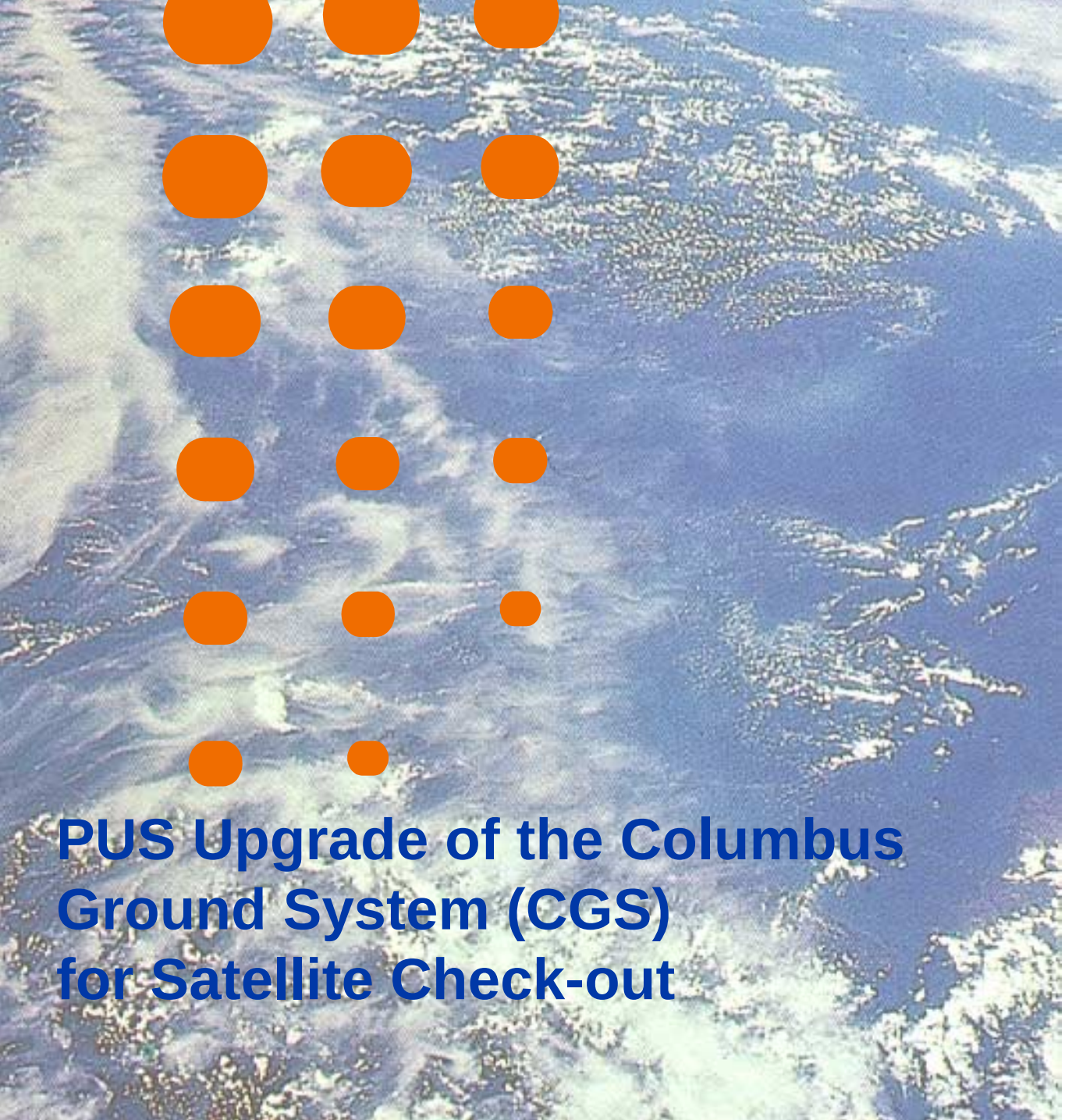


PUS Upgrade of the Columbus Ground System (CGS) for Satellite Check-out

astrium

an EADS joint Company with BAE SYSTEMS

Jürgen Hummel
Space Infrastructure
Astrium Bremen

Thomas Schuler
Earth Observation & Science
Astrium Friedrichshafen

February 2003

CGS/PUS Heritage

The Columbus Ground Software (CGS) toolset was developed as the support system for the whole system life cycle of the Columbus starting with requirements engineering through early simulation of non existing hardware, design, implementation, system integration and verification up to the operational phase

Today CGS supports applications in various projects such as International Space Station (ISS), Columbus module , ATV (Automated Transfer Vehicle) and satellite projects (MetOp, CryoSat and GOCE)

