# "IMPLEMENTATION OF THE ECSS-E-ST-40C PROCESSES USING A MODEL-BASED PARADIGM"

**Tiago Jorge [1], Régis de Ferluc [2], Iris Ávila [2], Poul Hougaard [3], Richard Melvin [4], Andreas Jung [5], Javier Fernández [5], Elena Alaña [1]**

*(1) GMV Aerospace and Defence (GMV), Spain, E-mail:tmdasilva@gmv.com, ealana@gmv.com*
*(2) Thales Alenia Space France (TAS-F), France, E-mail: regis.deferluc@thalesaleniaspace.com, iris.avila-martinez@thalesaleniaspace.com*
*(3) Terma A/S (Terma), Denmark, E-mail: poh@terma.com*
*(4) CGI IT UK LTD (CGI), U.K., E-mail: richard.melvin@cgi.com*
*(5) European Space Agency (ESA), ESTEC Noordwijk – The Netherlands, E-mail: Andreas.Jung@esa.int, Javier.Fernandez.Salgado@esa.int*

**Abstract**: This paper presents an implementation of a model-based paradigm for on-board software developments being compliant with the ECSS-E-ST-40C standard. A set of model-based artefacts, their exchange formats, roles and milestones are presented. This paper is the result of the MODEX activity ("*Model Exchange for Software Engineering*").

**Keywords**: Data Hub, ECSS, Embedded Software, Exchange Format, OBSW, Software Factory, Software life cycle, Work Product.

## 1. INTRODUCTION

Models are more and more used in the industry for the development of On-Board Software (OBSW), and sometimes replace documentation which instead is generated automatically. It is thus necessary to analyse the impact on the process to reflect this new trend.

The MODEX study [1] formalizes the artefacts produced and exchanged along the software development process when models are used, including the following aspects:

- 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 Exchange Formats; and
- The relationship with the Software Factory and the Data Hub.

The ECSS-E-ST-40C standard [2] is the baseline, and MODEX builds upon it by focusing on the model exchanges taking place during a software development that follows a model-based approach.

The Product Assurance (PA) and Quality Assurance (QA) processes [3] are not in the scope of the MODEX study, but a similar analysis should also be possible for PA/QA processes.

## 2. MODEL-BASED IMPLEMENTATION

### 2.1. BASIC CONCEPTS

In order to fully understand the formalisation of the proposed model-based implementation, the following concepts need to be introduced:

*Work Product (WP)*: Bounded engineering artefact exchanged during the model-based software development process. These artefacts are mainly models but not limited to them, e.g. source code, binary files.

*Exchange Format*: Specification used for sharing a WP among stakeholders using different tools that support it. An example is the *Requirements Interchange Format* (*ReqIf*) [4].

*Software Factory*: Software Engineering supporting infrastructure that integrates those tools that allow the application of a model-based approach at software level. It plays a central role to produce the WPs.

*Data Hub*: Repository to exchange information (i.e. WPs) with other domains, and to some extent, within the software domain.

*Roles*: Stakeholders involved in the model-based software development process. The generic ECSS-E-ST-40C [2] roles have been detailed, including specific roles not only from the OBSW domain, but also from other domains interacting with it.

## 2.2. FORMALISATION

The model-based implementation of the ECSS-E-ST-40C standard has been formalized using two formats that are consistent and complementary:

- A *BPMN2.0* (Business Process Model Notation) *model.* BPMN represents a modelling method for business processes in the form of a diagram similar to a flowchart. In this project, BPMN diagrams model the work flow, the dependencies among Work Products and the interactions with the stakeholders (Figure 1). Here we are not just illustrating the processes but modelling them. Once consolidated and using a suitable tool, the processes can be executed in simulation to perform V&V on the process model.
- *Work Products Table* [Table 3] (Excel sheet automatically generated from the BPMN model) that includes various properties about each Work Product, e.g. purpose, ownership, Exchange Formats, required *Software Factory capabilities*, need for *exchange via the Data Hub*, target reviews, etc. This table is in particular useful to filter such information according to specific views of interest.

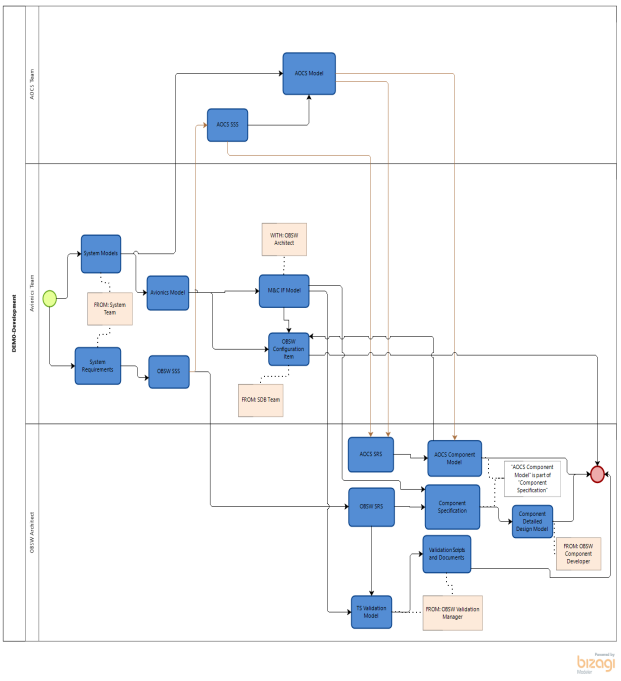


*Figure 1: Partial view of the formalized model-based software development process (WPs represented in blue)*

