MBSE2021
MODEL EXCHANGE FOR SOFTWARE ENGINEERING (MODEX)

September 29-30, 2021

Speakers
Elena Alaña (ealana@gmv.com)
Tiago Jorge (tiago.jorge@gmv.com)

OBJECTIVE

Definition of a global picture of the **data and models needed around the On-Board Software Reference Architecture (OSRA)** including:

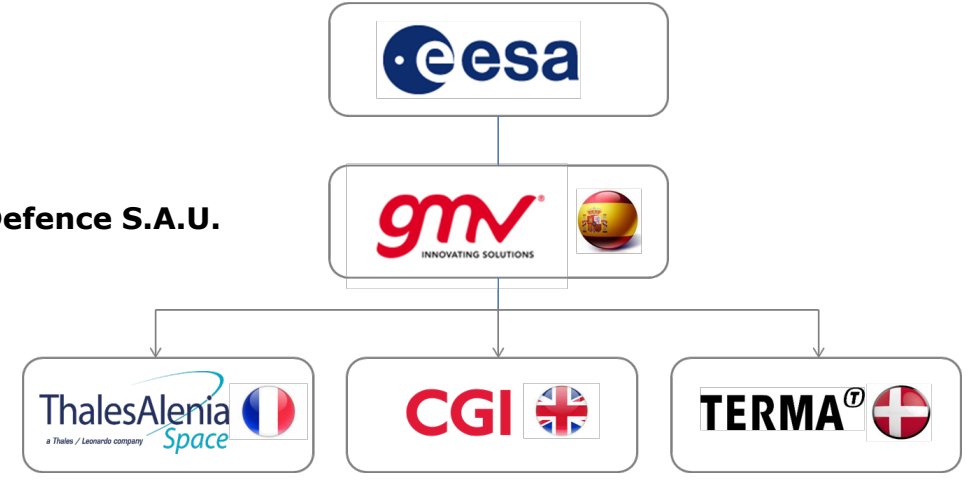
- Their relationship
- Their ownership
- The process to produce and use them
- The need for exchanging within and outside the software domain
- The data exchange items and their associated data exchange formats
- The relationship with the Software Factory

This will represent the **implementation of the *software development process* for the OSRA when a model-based approach is adopted**

⇨ The impact on the ECSS-E-ST-40C standard is analysed

MODEX PROJECT

- Duration: November 2018 – November 2020
- Technical Officer: **Andreas Jung**
- Consortium:
  - Prime Contractor: **GMV Aerospace and Defence S.A.U.**
  - Subcontractors:
    - **Thales Alenia Space France**
    - SCISYS UK Ltd (now **CGI**)
    - **Terma A/S**



ECSS AS BASELINE

- The implementation proposed in MODEX derives from **ECSS-E-ST-40C** in terms of activities and artefacts:
  - The standard is **used as a baseline**
  - MODEX **builds upon it by focusing on the model exchanges** during the software development following a model based approach
- The ECSS-E-ST-40C does not prescribe any particular implementation for the SW engineering life cycle. Hence, the **standard is still applicable**
  - The high-level process does not change, in the sense of WHAT sub processes shall be in place and WHAT shall be produced
  - This study instead prescribes **HOW the process could be implemented following the model-based approach**
- *Product Assurance (PA)* and *Quality Assurance (QA)* processes are not in the scope of the MODEX study, a *similar analysis* could be also conducted for OBSW PA/QA specific processes.

MODEX CONCEPTS

WORK PRODUCT (WP)

Any artefact exchanged among stakeholders during the SW development process when a model-based approach is adopted

EXCHANGE FORMAT

Format used for sharing a WP among stakeholders using different tools that support it
Example: *ReqIf*

SOFTWARE FACTORY

Software Engineering supporting infrastructure that integrates those tools that allow the application of a model-based approach at SW level

DATA HUB

Repository to Exchange information (i.e. WP) with other domains, and to some extend, within the software domain

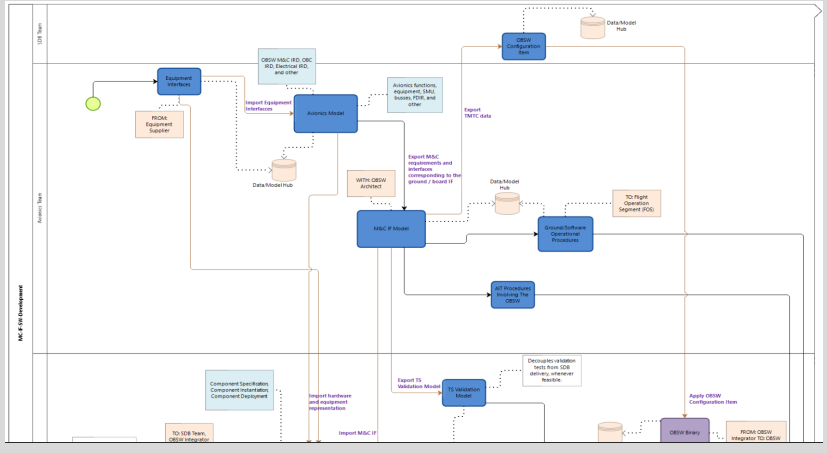
ROLES

Stakeholders involved in the model-based software development process. It also includes roles of other domains that interact with the OBSW

PROCESS FORMALISATION

- Process formalized via:

BPMN 2.0 model
(Business Process Model and Notation)



Work Products Table
Automatically generated from the BPMN model

A	B	D	L	M
WP ID	WP Name	Owner (exchange source)	DRL Item	SW Review
4	AOCS Component Model	OBSW Architect	ICD;SDD	SWRR;PDR;DDR;CDR
18	Component Deployment	OBSW Architect	SDD	PDR;DDR;CDR
19	Component Detailed Design Model	OBSW Component Developer	SDD	DDR
20	Component Instantiation	OBSW Architect	SDD	PDR;DDR;CDR
21	Component Specification	OBSW Architect	ICD;SDD;SCF	SWRR;PDR;DDR;CDR
24	Computational Model	OBSW Architect	SDD	PDR;DDR;CDR
26	EP Configuration	EP Provider	SCF	SWRR;PDR;DDR;CDR
27	EP Detailed Design	EP Provider	SDD	PDR;DDR;CDR
28	EP Implementation Model	OBSW Architect	SDD;SRF	PDR;DDR;CDR
30	EP SRS	EP Provider	SRS;ICD	SWRR;PDR;DDR;CDR
33	FDIR Model	Avionics Team	SRS;SCF	SWRR;PDR;DDR;CDR
37	M&C IF Model	Avionics Team	SSS;ICD	SRR;SWRR;PDR;DDR;CDR;QR;AR
44	OBSW Model	OBSW Architect	SRS;ICD;SDD	PDR;DDR;CDR
45	OBSW SRS	OBSW Architect	SRS;ICD	SWRR;PDR;DDR;CDR

