Avionics Data, Control and Software Systems
ADCSS-2015

Telemetry and Telecommand Packet Utilization

The ECSS-E-ST-70-41C

October 2015

Serge Valera
ESTEC/TEC-SWM
22 October 2015

ECSS-E-ST-70-41C WG – Main Purpose

01/2011 → 190 Change Requests raising the needs to:

- remove the PUS-A deficiencies
- inject lessons learned
- improve the standard to meet the need for future missions
- comply and ensure consistency with related ECSS and CCSDS standards
- comply with ECSS drafting rules

12/2014 → 620 Discrepancy Document Reports (DDR) resulting from the 4 months public review of ECSS-E-ST-70-41C DIR1

12/2015 → ECSS-E-ST-70-41C final draft delivery to ECSS…

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
   a. Generic Service Type
   b. Generic Service Deployment
6. Service Type System Requirements
7. Space/Ground Interface Requirements
   a. Packet field type code
   b. CCSDS Space packet
      – Packet data field for TM packet
      – Packet data field for TC packet
8. Service Type Interface Requirements

Annexes

A. normative Command Pulse Distribution Unit
B. informative IEEE and MILBUS real formats
C. informative CRC and ISO checksum
D. informative Request type & report type summary

The PUS foundation model in number of requirements	
Clause 5 - generic	**±150**
Clause 6 – system	**±2300**
Clauses 7 & 8 – interface	**±750**
	tailoring ±500

PUS-C versus PUS-A

1. A new vocabulary (involving types):
    a. formally specified in the PUS Foundation Model and
    b. consistently used through the overall system and interface requirements specifications.
2. A system requirement specification focusing on what is expected on-board

3. A drafting supported by the PUS Foundation Model to augment quality…

PUS- C standardized Service Types

Standard Service Types	
ST 1	request verification
ST 2	device access
ST 3	housekeeping
ST 4	parameter statistics
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 packet transfer
ST 14	real-time forwarding control
ST 15	on-board storage and retrieval
ST 17	test
…	

Standard Service Types	
…	
ST 18	on-board control procedure
ST 19	event – action
ST 20	parameter management
ST 21	telecommand sequencing
ST 22	position-based scheduling
ST 23	file management

ST2 – Device access

Standard Service Types		
ST 1	**request verification**	
ST 2	**device access**	√
ST 3	**housekeeping**	
ST 4	**parameter statistics**	
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 packet transfer**	
ST 14	**real-time forwarding control**	
ST 15	**on-board storage and retrieval**	
ST 17	**test**	
…		

New capabilities proposed for:

- register contents acquisition
- CPDU commands distributed by (on-board) software
- physical and logical low-level commands:
  - for device configuration and actuation
  - for data acquisition

ST3 – Housekeeping

Standard Service Types		
ST 1	**request verification**	
ST 2	**device access**	
ST 3	**housekeeping**	√
ST 4	**parameter statistics**	
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 packet transfer**	
ST 14	**real-time forwarding control**	
ST 15	**on-board storage and retrieval**	
ST 17	**test**	
…		

New subservice "The functional parameter reporting configuration"

It provides the capability to operate on dedicated sets of HK or diagnostic packets, i.e. reducing the number of requests needed to modify e.g. the current TM plan.

ST6 – memory management

Standard Service Types		
ST 1	**request verification**	
ST 2	**device access**	
ST 3	**housekeeping**	
ST 4	**parameter statistics**	
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 packet transfer**	
ST 14	**real-time forwarding control**	
ST 15	**on-board storage and retrieval**	
ST 17	**test**	
…		

The memory management service defines 4 subservices:

- the raw data memory management subservice
- the structured data memory management subservice (for memories containing e.g. files, on-board control procedures)
- the common memory management subservice (common to raw and structured data memories, e.g. abort all memory dumps)
- the memory configuration subservice that manage memories as wholes independently of their content and specific addressing scheme (e.g. enabling/disabling scrubbing, write protecting memories)

ST11 – time-based scheduling

Standard Service Types		
ST 1	**request verification**	
ST 2	**device access**	
ST 3	**housekeeping**	
ST 4	**parameter statistics**	
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 packet transfer**	
ST 14	**real-time forwarding control**	
ST 15	**on-board storage and retrieval**	
ST 17	**test**	
…		

The A version of this Service has been completely re-assessed resulting in 3 proposed new service types, i.e.

- ST11: the time-based scheduling
- ST21: the telecommand sequencing
- ST22: the orbit-based scheduling

The time-based scheduling service provides the capability to command on-board application processes using requests pre-loaded on-board the satellite and released at their due time.

It supports the concept of sub-schedules and groups.

This service is an *extension of the ground monitoring and control* *[As such, the application processes that execute the requests released by the telecommand sequencing service directly send the telecommand verification reports, if any, to the ground.]*

ST13 – large packet transfer

Standard Service Types		
ST 1	**request verification**	
ST 2	**device access**	
ST 3	**housekeeping**	
ST 4	**parameter statistics**	
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 packet transfer**	√
ST 14	**real-time forwarding control**	
ST 15	**on-board storage and retrieval**	
ST 17	**test**	
…		

A simplification of the PUS A large data transfer that offers only means to upload and download CCSDS packets of maximum 65 KB.

No retransmission

ST20 – parameter management

Standard Service Types		
…		
ST 18	**on-board control procedure**	
ST 19	**event – action**	
ST 20	**parameter management**	√
ST 21	**telecommand sequencing**	
ST 22	**position-based scheduling**	
ST 23	**file management**	

+/- ready

The parameter management service relates to the management of on-board parameters, providing the capability:

- to read their current values, and
- to set new values.

