SAVOIR
On-board Mass Memory Day

Telemetry and Telecommand Packet Utilization

The ECSS-E-ST-70-41C

Feb 2014 ECSS WG Draft

Serge Valera - ESA
Arnaud Bourdoux- Spacebel
Sante Candia - TAS
13 February 2014

ECSS-E-ST-70-41C WG – Main Purpose

- The Main Purpose of the PUS-A to PUS-C upgrade has been:

  processing 190 change requests…

  - to remove the PUS-A deficiencies and inject lessons learned
  - to improve the standard to meet the need for future missions
  - to acknowledge the existence of new ECSS and CCSDS standards and ensure consistency
  - to implement the ECSS drafting rules that apply to any ECSS Standards [e.g. naming each requirement to facilitate tailoring, traceability]
  - *to maintain backward compatibility when possible*

- This new version of the PUS has been developed by the major stakeholders including Space Agencies and Industry, with representatives of both the Space Segments and the Ground Segments

ECSS-E-ST-70-41C WG Members

Organisation	Representatives	
ESA	S. Valera *convenor* *M. de Lande Long* *book captain*	G.P. Calzolari M. Schön
ASTRIUM	R. Gessner P. Parmentier	J. Rueting
CNES	*P. Arberet* M.C. Charmeau	J.P. Loubeyre N. Pons
DLR	S. Zimmermann	
EUMETSAT	F. Croce	
SPACEBEL	A. Bourdoux	
TAS	S. Candia P. Fourtier	G. Garcia

PUS C – A new document structure

1. *Scope*
2. *Normative references*
3. *Terms, definitions and abbreviated terms*
4. Context and background
5. PUS foundation model
6. Service Type System Specification
7. Space/Ground Interface Requirements
   a. Space packet overview
   b. Packet Field Code (PTC/PFC)
   c. CCSDS Space Packet
8. Service Type Interface Specification

Annexes

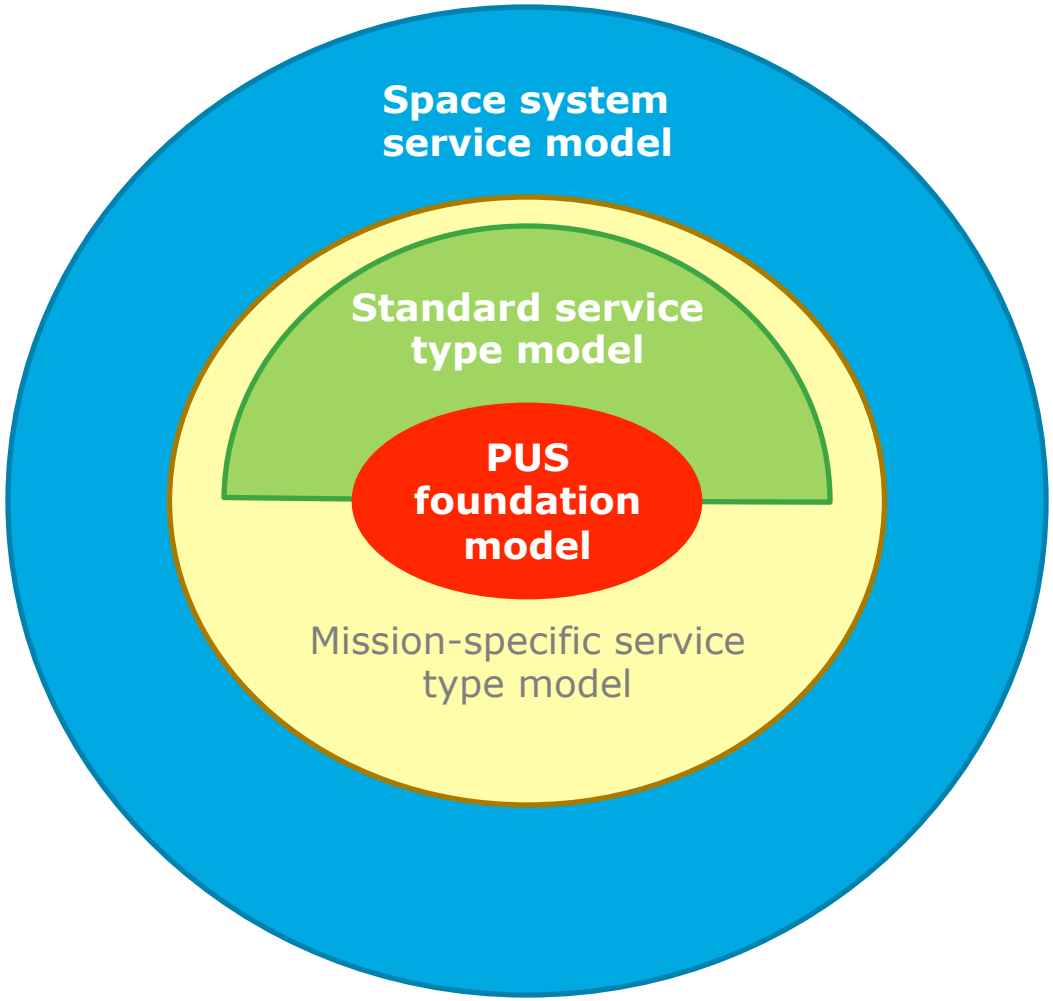
A. normative Command Pulse Distribution Unit
B. informative IEEE and MILBUS real formats
C. informative CRC and ISO checksum
D. informative Summary of requests and reports for PUS standard services

