Description of Needs for Automatic Service 1 Verification

Responsibility	
Written by	
Eduardo Pacheco (Critical Software SA)	Avionics IVVQ Engineer
Reviewed by	
Romaric Baudot	Avionics IVVQ Engineer
Fabrice Wallon	Avionics IVVQ Engineer
Approved by	
Philippe Cam	Head of Avionics IVVQ

CHANGE RECORDS

ISSUE	DATE	§ CHANGE RECORDS	AUTHOR
01	22/04/2016	First issue	E. Pacheco

TABLE OF CONTENTS

1. SCOPE AND APPLICABILITY

1.1 Scope

The purpose of this document is to describe the methodology to apply on the SC simulators or PFM interfaces in order to implement automatic verifications of TC acceptance TM reports.

1.2 Applicability

This document is applicable to any simulator or PFM interface where automatic verifications of TC acceptance TM reports are needed.

2. DOCUMENTS

	Title	Issue	Reference
[AD-1]	Generic Interface Control Document / PUS	15	100216228H

3. GENERAL DESCRIPTION

The goal is to achieve fully automatic validation of TC acknowledgments received during a sequence. All the information provided in the document will be based in the PUS standard (cf. **[AD-1]**) but it can also the extrapolated to most non-PUS missions has the functionalities are quite similar.

3.1 PUS TC acknowledgment

PUS standard uses service 1 (PUS SVC 1) for TC acknowledgment purposes.

The standard previews 4 different level of TC acknowledgment:

- TC Acceptance
- TC Execution Start
- TC Execution Progress
- TC Execution Complete

Although standard defines 4 levels of acknowledgment, each PUS instantiation (cf. **3.4**) will specify the applicability of each one of them for each mission/program.

It is also possible that in the same system different components/equipment's can apply different levels of acknowledgement, i.e., we can have a system where the CSW implements TC acceptance, execution start and execution complete reports and one PUS compatible equipment in that system that implements only TC acceptance and execution complete reports. This is mission dependant so it as to be analysed for each instantiation.

These levels have then the generation of the associated specific telemetry according to success/failure status of each level:

SVC/sub SVC	Description
TM(1,1)	TC acceptance report - success
TM(1,2)	TC acceptance report - failure
TM(1,3)	TC execution started report - success
TM(1,4)	TC execution started report - failure
TM(1,5)	TC execution progress report - success
TM(1,6)	TC execution progress report - failure
TM(1,7)	TC execution complete report - success
TM(1,8)	TC execution complete report - failure

Table 1 - TC ACK List

Moreover, TC acknowledgment is dependent on request. Each TC contains the TC ACK field in the data field header that indicates the requested of acknowledgments for that command. An example from PUS generic ICD:

