Executive Summary

Reference: SPB-PUSC-480-ES-001

Author: Project team

Version: 1.1

Date: 26/11/2019

OVERVIEW

This document is the executive summary for the study on the Deployment of the PUS-C Standard in Projects supported by an Automatic Generation Toolset.

TABLE OF CONTENTS

1 BACKGROUND

The version A of the PUS standard (i.e. the ECSS-E-ST-70-41A Telemetry and Telecommand Packet Utilization standard) was published in January 2003. It has been widely used in Europe. A new version of that standard (version C) has been released by mid of 2016. That new version reflects the lessons learned from using the PUS, including for example, the standardization of new services, used by recent missions, of interest for the future. It also takes into account the availability of new ECSS and CCSDS standards, for which compliance is required by European Space Projects and Organizations.

The PUS C standard defines the standardized monitoring and control logical view of PUS compliant Space Systems, including both ground and space segments but mainly focusing on the requirements that apply to the space segment, i.e. the spacecraft.

The PUS C standard specifies the conceptual model of that standardized monitoring and control information system. That conceptual model is called "the PUS foundation model".

In the clause 5 of that standard, the generic elements of the PUS foundation model are defined and organised in two "abstraction models", i.e.:

- The PUS service type abstraction model, and
- The PUS service abstraction model.

Clause 5 defines the generic rules that apply to:

- Any service type specification that claims compliance with the PUS standard;
- Any service specification developed for compliance with the PUS.

Clause 6 complies with the generic rules and defines the standardized service types of the service type abstraction model. These standardized service type requirement specifications can, from a modelling viewpoint, be seen as subtypes of the service type abstraction model generic concepts.

Populating the PUS foundation model with the "type" knowledge used to produce the clause 6 service types' definitions allows extracting the semantics of the standardized service types. Using that semantics allows automating the production of PUS compliant systems.

As for many other ECSS standards, the PUS-C standard is intended to become normative to many ESA projects. The size of this document, together with the number of requirements and tailoring options that it contains can be considered by many, a real difficulty. The formalism that has been used to produce the Standard allows software tools to be developed to support its utilization, e.g.:

- To help in producing on-board and ground software code to work with telecommands and telemetries (access to parameters, encoding and decoding functions, etc.)
- To help in producing documentation showing the layout of the packets.

2 OBJECTIVES AND SCOPE OF THE STUDY

In addition to the consolidation of existing services based on accumulated experience, and the standardisation of new ones, the C version of the packet utilisation standard also brings a complete overhaul of the standard's layout. Indeed, to comply with ECSS regulations, it is now properly expressed in terms of requirements which can be properly traced into the projects which implement this standard.

Three chapters of this new standard are of particular importance for this study: chapter 5, containing the PUS foundation model, chapter 6 expressing the standard services functional requirements, and chapter 8 providing the standard services interface requirements.

The PUS foundation model specifies the generic rules which are applicable to any service, be it standard or mission specific. It ensures the general architectural consistency of each service type.

The standard PUS services functional and interface sections respectively describe the semantic and functional specification of the standardized service types for the former, and the detailed packet structures which implement this specification for the latter.

The PUS foundation model is the primarily focus of this study. The first activity was to consolidate an existing ORM representation of this foundation model, while verifying its completeness and defining all the rules necessary to express the standard services requirements. This consolidated representation was then used to generate a relational database using the NORMA tool. Such a generation ensures that the relational database conforms to the requirements defined by the foundation model, while respecting databases normal forms. Finally, this database was populated with all information related to the standard services currently expressed in the PUS C.

Once created, this database served as the basis for tools capable of simplifying the tailoring and specification of missions using the PUS standard. These tools are able to perform a selection of standard services according to mission needs and to create new mission specific services and/or subservices. They are able to generate two kinds of outputs from the database contents:

- The mission specification, consisting in the tailored packet utilisation document for that mission, including all content of PUS C chapters 6 and 8 relevant to that mission, and extended with the potential mission-specific services;
- An ASN.1 description of the packet structures, which can then be used for further product generation (such as packet encoders and decoders, for example). This step also requires the introduction of the so-called 'PUS implicit knowledge', which is not part of the PUS requirements but still necessary to properly represent the packets in ASN.1 (such as the CCSDS header formats).

While not directly linked to the database software, another aspect of this study was the production of SDL state machines and MSC sequence charts to illustrate selected standard service types. The purpose of this activity was to verify the consistency of the specification (as the modelling in SDL/MSC verifies that the services can be implemented as specified).

Finally, in order to validate the produced tools, a representative mission use case was chosen. The established process for mission tailoring was exercised, and the results were compared the produced documentation to the mission PUS documentation and system specification. This allowed verifying the adequacy of the produced documentation on one side, and assessing the feasibility of the process on the other side.

3 Study Logic

The study consisted of four tasks, outlined below.

Task 1 was decomposed in sub tasks that were performed essentially in sequence.

The definition of the PUS Foundation Model had to be completed before the Relational Data Base Software could be developed. Similarly, the Data Base had to be set up and its Graphical User Interface be developed before it could be populated.

The development of the documentation automatic generation tool was performed as soon as the Data Base was available and in parallel with its feeding.

At the end of the activity the document was automatically generated from the populated data base and it was compared and verified against the published standard.

Task 2 consisted in the development of an ASN.1/ACN data model generator that reused the information of clause 8 captured in the PUS Foundation model in Task1 and completed it with the required ASN.1 tailoring data and the so called PUS implicit knowledge.

From the data model, the corresponding documentation and the encoder and decoder libraries corresponding to the new PUS-C were generated and their adequacy was verified.

Task 3 consisted in describing the state chart and the sequence chart diagrams corresponding to selected PUS Sub Services:

- PUS service 1
- PUS Service 6
- PUS service 12

Task 4 consisted in exercising the tools developed in Task1 and Task 2, in order to tailor the PUS services to a selected use case representative of an existing mission, and to generate the documentation, the data models and the encoders and decoders. The adequacy of these outputs was then verified.

4 CONCLUSIONS

The main conclusions of the study are the following:

- Roughly 25 change requests on PUS-C were raised as a direct result of this activity, including
  - Editorial errors;
  - Consistency errors (highlighted by the automatic generation processes used);
  - Packet definition errors (located during the population activity);
  - Functional errors (Parts of the standard that "do not work" as specified, in particular, service 1 reactions to failure notifications at instruction/request level, identified during MSC modelling).
- The tooling developed during the activity proves the feasibility of the automatic generation of such standards;
- In order to bring these tools to an industrial usage, improvements are necessary in terms of stability and user-friendliness;
- Long-term improvements are proposed in the lessons learned.