**SM&C Data Type Mapping to XTCE**

**Kevin Rice & Ron Jones**

**ASRC Inc, NASA-GSFC**

**Jan. 2015**

Table of Contents

[1 SMC Service Data Type Directly in XTCE 2](#_Toc410053620)

[1.1 MC 2](#_Toc410053621)

[1.2 Action 2](#_Toc410053622)

[1.2.1 ActionDefinition 2](#_Toc410053623)

[1.3 Parameter 4](#_Toc410053624)

[1.3.1 ParameterDefinition 4](#_Toc410053625)

[1.4 Alert 5](#_Toc410053626)

[1.4.1 AlertDefinition 5](#_Toc410053627)

[1.5 Check 7](#_Toc410053628)

[1.5.1 CheckDefinition 7](#_Toc410053629)

[1.5.2 ConstantCheck 7](#_Toc410053630)

[1.5.3 ReferenceCheck 8](#_Toc410053631)

[1.5.4 DeltaCheck 8](#_Toc410053632)

[1.5.5 LimitCheck 8](#_Toc410053633)

[1.6 Statistic 8](#_Toc410053634)

[1.6.1 StatisticFunctionDetails 8](#_Toc410053635)

[1.7 Aggregation 8](#_Toc410053636)

[1.7.1 AggregrationDefinition 8](#_Toc410053637)

[1.8 Conversions 9](#_Toc410053638)

[1.8.1 DiscreteConversion 9](#_Toc410053639)

[1.8.2 LineConversion 9](#_Toc410053640)

[1.8.3 PolyConversion 9](#_Toc410053641)

[1.8.4 RangeConversion 9](#_Toc410053642)

[1.9 Group 9](#_Toc410053643)

[1.9.1 GroupDefinition 9](#_Toc410053644)

[2 Appendix: Additional Thoughts 9](#_Toc410053645)

[Figure 2‑1 2](file:///C:\Users\jkrice1\Desktop\XTCE\ESA-XTCE\SMCtoXTCE_V2.docx#_Toc406682933)

[Figure 2‑2 5](file:///C:\Users\jkrice1\Desktop\XTCE\ESA-XTCE\SMCtoXTCE_V2.docx#_Toc406682934)

# SMC Service Data Type Directly in XTCE

## MC

Placeholder…

## Action

### ActionDefinition

“The ActionDefinition structure holds the definition information of an action.”

In XTCE this is a kind of MetaCommand.

Figure 2‑1

1..\*

MetaCommand

*<MetaCommand>*

*ActionDefinition*

< ActionDefinition>

@name

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **ActionDefinition** | **Type** | **XTCE -- MetaCommand** | **XTCE Type** |
| name | MAL::Indentifier | @name | <pattern value="[a-zA-Z0-9\_\-]\*"/> |
| description | MAL::String | @shortDescription or  LongDescription | xs:string |
| Severity | Severity  -INFORMATIONAL  -WARNING  -ALARM  -SEVERE  -CRITICAL | DefaultSignificance  @consequenceLevel  (pick 5 of 6) | -none  -watch  -warning  -distress  -critical  -severe |
| ProgressStepCount | MAL::UShort | AncillaryDataSet  AncillaryData | No type |
| Arguments | List<ArgumentDefinition> | ArgumentList  Arguments 1 … N | See Table 2 |
| ArgumentIds | List<MAL::Identifier> | ArgumentList  Arguments 1 … N  AliasSet  Alias | No Type |
|  |  | BaseMetaCommand  @metaCmdRef=”ActionDefinition” |  |

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **ArgumentDefinition** | **Type** | **XTCE -- AggregateArgumentType** | **XTCE Type/Note** |
|  |  | @name | <pattern value="[a-zA-Z0-9\_\-]\*"/> |
| rawType | MAL::Octet | ArgumentType | Map to one of XTCE’s ArgumentTypes: string, integer, float, binary, Boolean, enum, abs. time, rel. time, array and aggregate. |
| rawUnit | MAL::String |  | There is no rawUnit. There’s a UnitSet but it is not specifically define as being raw vs engineering. |
| conversionCondition | List<ConditionalReference> | AncillaryDataSet? | Although XTCE supports condition based calibrators, no calibrators are defined in this element. There is no good mapping in XTCE. |
| convertedType | MAL::Octet | ArgumentType/XXXDataEncoding | There are four choices for DataEncoding: integer, float, string and binary. |
| convertedUnit | MAL::String | AliasSet/Alias | These are optional, so they must go into Alias |

## Parameter

### ParameterDefinition

“The ParameterDefinition structure holds a parameter definition.”