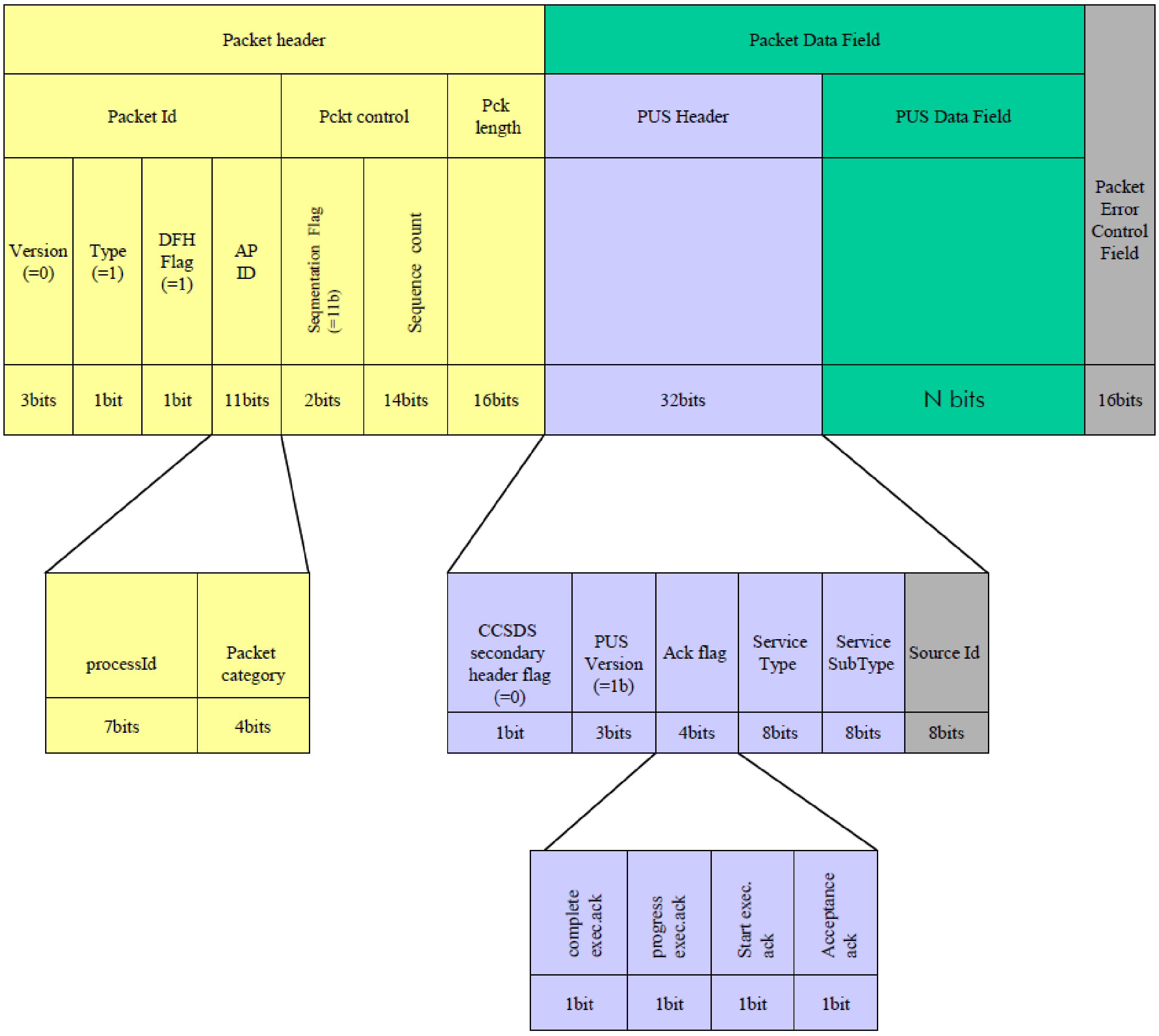

Figure 1 - Structure of PUS TC Header

Note that on each instantiation (cf. ***3.4****) the size and meaning of these fields can change. For each case always refer to mission/program specific PUS ICD.*

In resume, for each command the expected number of acknowledgment telemetries is dependent of:

- The PUS instantiation (cf. **3.4**) of the system/component
- The requested ACK on the command (TC ACK field)

3.2 TC to TM mapping

Any TC acknowledgment telemetry must contain information sufficient to uniquely identify the TC that generated it, examples:

- APID
- Destination ID
- TC Sequence counter
- TC Echo – Used for example in ExoMars mission which basically contains the first bytes of the TC (including for example TC date).

TC rejections (PUS sub-SVCs 2, 4, 6, and 8) must also contain at least a rejection reason, possibly can have other parameters to help ground better understand the rejection cause. This parameterization is reason dependent.

3.3 TC Origin

Not only ground commands can generate PUS SVC 1 telemetries. In generic PUS ICD we can find several OnBoard services that execute telecommands which also generate these telemetries. A list of PUS predefined TC origins:

Source	Description or SVC
Ground TC	TC received from ground (except HPTC)*
MTL	PUS svc 11
OBCP	PUS svc 18
EA	PUS svc 19
OBCP	PUS svc 131
AcSeq	PUS svc 132

** - HPTC (also called direct TCs) are HW commands, has these TCs are not treated by the CSW they do not generate PUS SVC 1 telemetries*

Note: these are services defined in generic PUS ICD. Each instantiation (cf. **3.4**) of the PUS STD will define the applicability of these services. Also new services can also be added depending on the DHS requirements. Moreover PUS compatible equipment’s can also add new cases to this list (cf. **3.5**).