PUS foundation model	**±30 pages**
ST System Specification	**± 320 pages**
ST Interface Specification	**± 180 pages**
ECSS-E-ST-70-41C	**± 630 pages**

The PUS Foundation Model

The **PUS Foundation Model** defines the PUS generic concepts, related terms and definitions and the business rules that:

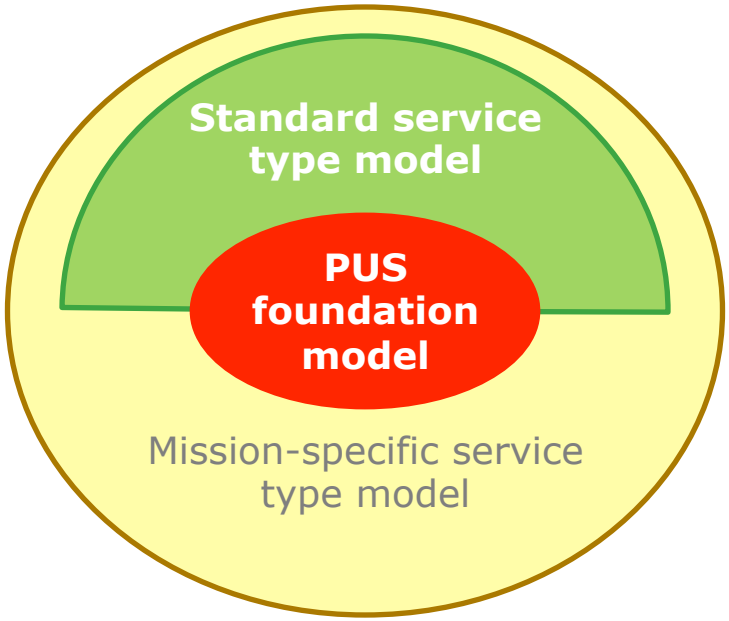
- have been used to produce the **"standard service type model"**;
- apply to each mission that makes applicable the Standard and define the **"mission-specific service type model"** by tailoring, i.e.:
  - adding new service types, subservice types, message types, …
  - adding capabilities to the "standard service types";
- apply to the architects of the mission-specific space system, i.e. both the space segment and the ground segment, that developing and instantiating the mission-specific service type model produce the **"space system service model"**



The PUS Foundation Model

The **PUS Foundation Model** defines two levels of abstractions, i.e.:

- The ***generic type abstraction level*** which specifies the set of generic object types and business rules that are required for ensuring the overall consistency of the service type model. This abstraction level includes all generic object types used to produce, by specialisation, the standardized and the mission specific service types.



The PUS Foundation Model

The **PUS Foundation Model** defines two levels of abstractions, i.e.:

- The ***generic type abstraction level*** which specifies the set of generic object types and business rules that are required for ensuring the overall consistency of the service type model. This abstraction level includes all generic object types used to produce, by specialisation, the standardized and the mission specific service types.

That abstraction level specifies:

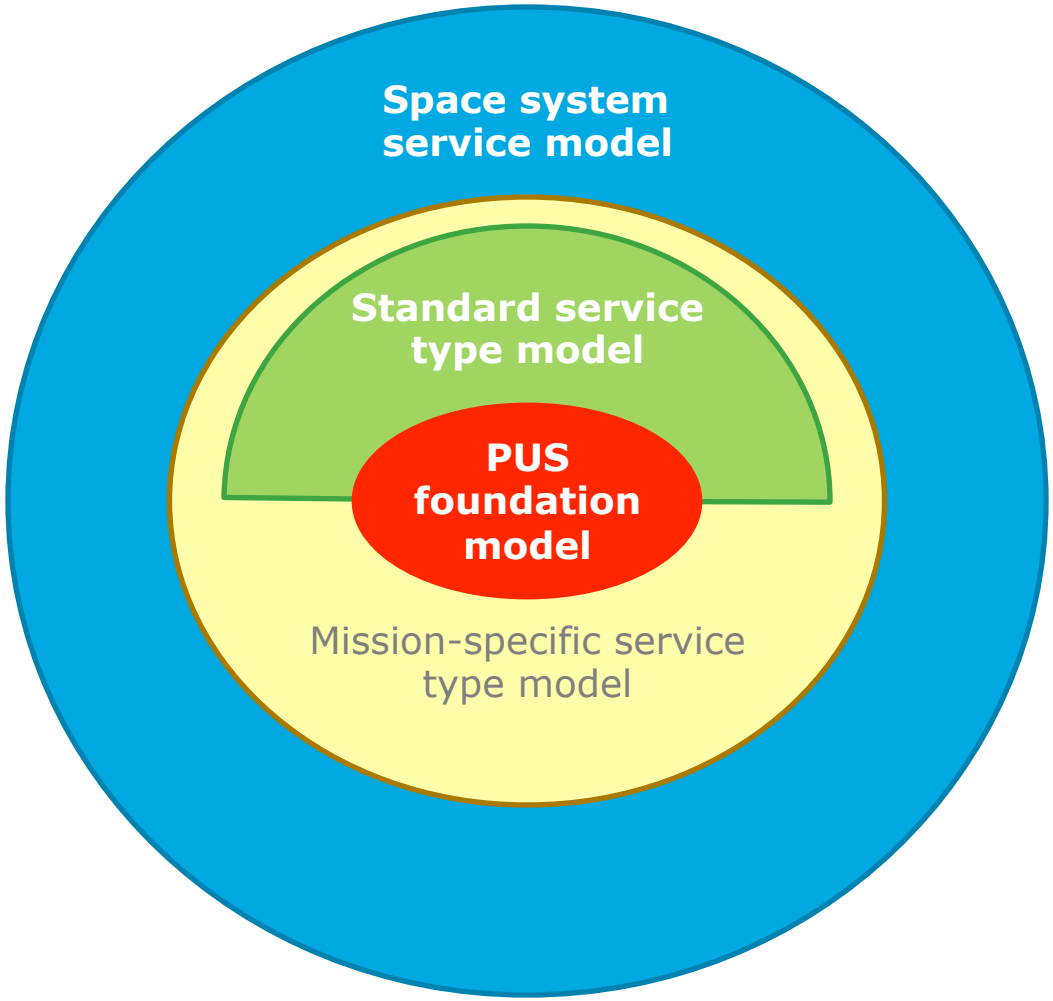
- the service type, the subservice type and the capability set type exposed by the subservice type;
- the message type, i.e.:
  - the request type and the instruction type,
  - the report type and the notification type;
- the transaction type and its nature dependent definition, i.e.:
  - if the nature is related to request, the request type, the associated execution notification type and to if some service data are generated in response to such request, the related indication report type;
  - if the nature is related to indication, the indication report type

The PUS Foundation Model

The **PUS Foundation Model** defines two levels of abstractions, i.e.:

- The ***generic type abstraction level*** which specifies the set of generic object types and business rules that are required for ensuring the overall consistency of the service type model. This abstraction level includes all generic object types used to produce, by specialisation, the standardized and the mission specific service types.

- The ***generic deployment abstraction level*** which specifies the set of generic object types and business rules that are required to capture the space system service model. This abstraction level includes all generic object types used to capture, by instantiation, the space system services resulting from the space system overall architecture



The PUS Foundation Model

The **PUS Foundation Model** defines two levels of abstractions, i.e.:

- The ***generic type abstraction level*** which specifies the set of generic object types and business rules that are required for ensuring the overall consistency of the service type model. This abstraction level includes all generic object types used to produce, by specialisation, the standardized and the mission specific service types.
- The ***generic deployment abstraction level*** which specifies the set of generic object types and business rules that are required to capture the space system service model. This abstraction level includes all generic object types used to capture, by instantiation, the space system services resulting from the space system overall architecture

That abstraction level specifies:

- the service system context

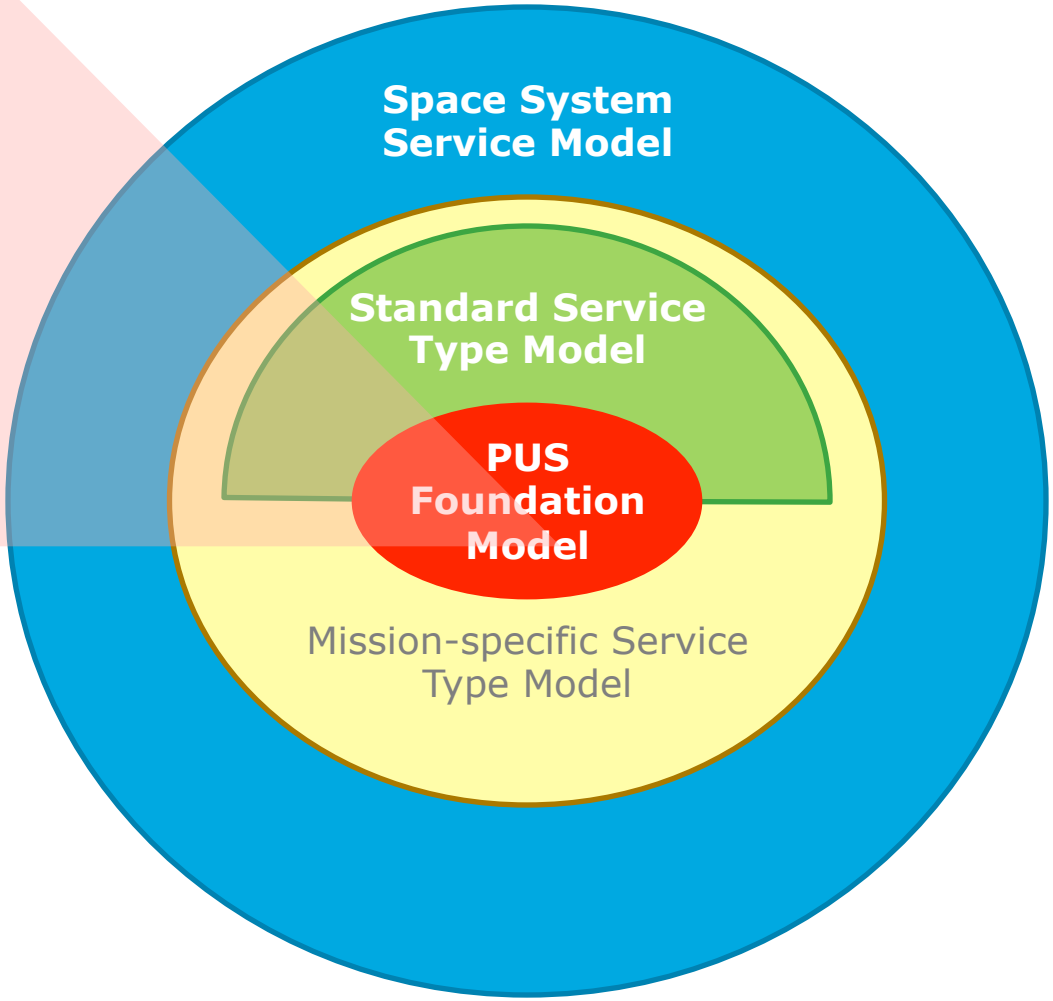
  in term of involved space system elements of relevance to the service functionality, e.g. the space segment, the ground segment, the application process, the on-board parameter, the on-board memory,
- the service, the subservice and the capability exposed by the subservice;
- the message, i.e.:
  - the request, the instruction slot and the instruction,
  - the report, the notification slot and the notification;
- the transaction.

The Service Type System Requirements

The **ST System Requirements**

addresses the "Semantic" of each service type including:

- the service type concept and related architecture
- the message type concept and related architecture
- the overall service type topology (focusing on message exchange between service users and service providers)