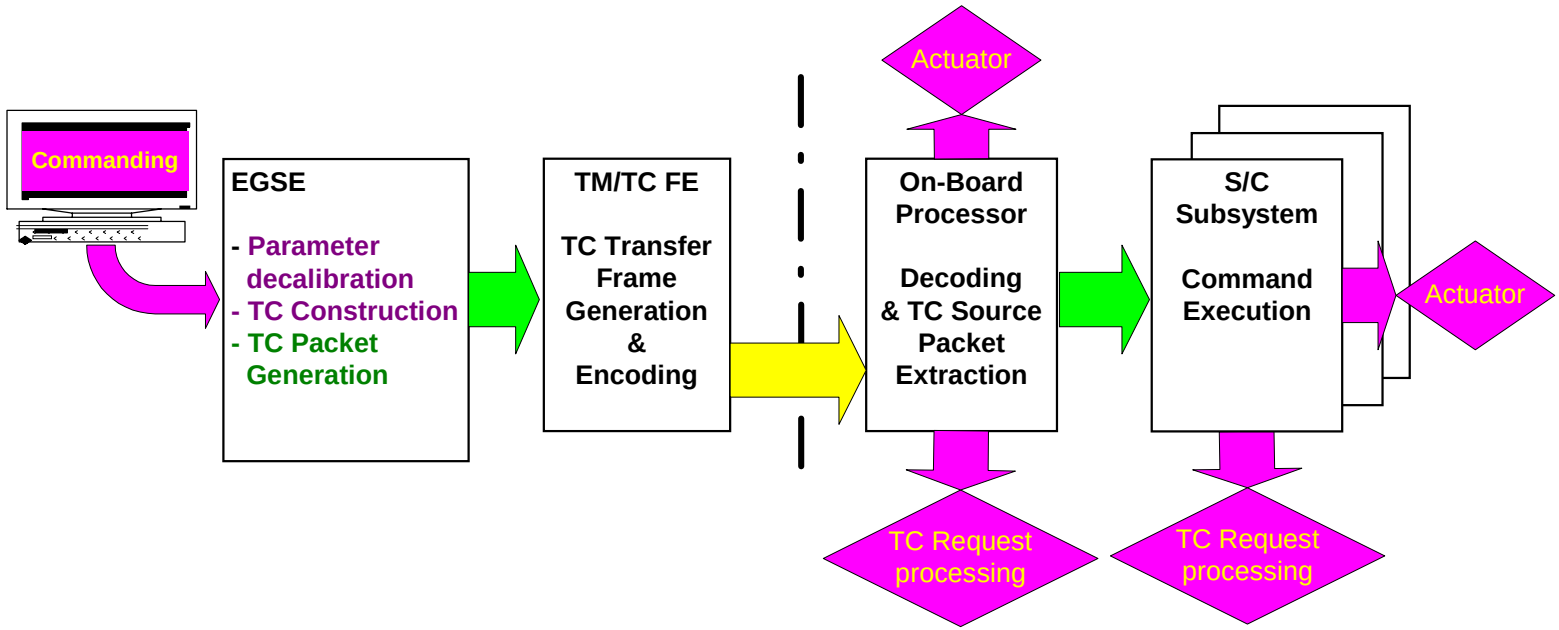
Packet Utilisation Standard support by CGS

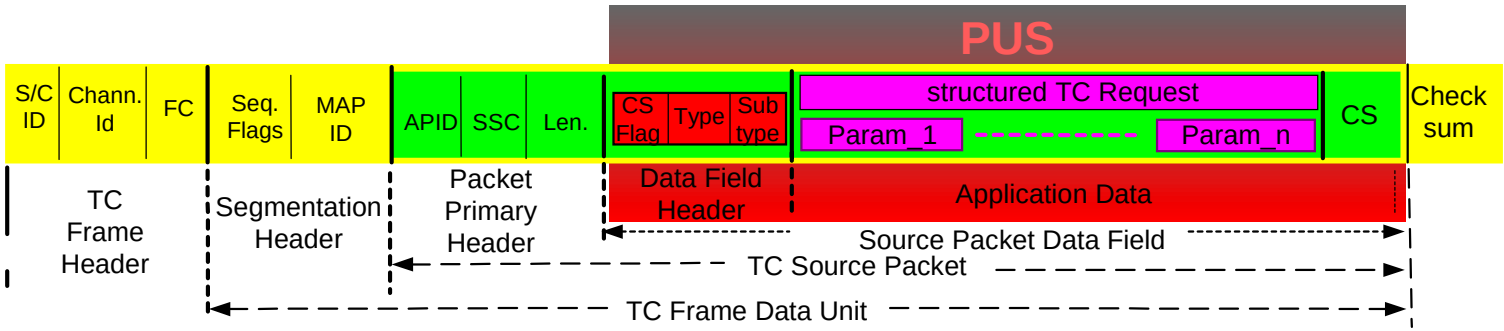
- **Some ideas of PUS were already implemented in the Columbus SW commanding**
- **ECSS PUS support has been implemented in a generic, re-usable way for the two missions CyroSat and GOCE in 2001**
- **ATV specific PUS support has been implemented in 2002**