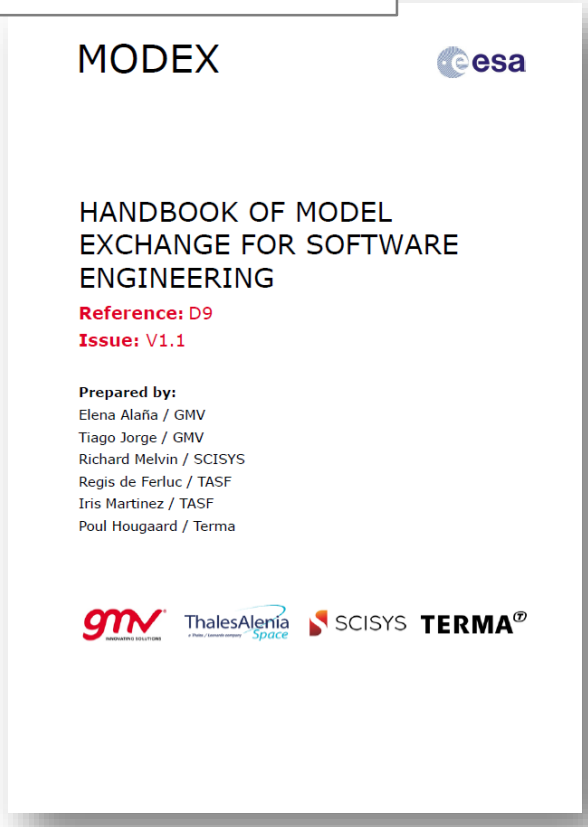
Both are ***consistent*** **and** ***complementary***

PROCESS FORMALISATION

- A ***BPMN 2.0 model***
  - BPMN represents a modelling method for illustrating *business processes* in the form of a diagram similar to a flowchart. It is activity-oriented, this means that the business process is defined as a *sequence of activities (work flow)*.
  - Composed of process diagrams
    - ***Global view*** of the process
    - ***Specific views*** on *Monitoring and Control* and *AOCS subsystem (more possible)*
  - Mainly intended to illustrate the *workflow*, the *dependencies* of Work Products, show the *interactions*, and useful for *communication* purposes
  - Nevertheless, the diagrams only depict a subset of the information to improve their readability
- ***Work Products Table***
  - BPMN model is *annotated* with the list of *properties* for each Work Product (not directly visible in the diagrams)
  - Provides all the *details* of each Work Product (e.g. *exchange formats, tools, owner, etc.*) and is recommended to adjust the information to give a *particular view of interest.* Depending on the stakeholders, they might require the access to the complete information defined in the model easily. Through this table, the reader can extract all data associated to any Work Products directly, e.g. applying filters

HANDBOOK

Self-contained **Handbook**



MODEX

esa

HANDBOOK OF MODEL EXCHANGE FOR SOFTWARE ENGINEERING

Reference: D9

Issue: V1.1

Prepared by:

Elena Alaña / GMV
Tiago Jorge / GMV
Richard Melvin / SCISYS
Regis de Ferluc / TASF
Iris Martinez / TASF
Poul Hougaard / Terma

gmv ThalesAlenia Space SCISYS TERMA

1. Introduction
2. Model-based software development process
   - 2.1. Overview
   - 2.2. ECSS standards
   - 2.3. Roles
   - 2.4. Work Product concept
   - 2.5. Exchange Formats
   - 2.6. Software Factory and Data Hub
3. Impact on ECSS-E-ST-40C
   - 3.1. ECSS-E-ST-40C applicability
   - 3.2. Mapping to the standard
   - 3.3. Contributions to the standard
4. Application of the Software Development process using a model-based approach
   - 4.1. Formalization of the software development process
   - 4.2. Software Factory and Data Hub
   - 4.3. Maturity and assumptions of the model-based process
5. Conclusions

ANNEX A. Software development process when a model-based approach is adopted
ANNEX B. Definition of "Model"
ANNEX C. Data Exchange Format standards
ANNEX D. Software Factory capabilities
ANNEX E. Software Factory interaction with the software development process
ANNEX F. Interactions between the Software Factory and Data Hub
ANNEX G. Demonstator
ANNEX H. Data Exchange Paradigm Assessment

esa gmv ThalesAlenia Space CGI TERMA

WORK PRODUCTS

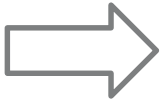
- After consolidation, ***56 WPs*** are identified
  - Models in bold
- List is ***not exhaustive***, but instead tailored to the needs of the activity
  - Other WPs to be considered in the future

WP ID	WP Name	WP ID	WP Name
1	Acceptance Test Report	29	EP Integration Tests
2	AIT Procedures Involving the OBSW	30	**EP SRS**
3	AOCS Algorithms Code	31	**EP SSS**
4	**AOCS Component Model**	32	**Equipment Interfaces**
5	**AOCS Model**	33	**FDIR Model**
6	**AOCS Model** (with design model information)	34	**Ground/SW Operational Procedures**
7	**AOCS OBSW IDS**	35	Interaction Layer
8	AOCS Ref-runs	36	**Logical and Interface Models**
9	**AOCS SRS**	37	**M&C IF Model**
10	**AOCS SSS**	38	Modelling Guidelines
11	AOCS Static Architecture Code	39	OBSW Binary
12	AOCS Unit Test Code	40	**OBSW Configuration Item**
13	**Avionics Model**	41	OBSW Installation Procedures
14	Non-Functional Code	42	OBSW Integration Test Report
15	Comparison Ref-runs With Code-runs	43	OBSW Integration Tests
16	Component Code	44	**OBSW Model**
17	Component Code Skeletons	45	**OBSW SRS**
18	**Component Deployment**	46	**OBSW SSS**
19	**Component Detailed Design Model**	47	Software Validation Report RB
20	**Component Instantiation**	48	Software Validation Report TS
21	**Component Specification**	49	SPRs and NCRs RB
22	Component Unit Test Results	50	SPRs and NCRs TS
23	Component Unit Tests	51	SVS RB
24	**Computational Model**	52	**System Models**
25	EP Code	53	**System Requirements**
26	**EP Configuration**	54	**Test Environment**
27	**EP Detailed Design**	55	**TS Validation Model**
28	**EP Implementation Model**	56	Validation Scripts and Documents

WORK PRODUCTS - DEFINITION

17 properties

WP ID
WP Name
WP Description
Owner (exchange source)
Collaborator (exchange target)
Receiver 1 (exchange target)
Receiver 2 (exchange target)
SW Factory
SW Factory Capability
SW Factory Automation
Software Domain
DRL Item
SW Review
Is Model
Exchanged via Data Hub
Exchange Format
Tool



Property	Description
WP ID	Numeric identifier.
WP Name	The name of the WP.
WP Description	Detailed description.
Owner (exchange source)	The Role that is the main responsible for the definition and production of the WP (e.g. a WP can be defined in collaboration with other roles, see *Collaborator* property). The possible Roles are the following: **Role** AOCS Team Avionics Team DHS Team EP Provider Equipment Supplier Flight Operation Segment OBSW Architect OBSW Component Developer OBSW Development Team OBSW Integrator OBSW Validation Manager SDB Team System Team Test Environment Engineer
Collaborator (exchange target)	A target Role that additionally can collaborate in the definition of the WP (since it is a target Role, it is assumed an exchange with the *Owner* shall take place).
Receiver 1 (exchange target)	A target Role that uses the WP (since it is a target Role, it is assumed exchange(s) with the Owner shall take place).
Receiver 2 (exchange target)	A target Role that uses the WP (since it is a target Role, it is assumed exchange(s) with the owner shall take place).

