PROCESS STANDARDIZATION AS PRELIMINARY STAGE FOR SYSTEM DESIGN AUTOMATION (USING THE EXAMPLE OF ECSS PUS-C)

Lena Stöcken, Stefan Wertheimer
Konzept Informationssysteme GmbH
Meersburg, Germany

August 30, 2018

Abstract

In the last years the terms "automation" and "digitization" came to widespread use in all industries. Through this concepts are neither new nor unknown, the "Industry 4.0" movement made the public aware of some of the initial thoughts behind electronic data processing. But to enable the full potential of the whole digitization approach, the underlying work flows and processes must be well defined and properly applied. Therefore technical norms and standards can help to digitize and automate process flows, since they lay a universal baseline and simplify the cooperation between institutions. Digitizing such a standard means building up a central management for contained data and rules to re-model the process in the digital world. Consequently the work and all emerging results can be stored. Every task done in the digital representation increases the efficiency and savings in work time and costs. This paper will elaborate ideas and thoughts for a future implementation of digitized and automated System Design Processes in Space Projects based on the recommendations of the European Cooperation for Space Standardization (ECSS).

Keywords
Automation, Digitization, ECSS, ESA, PUS, Standardization, System Design

1. APPLICABLE STANDARDS AND THEIR CURRENT USAGE

1.1. Why are standards necessary in aerospace

A high level regarding safety and dependability is necessary to fulfill the high expectations which come along with aerospace projects. This is especially necessary if automation and digitization should be established. The usage of standards is beside complete and complex testing one reliable possibility to guarantee these requirements.

This leads to a increasing usage of standards and also to the generation of completely new standards which become necessary because of the changes in contemporary times (e.g. pseudo-standardization of the CubeSat design pattern through *CalPoly* [1]). Beside the knowledge of the positive effects using standards in the development today, it is already prescribed to use some standards by major public authorities like *European Space Agency (ESA)*. The need to use standards in space projects cannot be seen as a constraint by ESA but as a proper foundation to implement high levels of safety and quality. The efforts to standardize space applications in Europe started in the 1980s (cf. [2] for the early history of Space Packet Utilization) by ESA, but soon it became visible that the task is huge.

In 1993 ESA started an initiative to found the *European Cooperation for Space Standardization (ECSS)* which should be responsible for the establishment of sound and up-to-date industrial standards for the space business in Europe. This was done to improve the quality, the collaboration ability and the competitiveness of the European space business. Some years later in 1996 the importance of such a movement was made clear by the failed first launch of Ariane 5 (V88 on 4th June 1996 with a cause damage of approx. 290 million Euros). In the retrospective it is now clear that project management and engineering failed on many different levels. So the only possible consequence was to improve craftsmanship and quality through better processes and more understanding. Through the years the ECSS enhanced its standards to a very good level and some of the documents could be used for automation purposes. The aspect of digitization was of less importance in the beginning and since only documents were published by the ECSS it was also hard to perform. Secondly many standards provided recommendations and specifications in the field of business processes (e.g. project management, quality assurance), manufacturing and (environmental) testing. Those standards are harder to transform in digital representation as the ones describing already digital aspects.

1.2. The PUS - An example for a space standard

The ECSS was quiet productive in writing and releasing standards and manages around 130 active documents at the moment. The standards reach from mechanical specifications over material analysis to project

doi:10.25967/480297

management. So one can see, that the whole development process is covered in detail by specialized work recommendations in addition to the universal norms released by organizations such as the *International Standards Organization (ISO)* or the *Deutsches Institut für Normung (DIN)* (German Institute for Standardization).

The following chapters will look at a special resort in Space Mission Engineering to limit this wide range down. ECSS has standardized the communication between the Space Segment and the Ground Segment through the so called "Space Packet Utilization". Based on *CCSDS 133.0-B-1* of the *Consultative Committee of Space Data Systems (CCSDS)* [3] it is global practice to use a packet-based approach to structure the data transmitted via the radio up- and downlink. The ESA-internal working group *Committee for Operation and EGSE Standardisation (COES)* made first efforts in 1987, which resulted in a couple of standards that came quickly into use (cf. [2]). Those documents were the baseline when in early 2000s the ECSS took over the task to modernize and harmonize the "Space Packet Utilization" into its growing network of engineering standards. The result was published as *ECSS-E-70-41A Ground systems and operations - Telemetry and telecommand packet utilization* [4] in 2003.