The ParameterDefinition maps best to XTCE’s Parameter and ParameterType.

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **ParameterDefinition** | **Type** | **XTCE – Parameter/ParameterType** | **XTCE Type/Note** |
| name | MAL::Identifier | @name | <pattern value="[a-zA-Z0-9\_\-]\*"/> |
| description | MAL::String | @shortDescription or  LongDescription | xs:string |
| rawType | MAL::Octet | ParameterType | Map to one of XTCE’s ArgumentTypes: string, integer, float, binary, Boolean, enum, abs. time, rel. time, array and aggregate. |
| rawUnit | MAL::String |  | There is no rawUnit. There’s a UnitSet but it is not specifically defined as being raw vs engineering. |
| generationEnabled | MAL::Boolean | AncillaryDataSet? | No Type |
| updateInterval | MAL::Boolean | AncillaryDataSet? | No Type |
| validityExpression | ParameterExpression | Parameter/ValidityCondition | Parameter operator value, converted or raw |
| conversion | ParameterConversion | ContextCalibrator | For parameter conversion, if a condition is evaluate to true – the actual conversion is looked up in some manner. In XTCE the conversion are held with the type, there is a way to define conditions associated with them. In theory the conversion could be “pre-looked up” and stored in XTCE in the ContextCalibrator – along with the conversion expression. |

## Alert

### AlertDefinition

“The AlertDefinition provides the definition of an alert including any argument definitions.”

An alert is not a command but an event. XTCE 1.1 does not have event syntax but XTCE 1.2 does. However this mapping describes it using XTCE 1.1. (see Section 2.1).

SequenceContainer

tlmFlags==true

*<SequenceContainer>*

*Telemetry Header*

APID < ?

APID >=?

<SequenceContainer>

Eng. Packet(s)

*<SequenceContainer>*

*AlertDefinition*

1..\*

APID==?

< AlertDefinition>

@name

|  |  |  |  |
| --- | --- | --- | --- |
| **AlertDefinition** | **Type** | **XTCE -- SequenceContainer** | **XTCE Type** |
| name | MAL::Indentifier | @name | <pattern value="[a-zA-Z0-9\_\-]\*"/> |
| description | MAL::String | @shortDescription or  LongDescription | xs:string |
| Severity | Severity  -INFORMATIONAL  -WARNING  -ALARM  -SEVERE  -CRITICAL | DefaultSignificance  @consequenceLevel  (pick 5 of 6) | -none  -watch  -warning  -distress  -critical  -severe |
| generationEnabled | MAL::Boolean | AncillaryDataSet  AncillaryData | No type |
| Arguments | List<ArgumentDefinition> | ArgumentList  Arguments 1 … N | Parameters, see below |
| ArgumentIds | List<MAL::Identifier> | ArgumentList  Arguments 1 … N  AliasSet  Alias | Parameter/AliasSet |
|  |  | BaseContainer  @containerRef=”ActionDefinition” | RestrictionCriteria APID==# |

Figure 2‑2

In XTCE, telemetry is always described using Parameter and ParameterType. The arguments above map then to Parameters in XTCE. Use Table 2 as a guide, but replace the argument nomenclature with Parameter.

The UML-like diagram above (Figure 2‑2) shows an derivation for purposes of example to explain the concepts. Since XTCE 1.1 does not have specific event syntax, the best that can be done is to create unique inheritance chain for that type of telemetry. In the diagram this is shown with various constraints (RestrictionCriteria) against the CCSDS APID (application id) but the specifics could change depending on implementation.

## Check

### CheckDefinition

“The CheckDefinition structure holds the definition of a check.”