...

WORK PRODUCTS - VERSIONS

WP ***versions***

- Several *versions* that target the different *SW reviews*
- This is marked by appending to the WP name, the target review between "< >"

Example:

Component Specification <SWRR>

Component Specification <PDR>

Component Specification <DDR>

Component Specification <CDR>

- The possible versions/reviews are:

  SRR, SWRR, PDR, DDR, CDR, QR, AR, and *ORR*

WP ↔ ECSS-E-ST-40C

- In order to guide the adopters in the implementation of the software development process, a ***mapping*** relating the *standard's main* ***sub processes / activities*** with the WPs is defined. Furthermore, a *mapping* of the WPs to the standard's ***Document Requirement List (DRL)*** and *vice-versa* is also provided.

ECSS-E-ST-40C process	Work Product
Software related system requirement process	**Avionics Model <SRR>** **AOCS SSS <SRR>** **EP SSS <SRR>** **Equipment Interfaces <SRR>** **M&C IF Model <SRR>** Modelling Guidelines <SRR> **OBSW Configuration Item <SRR>** **OBSW SSS <SRR>** **System Models <SRR>** **System Requirements <SRR>**
Software management process	*Out of scope*
Software requirements and architecture engineering process	**AOCS Component Model <SWRR>** **AOCS Component Model <PDR>** **AOCS Model <SWRR>** **AOCS Model <PDR>** **AOCS SRS <SWRR>** **Component Deployment <PDR>** **Component Instantiation <PDR>** **Component Specification <SWRR>** **Component Specification <PDR>** **Computational Model <PDR>** **EP Configuration <SWRR>** **EP Detailed Design <PDR>** **EP Implementation Model <PDR>** **EP SRS <SWRR>**

...

Related File	ECSS-E-ST-40C DRL Item	Work Products
Requirements baseline (RB)	Software system specification (SSS)	**AOCS SSS** **Avionics Model** **EP SSS** **M&C IF Model** **OBSW SSS** **System Models** **System Requirements**
	Interface requirements document (IRD)	**Avionics Model** **EP SSS** **OBSW SSS**
Technical specification (TS)	Software requirements specification (SRS)	**AOCS Model** **AOCS OBSW IDS** **AOCS SRS** **EP SRS** **FDIR Model** **Logical and Interface Models** **OBSW Model** **OBSW SRS**
	Interface control document (ICD)	**AOCS Component Model** **AOCS OBSW IDS** **Component Specification** **EP SRS** **Equipment Interfaces** **Logical and Interface Models** **M&C IF Model** **OBSW SRS**

...

ROLES

Role	SW domain	Description
AOCS Team	Outside	Is in charge of defining, the RB-level System-Software Specifications (SSS) related to the AOCS. AOCS Team is also in charge of developing the logical models based in Matlab/Simulink, including the GNC algorithms and the AOCS mode management logic. In addition, this team is in charge of producing the AOCS Software Requirements Specification (SRS) definition in collaboration with the OBSW Architect.
Avionics Team	Outside	Defines the high-level software system requirements and proposes standards to be used. The activity proceeds after the definition of the HW/SW perimeter for processing and the major output consists in the RB-level System-Software Specification (SSS). It is important to note that Avionics Team collaborates closely with the DHS Team and the AOCS Team.
DHS Team	Inside	Is in charge of defining, the RB-level System-Software Specifications (SSS) related to the DHS. They are also in collaboration with the Avionics Team. In addition, this team is in charge of producing the DHS Software Requirements Specification (SRS) definition in collaboration with the OBSW Architect.
EP Provider	Inside	Is in charge of performing the specification, design, implementation, integration and validation of the Execution Platform (EP) software. This role could be internal to the organization of the OBSW prime or external.
Equipment Supplier	Outside	Is in charge of equipment interfaces and shares this data with OBSW Architect and Avionics Team.
Flight Operation Segment	Outside	Is in charge on validating operational procedures, operating the satellite during In-Orbit Test (IOT) phase and ensuring in-orbit support when anomalies are detected. In most cases the Avionics Team is the interface between the OBSW Architect and the Flight Operation Segment.
OBSW Architect	Inside	Is in charge of the definition of the static architecture (components, interfaces, and connectors) and of the dynamic (real-time) architecture, and is responsible for the selection of the EP. He/she assists the Avionics Team in defining software system requirements. The OBSW Architect designs the architectural model, resource and implementation requirements and instantiates components. He/she also coordinates OBSW Component Developer, the EP Provider, and the OBSW Validation Manager.
OBSW Component Developer	Inside	Is in charge of developing the detailed design and the functional code of a given component implementation, based on the interfaces and functional specifications defined by the OBSW Architect. The component implementation is developed considering a set of requirements that are allocated to this component. The Component Developer is also in charge of the unitary tests (specification, source code and reports) of the component implementation. Software / Component integration and software validation are explicitly excluded from this role.
OBSW Development Team	Inside	Is in charge of the OBSW development activities according to the architecture defined by the OBSW Architect. Performs the analysis of requirements, the definition of the specification, and the allocation of requirements to the various OBSW components, together with the OBSW Architect. Responsible for the allocation of component to OBSW Component Developer.
OBSW Integrator	Inside	Is in charge of integrating components developed by one or more OBSW Component Developers in an incremental and iterative manner. Prepares the work for the OBSW validation.
OBSW Validation Manager	Inside	Is in charge of managing validation activities and the specification of the test plan in order to reach the validation objectives.
SDB Team	Outside	Is in charge of producing the missionisation of the Satellite Data Base from the description of the OBSW interfaces (telecommands, telemetries, configuration parameters, on-board parameters, etc.) provided by the OBSW Architect.
System Team	Outside	In charge of the system (mission) requirements and models. The Avionics Team then acts as intermediary between this role and the OBSW Architect, in particular to share any SW relevant system (mission) level data.
Test Environment Engineer	Inside	Is in charge of the development, support and maintenance of the tools used for software test activities.

