </complexType>
    <complexType name="BitOffsetType">
        <annotation>
            <documentation xml:lang="en">Denotes a bit field in a message that can be used to identify a particular message</documentation>
        </annotation>
        <choice>
            <element name="AbsoluteOffset" type="nonNegativeInteger" default="0" nillable="true">
                <annotation>
                    <documentation xml:lang="en">Absolute offset in bits.  Default is 0.</documentation>
                </annotation>
            </element>
            <element name="RelativeOffset" type="integer" default="0" nillable="true">
                <annotation>
                    <documentation xml:lang="en">Relative offset in bits from end of previous object, can be negative. Default is 0</documentation>
                </annotation>
            </element>
        </choice>
    </complexType>
    <complexType name="BusAttributesType">
        <annotation>
            <documentation xml:lang="en">Bus attributes describe how a particular piece of data is sent or received from some non-native, off-platform device.
(e.g. a spacecraft)</documentation>
        </annotation>
        <sequence>
            <element name="ErrorDetectCorrect" type="tc:ErrorDetectCorrectType" minOccurs="0"/>
            <element name="DefaultCalibrator" type="tc:CalibratorType" minOccurs="0"/>
            <element name="ByteOrder" minOccurs="0">
```

```
                <annotation>
                    <documentation>Used to describe an arbitrary byte order in multibyte parameters.  order is the order the byte is transmitted. </documentation>
                </annotation>
                <complexType>
                    <sequence maxOccurs="unbounded">
                        <element name="Byte">
                            <complexType>
                                <attribute name="appearanceInStream" type="nonNegativeInteger" use="required"/>
                                <attribute name="byteSignificance" type="nonNegativeInteger" use="required"/>
                            </complexType>
                        </element>
                    </sequence>
                </complexType>
            </element>
            <element name="ContextCalibrator" minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <documentation>Use when different calibrations must be used on the Parameter in different contexts.  Use the first one that tests true</documentation>
                </annotation>
                <complexType>
                    <sequence>
                        <element name="Context" type="tc:ComparisonCheckType"/>
                        <element name="Calibrator" type="tc:CalibratorType"/>
                    </sequence>
                </complexType>
            </element>
        </sequence>
        <attribute name="bitLength" type="unsignedInt" use="required"/>
        <attribute name="bitOrder" default="MostSignificantBitFirst">
            <simpleType>
                <restriction base="string">
                    <enumeration value="MostSignificantBitFirst"/>
                    <enumeration value="LeastSignificantBitFirst"/>
                </restriction>
            </simpleType>
        </attribute>
        <attribute name="busType" use="required">
            <simpleType>
                <restriction base="string">
                    <enumeration value="Binary"/>
                    <enumeration value="OnesComp"/>
                    <enumeration value="TwosComp"/>
                    <enumeration value="SignMag"/>
                    <enumeration value="Unsigned"/>
                    <enumeration value="IEEE754_1985Float"/>
                    <enumeration value="MILSTD_1750AFloat"/>
                    <enumeration value="ASCII_String"/>
                </restriction>
            </simpleType>
        </attribute>
    </complexType>
    <complexType name="SimpleParameterRefType">
        <annotation>
            <documentation xml:lang="en">A simple reference to a Parameter</documentation>
        </annotation>
        <attribute name="parameterName" type="tc:NameReferenceType" use="required"/>
        <attribute name="spaceSystemName" type="string"/>
    </complexType>
    <complexType name="ComparisonCheckType">
        <annotation>
            <documentation xml:lang="en">A simple boolean expresion</documentation>
```