In a PUS system each of the applicable origin must have a unique identifier (Source ID) present in TC Header (cf. **Figure 1**). This is them mapped to the PUS SVC 1 telemetries in a field called destination ID. An example from PUS generic ICD:

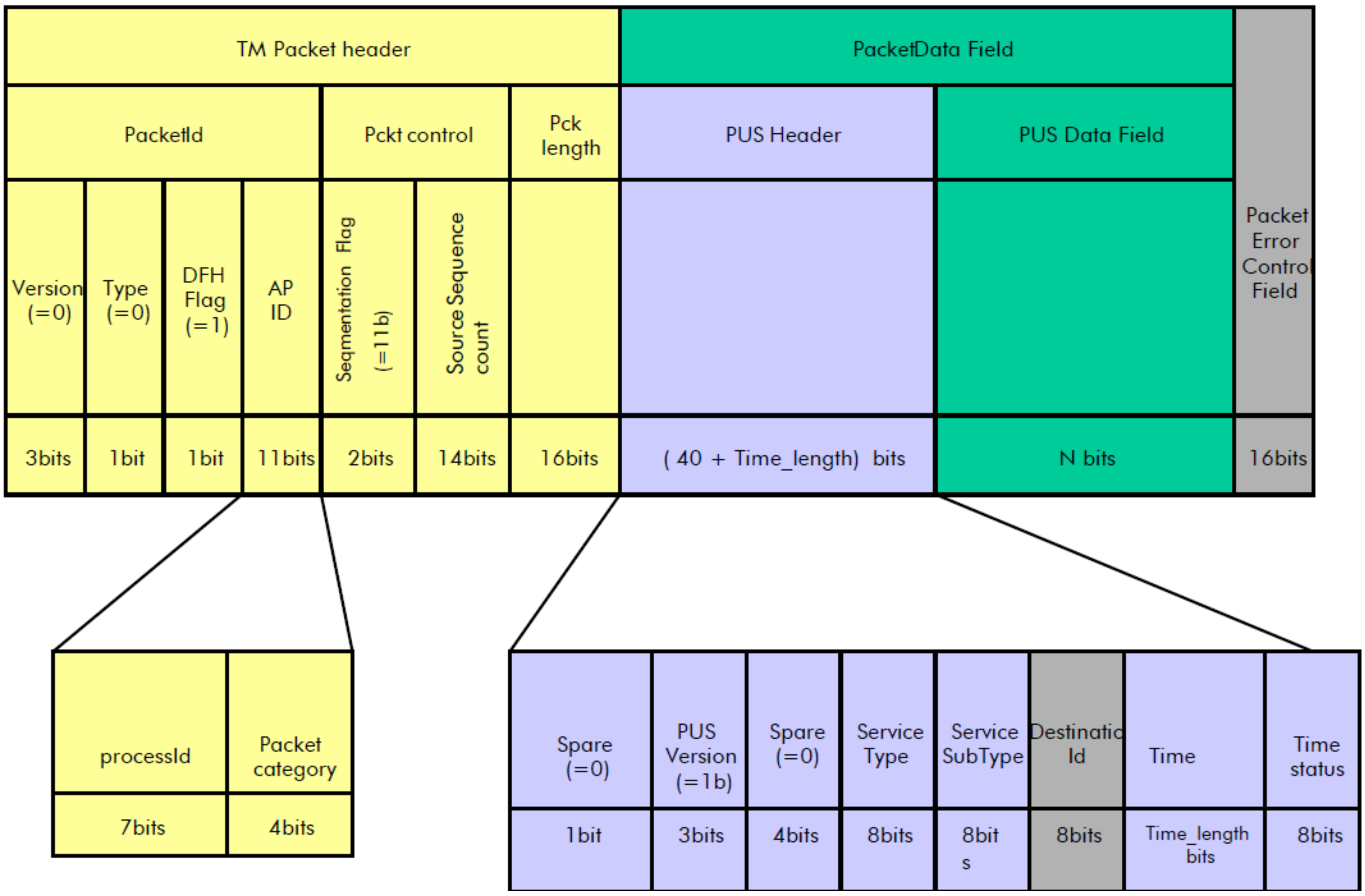

Figure 2 - Structure of PUS TM Header

Note that on each instantiation (cf. ***3.4****) the size and meaning of these fields can change. For each case always refer to specific PUS ICD.*

3.4 PUS Instantiation

Each PUS compatible system implements a specific PUS instantiation. This is composed by the complete set (or sub-set) of PUS generic services, plus a set of program and/or mission specific services.