N² CHART

- Help in determining, from a *global view* stand point, ***which Roles exchange which WPs*** among them

FROM / TO	AOCS Team	Avionics Team	DHS Team	EP Provider	Equipment Supplier	Flight Operation Segment	OBSW Architect	OBSW Component Developer	OBSW Development Team	OBSW Integrator	OBSW Validation Manager	SDB Team	System Team	Test Environment Engineer
AOCS Team		10					6 10		6		8 10			
Avionics Team	46		13			34	13 33 34 37 46			1 47 49	1 2 37 46 47 49 51	13 37		
DHS Team														
EP Provider							25 26 30			25 26 29				
Equipment Supplier		32					32							
Flight Operation Segment														
OBSW Architect	7 9 15 36 38	36		20 24 28 31				17 21	3 9 11 12 22 45	3 11 14 18 20 24 35	4 12 15 21 22 44 45	7		
OBSW Component Developer							16 19			16	23			
OBSW Development Team														
OBSW Integrator		39 41 42									39 42 43			
OBSW Validation Manager				50			48 55 56			48 50 56				
SDB Team										40				
System Team	52 53	52 53					52 53							
Test Environment Engineer				54						54				

WP IDs

EXCHANGE FORMATS

An Exchange Format is the specification used for sharing a specific WP among stakeholders using different tools that support it for import, export of both. If one tool can export to the Exchange Format, and another import from it, then those tools can work together.

'Standardization', 'Validation' and 'Test Case Availability' represent three key points that shall be considered for the potential Exchange Formats to be used in the *software development process*:

- **Standardization**: The specification of the format should be defined, documented and published as an unambiguous and freely available standard. This allows tools developers to work from that documentation when adding support for the format.
- **Validation**: The specification should clearly define what is and is not a valid usage of the format, in a way that can be automatically checked by a third party tool. This not only helps to avoid cases where tools differ in what they produce and accept, but also fairly assigns responsibility if such cases happen. Such a tool should automatically detect usage errors and inconsistencies, reporting them in enough detail that they can be straightforwardly fixed.
- **Test Case Availability**: Correct implementation of an Exchange Format specification is made considerably easier by a publicly available set of test data.

***Handbook** describes each candidate standard in detail*

Exchange Format	Standardization	Validation	Test Case Availability
ReqIF	Good	Adequate	Problematic
MATLAB	Poor	Poor	Problematic
FMI	Good	Adequate	Adequate
SMDL (SMP2)	Adequate	Adequate	Adequate
OSRA-SCM	Adequate	Adequate	Problematic
CCSDS/SAVOIR EDS	Good	Adequate	Adequate
EGS-CC CDM	Problematic	Problematic	Problematic
SEIM	Adequate	Adequate	Problematic
PUS-C FM	Adequate	Adequate	Problematic

PROCESS MODEL

Color code

Blue: WPs / Orange: Exchanges / Purple: SW Factory

Exchanges

WP owned by a certain Role depends on another WP owned by a different Role

Data Hub

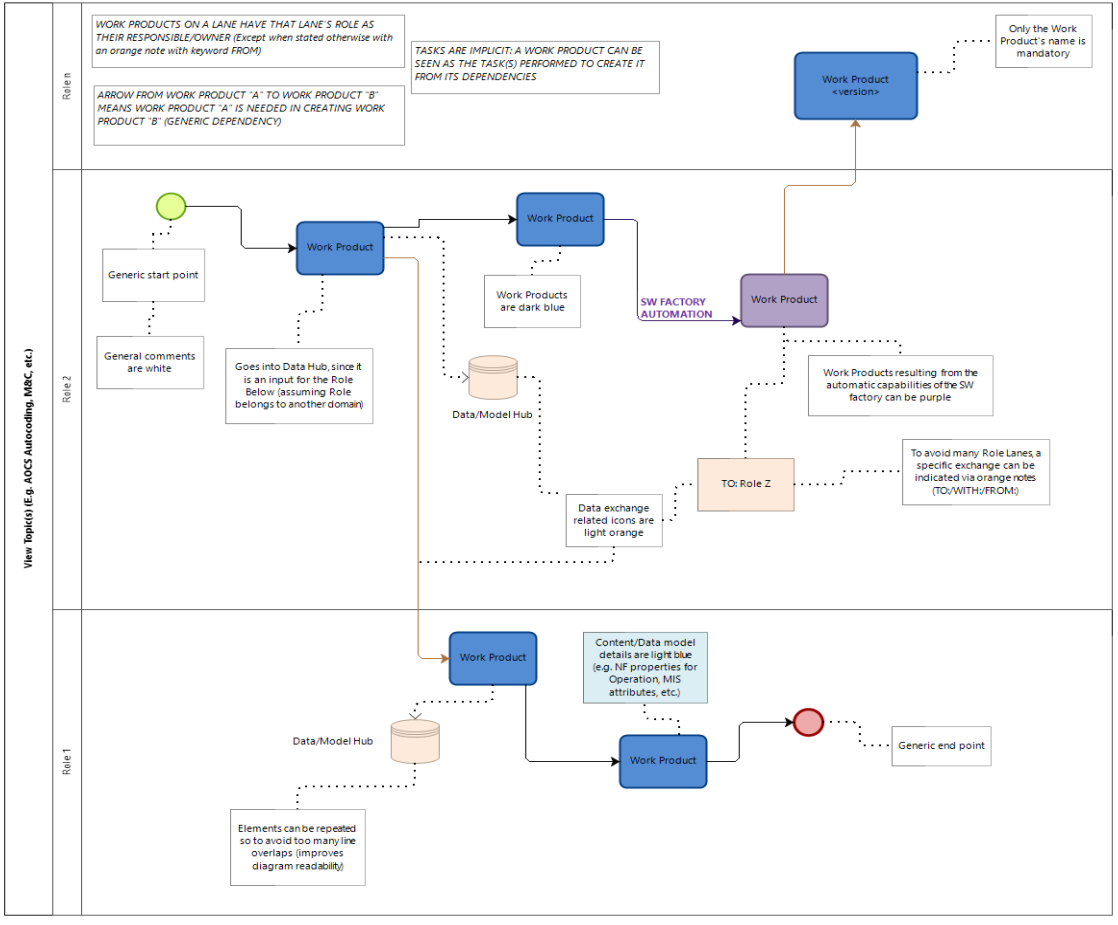
Link with the Data Hub if exchange involves a Role external to the software domain

Notes

Notes that clarify some aspects of the WP (e.g. contained information)