This document became known as "the PUS" (Packet Utilization Standard) and every ESA mission based its Space-to-Ground communication on it creating reams of so-called "Mission PUS" documents (or more officially "Space-to-Ground Interface Control Document"). The PUS was a plain write down of all supported Service Types and Services Subtypes only giving the technical information what shall be done. Since the PUS was by default intended to be tailored, this approach is quiet difficult. Without mentioning the reasons for some implementation recommendation by the ECSS, only experienced engineers can make reliable decisions for the tailoring. Defining the communication specification for a space craft was handwork and error prone. Many review cycles had to guarantee completeness and correctness.

Now this changed in 2016 when the ECSS released a new version of *ECSS-EST-70-41C Space engineering – Telemetry and telecommand packet utilization* [5]. After 13 years of usage the shortcomings and gaps became obvious. It was realized, that the structure of the old document was odd compared to all the other standards. Most of the other ECSS releases were divided in one part specification of requirements and another part re-commendations for application of the requirements. So the new PUS was re-structured in the same way, allowing the authors to explain hows and whys. At the same time cross-linking of the different packets that were connected to each other, was clarified. It is this new structure and the indirect mentioning of the intention to support digitization efforts from now.

1.3. Usage of the PUS and its references

The now introduced standards are some which are used in nearly every active space project and are based on completely different concepts of how they should be used. The first one is standard *CCSDS 133.0-B-1* [3].

Its active version was established in 2003. This standard defines the structure of packets which are transmitted between ground station and satellite. Each packet consists of following parts:

- Header with fixed length, fixed fields and partly fixed data
- Datafield which variable length and variable data
- Trailer

The header indicates the beginning of a new packet and defines its complete length. With the usage of this standard no discussions about topics like marking begin and end of transferred packets, where to specify the length, how to mark if the packet is uplink or downlink and so on can arise. When using this standard it is necessary to keep everything exactly like it is defined within it and not changing some definitions. If anything would be changed all advantages which come along with its usage would be gone. This is the reason *CCSDS 133.0-B-1* is defined completely applicable for all missions using it. The second introduced standard is taking a different way in how it should be used. The PUS (*ECSS-E-70-41A* [4] respectively *ECSS-E-70-41C* [5]) managed by the *European Cooperation for Space Standardization (ECSS)* elaborates the usage of the packets defined in the CCSDS Space Packet standard. It describes TC and TM packets with their content consisting of small datafields. A PUS packet is transmitted within the datafield of a space packet.

The current version is the C-Version (PUS-C) which was released in April 2016. Because it is not already available for a long time most currently active space projects still use the A-Version (PUS-A).

As against the CCSDS Space Packet standard the PUS allows the tailoring of all definitions within itself. The allowance of the tailoring is because not all space missions need the same communication between ground station and satellite. If all missions must use the same TMs and TCs both cases are probable:

Definition of too many messages

> The mission would define a lot of messages which will never be used. But if the mission must fit to the standard all messages must be implemented by the ground station and by the satellite. This would increase the development time and costs.

Necessary messages are missing

> It can happen that the mission needs special messages to fulfill the needed data transfer for special equipment. It can also be possible to evade these missing parts for example by using memory load and read functionalities. But this would increase the complexity which will also lead to the increasing of development time and costs.

During the whole tailoring procedure the creation of a consistent and valid result must be kept in sight. This was of very high importance with PUS-A, since some crosslinked requirements were not expressed directly and the knowledge of the system engineer performing the task was in high demand not to miss messages or datafield items.

An implicit process chain available on many satellites shall be used as an example for the work to be performed and the pitfalls to be evaded: Complex automatic reaction to on-board events.