In the context of this service, the parameter definition consists of:

- the mapped on-board memory address,
- the parameter type code (PTC), and
- the parameter format code (PFC).

ST21 – telecommand sequencing

Standard Service Types		
…		
ST 18	**on-board control procedure**	
ST 19	**event – action**	
ST 20	**parameter management**	
ST 21	**telecommand sequencing**	√
ST 22	**position-based scheduling**	
ST 23	**file management**	

The telecommand sequencing service provides the capability to release, one by one, the telecommands contained in an on-board sequence of telecommands.

Within a telecommand sequence, the delay between the release of a telecommand and the release of the next telecommand can be specified.

Several telecommand sequences can be running in parallel.

The service provides the capability to load a telecommand sequence from a file stored on-board or directly from ground.

This service is an *extension of the ground monitoring and control*.

ST22 – position-based scheduling

Standard Service Types		
…		
ST 18	**on-board control procedure**	
ST 19	**event – action**	
ST 20	**parameter management**	
ST 21	**telecommand sequencing**	
ST 22	**position-based scheduling**	√
ST 23	**file management**	

The (orbit) position-based scheduling service provides the capability to command on-board application processes using requests pre-loaded on-board the satellite and released when the Spacecraft reaches the associated position on the orbit.

It supports the concept of sub-schedules and groups.

This service is an *extension of the ground monitoring and control*.

ST23 – file management

Standard Service Types		
…		
ST 18	**on-board control procedure**	
ST 19	**event – action**	
ST 20	**parameter management**	
ST 21	**telecommand sequencing**	
ST 22	**position-based scheduling**	
ST 23	**file management**	√

The file management service provides the capability to manage on-board file systems and files.

File systems can either be:

- flat, where directory structures are not supported, or
- structured, where files are stored within directories.

Two subservices are defined for the file management service:

- The file handling subservice provides an interface to the on-board file handling system and includes requests for file management actions such as deleting a file, copying a file, creating a directory.
- The remote file copy subservice copies files, in either direction, between an on-board file system and a remote file system (e.g. on the ground). *This implies that the subservice provides a limited interface to a dedicated file transfer layer that can uplink and downlink files using, for example, the CCSDS CFDP protocol.*

The file management service is not concerned with the contents of the files that it manages.

2015 – Processing public review output

Discrepancy Review Reports	
2014 public review	464 DRR *by Agencies* 156 DRR *by Industry*
+ in 2015	44 DRR *by WG*
	= *664 DRR*

Criticality
72 major
413 minor
179 editorial

October 2015 status
623 disposed
41 opened

cl. 5	The PUS Foundation Model	72	ST 13	large packet transfer	23
ST 1	request verification	16	ST 14	real-time forwarding control	4
ST 2	device access	20	ST 15	on-board storage & retrieval	36
ST 3	housekeeping	43	ST 17	test	2
ST 4	parameter statistics reporting	8	ST 18	on-board control procedure	41
ST 5	event reporting	6	ST 19	event – action	12
ST 6	memory management	44	ST 20	parameter management	10
ST 8	function management	3	ST 21	request sequencing	23
ST 9	time management	8	ST 22	position-based scheduling	24
ST 11	time-based scheduling	38	ST 23	file management	40
ST 12	on-board monitoring	58			483

2014 Public Review main outputs

- size of the PUS-C standard and large number of requirements!
- need for more background information
- need for better explanations of the PUS foundation model, e.g.:
    a. how do services and subservices relate to application processes
    b. how do service messages (requests and reports) relate to TC/TM packets
    c. what is the exact meaning of an instruction compared to a request, a TC packet, activities
- need to simplify the ground segment model just introduce ground application processes
- improve consistency with ECSS-E-ST-70 standards E-70-01 OBCP steps, E-70-11, E-70-31
- need to simplify the way the minimum and additional capabilities are introduced
- need to offer more tailoring flexibility e.g. for the device access service type
- …

2014 Public Review main outputs, cont. 1

- avoid proliferation of type-specific requirements in clause 6 by generalizing in clause 5
- avoid inconsistencies in modelling requests and instructions e.g. "all" sometimes modelled as a specific request, sometimes as an instruction using "n = 0"
- allows utilization of CRC checksum as an alternative to ISO checksum
- allows CRC and ISO checksum on more than 16 bits
- add state machines and sequence diagrams
- ensure independencies between service types caused the real-time forwarding control service type to be completely remodelled
- etc.