PUS (Packet Utilisation Standard) – Short Summary

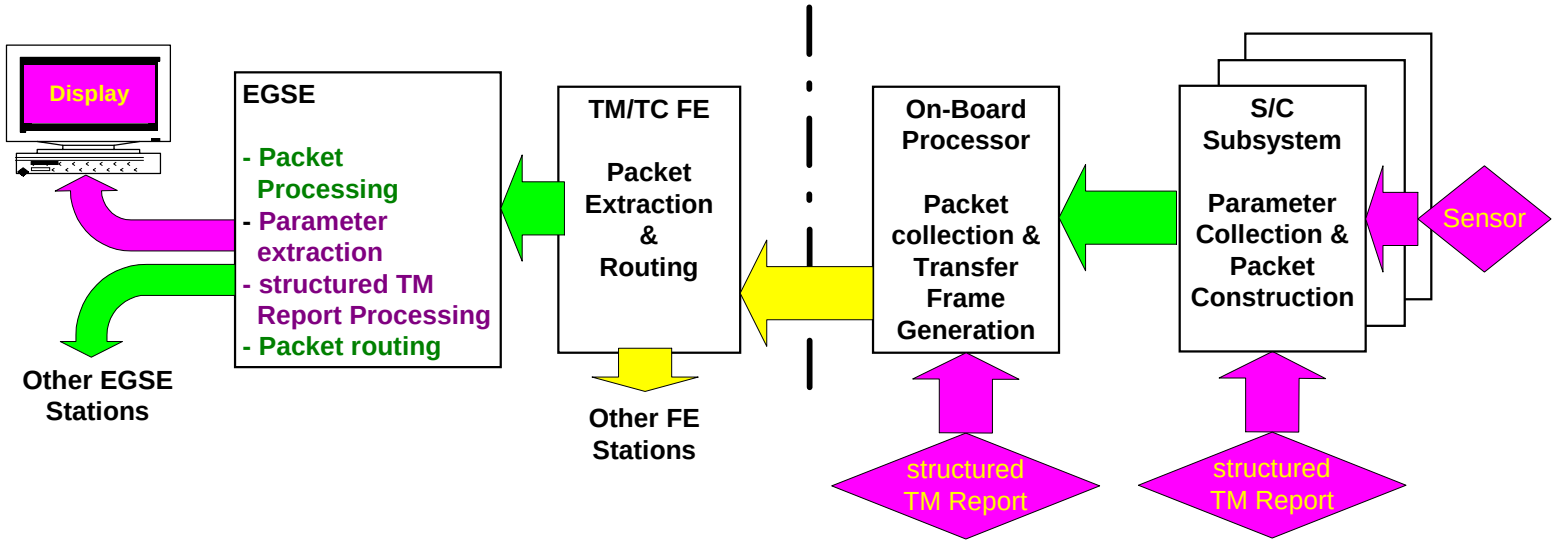
- PUS (Packet Utilisation Standard) standardizes the interface between onboard and ground on application level
- PUS structures the spacecraft command and monitoring in so called services (1-19)
- Each service defines the service request(s) and the service response(s)
- PUS structures the services and the respective data within CCSDS packets
- PUS is based on CCSDS packet standard (i.e. CCSDS packet structure definitions are valid)
- The services should be viewed as menu from which the applicable services should be selected for a given mission during the design phase