There are three to four Service Types involved in this scenario:

- ST[12] On-board Monitoring
- ST[5] Event Reporting
- ST[19] Event-Action
- ST[18] On-board Control Procedure

The forth Service Type is optional for real extensive actions, so it is omitted in the following diagram showing the interlinking.

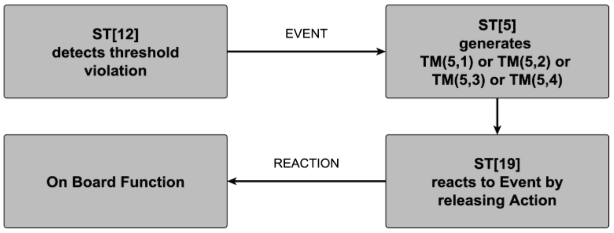

Figure 1: Interlinking of ST[5], ST[12] and ST[19]

Not only must every Service Type be defined consistent and valid by itself, but in this use case all required Service Types must be interoperable to form the process chain that allows the satellite to react autonomously to events. Taking a safety action in case of an anomaly as example, ST[12] has to be correctly defined and configured to detect some on-board parameter going out-of-range. The "On-board Monitoring" has to generate an event consistent with the adapted version of ST[5]. This event will be sent to ground, if defined and configured in the Mission PUS, and to the ST[19], the "Event-Action" Service Type. This component has now the duty to trigger an appropriate action, which has been set by the mission operator using the tailored TMs and TCs of the Service Type. If the to-be-performed action is too complex, the usage of an "On-board Control Procedure" will be required and ST[18] steps in as well. This is one of the most complex Service Types specified in the PUS since its functionality defines an on-board command interpreter that is able to write, store and execute scripts - in the current scenario even without human interaction.

If one takes into account that every service is quiet complex of its own through the definition of many interlinked or interlink-able packets, the combination of those four mentioned Service Types into a process chain requires sound experience and extensive peer review to create a safe and reliable implementation. As it can be seen now, the tailoring of the PUS for the usage of a mission can take a long time and many people interacting with each other until the "Space-to-Ground Interface Control Document (ICD)" for the mission is ready.

1.4. Improvements in PUS-C

As mentioned above, the PUS-A was a plain write down of all supported Service Types and Services Subtypes. The new PUS-C is now going one step further by allowing the tailoring and giving implementation recommendations based on the so-called PUS Foundation Model. This leads to more flexibility, more unique communication definitions for the single mission and the possibility to tailor the PUS also for not so experienced engineers, but will also increase the complexity and time for the tailoring, especially for the first missions which will be based on PUS-C.

Here the idea of digitization and automation comes into account. The digitization of PUS-C makes perfect sense after only looking on the page count: It has grown from approx. 220 pages (PUS-A) to over 660 pages of normative and informative material. Systems Engineers need to read through the whole document and understand the underlying concepts. Afterwards the new ways of doing tasks should be applied in the beginning to not lose any advantage through implementing things in the old, now outdated manner. To avoid the looming increase in the required learning and tailoring time automatic, digital tools implementing the PUS Foundation Model and assisting in the tailoring steps will become very helpful and be needed in the near future. Those tools can also perform the time consuming review cycles needed by completely manual processes.

The PUS Foundation Model is intended as a layered modeling approach (think of an onion) with the outer layer adaptable and definable by the end users of the standard (i.e. the operators and architects of a space mission).

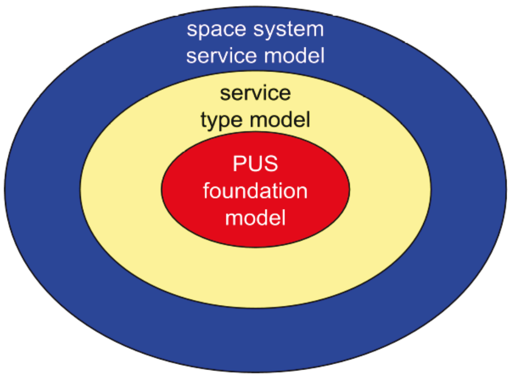

Figure 2: PUS Model

The core addresses generic and abstract definitions of the Service Type Model known from the old PUS-A. As an improvement there now shall be an indicator whether any Service Type is standardized or mission-specific. The same is applicable for the Service Subtypes. The PUS Foundation Model describes all requirements met by its predefined objects and makes them mandatory in use for all user-defined mission-specific objects as well. Through this rigorous approach in stipulating the requirements the whole model can easily be formalized in a digitized way.