3.5 PUS compatible equipment’s

Some equipment’s integrated in the system can be PUS compatible. These equipment’s implement their own instantiation of PUS standard. In these cases CSW only provides an interface ground/equipment were it routes commands/telemetries to/from these equipment’s, has equipment’s are able to treat the commands and generate telemetry packets in PUS format.

4. AUTOMATIC VERIFICATION OF TC ACKNOWLEDGEMENTS

In this chapter it will be provided a description of the needs, constraints, problems and possible solutions that can be found while building a tool to achieve the proposed goal.

4.1 Objectives

For each TC executed a set of acknowledgment telemetries should be received. The expected set depends on:

- The PUS instantiation (cf. **3.4**) of the system/component
- The requested ACK on the command (TC ACK field)

4.2 Ground TC

By ground TC it is considered any TC sent from ground executed without delay, except for cases of bufferization at TC reception.
The automatic verification of the acknowledgments of these TC should be straight forward has they are received just after TC is sent. Just a small warning for the services that can take long time to finish (seconds, minutes or even hours), in these cases a TC execution complete can take some time to be emitted. This is also applicable for OnBoard TCs.

Note: As an implementation decision in some cases for the longest duration services to avoid long term bufferization of services, CSW can emit the TC execution complete just after TC starts the execution.

4.3 OnBoard TC

By OnBoard TC it is considered any TC which execution is triggered OnBoard (MTL, AcSeq, etc…).

4.3.1 MTL TC

MTL TC is basically a delayed execution TC (time tagged TC). It is loaded associated to an OBT due date, at which moment (*) the TC is executed. These TCs will generate two sets of acknowledgments to verify:

- The TC to load MTL: This is a ground TC PUS SVC 11.
- The TC loaded: It is carried inside the TC PUS SVC 11.

(*) – MTL does not have to be executed at the exact moment OBT=due date. On each system a small window is defined around the due date on which the TC can be executed. Usually if for any reason this window is passed and the TC is not executed, it is considered obsolete and discarded (this behaviour and size of the window are mission dependant).

To verify the correct acknowledgment of the loaded TCs, it must not only verify that the set of PUS SVC 1 telemetries are generated but also the correct date is respected. Be aware of:

Destination ID: usually when building these TCs, tools use the default SDB data without changing Source ID parameter, in this cases the MTL TC destination ID will be ground as it is by default in the SDB. This can cause problems to later identify the origin of the TC. It is advisable that the tool that builds the TC, overwrites the Source ID value of the loaded TC to identify it as a MTL TC (PUS SVC 11 TC to load an MTL should remain as a ground TC as the execution is upon reception).

Due Date: Verify that an MTL TC is executed at the correct date can cause some issues as it forces the analysis tool to map the OBT with the bench date (cf. **4.5**). Moreover it as to maintain a bookkeeping of MTL TCs loaded with the associated date in order to perform this task.

Lost TCs: In some cases a MTL TC can be lost or discarded. An event can cause deletion of all or a sub-set the MTL TCs loaded (this is always an implementation decision, so each mission can have a different behaviour here). This means that in some cases not all loaded TCs will be executed and still the results will be correct.

Shifted due date: There are particular cases were internal or external mechanisms can shift a MTL due date. This is a very extraordinary scenario, possibility of validation of these should be analysed individually at mission level. If the scenario exists and how/if it can be automatically verified.

4.3.2 Action Sequence TC

An Action Sequence (AcSeq) is basically a set of TCs and Waits that can be triggered by ground or by an internal event.

TC execution from Action Sequences will also generate acknowledgments.
In order to correctly verify the acknowledgments the waits should also be taken in consideration.