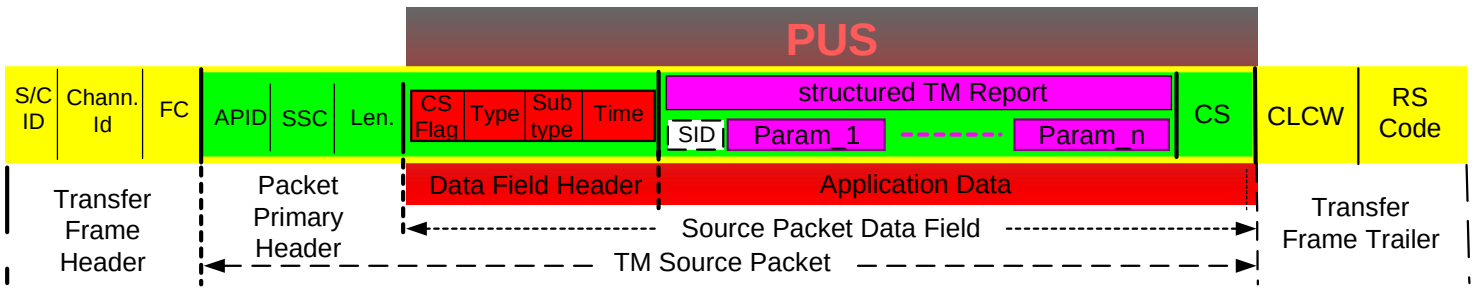
PUS Telecommand Data Flow





PUS Telemetry Data Flow





Summary PUS related CGS Extensions

- PUS packet definition in the configuration database
- Visualisation of command execution status in the command verification window (PUS Service 1 – TC Verification Service)
- HK Service processing (PUS Service 3 – HK Service)
- Trigger and TM Access Library
- Command Construction Library
- Visualisation support for repeated sub-structures
- Memory load and dump support

PUS Packet Definition

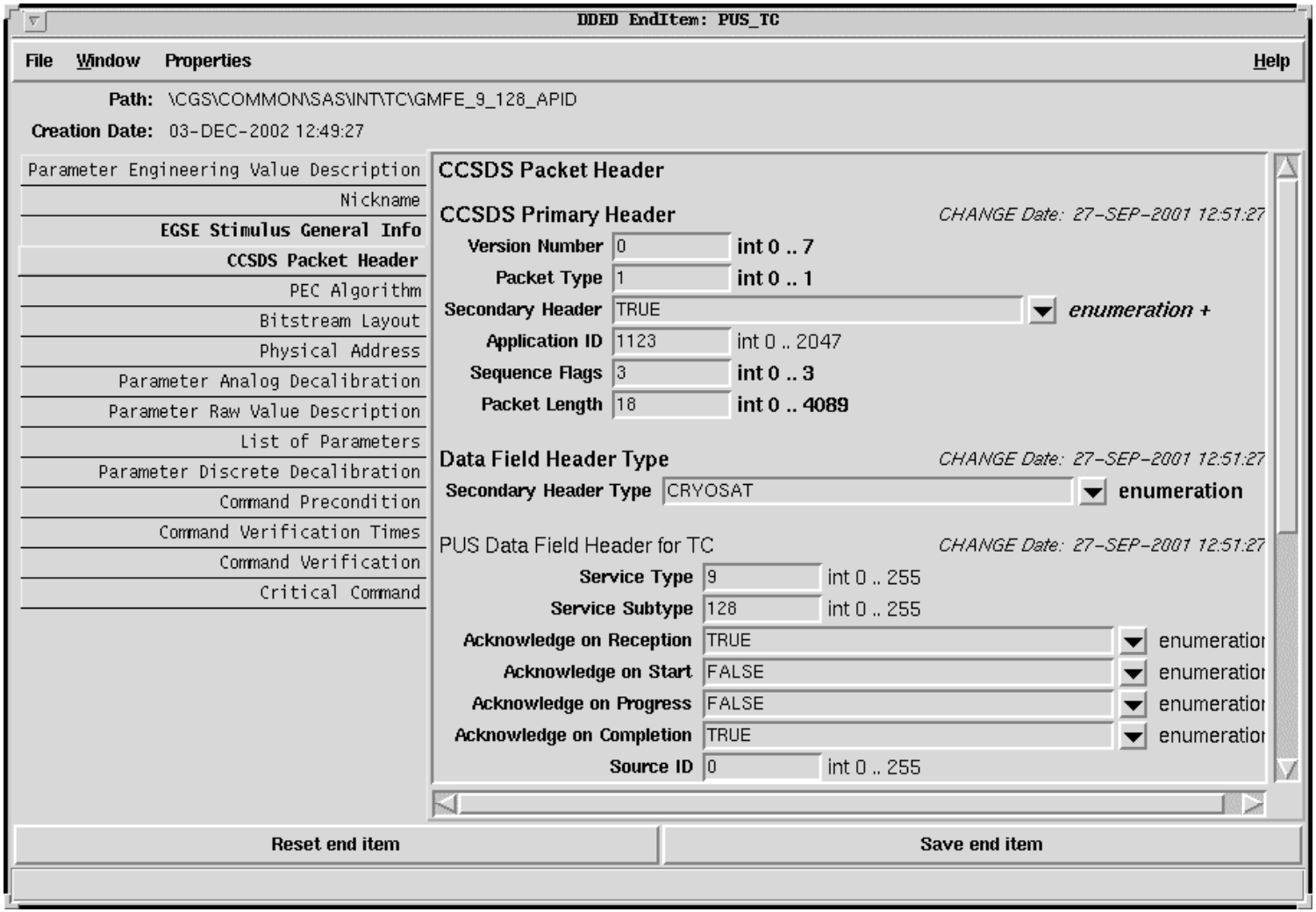
- PUS TC source packet definition is based on CCSDS command definition with the necessary extensions in the data field header
  - Service and sub-service type
  - Telecommand acknowledgments (on reception, start, progress or completion)