This has been performed by ECTEC that already transferred the content of the standard into an *Object Role Modelling (ORM)* repository that can be used by the *Natural ORM Architect (NORMA)* tool or any other software through exports and data transformation. In combination with an already existing macro-based Microsoft Word template written by ESTEC, it is possible to tailor a Space-to-Ground ICD and generate a document for further usage. The thoughts of the ESA engineers did not stop here. Consequently they made recommendations to use telecommunication specification languages like the *Specification and Description Language (SDL)* [6] and the *Abstract Syntax Notation One (ASN.1)* [7] to further define the data packets automatically. This rudimentary tool set and the desired approaches were compiled into an Invitation to Tender in early 2016 [10] getting the industry involved in the digitization and automation of the PUS.

2. REASONS FOR AUTOMATION

One can say, Modern Times started in the late years of the 18th century. Primly in Great Britain more and more craft production companies invested in machinery to support human labor. Through the increasing use of machinery the demands of different raw materials and consumables lead to the creation of whole new economic sectors. Needless to say a change in society followed and nowadays this historic period is called the *Industrial Revolution*. The increased use of machines and automates lead to an increase in production output, since the machinery could produce the goods faster and independent from the human daily routine. Less men power was required to keep the factories running as if the production would have been performed manually. Through the years the industrialization proceeded: The spiral in production increase, changes in society and invention of better and more efficient machinery lead to so called "Second Industrial Revolution". It profited from electrification and mass production (cf. Taylorism and Fordism). The invention and progressive development of micro-electronics beginning in the 1970s formed the tacit cornerstone for the 3rd revolution in industrialization. With the emerging (personal) computers many manual management tasks formerly performed by scribes and typists started to be performed with the new tools of trade. The "Electronic Data Processing" had the same effect to administrative, creative and intellectual tasks as the machinery on the factory floor had to manual labor.

The broadening of the “new ways of work” over the last decades – if not say centuries – lead to a re-thinking of the established processes: Why could the automation not be completely combined with the digitization? The term *Industry 4.0* originates from a high-tech strategy project of the German government. The final report on the thoughts of the governmental working group was first presented at the Hannover Fair in 2013 [8] and the quintessence was push forward on the international stage through the World Economic Forum 2015 and 2016 in Davos.

With global players sparking the flame automation and digitization are nowadays omnipresent. So it is no surprise to adapt those concepts on satellite production as well. Big companies as *Airbus Defence and Space* or *Lockheed Martin* talk about “satellite factories”, but it is interesting and important for smaller enterprises as well. Here the advantages in form of cost and time reduction are even of bigger impact compared to the big players since the fewer engineers performing the work can increase their productivity through automated and assisted processes.

Recent studies show that the increase of automation by using computer systems could supersede every second job [9]. In case of system design those jobs transferred to computer programs will mostly be the unimaginative but fault-prone tasks. Outsourcing those tasks to an automated system that does not suffer of tiredness or lack of concentration enhances the quality of the results. As an example the review processes of textual documents by human coworkers can be eliminated through an automated system. If the standardization of processes is rigorously pursued by organizations, this effect will spread through the whole system design task since well-defined requirements can be fed into the computer systems. Taking the thought even a step further will lead to machine leaning and artificial intelligence, so the system can make its own decisions in the near future. At this stage the system can assist the system engineer even more and take over simple tasks, so he or she can handle the demanding tasks.