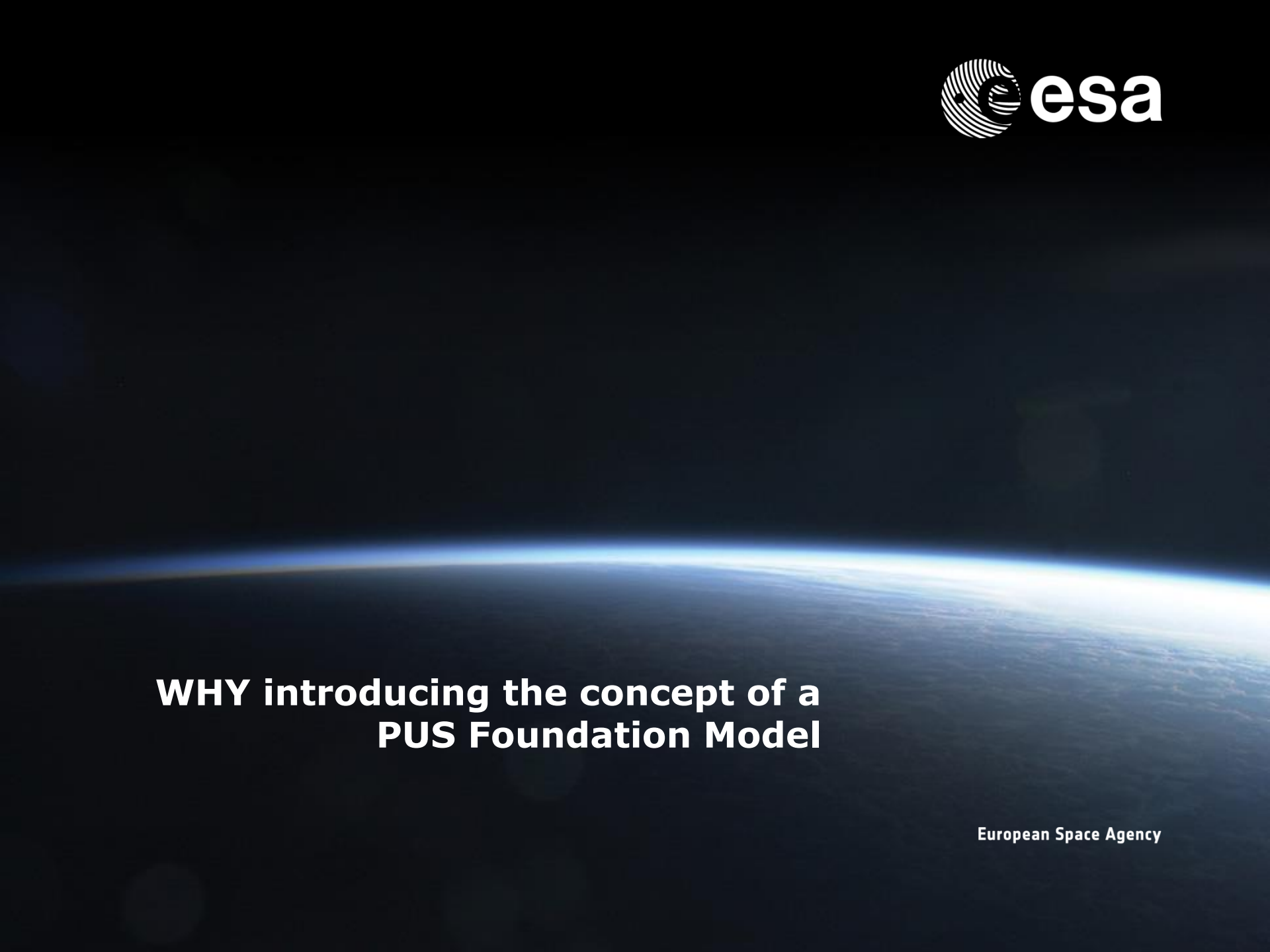
not to forget…

PUS-C mainly only addresses the on-board capabilities

an assumption is made that generic ground segments will implement all standardized PUS service types' & service deployment's capabilities

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



WHY introducing the concept of a PUS Foundation Model

European Space Agency

The PUS-C Standard

An ECSS drafting exercise supported by

a formal information modelling technique *the fact based modelling*

with the aims

- to remove the PUS-A ambiguities, starting from properly introducing the PUS concepts *including defining the corresponding terms and definitions, the relations between these concepts, the constraints that apply, etc.*
- to ensure the overall consistency of all system and interface requirements
- to ease the tailoring of the PUS by the mission, i.e. establishing rules:
  - for selecting the subsets of relevance
  - for adding new capabilities

but also

to enable the automatic generation of software and documentation !

PUS Foundation Model - Requirement View

Service type abstraction

5.3.1 General

a. Each service type shall be uniquely identified by exactly one service type name.

b. Each service type shall be uniquely identified by exactly one service type identifier that is an unsigned integer greater than or equal to 1 and less than or equal to 255.

NOTE The service type identifiers are used in the telemetry packet secondary header (refer to clause 7.4.3.1) and in the telecommand packet secondary header (refer to clause 7.4.4.1), together with a message subtype identifier to uniquely identify a message type.

c. Each standard service type shall have a service type identifier less than or equal to 127.

NOTE The standard service types are specified in the different versions of this Standard. When mission specific functionalities, identified by a mission specific service type, are considered adequate for being standardized, a new standard service type is created. When a standard service type is no longer considered adequate for remaining a standard, that service type is removed from the Standard; its service type identifier is not reused.

d. Each mission specific service type shall be associated with a service type identifier greater than or equal to 128.

5.3.2 Subservice type

a. Each service type shall define at least one subservice type.

NOTE This Standard introduces the concept of subservices that group and isolate the functions of a service.

b. Each subservice type shall be defined by exactly one service type.

c. Each subservice type shall be uniquely identified by exactly one subservice type name.

d. For each subservice type, whether the realization of that subservice type is implicitly required for each realization of the service type or required by tailoring shall be declared when specifying that subservice type.

Service abstraction

5.4.1 Introduction

The services are functional entities that involve both ground elements and on-board elements.

A service is composed of one or more subservices. Each subservice involves:

- one or more subservice users, each one hosted by an application process that resides on-ground or on-board, and
- exactly one subservice provider that is usually hosted by an on-board application process.

The communication between the subservice entities (i.e. a subservice user and a subservice provider) consists of exchanging messages between these entities. When messages are exchanged between the ground segment and the space segment, these messages are transported in CCSDS packets as specified in clause 7.

5.4.2 Application process

5.4.2.1 General

a. Each application process shall either be:

1. an on-board application process, or
2. a ground application process.

b. Each application process shall have exactly one application process identifier.

c. The application process identifier shall be used to uniquely identify the destination of any request and the source of any report.