The Service Type Interface Requirements

The **ST System Requirements**

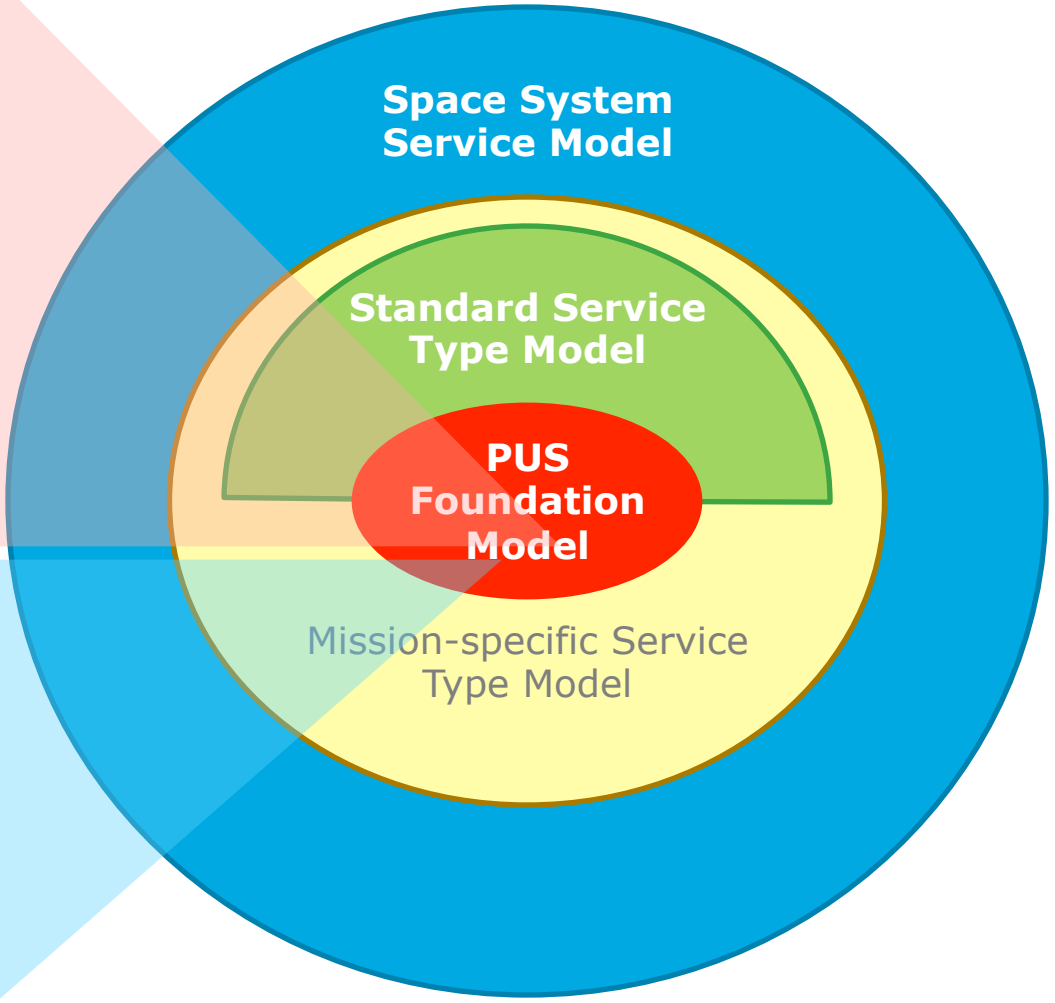
addresses the "Semantic" of each service type including:

- the service type concept and related architecture
- the message type concept and related architecture
- the overall service type topology (focusing on message exchange between service users and service providers)

The **service type interface requirements**

specifies the Ground/Space Link message exchange protocol i.e.:

- how requests are transported within CCSDS SPP & PUS telecommand packets;
- how reports are transported within CCSDS SPP & PUS telemetry packets.



The Standard Service Type Model

Standard Service Types		
ST 1	**request verification**	**…**
ST 2	**device access**	√
ST 3	**Housekeeping**	√
ST 4	**parameter statistics reporting**	√
ST 5	**event reporting**	√
ST 6	**memory management**	√
ST 8	**function management**	√
ST 9	**time management**	√
ST 11	**time-based scheduling**	√
ST 12	**on-board monitoring**	√
ST 13	***large data transfer***	**TBC**
ST 14	**real-time forwarding control**	√
…		

Standard Service Types		
…		
ST 15	**on-board storage and retrieval**	**…**
ST 17	**test**	√
ST 18	**on-board control procedure**	√
ST 19	**event – action**	√
new service types:		
ST 20	**parameter management**	√
ST 21	**request sequencing**	√
ST 22	**position-based scheduling**	√
ST 23	**file management**	√
ST 24	***large packet transfer***	**TBC**

Transfer of large data, “large” packets

Standard Service Types		
ST 1	request verification	…
ST 2	device access	√
ST 3	Housekeeping	√
ST 4	parameter statistics reporting	√
ST 5	event reporting	√
ST 6	memory management	√
ST 8	function management	√
ST 9	time management	√
ST 11	time-based scheduling	√
ST 12	on-board monitoring	√
ST 13	***large data transfer***	**TBC**
ST 14	real-time forwarding control	√
…		

Standard Service Types		
…		
ST 15	on-board storage and retrieval	…
ST 17	test	√
ST 18	on-board control procedure	√
ST 19	event – action	√
new services :		
ST 20	parameter management	√
ST 21	request sequencing	√
ST 22	position-based scheduling	√
ST 23	file management	√
ST 24	***large packet transfer***	**TBC**