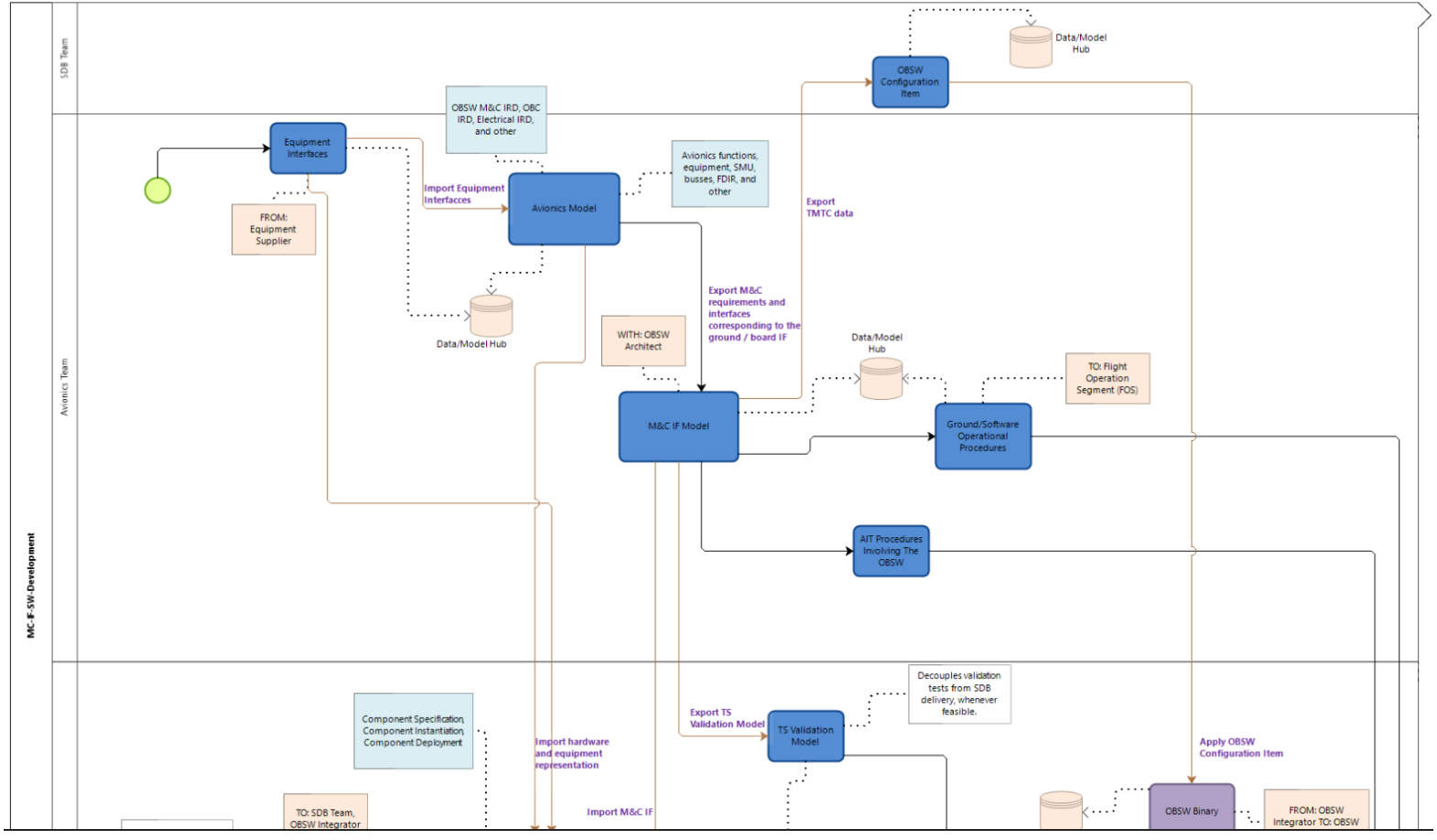
Multi perspective view

In a single view: *Roles* + *WPs* + *workflow* + *Exchanges* + key concepts (e.g. *Data Hub*) + notes