NOTE This Standard acknowledges that the same application process identifier can be used to identify several application processes. This is for example the case during the space system development where different representations of a given application process are used, e.g preliminary versions (e.g. simulations) or final version of an application process, but also during operations, e.g. in case of cold redundancy.

d. For each report that it generates, each on-board application process shall time tag that report using the on-board reference time.

e. For each application process, whether that application process time-tags the reports before collecting the values of the constituting parameters or after shall be declared when specifying that subservice.

PUS Foundation Model – FBM View

ECSS requirements

5.3.1 General

a. Each service type shall be uniquely identified by exactly one service type name.

b. Each service type shall be uniquely identified by exactly one service type identifier that is an unsigned integer greater than or equal to 1 and less than or equal to 255.

c. Each standard service type shall have a service type identifier less than or equal to 127.

d. Each mission specific service type shall be associated with a service type identifier greater than or equal to 128.

5.3.2 Subservice type

a. Each service type shall define at least one subservice type.

NOTE This Standard introduces the concept of subservices that group and isolate the functions of a service.

b. Each subservice type shall be defined by exactly one service type.

c. Each subservice type shall be uniquely identified by exactly one subservice type name.

d. For each subservice type, whether the realization of that subservice type is implicitly required for each realization of the service type or required by tailoring shall be declared when specifying that subservice type.

ORM graphical view

NORMA verbalization view

Assertions

a. Each service type has exactly one service type name.
b. Each service type name is of exactly one service type.
c. Each service type has exactly one service type ID.
d. Each service type ID is of exactly one service type.
e. The possible values of service type ID are at least '1' to at most '255'.

Derivations

1. Each standard service type is by definition some service type that has some service type ID where the possible values of that service type ID are at least '1' to at most '127'.
2. Each mission specific service type is by definition some service type that has some service type ID where the possible values of that service type ID are at least '128' to at most '255'.

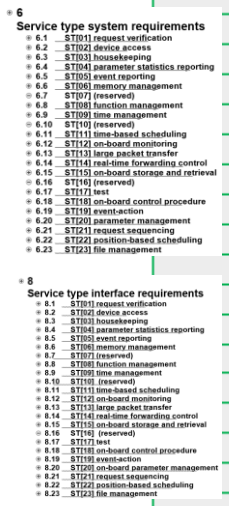
PUS Foundation Model – FBM View

Service type **is an entity type**.
Reference Scheme: service type has service type ID.
Reference Mode: ServiceTypeID.

a. Each service type has exactly one service type name.
b. Each service type name is of exactly one service type.

Table 4-1: The Standard service types

ServiceTypeID	ServiceTypeName
1	request verification
2	device access
3	housekeeping
4	parameter statistics reporting
5	event reporting
6	memory management
8	function management
9	time management
11	time-based scheduling
12	on-board monitoring
13	large packet transfer
14	real-time forwarding control
15	on-board storage and retrieval
17	test
18	on-board control procedure
19	event-action
20	parameter management
21	request sequencing
22	position-based scheduling
23	file management