Transfer of large data, "large" packets

Two issues

- **Up to 65 KB, for CCSDS packets transfer**
- **More than 65 KB, for large memory area**

Several views under assessment by the PUS C WG

1. standardize only PUS A service 13 large data transfer
   a. in full compliance
   b. without extended packet support, i.e. limited to 65 KB packets
2. standardize only the new service 24 large packet transfer, i.e. limited to 65 KB packets
   a. with retransmission capability
   b. without retransmission capability
3. standardize both service 13 and service 24
4. …

Transfer of large data, "large" packets

Several views under assessment by the PUS C WG

1. *standardize only PUS A service 13 large data transfer*
2. *standardize only the new service 24 large packet transfer, i.e. limited to 65 KB packets*
3. *standardize both service 13 and service 24*

4. limit the scope of PUS to address only the application layer, and not anymore any lower level transfer protocol issues **+**
   a. All missions shall support both uplink and downlink of any CCSDS packets up to 65 KB
   b. Each mission that limits the maximum size of the exchanged TM and/or TC packets to a value that is less than the CCSDS SPP maximum packet length, shall identify a low level mechanism that enable the transfer and reconstruction of source packets, *i.e. using the ECSS-E-50 and the CCSDS Space Data Link protocol standards or any other adequate mechanism*

Service Type 6 Memory Management

Standard Service Types		
ST 1	request verification	…
ST 2	device access	√
ST 3	Housekeeping	√
ST 4	parameter statistics reporting	√
ST 5	event reporting	√
ST 6	**memory management**	√
ST 8	function management	√
ST 9	time management	√
ST 11	time-based scheduling	√
ST 12	on-board monitoring	√
ST 13	*large data transfer*	TBC
ST 14	real-time forwarding control	√
…		

Standard Service Types		
…		
ST 15	on-board storage and retrieval	…
ST 17	test	√
ST 18	on-board control procedure	√
ST 19	event – action	√
	new services :	
ST 20	parameter management	√
ST 21	request sequencing	√
ST 22	position-based scheduling	√
ST 23	file management	√
ST 24	*large packet transfer*	TBC

Service Type 6 Memory Management

Provide support for 'flat' and 'structured' file systems.

Three subservices

- **The raw data memory management subservice**, for memory that content is not implicitly known
- **The structured data memory subservice**, for memories containing memory objects such as files (containing e.g. on-board control procedures, request sequences), configuration tables
- **The common memory management subservice**, common to raw and structured data memories, that manage memories as "wholes", i.e. independently of their content and specific addressing scheme, e.g. for enabling/disabling scrubbing, write protecting memories

Service Type 23 File Management

Standard Service Types		
ST 1	request verification	...
ST 2	device access	√
ST 3	Housekeeping	√
ST 4	parameter statistics reporting	√
ST 5	event reporting	√
ST 6	memory management	√
ST 8	function management	√
ST 9	time management	√
ST 11	time-based scheduling	√
ST 12	on-board monitoring	√
ST 13	*large data transfer*	TBC
ST 14	real-time forwarding control	√
...		

Standard Service Types		
...		
ST 15	on-board storage and retrieval	...
ST 17	test	√
ST 18	on-board control procedure	√
ST 19	event – action	√
	new services :	
ST 20	parameter management	√
ST 21	request sequencing	√
ST 22	position-based scheduling	√
ST 23	**file management**	√
ST 24	*large packet transfer*	TBC

Service Type 23 File Management

Two subservices

- **The file handling subservice**, providing the operations to handle on-board files and directories
- **The file copy subservice**, providing the operations:
  - for copying files within and between file systems
  - for controlling the on-going copies, i.e. suspend, resume and abort

This Standard does not impose a specific file delivery protocol (e.g. CFDP).

The choice for the protocols is implementation-dependent and depends on the location of the source file and its destination, i.e.:

- *on-board, i.e.*
  - *both source and destination in the same file system*
  - *source and destination in different file systems or*
- *located into remote file systems, e.g.*
  - *source on ground and destination on-board*
  - *source in one spacecraft and destination in another spacecraft*

Service Type 23 File Management

Requests and reports

- **The file handling subservice:**
  - create and delete files, lock and unlock files
  - find files,
  - report file attributes
  - create, delete and rename directories
  - report repository content
- **The file copy subservice:**
  - copy and move files
  - suspend, resume and abort copy/move operations
  - suspend all, resume all and abort all copy/move operations
  - start and stop the status reporting of file copy/move operations progress

