Reaching into space TOGETHER

Deployment of the PUS-C Standard in Projects supported by an Automatic Generation Toolset (PUS-Gen)

Maxime Perrotin, Serge Valera (ESA) Michal Kurowski (N7Space) Arnaud Bourdoux (Spacebel)

ADCSS 2018



Outline

- PUS-Gen Context and Overview
- Utilization and extension of PUS-C foundation model
- PUS-Gen toolset
- Outcome and conclusion



2

Project Context

- PUS-C standard released in April 2016
- More than 600 pages of normative and informative material
- Formal representation of the PUS concepts to avoid misinterpretations
- Cleanup and simplification of the PUS-A services and introduction of new types of services



Project Context

The PUS-C contains more than 3000 requirements.

How to

- Verify the PUS-C overall consistency ?
- Make sure it fulfils operational projects needs ?



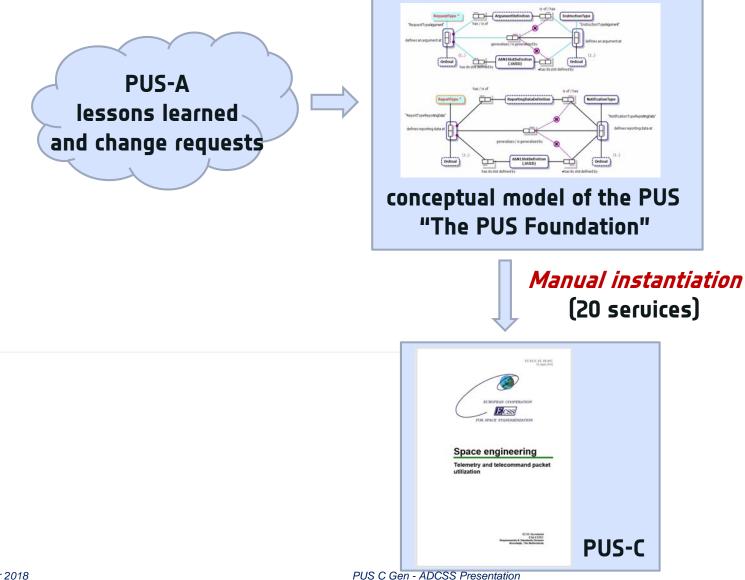
Project Objectives

- Verify, validate and extend the **PUS-C foundation model**
- In support to Projects, develop
 - Tools to support the **tailoring of PUS-C**
 - Document generators to produce mission-specific ICDs
 - Code generators to automate on-board and ground software production
- Model with formal state machines the PUS-C Services **behavior**
- Verify the developed tools using a realistic **use case**



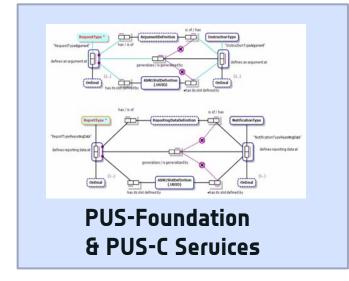


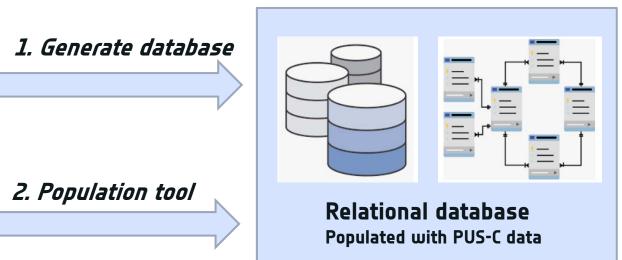
PUS-C Creation Logic





Study logic (1)