verbalize

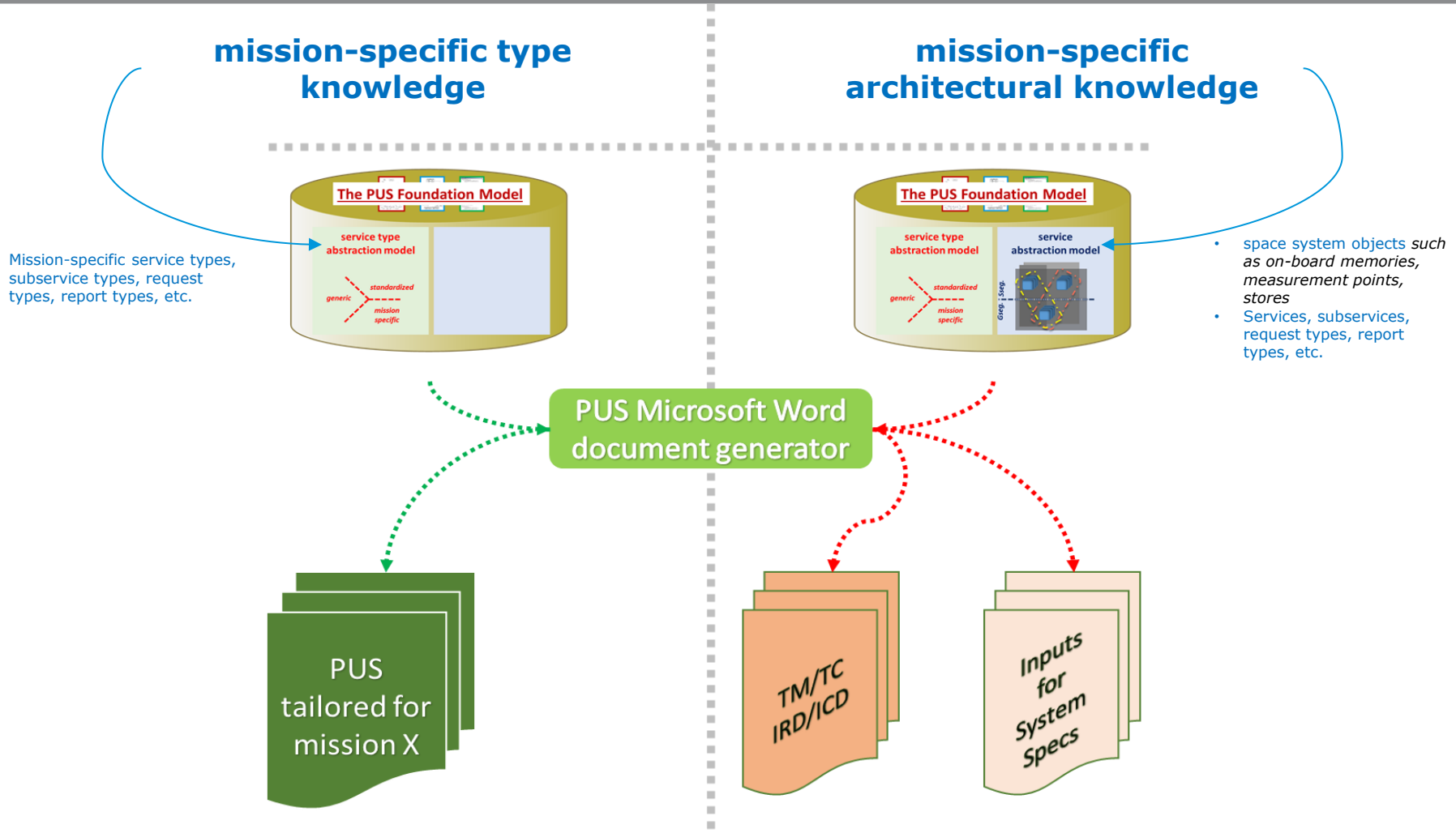
populate

automate

Service type '1' has service type name 'request verification'.
Service type '2' has service type name 'device access'.
Service type '3' has service type name 'housekeeping'.
Service type '4' has service type name 'parameter statistics reporting'.
Service type '5' has service type name 'event reporting'.
Service type '6' has service type name 'memory management'.
Service type '8' has service type name 'function management'.
Service type '9' has service type name 'time management'.
Service type '11' has service type name 'time-based scheduling'.
Service type '12' has service type name 'on-board monitoring'.
Service type '13' has service type name 'large packet transfer'.
Service type '14' has service type name 'real-time forwarding control'.
Service type '15' has service type name 'on-board storage and retrieval'.
Service type '17' has service type name 'test'.
Service type '18' has service type name 'on-board control procedure'.
Service type '19' has service type name 'event-action'.
Service type '20' has service type name 'parameter management'.
Service type '21' has service type name 'request sequencing'.
Service type '22' has service type name 'position-based scheduling'.
Service type '23' has service type name 'file management'.

PUS Foundation Model – FBM View

PUS tailored for a mission



Using ASN.1 to automate SW production

ASN.1 - Abstract Syntax Notation

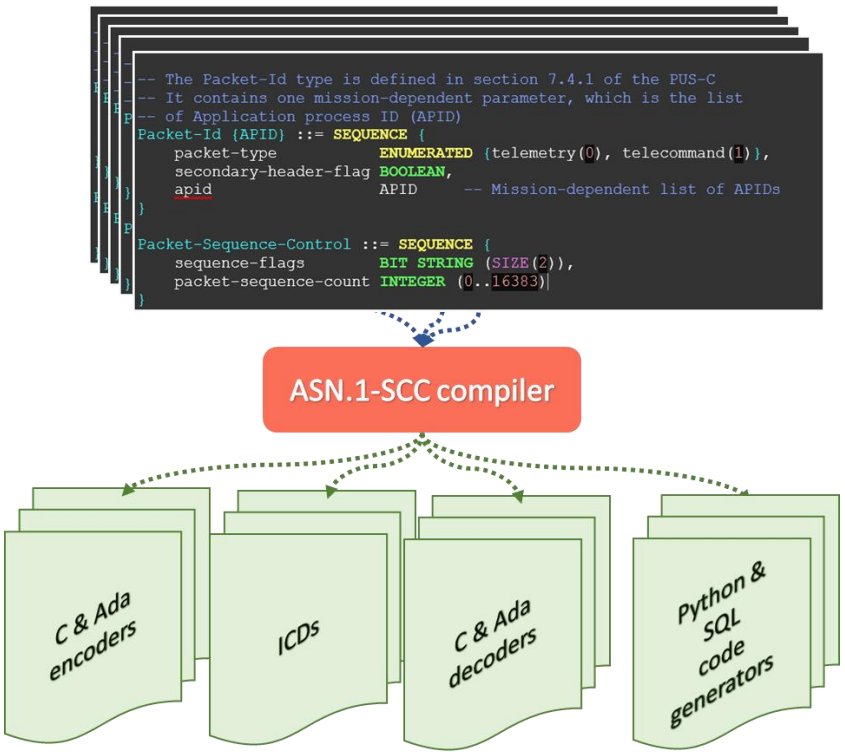
A formal language to describe data structures and physical encodings for the purpose of communication between heterogeneous networks

used at ESA to automate the production of ground and on-board PUS packets encoding and decoding functions and corresponding documentation