Service Type 23 File Management

Service types that include standard request types that operate with files

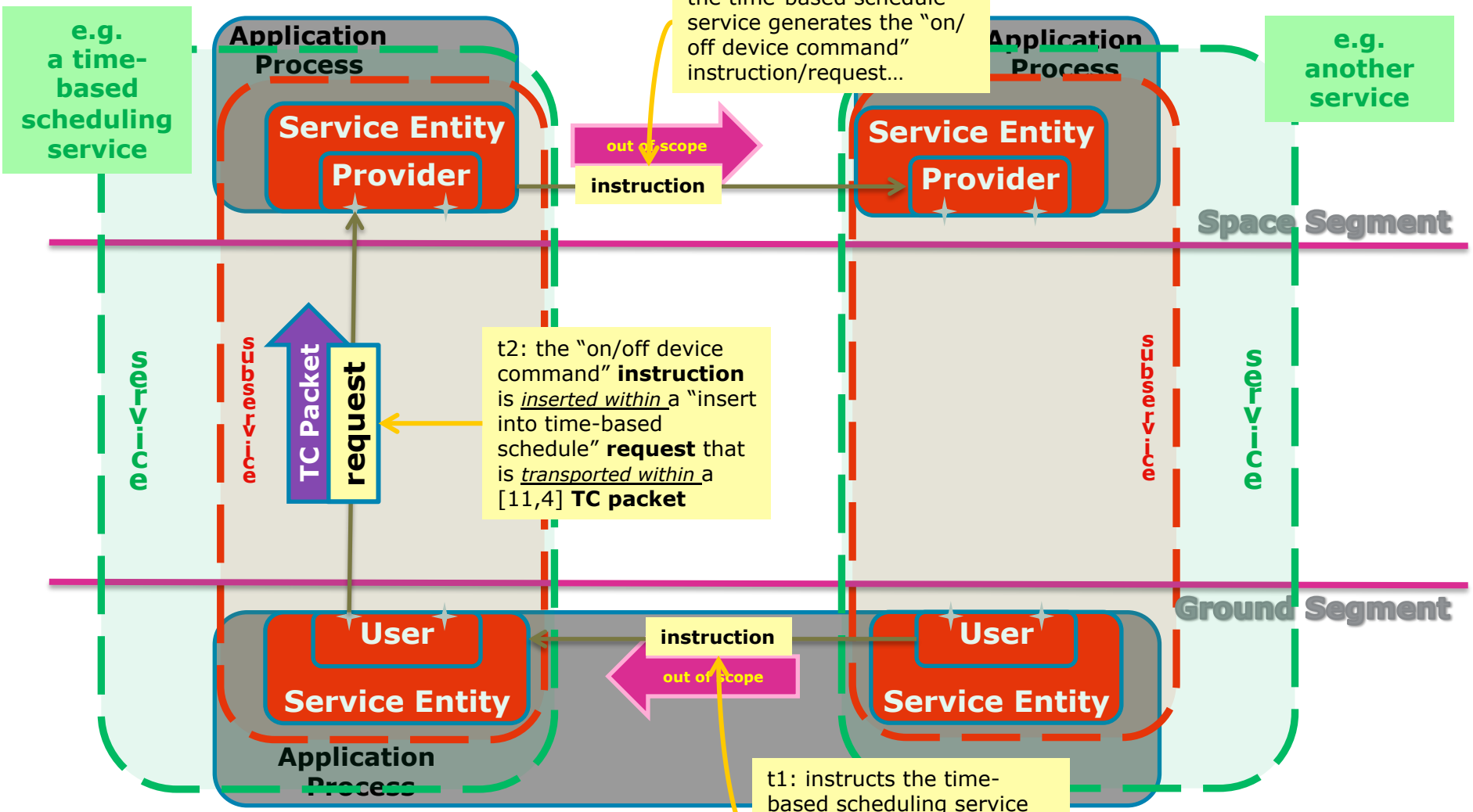
- **ST18 on-board control procedure**
  - load an OBCP by reference
- **ST21 request sequencing:**
  - load a request sequence by reference

Service types that interact with files

- **Service 6 memory management:**
  - the memory base plus offset concept has been extended to access memory objects such as files. This can be used to e.g. patch files, compute their checksum
- ***Service 20 parameter management***
  - get, set parameter values, dynamic *creation (!)* of new parameters based on raw or structured data memory access


Any questions?
European Space Agency

A time-based scheduling service interacting with other services



A specification view of a request

PUS system requirements only address the *what*, i.e. the information from a semantic viewpoint

Request type [x, y]