- PUS TM source packet definition is based on CCSDS telemetry definition with necessary extensions in the data field header
  - Service and sub-service type
  - SID (structure identifier) specifying the TM parameters in the packet following the SID

- PUS SID describes the packet contents
  - By a TM parameter list
  - Value alignment (byte, word, longword, unaligned)

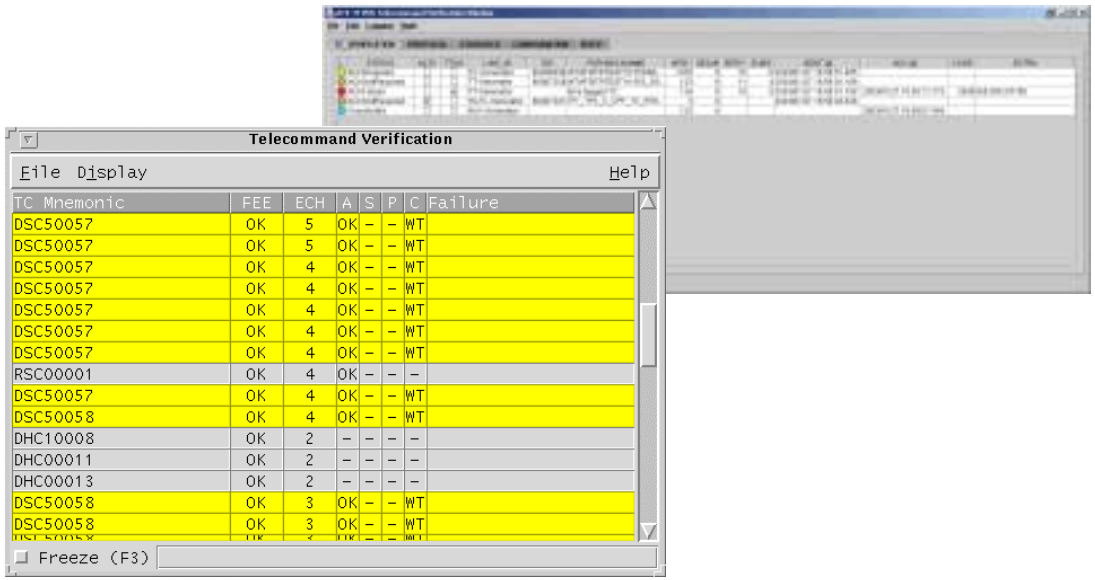
- Standard packet layout can be applied to PUS packets additionally

Example PUS TC



PUS TC Verification Window (PTV)

- PTV provides visualisation of the PUS Telecommand Verification Service
- When issuing a telecommand an entry will be created in the PTV and the system asynchronously from now on waits for acknowledgments (TM/TC Front End) and onboard responses to the issued telecommand
- The PTV entries will be updated with received responses to show the actual command execution status to the operator
- PTV is available as TCL/TK and Java application



Housekeeping Service

- The contents of a PUS TM packet is exactly described by APID (Primary Header), PUS type, PUS sub-type (Data Field Header) and Structure Id (Data Field)
- CGS supports the TM parameter extraction on basis of an identifier. The location of the identifier in the TM packet can be specified in the database. The identifier may be the SID (Structure Id), the EID (Event Id) or any other identifier.
- The identifier may vary from packet to packet in offset and length
- The identifier represents a list of TM parameters defined in the CGS database
- Subsequent processing of the TM parameter (calibration, monitoring) is performed by CGS as for any other TM parameter

Trigger and TM Access Library

- PUS services often contain TM packet definitions with repeated sub-structures or dynamic contents where it is not convenient or sometimes not possible to work with TM parameter (e.g. Onboard Min/Max Statistics)

- Processing of these TM packets can be done by using the "Trigger and TM Access Library". This Library allows to wait asynchronously for a dedicated TM packet. When the packet arrives a CGS automated procedure will be started automatically which processes/evaluates(*) the packet contents