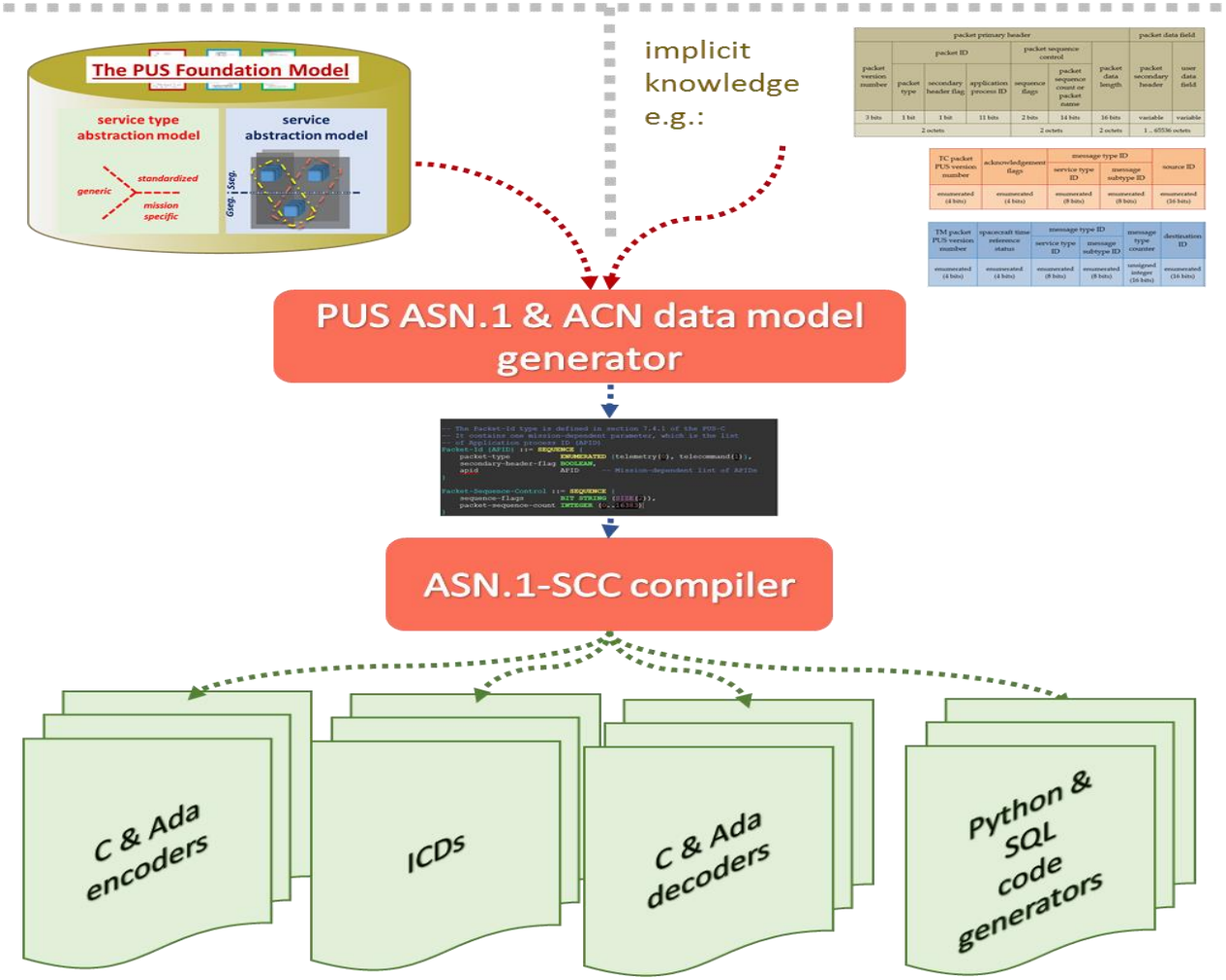
e.g.:

- *PROBA-3 ASPIICS payload*
- *ESTEC/TEC-SW TASTE toolkit, on-board software development platform*

ASN.1-SCC compiler requires as inputs configuration files describing the PUS data structures used by the mission, specified in the ASN.1/ACN languages