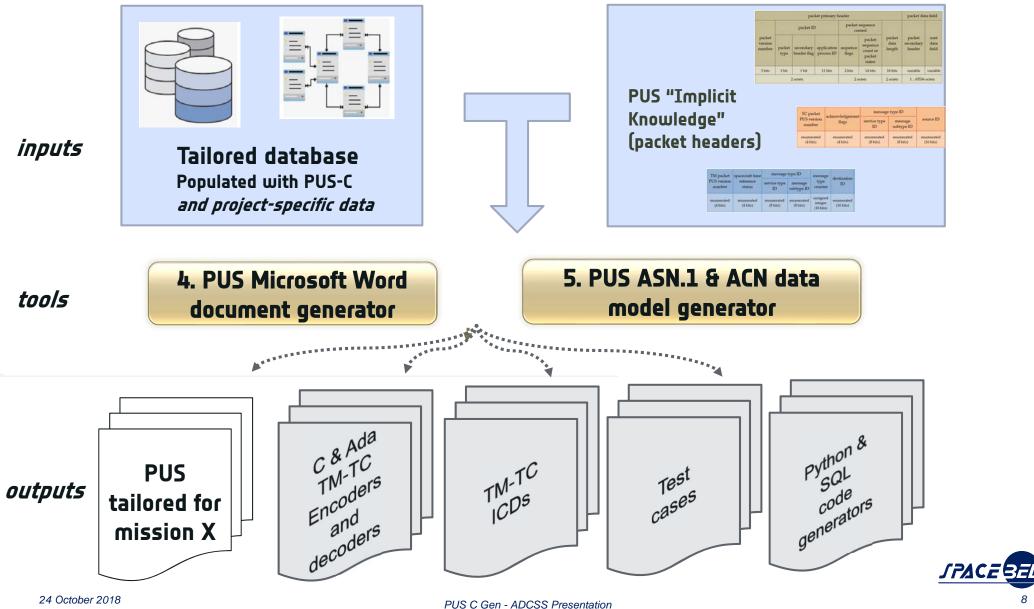
3. Tailoring tool:

- Add/Remove services
- Set project-specific data





Study logic (2)



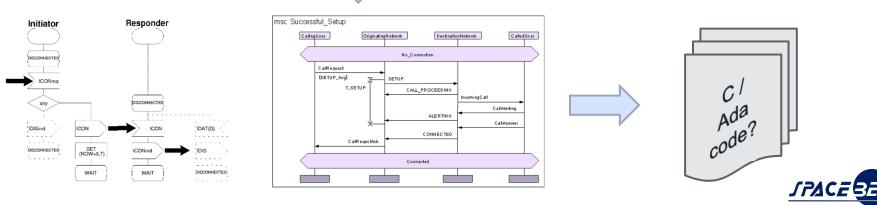
Copyright © 2016 by SPACEBEL – All rights reserved

Study logic (3)

PUS addresses the packet contents but also the services behavior

Can this also be formalized and automated ?

- e. For each valid instruction to enable the functional monitoring function, the functional monitoring subservice shall:
 - 1. set the FMON function status to "enabled";
 - 2. for each functional monitoring definition that is enabled:
 - (a) set its FMON checking status to "unchecked";
 - 3. start immediately the monitoring of the enabled functional monitoring definitions.

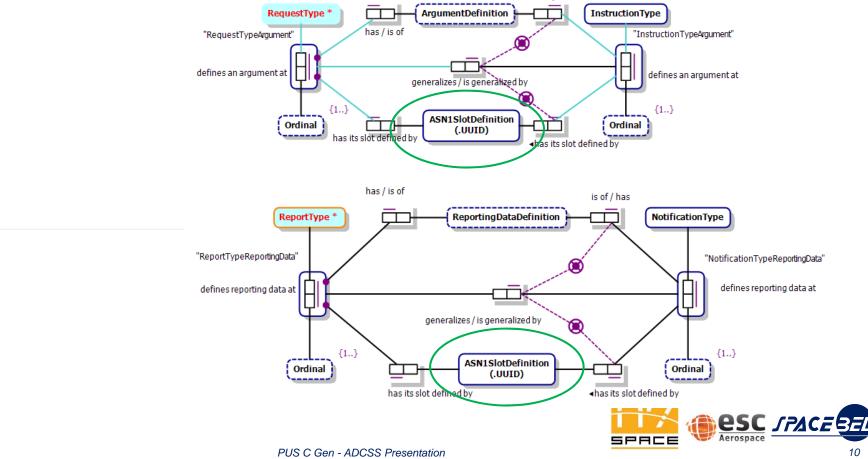


PUS C Gen - ADCSS Presentation

PUS foundation model (Extensions)

Baseline model used for PUS-DBI software

 Contains ASN.1 references which are necessary for the ASN.1 but may be reconsidered for other uses of the PUS-DBI



PUS C Toolset

The toolset consists of 3 applications that work together:

Population Tool

population definition and tailoring

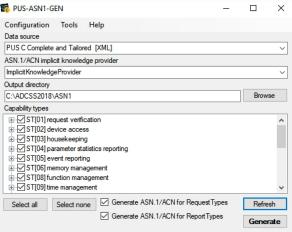
DOC-GEN

ECSS-E-ST-70-41C compatible document generation

ASN1-GEN

population tailoring and ASN.1/ACN generation

PUS-C Population Tool		- 🗆 ×	🔀 PUS-DOC-GEN — 🗆 >	<
Eile Help Service capabilities	Detail: [capability to acquire data from physical devices (TC[2.8] and TM[2.9])]	Reference data values	Configuration Tools Help	
encice capabilities ■ 517011 equate verterStation ■ 517021 equate verterStation ■ 517021 equate verterStation ■ Corcup(41 lead core of capability for - capability to accure date from top - capability to accure date from top - capability to accure date from top - capability to datebulk cryb(10 date - capability to datebulk cryb(10 datebulk - capability to d	Detail: populativy to acquire data from physical devices (TC(2.8) and TM(2.9)) General: Request Report: Renerquired capabilities Inplied capabilities E * * Capability Name Inclusion: Cata from physical devices Inclusion: prove the minimum applicability constraint. Image: field and Themaciton Type	Perference data values System Object Types System Objects Enumanidad Values System Object Types System Objects CCSDS Packet Types Astract Types ASN Type Definations Astract Types ASN Type Definations Astract Types System Control Astract Types Astract Types System Control Astract Types Astract Types Types Astract Types Types Astract Types Types Astract Types Types Communication Communication Distance Control Astract Control Astract Control Astract Control Distance Control Astract Control Control Control Distance Control Control Control Distance Control Control Control Distance Control Distance Control Control Distance Control Control Distance Cont	Configuration Tools Help Data source MS-2 Population [XML] Document template Standard template Output document OpenXML C:\ADCSS2018\Docs\FormationManagement.docx Browse Service types STI01] Inquest verification STI02] device access STI03] housekceping STI03] housekceping STI03] function management StI05] event reporting STI03] function management Select all Select none Generated document elements Clause 8 "Service type interface requirements" Clause 8 "Service type interface requirements" Clause 8 "Service type interface requirements" Arnex C "Symany of the pends for PUS standard services" Arnex D "System and interface specification index"	





PROBA3 "themed" Service Type – Formation Management

A high level overview of the functional requirements:

- enter formation at desired offset
- leave formation
- report formation status
- **Reporting optional**

Algorithm – TBD (different missions can have different ones)

Don't define the requirements directly. Capture the essence of the Formation Management Service Type (a class of services) using the Foundation Model. PUS C compliant requirements will be generated automatically.



- Define Service Type (with name and scope)
- Define Subservice Type (with name and scope)

a Source: Formation Management [XML]			✓ Q R
rice capabilities	Detail: IEXT[177] formation management]	_	Reference data values
EXT[177] formation management	🐻 Add Subservice Type [Service Type: EXT[177] formation management]	×	System Object Types System Objects
	Subservice Name		Enumerated Value Types
	static formation management		CCSDS Packet Types
	Subservice Scope		Abstract Types ASN.1 Type Definition
		ai	
	Is mandatory for defining Service Type Realisation is part of the inclusive-or service dependency Realisation is required by tailoring		
	Allows multiple subservice provider realisations within a single service		
	OK Cance		
	Demands that all subservice providers are hosted by a single application process		





- Define known types (vector, status) using ASN.1/ACN
- Define types to be tailored (algorithm) using name and PTC

a Source: FormationManagem	nt [XML]	V 🗘 Rela
ice capabilities	ASN.1 Type Definition -	lata values
EXT[177] formation managemer L Static formation managemer		ject Types System Objects
	General Abstract Types Tailoring Parameters	d Value Types
	ASN.1 Type Definition	cket Types
	Vector3d	vpes ASN.1 Type Definitions
	ASN.1 Definition Block	1
	x REAL, y REAL, z REAL } ACN Definition Block Vector3d [] { x [encoding IEEE7754-1985-32], y [encoding IEEE7754-1985-32], z [encoding IEEE7754-1985-32] }] I	
	OK Cancel	
	Realisation is part of the inclusive-or service dependency	<u>1</u>
	Realisation is required by tailoring	

Data changed 📕 🔹



14

• Define Capability Types – with Transaction Types, Instruction Types, Instruction Type Arguments...