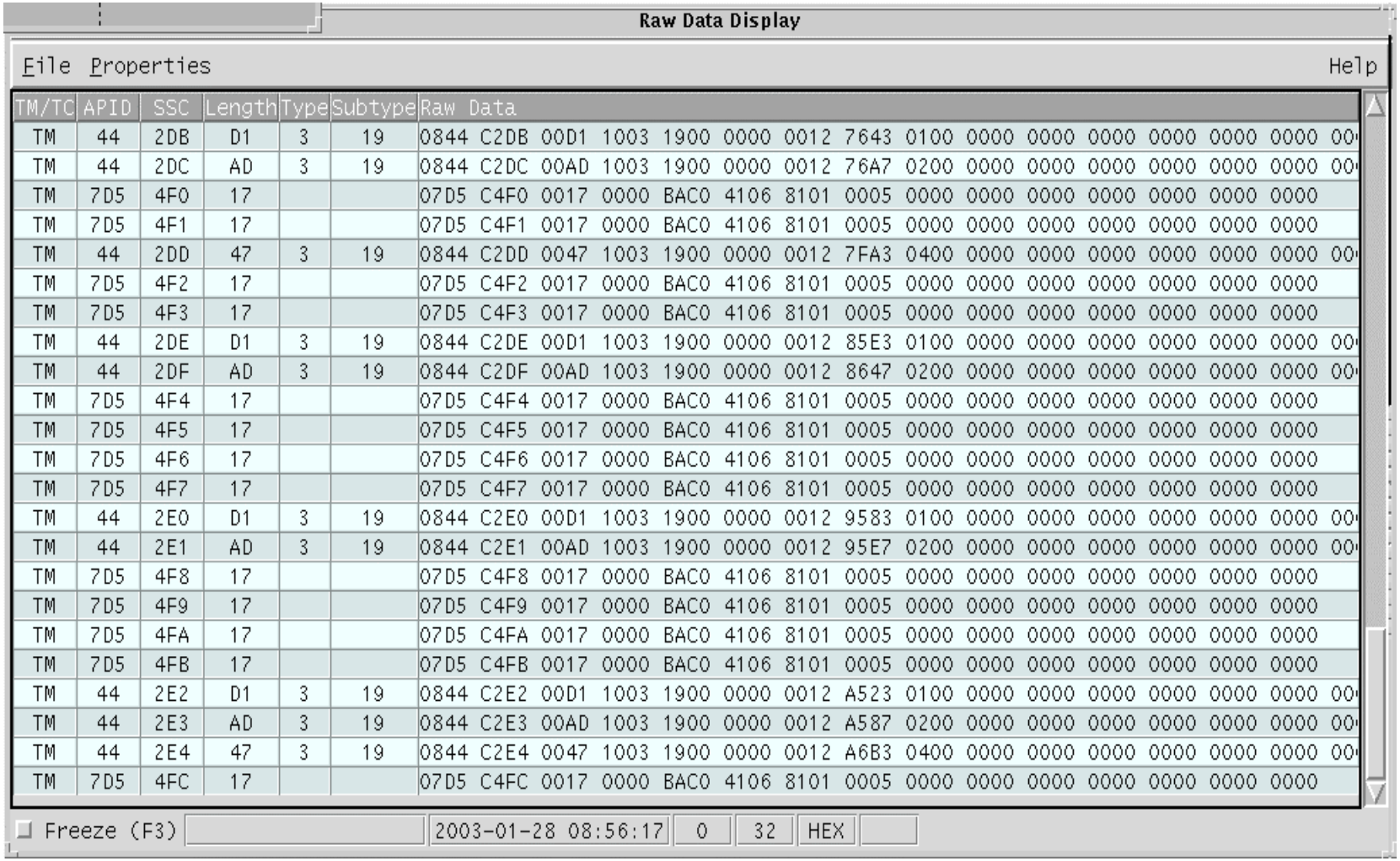
  (*) create log entry, display message in window, open notification window, other processing

- Access is granted to the entire TM packet, i.e. primary header, data field header (secondary header) and user data

Construction Lib/Visualisation/Memory Access

- Alternative to TM packet processing with the “Trigger and TM Access Library” the repeated sub-structures can be visualised in a dedicated window
- To test insertion of telecommands into the onboard master timeline queue (Service 11) a command construction library is available that allows to encapsulate telecommand(s) into an envelope telecommand sent to the onboard system
- The command construction library can also be use to build erroneous command(s) to test the onboard system
- Memory load and dump functionality has been implemented by CGS automated procedure/libraries
- Advantage of this implementation is that the AIT personal (not the EGSE provider) can modify the SW in case needed