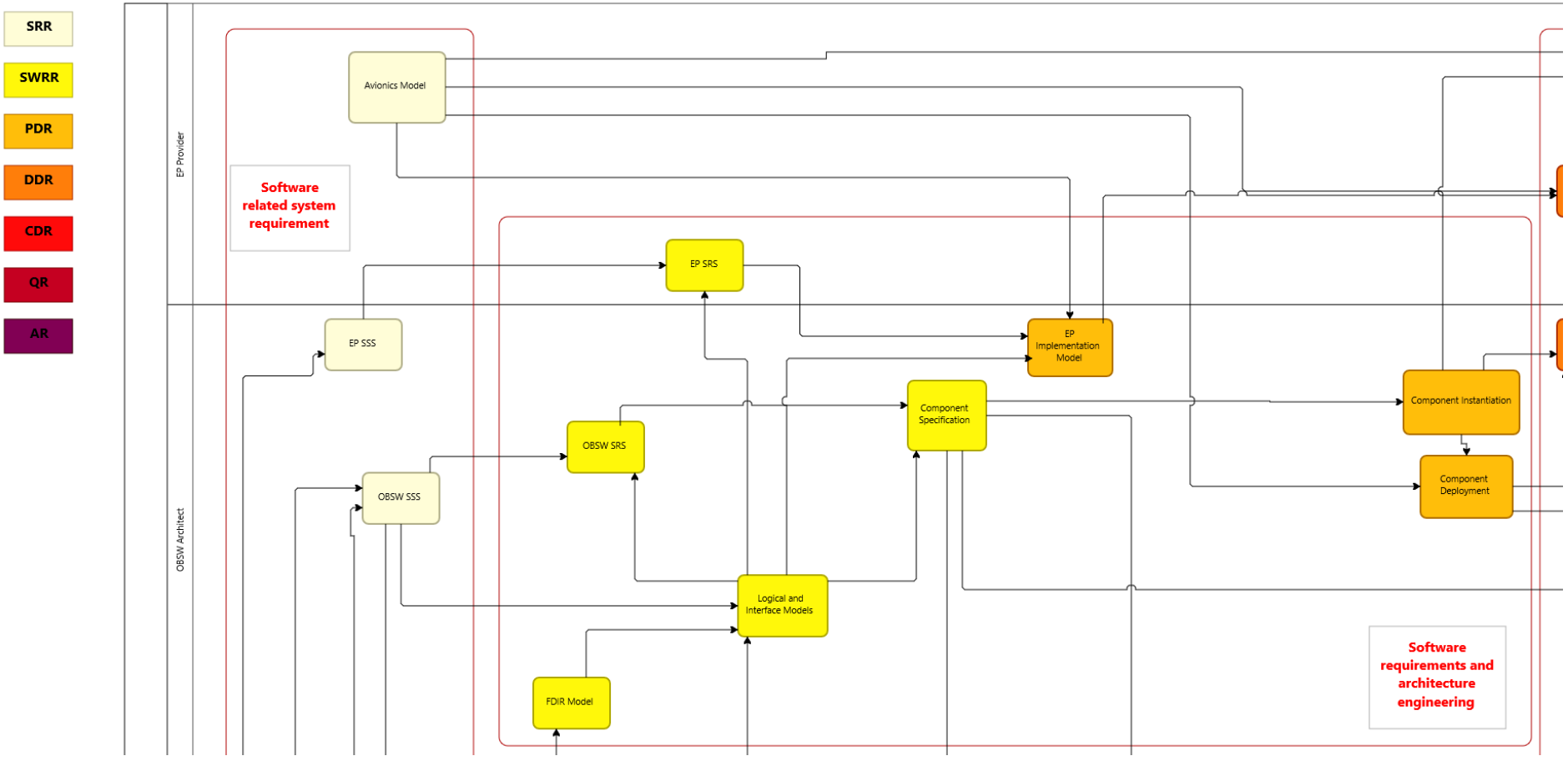
PROCESS MODEL

(part of) ***Specific*** *view on* ***Monitoring and Control***



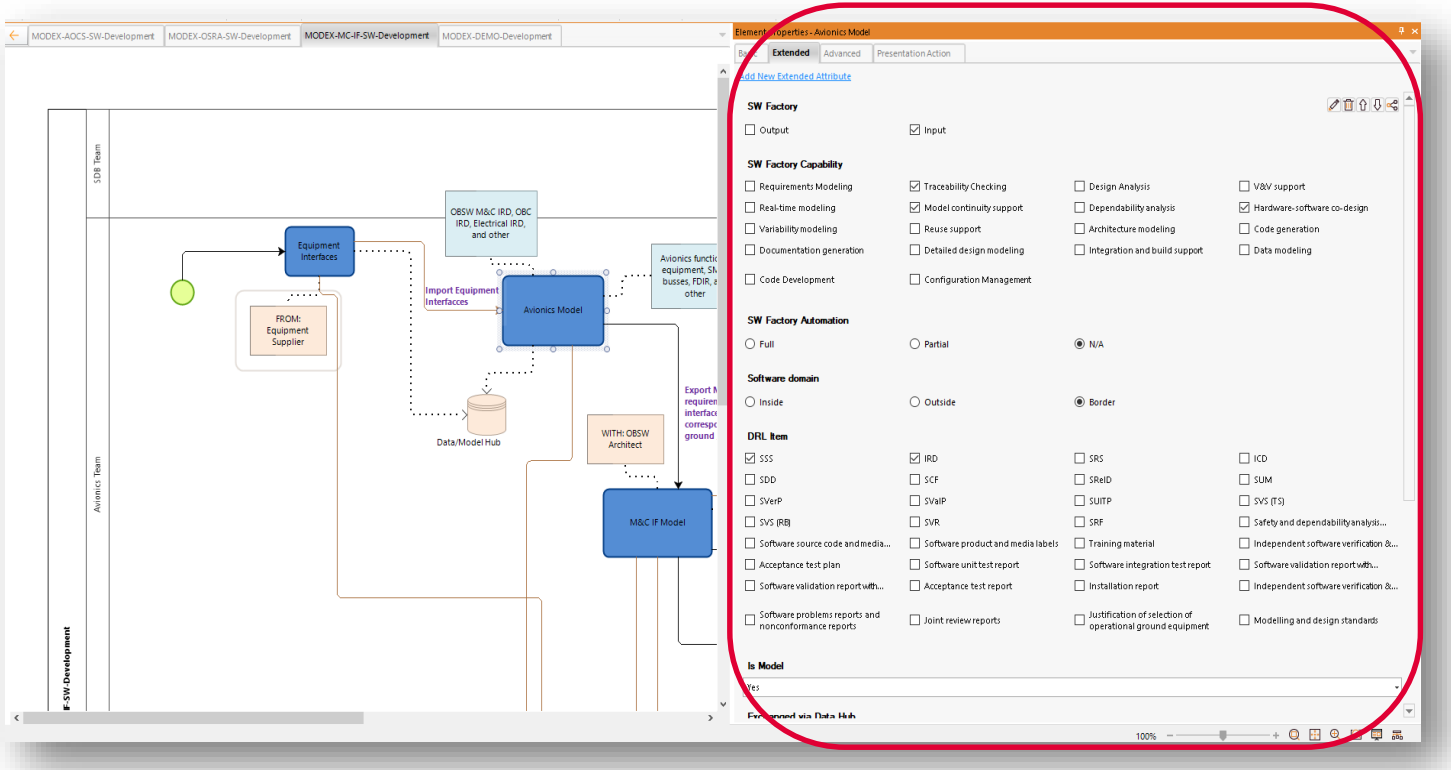
PROCESS MODEL

(part of) ***Global*** *view +* ***ECSS mapping*** *(reviews and processes)*



PROCESS MODEL

BPMN model is ***annotated*** with the list of ***properties*** for each *Work Product*



PROCESS MODEL

*Work Products **inside** the software domain*

One of the potential uses of the *WP Table* is to adjust the WPs and their properties to give a ***particular view* of interest**

Hiding/filtering

columns/*WP properties*

WP ID	WP Name	WP Description	Software domain
3	AOCS Algorithms Code	AOCS code generated from the logic	Inside
4	AOCS Component Model	The AOCS component is represente	Inside
9	AOCS SRS	Can be generated from the AOCS M	Inside
11	AOCS Static Architecture Code	Code generated from the AOCS Mo	Inside
12	AOCS Unit Test Code	Source code of unitary tests for the s	Inside
14	Non-Functional Code	Non-Functional code generated from	Inside
16	Component Code	Code generated from the logical mo	Inside
17	Component Code Skeletons	Code generated from the OBSW cor	Inside
18	Component Deployment	The deployment phase is used to ma	Inside
19	Component Detailed Design Model	Component Developer performs the	Inside
20	Component Instantiation	A component instance is an instantia	Inside
21	Component Specification	The OBSW Architect defines compo	Inside
22	Component Unit Test Results	Component implementations unit tes	Inside
23	Component Unit Tests	Source code of unitary tests for the s	Inside
24	Computational Model	A central principle of the OSRA is the	Inside
25	EP Code	Code generated from the Execution	Inside
26	EP Configuration	Part of the build tool, providing pseu	Inside
27	EP Detailed Design	The Execution Platform provider pro	Inside
28	EP Implementation Model	The OBSW Architect defines the imp	Inside
29	EP Integration Tests	Specification of the Integration tests	Inside
30	EP SRS	OBSW Architect supports the Execu	Inside
31	EP SSS	From the analysis of the Avionics SS	Inside
35	Interaction Layer	Code generated from the OBSW dep	Inside
42	OBSW Integration Test Report	OBSW integration test reports to doc	Inside
43	OBSW Integration Tests	Specification of the OBSW Integratic	Inside
44	OBSW Model	Represents the structure/architectur	Inside
45	OBSW SRS	OBSW Architect and OBSW Develop	Inside
48	Software Validation Report TS	OBSW validation report with respect	Inside
50	SPRs and NCRs TS	List of Non-Compliances and associ	Inside
55	TS Validation Model	Software validation team uses the O	Inside
56	Validation Scripts and Documents	Validation Documents is the specific	Inside

PROCESS MODEL

WPs to be reviewed at **DDR**

A	B	D	L	M
WP ID	**WP Name**	**Owner (exchange source)**	**DRL Item**	**SW Review**
4	AOCS Component Model	OBSW Architect	ICD;SDD	SWRR;PDR;DDR;CDR
18	Component Deployment	OBSW Architect	SDD	PDR;DDR;CDR
19	Component Detailed Design Model	OBSW Component Developer	SDD	DDR
20	Component Instantiation	OBSW Architect	SDD	PDR;DDR;CDR
21	Component Specification	OBSW Architect	ICD;SDD;SCF	SWRR;PDR;DDR;CDR
24	Computational Model	OBSW Architect	SDD	PDR;DDR;CDR
26	EP Configuration	EP Provider	SCF	SWRR;PDR;DDR;CDR
27	EP Detailed Design	EP Provider	SDD	PDR;DDR;CDR
28	EP Implementation Model	OBSW Architect	SDD;SRF	PDR;DDR;CDR
30	EP SRS	EP Provider	SRS;ICD	SWRR;PDR;DDR;CDR
33	FDIR Model	Avionics Team	SRS;SCF	SWRR;PDR;DDR;CDR
37	M&C IF Model	Avionics Team	SSS;ICD	SRR;SWRR;PDR;DDR;CDR;QR;AR
44	OBSW Model	OBSW Architect	SRS;ICD;SDD	PDR;DDR;CDR
45	OBSW SRS	OBSW Architect	SRS;ICD	SWRR;PDR;DDR;CDR