```
            </annotation>
            <sequence>
                <element name="ParameterRef" type="tc:SimpleParameterRefType"/>
                <element name="ComparisonOperator" type="tc:ComparisonOperatorsType"/>
                <choice>
                    <element name="ArgumentRef" type="tc:SimpleParameterRefType">
                        <annotation>
                            <documentation>Takes the value of a command argument, is assumed to be of the same type as the comparison Parameter</documentation>
                        </annotation>
                    </element>
                    <element name="ParameterRef" type="tc:SimpleParameterRefType">
                        <annotation>
                            <documentation>Parameter is assumed to be of the same type as the comparison Parameter</documentation>
                        </annotation>
                    </element>
                    <element name="Value" type="string">
                        <annotation>
                            <documentation>Value is assumed to be of the same type as the comparison Parameter</documentation>
                        </annotation>
                    </element>
                </choice>
            </sequence>
            <attribute name="calibratedComparison" type="boolean" default="false">
                <annotation>
                    <documentation>Should the compaison be based on the calibrated format or not, if true and a constant value is used this should be in calibrated format too.</documentation>
                </annotation>
            </attribute>
        </complexType>
        <simpleType name="ComparisonOperatorsType">
            <annotation>
                <documentation xml:lang="en">Operators to use when testing a boolean condition for a validity check</documentation>
            </annotation>
            <restriction base="string">
                <enumeration value="=="/>
                <enumeration value="!="/>
                <enumeration value="&amp;lt;"/>
                <enumeration value="&amp;lt;="/>
                <enumeration value=">"/>
                <enumeration value=">="/>
            </restriction>
        </simpleType>
        <!-- ErrorDetectCorrect -->
        <complexType name="ErrorDetectCorrectType">
            <annotation>
                <documentation xml:lang="en">A simple element that provides for simple, but common error checking and detection.</documentation>
            </annotation>
            <choice>
                <element name="Parity">
                    <annotation>
                        <documentation xml:lang="en">Bit position starts with 'zero'.</documentation>
                    </annotation>
                    <complexType>
                        <attribute name="type" use="required">
                            <simpleType>
                                <restriction base="string">
                                    <enumeration value="Even"/>
                                    <enumeration value="Odd"/>
                                </restriction>
                            </simpleType>
                        </attribute>
```

```
                        <attribute name="bitsFromStart" type="nonNegativeInteger" use="required"/>
                    </complexType>
                </element>
                <element name="CRC">
                    <annotation>
                        <documentation xml:lang="en">Cyclic Redundancy Check definition. Legal values for coefficient's are 0 or 1.Exponents must be integer values.</documentation>
                    </annotation>
                    <complexType>
                        <sequence>
                            <element name="PolynomialType" type="tc:PolynomialType"/>
                        </sequence>
                        <attribute name="bitsFromStart" type="nonNegativeInteger"/>
                    </complexType>
                </element>
            </choice>
        </complexType>
        <simpleType name="FormatType">
            <annotation>
                <documentation xml:lang="en">Add</documentation>
            </annotation>
            <restriction base="string">
                <enumeration value="SignedInteger"/>
                <enumeration value="UnsignedInteger"/>
                <enumeration value="Real"/>
            </restriction>
        </simpleType>
        <!-- Header -->
        <complexType name="HeaderType">
            <annotation>
                <documentation xml:lang="en">Schema for a Header record.  A header contains general information about the document or subdocument.</documentation>
            </annotation>
            <sequence>
                <element name="Author" type="string" minOccurs="0" maxOccurs="unbounded"/>
                <element name="Note" type="string" minOccurs="0" maxOccurs="unbounded"/>
                <element name="History" type="string" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
            <attribute name="version" type="string"/>
            <attribute name="date" type="string"/>
            <attribute name="classification" default="Unclassified">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="Unclassified"/>
                        <enumeration value="Confidential"/>
                        <enumeration value="Secret"/>
                        <enumeration value="TopSecret"/>
                        <enumeration value="Proprietary"/>
                    </restriction>
                </simpleType>
            </attribute>
        </complexType>
        <simpleType name="IdentificationNumberType">
            <annotation>
                <documentation xml:lang="en">Add</documentation>
            </annotation>
            <restriction base="integer">
                <minInclusive value="1"/>
                <maxInclusive value="9999"/>
            </restriction>
        </simpleType>
        <simpleType name="LimitCheckType">
```

