PUS-packet model with ASN.1 and ACN

This document is supposed to give a short overview about the example TM/TC format and its representation using ASN.1 and ACN. The ECSS-E-70-41A standard was used as a reference for the packet structure and payload data format. In this example ASN.1 is used to describe the data structures of the packet and the payload data and ACN provides the means to define the encoding of the data structures on the bit-level and automatically generate de- and encoding source code for the C and ADA programming languages as well as an ICD of the packet of the protocol.

Organisation

The whole example is divided into 3 modules:

* **base-types**: contains type definitions for basic types which are common for telecommands and telemetry, e.g. integer types of different range, some common ENUMERATED, etc.
* **telecommand**: contains all structures related to the telecommands of the protocol
* **telemetry**: contains all structures related to the telemetries of the protocol

Each module is represented by 2 files, one containing the data types in ASN.1 (\*.asn) and one containing the ACN-encoding properties (\*.acn) following the ACN-encoding guidelines[[1]](#footnote-1). In the following a short overview of the telecommand and telemetry module is given. The structure and naming of data types follows the description in the PUS standard as closely as possible. The base types module is considered quite self-explanatory and won’t be mentioned further.

Telecommand module

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Packet Header (48 Bits) | | | | | | | Packet Data Field (Variable) | | | |
| Packet ID | | | | Packet Sequence Control | | Packet Length | Data Field Header (Optional) (see Note 1) | Application Data | Spare | Packet Error Control (see Note 2) |
| Version Number (=0) | Type (=1) | Data Field Header Flag | Applica- tion Process ID | Sequence Flags | Sequence Count |  |  |  |  |  |
| 3 | 1 | 1 | 11 | 2 | 14 |  |  |  |  |  |
| 16 | | | | 16 | | 16 | Variable | Variable | Variable | 16 |

Figure 1 Telecommand Packet structure

Figure 1 presents the general Telecommand packet layout according to the PUS. The corresponding ASN.1 and ACN representation is shown in Listing 1 and Listing 2. The layout of the packet header matches exactly the one of Figure 1 with the subordinate structures Packet ID, Packet Sequence Control and Packet Length, because the Packet Header as a whole or parts of it are often return values in telemetry packets (e.g. the PUS service 1). It allows the application process to easily copy e.g. the Packet ID field of a TC to the respective field of a TM using a simple assignment operation instead of copying each value individually and without any knowledge of the encoding settings.

However, no metatype for the Packet Data Field is defined, but instead Data Field Header, Application Data and Packet Error Control are defined directly in the Telecommand type. The main reason is to avoid a too fine grained separation and thus cluttered tree structure for the developer of the application process, especially since the Data Field Header and the Packet Error Control field are of the same structure for every Telecommand packet and only the Application Data field varies for each TC. Nevertheless, if the main goal is to match the PUS exactly it would be easy to define an additional type for the Packet Data Field. Of course, this does not affect the encoding which is PUS compliant in any case. In the ASN.1 definition of the Packet ID only the APID field is explicitly defined, all other fields are constant for all TCs and therefore the respective bit patterns are defined in the ACN-file.

T-telecommand ::= SEQUENCE

{

packet-header TC-packetHeader,

data-field-header T-tc-dataFieldHeader,

application-data T-tc-applicationData,

crc T-uint16

}

TC-packetHeader ::= SEQUENCE

{

packet-id T-tc-packetID,

packet-sequence-control T-packetSequenceControl,

packet-length T-uint16

}

T-tc-packetID ::= SEQUENCE

{

-- versionNumber INTEGER(0..7), always 0

-- packetType T-packetType, always 1 for telecommand

-- hasDataFieldHdr BOOLEAN, always 1

apid T-apid

}

Listing 1 ASN.1 definition of the Telecommand Packet structure

T-telecommand[]

{

packet-header [],

data-field-header [] ,

application-data<data-field-header.service-type, data-field-header.service-subtype>[],

crc []

}

TC-packetHeader []

T-tc-packetID []

{

ccsds-version-number NULL [pattern '000'B], -- fixed according to PUS

packet-type NULL [pattern '1'B] , -- dito

has-data-fieldhdr NULL [pattern '1'B], -- dito

apid []

}

Listing 2 ACN definition of the Telecommand Packet structure

The main logic of the protocol is behind the definition of the Application Data field. It consists of a CHOICE-type which maps for each valid TC a label to a type definition with the corresponding data layout of the TC. The Service Type and Service Subtype field of the Data Field Header determine the kind of TC, i.e. what CHOICE parameter to choose. They are passed to the Application Data field in the ACN file (see Listing 2). The definition of the CHOICE-type and the mapping of Service Type and Service Subtype to a certain TC can be seen in Listing 3 and Listing 4. Note that several TC only have a Data Field Header but no actual Application Data. These are defined of type T-NULL which is imported from base-types and is encoded with 0-bits. For each TC which does transport Application Data a separate type is defined which is a straight forward task.