PROCESS AND SOFTWARE FACTORY/DATA HUB

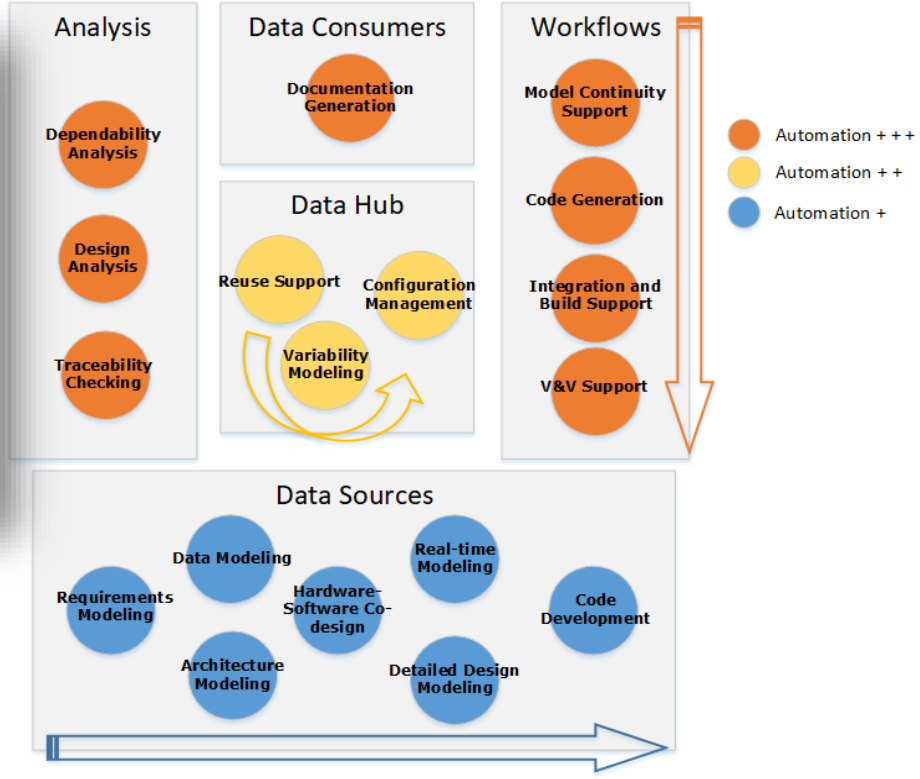
The arrows point the approximate ***chronological order*** of the SW development lifecycle (iteration)

The **18** capabilities of the *Software Factory*:

1. **Requirements Modelling:** Elaboration of the technical specifications or requirements baselines using model-based tools (e.g. DOORS, Capella, SysML, Eclipse RMF, etc.);
2. **Traceability Checking**: Verification of the traceability along the SW development process (requirements → design → verification cases → verification results → etc.). The traceability can be maintained between artefacts of the same type but across different levels and phases (e.g. RB and TS requirements), or between different artefact types (e.g. TS requirements and a SDD related design model, and from the latter to e.g. automatically generated source code);
3. **Design Analysis:** Analyses executed on SW design models, e.g. data, functional, schedulability, performance (e.g. communication latency, resource usage), consistency, completeness, feasibility, trade-off and impact, simulations, etc.;
4. **V&V Support:** Support to the elaboration and usage of models supporting the V&V activities. Production and maintenance of V&V results/evidences. Verification and validation (V&V) activities are expected to benefit from a model-based development approach, in this way reducing the associated effort and risks. Encompasses e.g. model based testing;
5. **Real-time Modelling:** Formalization of the real-time properties in a design model, e.g. deadlines, WCET. Description of the behaviour and the interactions of the different processes or tasks, and the ways shared resources are accessed. Defines real-time constraints and real-time environment needs;
6. **Dependability Analysis:** Specific SW safety and dependability (RAMS) analysis (e.g. FMEA,

...