```
        <annotation>
            <documentation xml:lang="en">Identifies the type of limit set check</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="warning"/>
            <enumeration value="critical"/>
            <enumeration value="statusConsistency"/>
            <enumeration value="eventGeneration"/>
        </restriction>
    </simpleType>
    <simpleType name="LongDescriptionType">
        <annotation>
            <documentation xml:lang="en">An unounded description</documentation>
        </annotation>
        <restriction base="string"/>
    </simpleType>
    <simpleType name="MathOperatorsType">
        <annotation>
            <documentation xml:lang="en">Mathematical operators</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="+"/>
            <enumeration value="-"/>
            <enumeration value="mult"/>
            <enumeration value="div"/>
            <enumeration value="mod"/>
            <enumeration value="exp"/>
            <enumeration value="bitor"/>
            <enumeration value="bitand"/>
            <enumeration value="bitxor"/>
        </restriction>
    </simpleType>
    <complexType name="MathOperationType">
        <annotation>
            <documentation xml:lang="en">A simple math operation</documentation>
        </annotation>
        <sequence>
            <choice>
                <element name="ParameterRef" type="tc:NameReferenceType"/>
                <element name="Value" type="string">
                    <annotation>
                        <documentation>Value is assumed to be of the same type as the comparison Parameter</documentation>
                    </annotation>
                </element>
            </choice>
            <element name="Operator" type="tc:MathOperatorsType"/>
            <choice>
                <element name="ParameterRef" type="tc:NameReferenceType"/>
                <element name="Value" type="string">
                    <annotation>
                        <documentation>Value is assumed to be of the same type as the comparison Parameter</documentation>
                    </annotation>
                </element>
            </choice>
        </sequence>
    </complexType>
    <!-- NameType -->
    <simpleType name="NameType">
        <annotation>
            <documentation xml:lang="en">Used for "directory" style unique names.  Only letters, digits, '_', ' ' and "-" are allowed </documentation>
        </annotation>
        <restriction base="string">
```

```
            <pattern value="[a-zA-Z0-9_\- ]*"/>
        </restriction>
    </simpleType>
    <!-- NameReferenceType -->
    <simpleType name="NameReferenceType">
        <annotation>
            <documentation xml:lang="en">Used when referencing a directory style "NameType".</documentation>
        </annotation>
        <restriction base="string">
            <pattern value="[a-zA-Z0-9_\- /]*"/>
        </restriction>
    </simpleType>
    <complexType name="NonStandardDataType">
        <annotation>
            <documentation>An any type that may be used in certain key locations within the database to hold mission unique data.</documentation>
        </annotation>
        <complexContent>
            <extension base="anyType">
                <attribute name="usage" type="string" use="required"/>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="PolynomialType">
        <annotation>
            <documentation xml:lang="en">A polynomial expresssion.  For example: 3 + 2x</documentation>
        </annotation>
        <sequence>
            <element name="Term" maxOccurs="unbounded">
                <annotation>
                    <documentation xml:lang="en">A term in a polynomial expresssion. </documentation>
                </annotation>
                <complexType>
                    <attribute name="coefficient" type="double" use="required"/>
                    <attribute name="exponent" type="double" use="required"/>
                </complexType>
            </element>
        </sequence>
    </complexType>
    <complexType name="PropertyType">
        <annotation>
            <documentation>Used for custom user properties</documentation>
        </annotation>
        <sequence minOccurs="0" maxOccurs="unbounded">
            <element name="Property" type="tc:PropertyType"/>
        </sequence>
        <attribute name="name" type="tc:NameReferenceType" use="required"/>
        <attribute name="shortDescription" type="string"/>
        <attribute name="longDescription" type="string"/>
        <attribute name="value" type="string" use="required"/>
    </complexType>
    <simpleType name="RadixType">
        <annotation>
            <documentation xml:lang="en">Specifies the number base</documentation>
        </annotation>
        <restriction base="string">
            <enumeration value="Decimal"/>
            <enumeration value="Hexadecimal"/>
            <enumeration value="Octal"/>
            <enumeration value="Binary"/>
        </restriction>
    </simpleType>
    <!-- RangeType -->
```