Received TM Source Packet Display



Raw Data Display

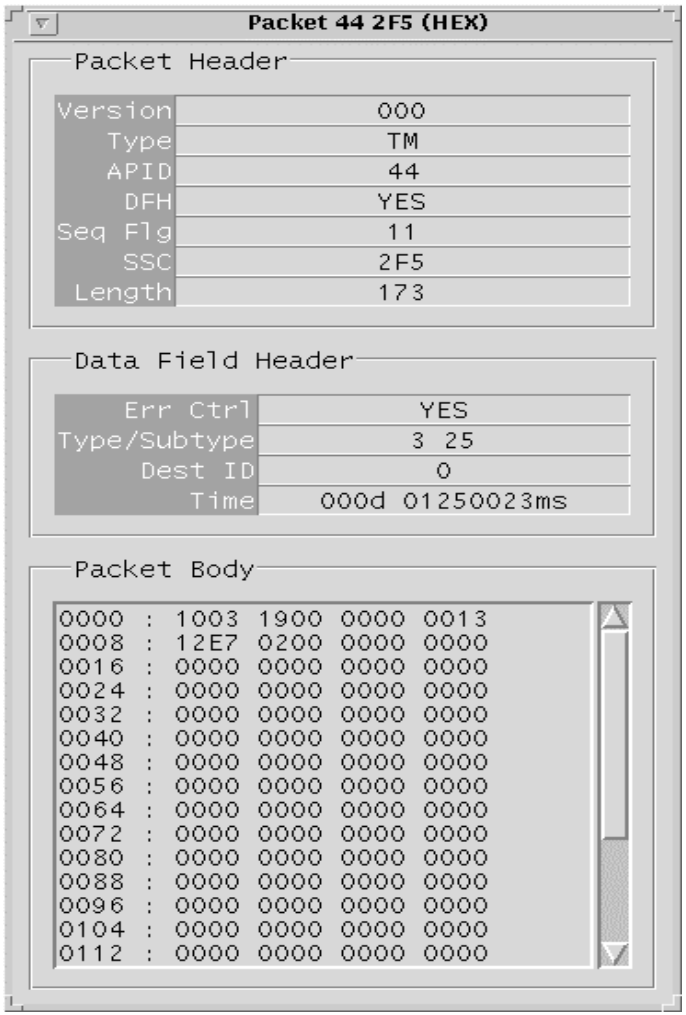
File Properties Help

TM/TC	APID	SSC	Length	Type	Subtype	Raw Data
TM	44	2DB	D1	3	19	0844 C2DB 00D1 1003 1900 0000 0012 7643 0100 0000 0000 0000 0000 0000 0000 00
TM	44	2DC	AD	3	19	0844 C2DC 00AD 1003 1900 0000 0012 76A7 0200 0000 0000 0000 0000 0000 0000 00
TM	7D5	4F0	17			07D5 C4F0 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4F1	17			07D5 C4F1 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	44	2DD	47	3	19	0844 C2DD 0047 1003 1900 0000 0012 7FA3 0400 0000 0000 0000 0000 0000 0000 00
TM	7D5	4F2	17			07D5 C4F2 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4F3	17			07D5 C4F3 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	44	2DE	D1	3	19	0844 C2DE 00D1 1003 1900 0000 0012 85E3 0100 0000 0000 0000 0000 0000 0000 00
TM	44	2DF	AD	3	19	0844 C2DF 00AD 1003 1900 0000 0012 8647 0200 0000 0000 0000 0000 0000 0000 00
TM	7D5	4F4	17			07D5 C4F4 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4F5	17			07D5 C4F5 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4F6	17			07D5 C4F6 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4F7	17			07D5 C4F7 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	44	2E0	D1	3	19	0844 C2E0 00D1 1003 1900 0000 0012 9583 0100 0000 0000 0000 0000 0000 0000 00
TM	44	2E1	AD	3	19	0844 C2E1 00AD 1003 1900 0000 0012 95E7 0200 0000 0000 0000 0000 0000 0000 00
TM	7D5	4F8	17			07D5 C4F8 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4F9	17			07D5 C4F9 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4FA	17			07D5 C4FA 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	7D5	4FB	17			07D5 C4FB 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000
TM	44	2E2	D1	3	19	0844 C2E2 00D1 1003 1900 0000 0012 A523 0100 0000 0000 0000 0000 0000 0000 00
TM	44	2E3	AD	3	19	0844 C2E3 00AD 1003 1900 0000 0012 A587 0200 0000 0000 0000 0000 0000 0000 00
TM	44	2E4	47	3	19	0844 C2E4 0047 1003 1900 0000 0012 A6B3 0400 0000 0000 0000 0000 0000 0000 00
TM	7D5	4FC	17			07D5 C4FC 0017 0000 BAC0 4106 8101 0005 0000 0000 0000 0000 0000 0000 0000