PUS-C Population Tool		- 0			
File Help					
Data Source: FormationManagement [XML]		~ ₹ 2 Rel			
rvice capabilities Detail: [capability to report formation status (TC[177,10] and TM[177,11])] Reference data values					
EXT[177] formation management Static formation management subservice -capability to enterformation (TC[177,1]) -capability to leave formation (TC[177,2]) -capability to report formation status (TC[177,10] and TM[177,11])	General Request Report Prerequired capabilities Implied capabilities Excluded capabilities General Instructions Request Arguments On Board Conditions Before Instruction Functionalities After In Name report formation status	System Object Types System Objects Enumerated Value Types CCSDS Packet Types Abstract Types Abstract Types FormationFlyingStatus Vector3d			





PUS C Toolset DEMO

Video:

- Service Type definition (already presented skip)
- Document generation
- Document presentation
- ASN.1/ACN tailoring
- Integration of the custom Service Type with the standard
- ASN.1/ACN generation
- HTML ICD and C code generation
- HTML ICD presentation
- Generated ASN.1 presentation in asn1scc.IDE



PUS C Toolset

Benefits:

- Support for ensuring standard document consistency and PUS Foundation Model compliance
- Automated documentation generation SDD, SRS, ICD
- Automated flight code generation (C and Ada)
- Automated test code generation (Python)

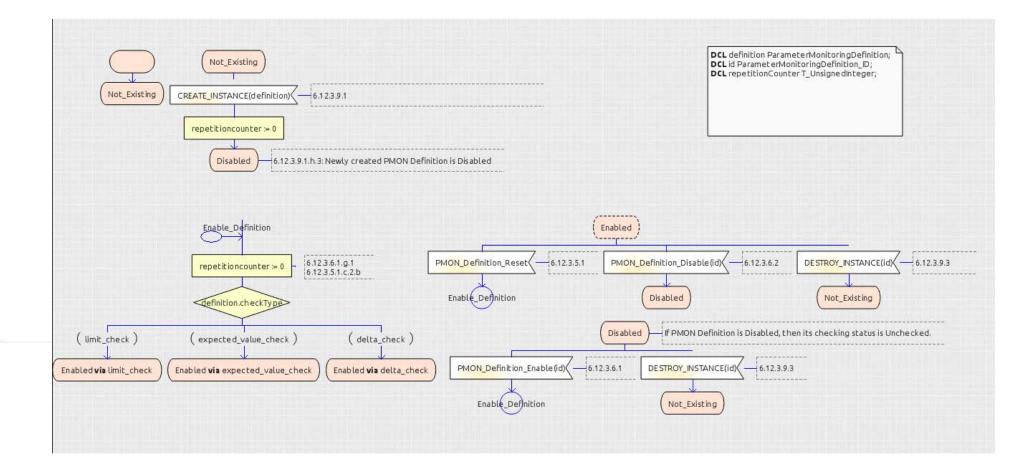
Parts of the process already deployed:

- PROBA3 (instrument payload BSW and ASW)
- ARM BSP with CANOpen (BSW)

Possibly more applications in the future – depending on the ASN1SCC/TASTE ecosystem development



Behavioral modelling (in progress)





Modelling activity outcomes

Lessons learned document produced gathering:

- Valuable feedback on using the NORMA tool for performing conceptual modelling
- 21 change requests towards the PUS-C standard, out of which
 - 7 are considered major (The specification has a functional problem)
 - 7 are considered minor (The specification lacks internal consistency, or is not covered by the foundation)
 - 7 are considered editorial only.

Relational database directly generated from the ORM model to support the toolset development.





Conceptual Modelling in ECSS

European Space Agency

24 October 2018

PUS C Gen - ADCSS Presentation

ECSS-E-ST-70 standards status

- E-ST-70-41C changes implies revisiting all E-ST-70C standards, i.e.:
- E-ST-70-11C Space segment operability
- E-ST-70-31C Monitoring and control data definition
- E-ST-70-32C Test and operations procedure language
- E-ST-70-01C Spacecraft on-board control procedure
- E-ST-70C Ground systems and operations
- → 2017-2019 change requests and lessons learned production





E-TM-10-23A Space System Data Repository

- objective: "Semantic Interoperability"
- pre-requisite:
 - modelling language (and tool) for the WHAT !
 - Language: Object Role Modelling/Fact based Modelling
 - Tool: NORMA Pro with FAMOUS-2 extension
- approach:
 - "global overall space system conceptual model"

= "Space System Ontology"

 conceptual modelling "products" by selecting from the global model



Conceptual modelling in ECSS

2014-2016 PUS Foundation

first ECSS use of "fact based modelling"

2017-2018 ECSS MasterDB ECSS requirement management system See EMITS Call for Information Information Day at ESTEC, on 30/10/2018

2018-2019 TRP - Extending the "PUS Foundation" to produce the "Monitoring and Control Foundation"

preparing for the update of:

ECSS-E-ST-70-31 Monitoring and control data definition ECSS-E-ST-70-32 Test and Operations procedure language

2019-2020 TRP – Space System Ontology Development

activity proposal will be published in EMITS