3. MORE EFFICIENT WAYS TO WORK

Beside any automation effort which increases the working efficiency, standards contain many small other points leading to an improvement of the status quo. Firstly digitization has to be mentioned independently from automation - which otherwise often goes hand by hand. A digitized standard can be available from all working places where it might be required. It also helps to simplify searching and indexing of its content. If provided by the organization, the people working with it can check something again and again much easier and faster in a central digitized version than heading over to the library (or even searching for it on file servers or intranet places). The centralized and omnipresent approach will decrease the required lead time freeing it for other work tasks. This whole thought might sound profane nowadays since PDF versions of many standards are available, but even within those documents some limitation in the field of searching and indexing can be found.

All other points mostly address the human aspects of working with standards. This includes the personal working behavior and the comfort gain of the working people.

Increasing the knowledge about the standard

The more the people working with a standard know about it, the easier their work with it gets. Well-founded and usable output can be expected can only be expected, if everyone working with a standard has knowledge about its content. Professional training can be used to acquire at least this knowledge. Afterwards knowledge exchange with colleagues about different topics of a standard can be helpful to deepen the learned basics. This knowledge transfer works best in smaller and mixed teams so everybody can learn from the other ones. It is important that no one in this group is seen or acts as the "omnipotent expert", who knows everything. It can always happen that a person working with the same standards and tools for years can learn something new triggered by a different point of view of another team member.

Providing short and helpful information

It cannot be expected, that everybody in a project team knows everything from a complex standard like the PUS-C with its over 600 pages of content. So people have to re-read some topics within the standard, while working with it. If it is necessary to look around these 600 pages every time to find just the small information that was missing, work gets protracted and frustrating. This will not increase the efficiency of the team, but definitely decrease it. The extraction off all basic and often looked up information can be used to invert this problem into an advantage that could lead to higher efficiency. This extracted information should be available in a digital way (like in the best case the complete standard) and also in an analog way like for example as a poster in the office of the project team. If automation is already in use, it can also be helpful to provide the important information and even some special cases within the used automation tool.

Using standards in daily working life

It is unnecessary to say that satisfied employees are more effective than discontented ones. If working with standards is seen as something disruptive, which costs too much time and energy, it will be done with less vigor and conviction than other tasks. So it is necessary to recognize the standards as part of the daily working routine. It has to be applicable for all people on all operational levels and not as something that is prescribed by the supervisors - or even worse as "black magic" kept by the system experts in locked bookshelves. Everyone on the team should have the possibility to look up the details in standards, when he or she feels the need.

4. A PROCESS AUTOMATION EXAMPLE: PERIGEE

PERIGEE is a self-developed software application by *Konzept Informationssysteme GmbH* and has been worked on over the past two years. The initial idea was derived from two things: First the ever-present carving for a centralized and standardized tool by colleagues working on Space Missions in different project stages, and second an Invitation to Tender issued by the *European Space Research and Technology Centre (ESTEC)* early in 2016 [10]. With the new version of the PUS (*ECSS-E-ST-70-41C* [5]) to be released in April of this year, ESTEC wanted to kick-off a study on new and efficient ways to work with the new standard before it was even applied in the first project. Konzept Informationssysteme GmbH delivered a proposal that was unfortunately not pursued by ESTEC. Soon after, the cornerstones of the proposal were interrelated with the ideas of some colleagues (combining several decades of space project knowledge) and the know-how in software development as well as UI design. A vision was formed and it is lived up to now: Ease the way how Space Missions are designed and managed for all kinds of players and take into account the advancing digitization. Through digitization and centralization of data, the processes involved will be qualitatively improved and accelerated. The first version of PERIGEE was released in December 2017 and it builds the foundation for the planned application family.

So how would a satellite development process with PERIGEE look like and what can be done right now out-of-the-box?

First of all the initial product version lets you tailor the communication protocol between Space Segment and Ground Segment and each component on-board the Space Segment communicating through the PUS. The application fully supports the new version PUS-C and lets the user base a new "Mission" on it. The term "Mission" is synonymous with project if working with PERIGEE, but only some aspects are covered other than a real complete Space Mission. After defining some basic parameters of the new project, a so-called PUS Architect may tailor the PUS to the needs of the satellite’s mission. Therefore an intuitive and well-arranged UI is used facilitate by modern and well-known operational concepts (e.g. drag-and-drop for Packet Data Field manipulation).