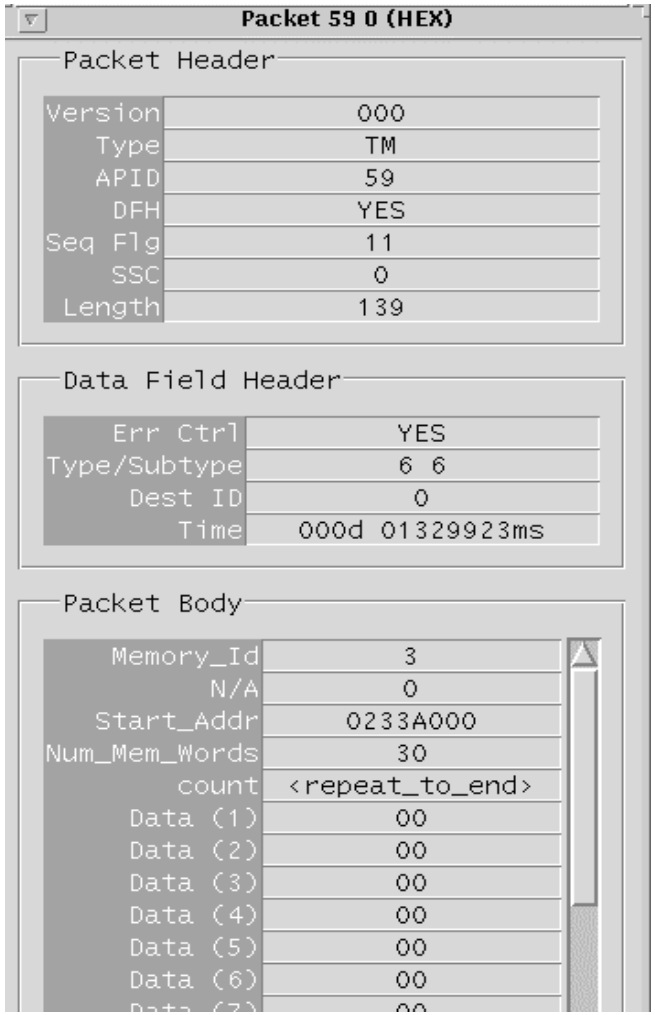
Freeze (F3) | 2003-01-28 08:56:17 | 0 | 32 | HEX

TM Source Packet De-commutation

Raw Packet Hex-Dump



Structured Packet Display



Experience in CryoSat Check-Out - (1)

Remarks to the PUS

PUS Capabilities not implemented / not implementable

- on-line HK TM Packet modification / definition
  - TM Packets are defined in Database, therefore DB needs to be changed and Run-Time Environment re-generated

PUS Flexibility causing Trouble

- flexibility given by the PUS “Id” + “Parameter list (any parameter)” leads to an inflation of TM Packet definitions.
  - » to avoid problems project-specific tailoring requires to define how to use “Error-Id” / “Failure-Id” Parameter Lists

Experience in CryoSat Check-Out - (2)

Implementation Approach for CryoSat Check-Out

TM Event Packets

- captured by CGS Trigger & Access Library Function
- reporting via user-defined Test Sequences (AP´s)
- TM Parameter Monitoring by CGS Standard Functions

variable length TM Report Packets

- capturing by CGS Trigger & Access Library Function
- “Sub-Structure” Decommutation & Reporting via AP´s

Import of Spacecraft TM/TC Definitions from “Satellite Reference Database” C-SDB - SCOS 2000 Access DB

Recommendations

- mission specific tailoring of PUS services used is mandatory
- services selected must be used as defined by ECSS-E-70-41A
- additional mission-specific (Sub-) Services may be defined as appropriate
  - never re-define Sub-Services defined by ECSS-E-70-41A
- TM/TC Source Packet definition must be used according to Standard (ESA Standard / CCSDS Recommendations refer)
  - never re-define Source Packets Primary Header
- TM & TC Data Field Headers must be used according to Standard
  - never change the basic structure given by the first 24 bits
- Time Stamp in TM DFH has to be fixed to length 48 bit
  - ⇨ time format is mission-specific
  - ⇨ 48 bit considered sufficient for TM Source Pkt time stamping
  - ⇨ more accurate time for synch purpose may be povided by dedicated “Time Packet” (APID = 0, Service,Sub-Service = (9,2))