4.3.2.1 *Inputs for validation*

Inputs for this validation cannot be taken from test script (except for cases described in **4.3.2.6**) like the cases described above as action sequences are included in the binary. The content of the action sequences can be retrieved from PAR_EVENT.DAT (default SDB file).

4.3.2.2 *AcSeq start/stop signalling*

As in most cases AcSeqs are triggered by internal events, predict when they will exactly execute is very hard if not impossible.
Automation in these cases should be based in the AcSeq Start/Stop signalling. This is basically an asynchronous telemetry that indicates start and end of an event sequence (example PUS SVC 5, but not mandatory to be this SVC). This telemetry also indicates the AcSeq ID that triggered it.
A small note here to remark that as acknowledgment telemetries and AcSeq event can managed by different tasks inside the CSW, the 1st ACK of the 1st TC in an AcSeq can arrive before (in the CADU) the AcSeq start signal, but usually with the same OBT (generation date).

4.3.2.3 *AcSeq chaining*

Some missions can allow chaining of AcSeq (AcSeqs that execute other AcSeqs). This is always mission dependent. In these cases the sub-AcSeq will also generate the Start/Stop signalling.
Maximum AcSeq chaining level is defined at the mission level.

4.3.2.4 *Conditional AcSeqs*

AcSeqs have usually an enabled/disabled status that allows/inhibits their execution upon occurrence of the event that should trigger it (TC or internal event).
So called conditional AcSeqs are chained AcSeqs that execute or not depending on the enabled/disabled status. This feature provides a higher flexibility and reuse of AcSeqs throughout mission life time, allowing the AcSeq sequence to change according to specific temporary mission requirements.

4.3.2.5 *AcSeq waits*

In order to verify the correct execution of an AcSeq the waits between each TC should also be verified. The value of the wait is defined in inside the AcSeq and can be retrieved from PAR_EVENT.DAT file.

The verification of this wait can be easily done by comparing the delta between two consecutive TC acceptance reports. For this purpose the TM packet OBT date should be used, this will workaround any delay in TM reception caused by TME bufferization.

It is important to use TC acceptance report for this and not another layer of acknowledgment due to services execution time (cf. **4.2**). Note that for implementation decisions this can change, if for example it is decided that no TC execution starts before the previous as finished. This has to be analysed at mission level.

Also for each mission an acceptance window must be defined, the wait might not be fully precise due to scheduler constraints, OBT precision, etc.

4.3.2.6 *AcSeq Creation/Modification*

Although AcSeqs are included in the binary, it is always possible to add new, remove or modify existing AcSeqs. So a feature that allows user to modify expected AcSeq could be interesting.

4.3.2.7 *Execution of TCs from other origin during AcSeq execution*

In some missions the architecture can allow execution of TC from different sources while an AcSeq is being executed. This is mission dependant and should be analysed at each case.

4.3.2.8 *Destination ID*

As for the MTL (cf. **4.3.1**) Destination ID parameter is important to identify that the acknowledgment was originated by a TC executed in an AcSeq.

Also as the MTL the AcSeq generation tool might use default SDB in order to create the TCs. An effort at DHS level should be done here in order to ensure that AcSeq TC Source ID is overwritten to correctly identify his origin.

4.3.3 Event Action TC

Event Action (EA) are single events that can be an execution of one TC on which case will generate the normal acknowledgment.
These are triggered by internal events reported in PUS SVC 5 which can be uniquely identified by his report ID (RID). Each EA is associated to a specific pairing of APID+RID and will trigger only when an internal event occurs that fits this pairing.

4.3.3.1 *Inputs for validation*

Inputs for this validation cannot be taken from test script (except for cases described in **4.3.3.3**) like the cases described for the AcSeqs, EA are included in the binary. The content of the EA can be retrieved from PAR_EA.DAT (default SDB file).

4.3.3.2 *EA execution signalling*

The execution of an EA is triggered by an internal event. This events are reported to ground in a PUS SVC 5 TM packet with a specific RID. This packet can be used as a signal to know when the EA should occur.

Note: as for the AcSeqs, the EAs have an enabled/disabled status.

4.3.3.3 *EA Creation/Modification*

Similar to what was described for the AcSeqs in **4.3.2.6**.