Throughout the whole process the application provides first-hand information and hints on the meaning and usage of all displayed elements as well as direct cross references into the underlying standard, if the original text as required for consultation. If new telemetry or telecommand packets have to be defined, there is the possibility to define mission-specific packets from scratch. The basic requirements on PUS packets are enforced to assess compatibility and conformity. Afterwards the tailored PUS can be applied and broken down onto "Application Processes", which will perform the required functionality in the future space mission. All these steps are guided and assisted by hints and helping texts as well as automated checks minimizing unintended deviations from the standard. All made deviation are fully traceable in the resulting data set and have to be justified by the user.

After the tailoring process is finished, the resulting mission-specific PUS can be altered at any time and through the integration of a configuration management system in the application the version control and tractability of changes is secured. A document generator is integrated as well to allow the provision of up-to-date versions of the resulting Space-To-Ground ICD in the classical form of a document. This document can be generated any time and conforms always to the requirements and recommendations of ECSS-E-ST-70-41C.

In future releases PERIGEE will allow the deviation of "real" telemetry and telecommand packets out of the specifications made in the Space-To-Ground ICD. It will take into account all knowledge and parametrization deposited for the satellite and its Application Processes. This will even enable the automatic generation of software source code for packet encoders and decoders. The generated source code will follow fundamental quality guidelines provided by other ECSS norms (e.g. *ECSS-E-ST-40C* [11] and *ECSS-Q-ST-80C* [12]) and be paired with the corresponding test cases for verification. So the developer only needs to integrate the generated code into the corresponding software modules which need the encoder or decoder and does not need to worry about conformity and code quality. Another future improvement will take this automation possibility even further:

The integration of a simple cable harness editor will allow the user to define the interconnection of satellite sub-systems. The emerging information can not only be used to create parts lists for electrical manufacturing, but to generate software drivers were necessary as well. Corresponding tests can be derived from the stored data, too, further increasing the quality of the overall system design.

PERIGEE is aimed to support any size of organization in the field of satellite system design and will be improved an extended on user feedback.

REFERENCES

[1] California Polytechnic State University (2014) CubeSat Design Specification Rev. 13

[2] Kaufeler, J.-F.; Parkes, A.; Pidgeon, A. (1992) The European Space Agency Standard for Space Packet Utilization

[3] CCSDS Secretariat (2003). CCSDS 133.0-B-1 “Space Packet Protocol (Blue Book)”

[4] ECSS Secretariat (2003) ECSS-E-70-41A “Space engineering: Ground systems and operations - Telemetry and telecommand packet utilization”

[5] ECSS Secretariat (2016) ECSS-E-ST-70-41C “Space engineering: Telemetry and telecommand packet utilization”

[6] Telecommunication Standardization Sector of ITU (2016) Z.100 “Specification and Description Language”

[7] Telecommunication Standardization Sector of ITU (2018) X.680 “Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation“

[8] Kagermann, H. et al. (2013) Abschlussbericht des Arbeitskreises Industrie 4.0

[9] Frey, C. B.; Osborne, M. A. (2013) The future of employment: How susceptible are jobs to computerisation?

[10] European Space Research and Technology Centre (ESTEC) (2016) Appendix 1 to ESA ITT AO/1-8446/16/NL/FE - Statement of Work: Deployment of the PUS-C standard in projects supported by an automatic generation toolset

[11] ECSS Secretariat (2009) ECSS-E-ST-40C “Space engineering: Software

[12] ECSS Secretariat (2017) ECSS-Q-ST-80C Rev.1 “Space product assurance: Software product assurance”

LIST OF FIGURES

CONTACT INFORMATION

Dipl.-Ing. (BA) Stefan Wertheimer

Konzept Informationssysteme GmbH
Am Weiher 13, 88709 Meersburg, Germany
E-Mail: stefan.wertheimer@konzept-is.de