```
    <complexType name="RangeType">
        <annotation>
            <documentation xml:lang="en">A range of numbers.  "minInclusive", "minExclusive", "maxInclusive" and "maxExclusive" attributes are borrowed from the W3C schema language.</documentation>
        </annotation>
        <attribute name="minInclusive" type="string"/>
        <attribute name="minExclusive" type="string"/>
        <attribute name="maxInclusive" type="string"/>
        <attribute name="maxExclusive" type="string"/>
        <attribute name="calibrated" type="boolean" default="false"/>
    </complexType>
    <simpleType name="ShortDescriptionType">
        <annotation>
            <documentation xml:lang="en">A bounded description</documentation>
        </annotation>
        <restriction base="string">
            <maxLength value="32"/>
        </restriction>
    </simpleType>
    <complexType name="SplinePointType">
        <annotation>
            <documentation xml:lang="en">a spline is a set on points from which a curve may be drawn to interpolate raw to calibrated values</documentation>
        </annotation>
        <attribute name="order" type="positiveInteger" default="1"/>
        <attribute name="raw" type="double" use="required"/>
        <attribute name="calibrated" type="double" use="required"/>
    </complexType>
    <complexType name="SystemNameType">
        <annotation>
            <documentation xml:lang="en">Sub-system name.</documentation>
        </annotation>
        <simpleContent>
            <extension base="string"/>
        </simpleContent>
    </complexType>
    <complexType name="ToStringType">
        <annotation>
            <documentation xml:lang="en">There are two ways numeric data can be changed to string data: using a C style format string, or using an enumerated list.  Enumerated lists can be assigned to a single value or a value range.</documentation>
        </annotation>
        <choice>
            <choice maxOccurs="unbounded">
                <element name="ValueEnumeration">
                    <annotation>
                        <documentation xml:lang="en">A number or range assigned to a string.</documentation>
                    </annotation>
                    <complexType>
                        <attribute name="value" type="integer" use="required"/>
                        <attribute name="stringValue" type="string" use="required"/>
                    </complexType>
                </element>
                <element name="RangeEnumeration">
                    <annotation>
                        <documentation xml:lang="en">A string value associated with a numerical range.</documentation>
                    </annotation>
                    <complexType>
                        <sequence>
                            <element name="Range" type="tc:RangeType"/>
                        </sequence>
                        <attribute name="stringValue" type="string" use="required"/>
                    </complexType>
                </element>
```