T-tc-dataFieldHeader[]

{

ccsds-2nd-header-flag NULL [pattern '0'B], -- fixed = 0

packet-pus-version NULL [pattern '001'B], -- fixed = 001

ack-execution-completion [true-value '1'B],

ack-execution-progress [true-value '1'B] ,

ack-execution-start [true-value '1'B] ,

ack-acceptance [true-value '1'B] ,

service-type base-types.T-uint8[],

service-subtype base-types.T-uint8[],

source-id []

}

-- Table which maps the pusType and subtype to the corresponding

-- packet payload data

T-tc-applicationData<base-types.T-uint8:pusType, base-types.T-uint8:pusSubType> []

{

tc-3-27-update-hk-period [present-when pusType== 3 pusSubType== 27 ],

tc-6-2-load-memory [present-when pusType== 6 pusSubType== 2 ],

tc-6-5-dump-memory [present-when pusType== 6 pusSubType== 5 ],

tc-6-9-check-memory [present-when pusType== 6 pusSubType== 9 ],

tc-6-129-transfer-image [present-when pusType== 6 pusSubType==129 ],

tc-210-3-reset-dpu [present-when pusType==210 pusSubType== 3 ],

tc-210-4-enable-watchdog [present-when pusType==210 pusSubType== 4 ],

tc-210-5-disable-watchdog [present-when pusType==210 pusSubType== 5 ],

tc-210-6-boot-iasw [present-when pusType==210 pusSubType== 6 ],

tc-197-2-report-boot [present-when pusType==197 pusSubType== 2 ]

}

Listing 3 ACN definition of Application Data

T-tc-dataFieldHeader ::= SEQUENCE

{

-- ccsdsHeaderFlag BOOLEAN, set to 0

-- packetPusVersion INTEGER(0..7), set to 1

ack-execution-completion BOOLEAN,

ack-execution-progress BOOLEAN,

ack-execution-start BOOLEAN,

ack-acceptance BOOLEAN,

source-id T-sourceId

}

T-tc-applicationData ::= CHOICE

{

tc-3-27-update-hk-period TC-UPDATE-HK-PERIOD,

tc-6-2-load-memory TC-LOAD-MEMORY,

tc-6-5-dump-memory TC-DUMP-MEMORY,

tc-6-9-check-memory TC-CHECK-MEMORY,

tc-6-129-transfer-image TC-TRANSFER-IMAGE,

tc-210-3-reset-dpu T-NULL,

tc-210-4-enable-watchdog T-NULL,

tc-210-5-disable-watchdog T-NULL,

tc-210-6-boot-iasw TC-BOOT-IASW,

tc-197-2-report-boot T-NULL

}

Listing 4 ASN.1 definition of Application Data

Telemetry module

The Telemetry Packet structure is generally the same as the one from for the Telecommand Packages with the exception that a few constant values differ and the Data Field Header Header has an additional field. Therefore the rationale of the TC apply here in the very same way.

A main difference appears in the definition of the Application Data. First, the TM-packets of the Telecommand verification service (Service type 1) use types defined in the Telecommand module (namely the Packet ID and Packet Sequence Control) which have to be imported. More important are the differences for the event reporting service. A TM-packet with a certain service type and subtype can still have different data layouts. For example TMs which report errors with high severity (service type 5, subtype 4) can transmit different parameters depending on the Report ID. In the ASN.1 notation this corresponds to another CHOICE; this time depending on an ENUMERATED. Listing 5 and Listing 6 show an example definition in ASN.1 and ACN for the mentioned report type. Report types for low and medium severity errors are defined in the same manner.

T-tm-applicationData ::= CHOICE

{

tm-1-1-acc-sucess TM-ACC-SUCCESS,

[…]

tm-5-2-event-low-severity TM-EVENT-ANOMALY-REPORT-LOW-SEVERITY,

tm-5-3-event-medium-severity TM-EVENT-ANOMALY-REPORT-MEDIUM-SEVERITY,

tm-5-4-event-high-severity TM-EVENT-ANOMALY-REPORT-HIGH-SEVERITY

}

-- Type defining layout common to all reports

TM-EVENT-ANOMALY-REPORT-HIGH-SEVERITY ::= SEQUENCE

{

-- event-report-id set in ACN

event-error-code T-event-error-code,

event-error-cnt T-uint8,

report-data T-report-data-high-severity

}

T-report-data-high-severity ::= CHOICE

{

he-dpu-reset T-event-he-dpu-reset,

he-dpu-boot T-event-he-dpu-boot

}

T-event-he-dpu-reset ::= T-event-le-dpu-reset

T-event-he-dpu-boot ::= SEQUENCE

{

first-seg-addr T-address,

last-seg-addr T-address,

seg-header-1 T-address

}

-- Used in the ACN-file for determination of the packet type

T-event-report-id-he ::= ENUMERATED

{

he-dpu-reset (9),

he-dpu-boot (10)

}

Listing 5 Example of event report definition in ASN.1

T-tm-applicationData<base-types.T-uint8:pusType, base-types.T-uint8:pusSubType> []

{

tm-1-1-acc-sucess [present-when pusType== 1 pusSubType== 1],

[…]

tm-5-2-event-low-severity [present-when pusType== 5 pusSubType== 2],

tm-5-3-event-medium-severity [present-when pusType== 5 pusSubType== 3],

tm-5-4-event-high-severity [present-when pusType== 5 pusSubType== 4]

}

TM-EVENT-ANOMALY-REPORT-HIGH-SEVERITY []

{

event-report-id T-event-report-id-he [],

event-error-code [],

event-error-cnt [],

report-data <event-report-id>[]

}

-- Use event report id to determine the type of the report

T-report-data-high-severity<T-event-report-id-he:id> [determinant id]

Listing 6 Example of event report definition in ACN

1. <http://semantix.gr/asn1scc/ACN-UM-v-2.0x.pdf> [↑](#footnote-ref-1)