4.3.3.4 *Destination ID*

Similar to what was described for the AcSeqs in **4.3.2.8**.

4.3.4 OBCP

Missing information about these cases. Need to be done in the future.

4.3.5 OPS

Missing information about these cases. Need to be done in the future.

4.3.6 TC Files

As an example of a mission specific case not previewed in generic PUS STD can be TC Files used in ExoMars mission.

TC files are basically a set of commands uploaded into SC memory (MM or SGM). These are uploaded using large data transfer (LDT) PUS features (PUS SVC 11).

The execution of TC File is then triggered by TC request or by an internal event.

The validation of the ACK of the TCs executed from a TC file has a similar set of constraints as the ones presented for AcSeqs (cf. **4.3.2**).

4.3.7 Chaining of different services and features

It is important to refer that all of the different TC sources presented above can also be chained. In an extreme example we could imagine a case where an Event Action triggers the execution of a TC file, which contains a TC to execute an AcSeq which contains several load MTL TCs.

4.4 Mass Memory Stored Acknowledgments

In some missions, the asynchronous telemetries can be stored in the mass memory (MM). This may include the acknowledgment telemetries. So upon dumping the content of the MM the acknowledge telemetries will be duplicated with a reception date much older than the generation date (in opposite to direct telemetry).

In these cases it is important to ensure that these are all correct. Moreover it is important to understand these are acknowledgments that could be received in duplicate.

For this it is important to keep in mind the following points:

Not all ACK are in MM: As these are stored in MM, not all ACK telemetries of a test sequence are in the MM. MM storage starts after MM initialization meaning that first ACK telemetries received will never be stored. Moreover only ACK generated until the dump moment will show here.

Differentiating them: As MM downlink normally uses a different virtual channel (VC) for their telemetry than the one used for direct TM, the differentiation can be done at CADU level identifying the VC of origin.

Loss of direct TM: These features are usually associated to missions where the loss of direct telemetry is frequent (due to large periods of ground station blinding, specific mission payload that requires SC pointing which can interfere with communications pointing requirements, etc…). In these cases the ACK TM is not be in duplicate. Moreover, the telemetry stored in MM can and should be used to confirm the TC was correctly acknowledged.

In order to correctly analyse the ACK TM packets dump from the MM it is mandatory to use TM packet OBT date (generation date) and not ground reception date (cf. **4.5**).

4.5 OBT and Bench Date Mapping

As most current analysis tool use bench date for their analysis and considering all that was presented above in this document the mapping between bench date and OBT is an important requirement.
The most important point to be aware and perhaps the most challenging of implementing this mapping is the fact that OBT might not be linear during a sequence. Generic PUS STD introduces “Time management service” (SVC 9). This allows testers to modify the OBT value upon a TC request.

This feature is not very common in flight, but it is extensively used in tests scenarios. It is normally done once per test to set the OBT to a value similar to flight date scenario under test. This causes a discontinuity in the OBT timeline. This discontinuity can be forward or backward. For example in the ATB (using a SMU FUMO) the TTRM board is continuously working even if no test is running, this means that if two tests are launch consecutively (without an electrical reboot in between) the 2nd test initial OBT value will be a continuation of the value of the end 1st test.

5. WHAT IS EXPECTED

It is expected to have a listing of all TCs executed with:

- Compete TC with all parameterization (decomuted if possible)
- Execution date
- List should be grouped by origin (ground, acseq, …)

For each TC, to have a list of the ACK received:

- Direct/MM TM ACK
- Reception and generation dates
- In case of rejection:
  - Rejection reason
  - All parameters whenever applicable

6. FINAL NOTE

As stated at the beginning the goal is to have a full automatic verification of TC ACK telemetries. Considering all the details and constraints presented throughout this document it is easily understandable that a fully automatic validation for every case can be hard, if not impossible to do in some cases. Moreover, it will never be a tool that will be generic and applicable for everyone has his, instead it will always require an analysis and update done for each mission.

It is also relevant to state that the kind of constraints detailed throughout this document can evolve during the development cycle of a mission. Initial strategies can change to correct or improve system behaviour.

END OF DOCUMENT