```
            </choice>
            <element name="NumberFormat">
                <complexType>
                    <attribute name="numberBase" type="tc:RadixType" use="optional"/>
                    <attribute name="minimumFractionDigits" type="nonNegativeInteger" use="optional"/>
                    <attribute name="maximumFractionDigits" type="nonNegativeInteger" use="optional"/>
                    <attribute name="minimumIntegerDigits" type="nonNegativeInteger" use="optional"/>
                    <attribute name="maximumIntegerDigits" type="nonNegativeInteger" use="optional"/>
                    <attribute name="negativeSuffix" type="string" use="optional"/>
                    <attribute name="positiveSuffix" type="string" use="optional"/>
                    <attribute name="negativePrefix" type="string" use="optional" default="-"/>
                    <attribute name="positivePrefix" type="string" use="optional"/>
                    <attribute name="showThousandsGrouping" type="boolean" use="optional" default="true"/>
                    <attribute name="notation" use="optional" default="normal">
                        <simpleType>
                            <restriction base="string">
                                <enumeration value="normal"/>
                                <enumeration value="scientific"/>
                                <enumeration value="engineering"/>
                            </restriction>
                        </simpleType>
                    </attribute>
                </complexType>
            </element>
        </choice>
    </complexType>
    <complexType name="UnitType" mixed="true">
        <annotation>
            <documentation>Used to hold the unit(s) plus possibly the exponents for the units</documentation>
        </annotation>
        <attribute name="power" type="decimal" use="optional" default="1"/>
        <attribute name="description" type="string"/>
    </complexType>
    <simpleType name="RelativeTimeType">
        <annotation>
            <documentation>Used to describe a relative time.  Normally used for time offsets.  A Relative time is expressed as PnYn MnDTnH nMnS, where nY represents the number of years, nM the number of months, nD the number of days, 'T' is the date/time separator, nH the number of hours, nM the number of minutes and nS the number of seconds. The number of seconds can include decimal digits to arbitrary precision.  For example, to indicate a duration of 1 year, 2 months, 3 days, 10 hours, and 30 minutes, one would write: P1Y2M3DT10H30M. One could also indicate a duration of minus 120 days as: -P120D.</documentation>
        </annotation>
        <restriction base="duration"/>
    </simpleType>
    <complexType name="RealRangeType">
        <annotation>
            <documentation>A range for real numbers.  "minInclusive", "minExclusive", "maxInclusive" and "maxExclusive" attributes are borrowed from the W3C schema language.</documentation>
        </annotation>
        <attribute name="minInclusive" type="float"/>
        <attribute name="minExclusive" type="float"/>
        <attribute name="maxInclusive" type="float"/>
        <attribute name="maxExclusive" type="float"/>
    </complexType>
    <complexType name="OccursType">
        <annotation>
            <documentation xml:lang="en">Hold a structure that can be repeated X times, where X is held in the supplied parameter reference</documentation>
        </annotation>
        <sequence>
            <choice>
                <element name="CountRef" type="tc:OffsetParameterRefType">
                    <annotation>
                        <documentation xml:lang="en">Parameter that contains the count of repeated structures, implies variable length structure</documentation>
```

```
                    </annotation>
                </element>
                <element name="Count">
                    <annotation>
                        <documentation xml:lang="en">Fixed value that contains the count of repeated structures, implies fixed length structure</documenta-
tion>
                    </annotation>
                    <complexType>
                        <attribute name="NumOfOcc" type="positiveInteger" use="required">
                            <annotation>
                                <documentation xml:lang="en">Number of occurances of structure in container</documentation>
                            </annotation>
                        </attribute>
                        <attribute name="RelativeOffsetInBits" type="nonNegativeInteger" use="optional">
                            <annotation>
                                <documentation xml:lang="en">Number of bits between the start of two consecutive structures</documentation>
                            </annotation>
                        </attribute>
                        <attribute name="TimeOffset" type="integer" use="optional" default="0">
                            <annotation>
                                <documentation xml:lang="en">Time delay in milliseconds between two consecutive occurances</documentation>
                            </annotation>
                        </attribute>
                    </complexType>
                </element>
            </choice>
            <element name="OffsetInBits" type="tc:BitOffsetType" minOccurs="0">
                <annotation>
                    <documentation xml:lang="en">Offset in bits from the start of the container scope</documentation>
                </annotation>
            </element>
        </sequence>
        <attribute name="type" default="Occurs">
            <simpleType>
                <restriction base="NMTOKENS">
                    <enumeration value="minOccurs"/>
                    <enumeration value="maxOccurs"/>
                    <enumeration value="Occurs"/>
                </restriction>
            </simpleType>
        </attribute>
    </complexType>
    <complexType name="IntegerRangeType">
        <annotation>
            <documentation>A range for integers</documentation>
        </annotation>
        <attribute name="min" type="long"/>
        <attribute name="max" type="long"/>
    </complexType>

                                        </schema>
```