ASN.1 & ACN data model generator



Using the PUS Foundation Model in support to PUS compliant Missions

ESA Intended Invitation to Tender

15.132.06

deployment of PUS-C standard in projects

supported by an automatic generation toolset

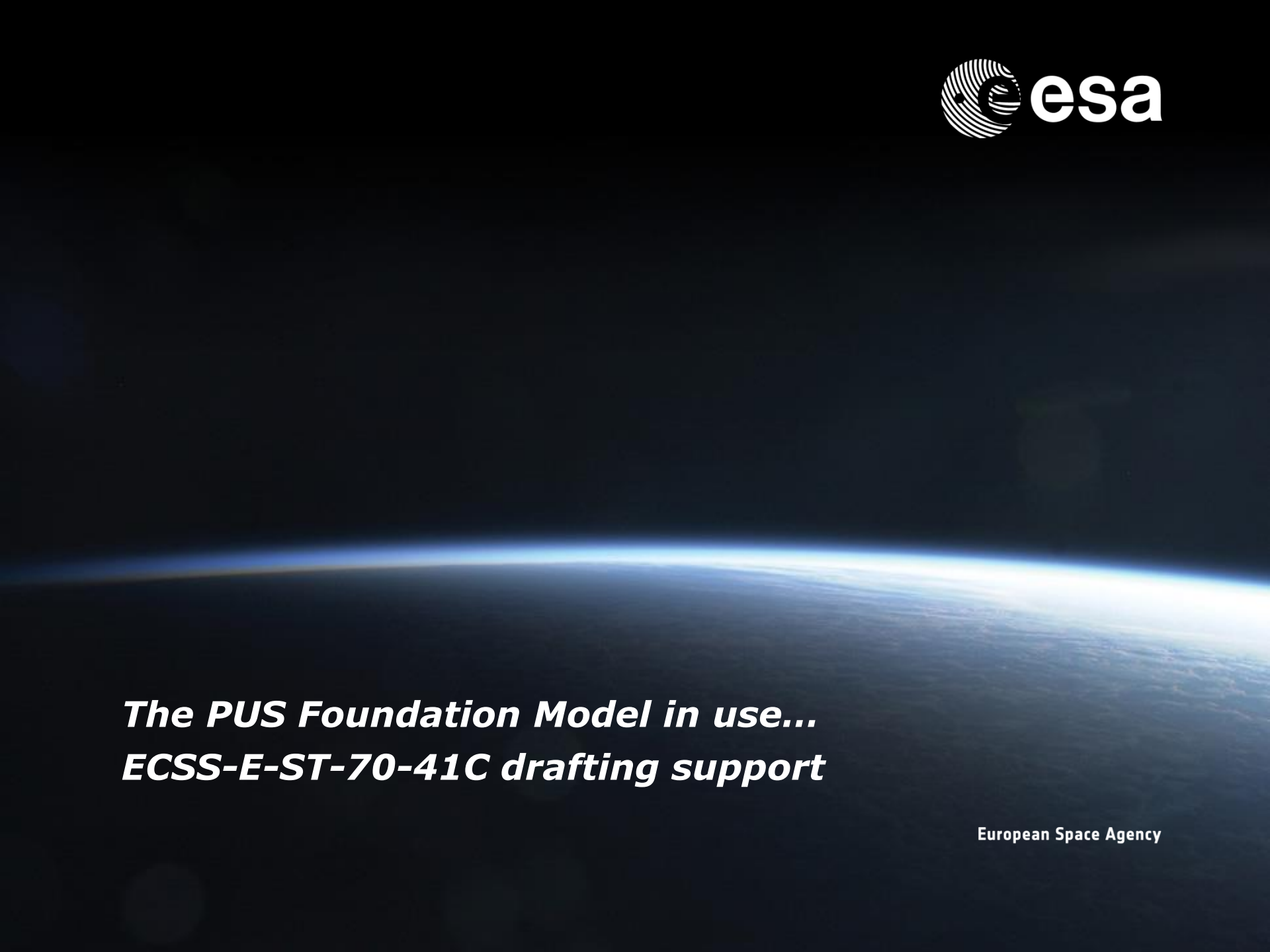
TRP - Technology Research Program

open competition

Objectives:

- to assess means to optimize the use of the new PUS-C by ESA project
- to prototype the development of a code generator framework able to automate the production of PUS products

ITT publication foreseen in December 2015



The PUS Foundation Model in use…
ECSS-E-ST-70-41C drafting support

European Space Agency

The PUS-FDB proto

PUS-FDB proto

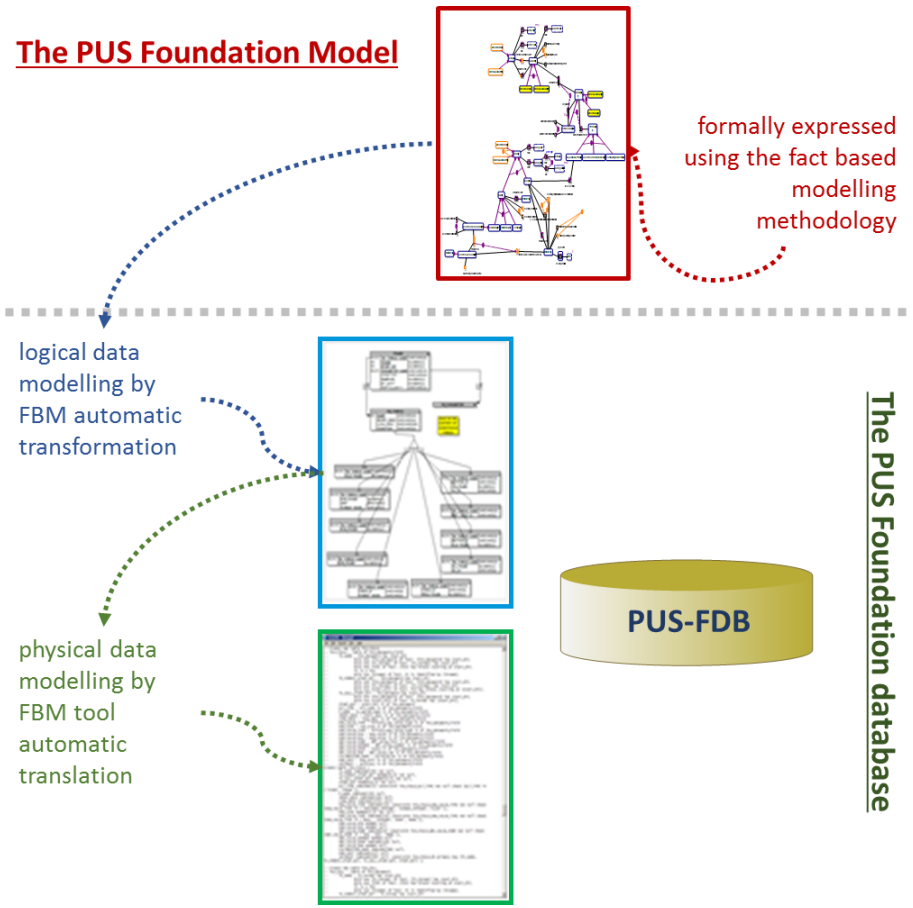
a first prototyped implementation of the PUS Foundation Database

---

Objective ECSS-E-ST-70-41C WG drafting support by verifying the consistency of the system and interfaces requirements related to message (request and report) types

---

1. PUS-FDB conceptual data model a subset of ECSS-E-ST-70-41C Clause 5
2. Automatic *NORMA* transformation to:
   a) logical data model – *Relational*
   b) physical data model – *ORACLE SQL*



The PUS-FDB proto, cont. 1

PUS-FDB proto

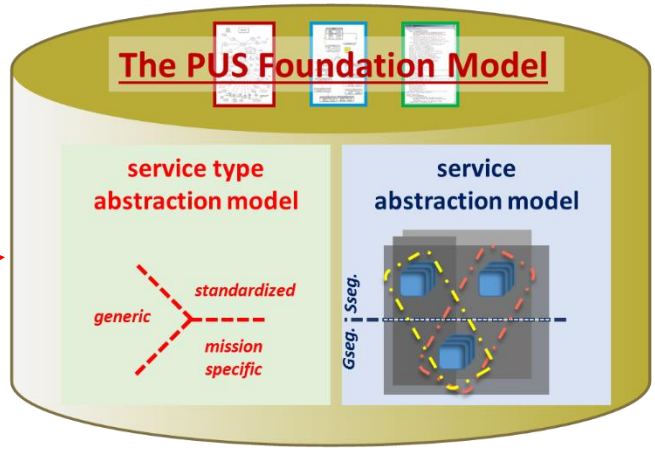
a first prototyped implementation of the PUS Foundation Database