An abstract “root” for limit or alarm checks. In XTCE the limit/alarm is defined at the telemetry parameter type. The available forms of alarms are dependent on the data type of the parametertype (i.e. string, enum, integer, float, binary, boolean, absolute, relative time).

|  |  |  |  |
| --- | --- | --- | --- |
| **CheckDefinition** | **Type** | **XTCE -- MetaCommand** | **XTCE Type/Notes** |
| name | MAL::Indentifier | @name | <pattern value="[a-zA-Z0-9\_\-]\*"/> |
| description | MAL::String | @shortDescription or  LongDescription | xs:string |
| checkSeverity | Severity  -INFORMATIONAL  -WARNING  -ALARM  -SEVERE  -CRITICAL |  | -none (assumed to inside lowest alarm range)  -watch  -warning  -distress  -critical  -severe |
| maxReportingInterval | MAL::Duration | AncillaryDataSet  AncillaryData | No type |
| nominalCount | MAL::UInteger | Not in XTCE 1.1, but in XTCE 1.2 | Ancillary in XTCE1.1, supported in XTCE 1.2. |
| violationCount | MAL::UInteger | @minViolations | xs:integer |

### ConstantCheck

“The ConstantCheck structure holds the constant values to compare against for a consistency check.”

In XTCE the best match for a SINGLE constant check would be an AlarmCondition where the operator is “equals” (“==”). If there are several values to check with different levels of severity then up to 5 could be specified in a single AlarmCondition.

If a list of values is does not meet this criteria then the “ContextAlarmList” is the best fit – although coming up with “contexts” (conditions) that must simply evaluate to true in all cases to work through the list is not ideal.

XTCE’s alarm model is that of a “switch” statement. There’s a default alarm, and a context list which represents each switch case. Thus any list of items that needs to be checked must be mapped to the switch/case/default model.

The AlarmCondition in XTCE is available for all scalar parameter types.

Although AlarmCondition is available for all scalars – string and enum also have separate alarms form where a list of value can be supplied – the StringAlarmList and EnumAlarmList.

### ReferenceCheck

“The ReferenceCheck structure holds the key to another entity to compare against for a consistency check.”

In XTCE AlarmCondition as well as others can refer to any parameter, not just the one the alarm is defined in.

### DeltaCheck

In XTCE the numeric parameter types (integer, float) have a ChangeAlarm which can either be by sample or by percentage – and a range can be given as well.

### LimitCheck

In XTCE the numeric parameter types (integer, float) have a RangeAlarm which allows one to specific least severe to most severe ranges, where the inner most range is “normal/nominal/green” – some call these “inside alarms”. In XTCE 1.2 both inside and outside alarms are available where an outside alarm has the most severe range as the inner most range.

## Statistic

### StatisticFunctionDetails

There are no specific statistic functions in XTCE.

## Aggregation

### AggregrationDefinition

In XTCE there is a complex data type called an AggregrateParameterType (there is a similar type called AggregrateArgumentType on the command side).

## Conversions

### DiscreteConversion

A conversion between raw and calibrated points.

XTCE has a SplineCalibrator for numeric however it defines (order = 1) a line in which it is assume the value are interpolated between them.

### LineConversion

A conversion between raw and calibrated points forming a line in which value between points are interpolated, the lines ends may be extrapolated.

XTCE has a SplineCalibrator for numeric however it defines (order = 1) a line in which it is assume the value are interpolated between them.

There is also an extrapolate attribute applied to values beyond the line ends.

### PolyConversion

Defines a polynomial conversion.

XTCE has a polynomial conversion for numerics.

### RangeConversion

Map between a range and single value.

Range conversion are often seen in relation to enumerated definitions. XTCE 1.1 does not support range enums but XTCE 1.2. does support them.

In theory one could define this with XTCE’s more open MathOperationCalibrator.

In order to accommodate a list of such definitions, a context calibrator would have to be defined.

However this is not an ideal mapping and may prove cumbersome, inefficient or unworkable.

## Group

### GroupDefinition

XTCE does not support such a concept… N/A.

# Appendix: Additional Thoughts

XTCE describes binary blocks associated with telemetry and commanding such as CCSDS packets (or PUS).

We posit the relationship is as shown the in the diagram, although not all aspects are fully understood at this time.

Given that, it seems to us another approach would be to define specific XML schemas for each SMC item presented below in Section 2, then use instances of these to communicate the specific behaviors desired to/from the spacecraft. These would then mapped to tlm/cmd packet by defines a binary mapping of that information.

The binary mapping then could then be described w/XTCE although whether this is beneficial or not is another matter.

*SMC Tables from Blue Book*

Defines the semantics of

Informs

Defines Mapping

XTCE Descriptions

Maps to

Instances of

SMC Message Instances