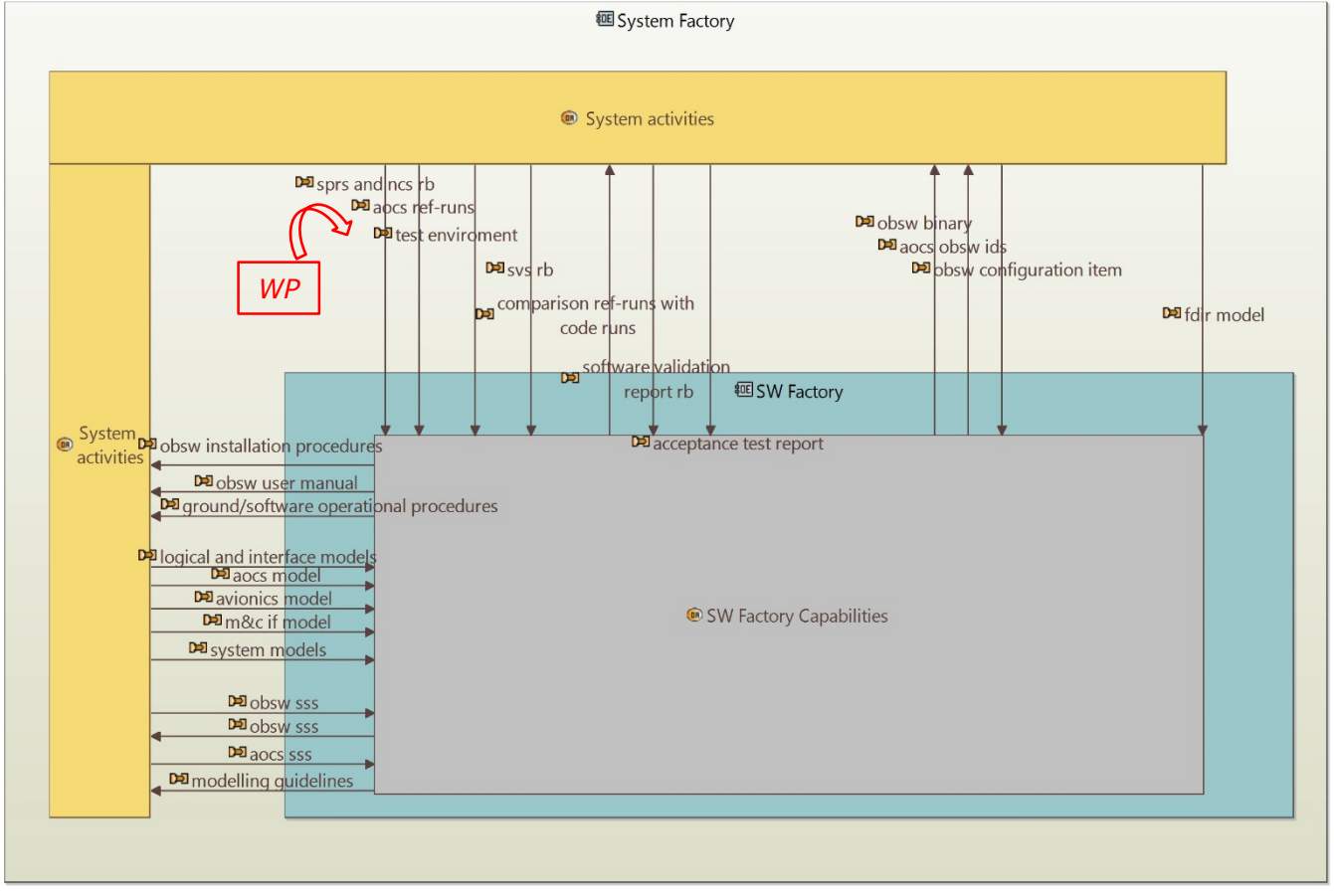
Handbook *describes each capability*



PROCESS AND SOFTWARE FACTORY/DATA HUB

Software Factory ***external*** *interfaces*
(with *Avionics/System* level)

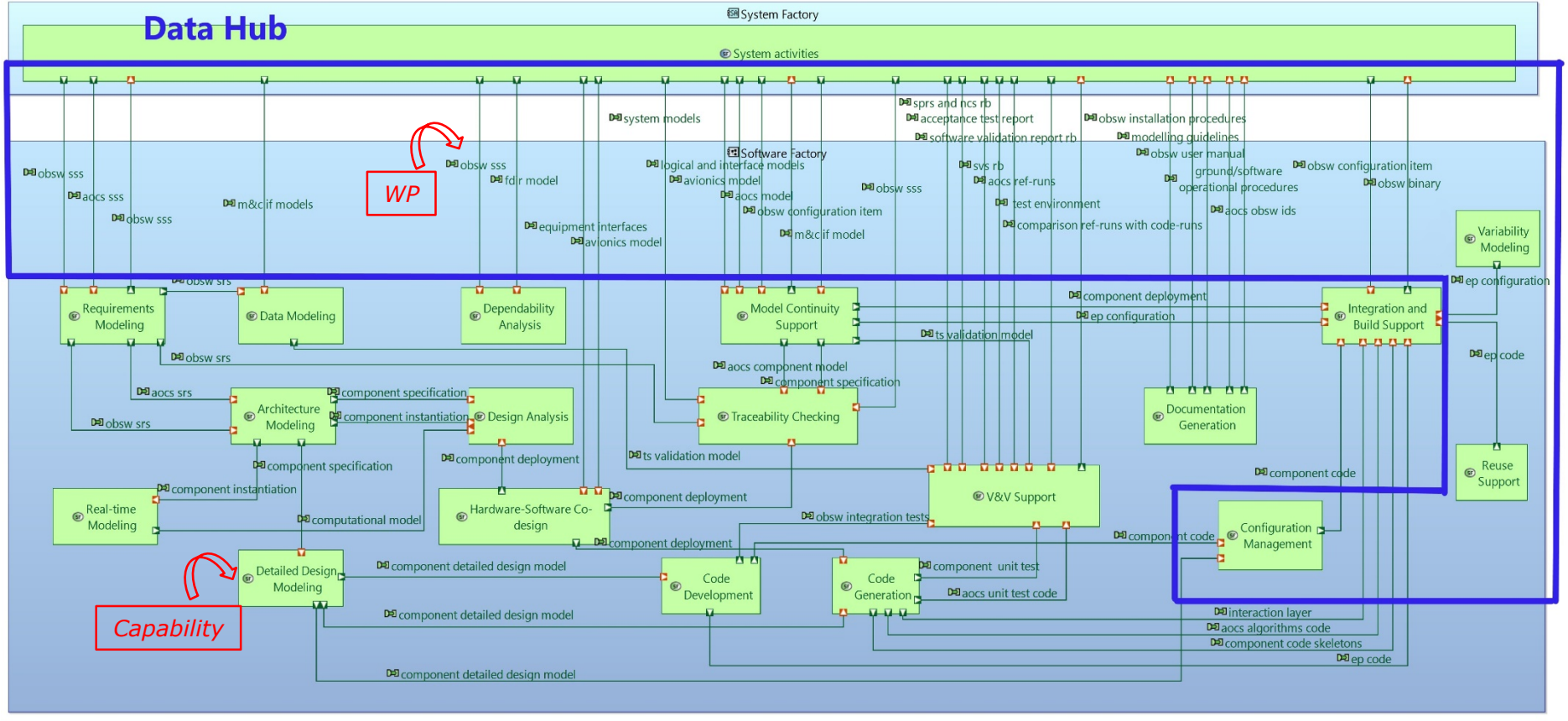
Obtained from the *WP Table* using *filters* on *SW Factory* related *properties*



PROCESS AND SOFTWARE FACTORY/DATA HUB

*Software Factory **internal** functions and interfaces + **Data Hub***
(Functional Analysis)

Obtained from the *WP Table* using *filters* on *SW Factory* related *properties*



CONCLUSIONS

- *ECSS-E-ST-40C* is not to be modified i.e. activities, milestones, etc. remain the same. The information exchanged is similar but the key point is that the exchanges are produced in a ***different way*, *different granularity* and *different format***
- Defining a process in terms of models, together with candidate formats and tools should ***facilitate the deployment of model-based practices***
- Focus on cross-domain exchanges should enable ***better coordination between disciplines***
- By formalizing a process it becomes easier to use it and make it evolve (opening all kind of ***opportunites for automation***)
- The formalization of process artefact (WP) properties and relationships among them (e.g. dependency) enables an advanced ***analysis on the process*** and supports the deployment of model-based development (e.g. digital continuity, traceability)
- The identification of detailed roles, ownership and exchanges should enable an efficient and ***guided deployment of process***
- There is a need to show ***both global and specific views*** on the process
- The specification of the process can and should make the ***bridge with the SW Factory and Data Hub*** (process based on advanced MBSE infrastructures)

FUTURE WORK

- Consolidation of the Work Products: ***more precise*** definitions, properties and relationships
- Perform a ***semantic mapping*** between dependent consolidated WPs
- Elaborate on ***relationship with Space Systems Ontology***
- Further ***validation*** of process using representative mission models
- To extend the process to ***cover out of scope areas***
- A more **rigorous mapping of WP to the *ECSS-E-ST-40C*** is needed (e.g. mapping with DRD sections)
- Expand and ***improve diagrams*** (communication is still challenging)
- ***Extending and refining SW Factory capabilities***
- Refine ***SW Factory / Data Hub functional architecture***, and ***define logical and candidate physical architectures*** as well
- ***Extend an review process Roles***

Thank you

MODEX team

Elena Alaña (ealana@gmv.com)
Tiago Jorge (tiago.jorge@gmv.com)