---

Objective ECSS-E-ST-70-41C WG drafting support by verifying the consistency of the system and interfaces requirements related to message (request and report) types

---

1. PUS-FDB proto reuses a subset of the conceptual data model
2. NORMA automates the production of the logical *relational* data model
3. NORMA automates the production of the physical *SQL for Oracle RDBMS* data model
4. Populate the PUS-FDB with the *standardized message types knowledge*



populate

The PUS-FDB proto, cont. 2

PUS-FDB proto

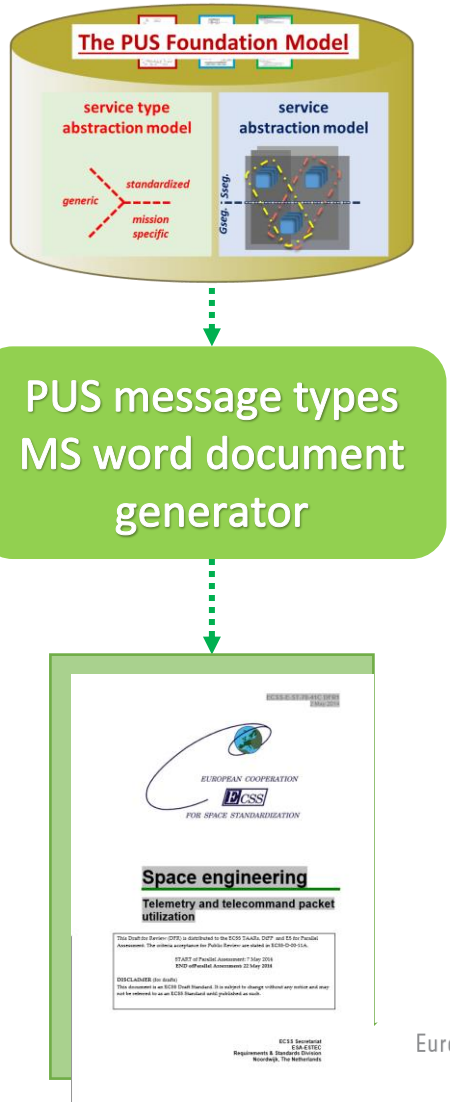
a first prototyped implementation of the PUS Foundation Database

---

Objective ECSS-E-ST-70-41C WG drafting support by verifying the consistency of the system and interfaces requirements related to message (request and report) types

---

1. PUS-FDB proto reuses a subset of the conceptual data model
2. NORMA automates the production of the logical *relational* data model
3. NORMA automates the production of the physical *SQL for Oracle RDBMS* data model
4. Populate the PUS-FDB with the standardized message types knowledge
5. *Automate the production of*
   a) *message type system requirements sub-clauses* *of clause 6*
   b) *Message type interface requirements sub-clauses* *of clause 8*
   c) *Annex D – Message types (with tailoring conditions) summary*



The PUS-FDB proto, cont. 3

1.1.3 On/off device

1.1.3.1 Distribute on/off device commands

a. The device access subservice capability to distribute on/off device commands shall be declared when specifying that subservice.

NOTENOTE 1 The corresponding requests are of message type "TC[2,1] distribute on/off device commands".

NOTE 2 For that declaration, refer to requirement 6.2.3a.

b. Each request to distribute on/off device commands shall contain an ordered list of one or more instructions to distribute an on/off device command.

NOTE The delay to apply between two consecutive instructions is dependent on the spacecraft on-board architecture.

c. Each instruction to distribute an on/off device command shall include:

1. the device address.

d. The device access subservice shall reject any request to distribute on/off device commands if:

1. that request contains an instruction that refers to an unknown device address.

e. For each request to distribute on/off device commands that is rejected, the device access subservice shall generate a failed start of execution notification.

f. For each request to distribute on/off device commands that contains only valid instructions, the device access subservice shall execute the related instructions in the order of their appearance in that request.

g. For each valid instruction to distribute an on/off device command that is not rejected, the device access subservice shall:

1. distribute the related on/off command to the related device address.

1.1.4 Register

1.1.4.1 Distribute register load commands

a. The device access subservice capability to distribute register load commands shall be declared when specifying that subservice.

NOTENOTE 1 The corresponding requests are of message type "TC[2,2] distribute register load commands".

NOTE 2 For that declaration, refer to requirement 6.2.3a.

b. Each request to distribute register load commands shall contain an ordered list of one or more instructions to distribute a register load command.

c. Each instruction to distribute a register load command shall include:

1. the register address;
2. the data for the register fields.

d. The device access subservice shall reject any request to distribute register load commands if any of the following conditions occurs:

1. that request contains an instruction that refers to an unknown register address;

1.2. that request contains an instruction that fails its register consistency checks.

e. For each request to distribute register load commands that is rejected, the device access subservice shall generate a failed start of execution notification.

NOTE A partial load can result in an unknown or inconsistent device status.

f. For each request to distribute register load commands that contains only valid instruction, the device access subservice shall execute the related instructions in the order of their appearance in that request.

g. For each valid instruction to distribute a register load command, the device access subservice shall:

1. distribute the command to the register.


esa
Any questions?
European Space Agency

see also

- http://www.factbasedmodeling.org
- http://www.orm.net
- http://www.ormfoundation.org
- https://en.wikipedia.org/wiki/Abstract_Syntax_Notation_One
- https://github.com/ttsiodras/asn1scc