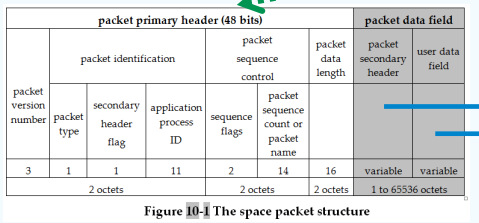
RT instruction

RX provides multiple instruction slots = TRUE

instruction 1 of type [x, y] | **instruction 2 of type [x, y]** | ... | **instruction N of type [x, y]**

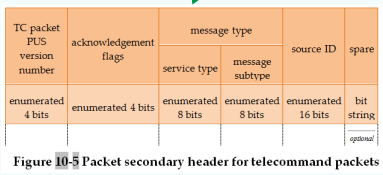
Request of type [x, y]

instruction slot 1 | **instruction slot 2** | ... | **instruction slot N**



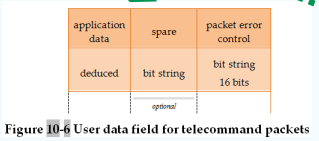
packet primary header (48 bits)							packet data field	
packet version number	packet identification			packet sequence control		packet data length	packet secondary header	user data field
	packet type	secondary header flag	application process ID	sequence flags	packet sequence count or packet name			
3	1	1	11	2	14	16	variable	variable
2 octets				2 octets		2 octets	1 to 65536 octets	

Figure 10-1 The space packet structure



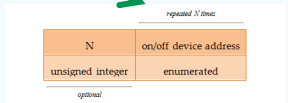
TC packet PUS version number	acknowledgement flags	message type		source ID	spare
		service type	message subtype		
enumerated 4 bits	enumerated 4 bits	enumerated 8 bits	enumerated 8 bits	enumerated 16 bits	bit string
					optional

Figure 10-5 Packet secondary header for telecommand packets



application data	spare	packet error control
deduced	bit string	bit string 16 bits
	optional	

Figure 10-6 User data field for telecommand packets



repeated N times

N	on/off device address
unsigned integer	enumerated

optional

Figure 11-1 Distribute on/off device commands

PUS interface requirements address the how, the data representation of the information

A specification view of a report

PUS system requirements only address the *what*, i.e. the information from a semantic viewpoint

Report type [x, y]

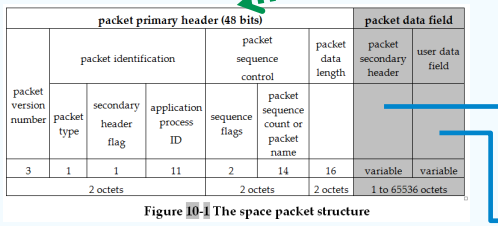
RT notification

RT provides multiple instruction slots = TRUE

notification 1 of type [x, y] | **notification 2 of type [x, y]** | ... | **notification N of type [x, y]**

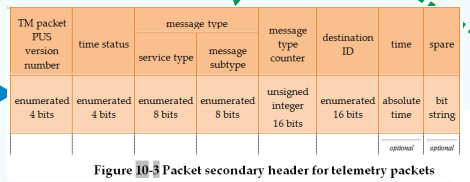
Report of type [x, y]

notification slot 1 | **notification slot 2** | ... | **notification slot N**



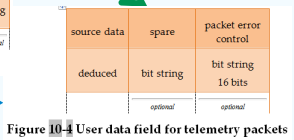
packet primary header (48 bits)							packet data field	
packet identification				packet sequence control		packet data length	packet secondary header	user data field
packet version number	packet type	secondary header flag	application process ID	sequence flags	packet sequence count or packet name			
3	1	1	11	2	14	16	variable	variable
2 octets				2 octets		2 octets	1 to 65536 octets	

Figure 10-1 The space packet structure



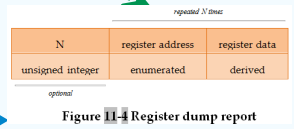
TM packet PUS version number	time status	message type		message type counter	destination ID	time	spare
		service type	message subtype				
enumerated 4 bits	enumerated 4 bits	enumerated 8 bits	enumerated 8 bits	unsigned integer 16 bits	enumerated 16 bits	absolute time	bit string
						optional	optional

Figure 10-3 Packet secondary header for telemetry packets



source data	spare	packet error control
deduced	bit string	bit string 16 bits
	optional	optional

Figure 10-4 User data field for telemetry packets



repeated N times

N	register address	register data
unsigned integer	enumerated	derived

optional

Figure 11-4 Register dump report

PUS interface requirements address the how, the data representation of the information