## 3. IMPACT ON ECSS-E-ST-40C

The ECSS-E-ST-40C [2] standard is agnostic to the implementation details of the software engineering lifecycle. The standard includes a set of requirements to perform the development process. Projects are free to choose the paradigm that suits the needs of the project.

In case of adopting a model-based approach, the standard is still fully applicable and the high-level process does not change, in the sense of *WHAT* sub processes shall be in place and *WHAT* outputs ultimately shall be produced. The results of this activity instead propose *HOW* the process could be implemented following a model-based approach, where models become the main vehicle along the development.

### 3.1. MAPPING TO ECSS-E-ST-40C

Two mapping tables are produced to guide the adopters in the usage of a model-based approach. Both mappings facilitate the adoption and understanding of the new implementation.

1. *Mapping of ECSS-E-ST-40C processes to Work Products*. Table 1 shows an extract of this table where Work Products in **bold** are considered models (or strong candidates), and the specific Work Product version (target review), is indicated between "<>" in the WP's name.

*Table 1: Part of the E-40 process vs. WP mapping table*

| ECSS-E-ST-40C Process | Work Product |
|---|---|
| Software related system requirement process | **Avionics Model <SRR>**<br>**AOCS SSS <SRR>**<br>**EP SSS <SRR>**<br>**Equipment Interfaces <SRR>**<br>**M&C IF Model <SRR>**<br>Modelling Guidelines <SRR><br>**OBSW Configuration Item <SRR>**<br>**OBSW SSS <SRR>**<br>**System Models <SRR>**<br>**System Requirements <SRR>** |

2. *Mapping of Work Products to the standard's Document Requirement List (DRL) Item and vice-versa.* This mapping could be extended to the respective Document Requirements Definition (DRD). A precise mapping to DRD sections could pave the way to e.g.

demonstrate how such standard documentation could be automatically generated from, and kept consistent with, specific WP parts.

*Table 2: Part of the DRL Item vs. WP mapping table*

| Related File | ECSS-E-ST-40C DRL Item | Work Products |
|---|---|---|
| Requirements baseline (RB) | Software system specification (SSS) | **AOCS SSS**<br>**Avionics Model**<br>**EP SSS**<br>**M&C IF Model**<br>**OBSW SSS**<br>**System Models**<br>**System Requirements** |
| | … | … |
| Technical specification (TS) | Software requirements specification (SRS) | **AOCS Model**<br>**AOCS OBSW IDS**<br>**AOCS SRS**<br>**EP SRS**<br>**FDIR Model**<br>**Logical and Interface Models**<br>**OBSW Model**<br>**OBSW SRS** |

### 3.2. CONTRIBUTIONS TO THE ECSS

The WP concept as defined in the proposed software development process can complement/improve the ECSS-E-ST-40C standard [2], the ECSS-E-HB-40A software engineering handbook [4] or even other ECSS standards/handbooks since some WPs are exchanged between various domains. Examples of these contributions are the following:

1. Defining an engineering artefact internal to the process in a more precise way, identifying dependencies among Work Products, and their owners;
2. Pushing a MBSE (Model-Based Systems and Software Engineering) approach by directly representing of linking MBSE related concepts, e.g. properties related to the Software Factory and Data Hub that help in defining their interfaces and functional architectures;
3. Providing objective process artefacts that can help identifying all the needed engineering exchanges, to different levels of details, e.g. the WPs elaborated in collaboration shall be exchanged; or the inputs necessary to define a WP shall be received;
4. The detailed semantic mapping between dependent WPs contributes to identify refined data exchange items, consistently with an Ontology for Space Systems [6];
5. Via their dependency/exchange relationships with other WPs be a concrete definition of digital continuity between the various process artefacts. That is saying e.g. that any refinements or transformations between WPs becomes explicitly defined.

### 3.3. PROJECT REVIEWS

The software development process relies on fix and well-defined milestones with defined Document Requirements List (DRL) Items. In case of a model-based paradigm, WPs are also linked to the project reviews. On the one hand, each milestone will have a defined set of WPs to be reviewed. On the other hand, each WP might have different versions associated to it, in particular dedicated versions targeting the different applicable reviews. For instance, Table 3 lists the WPs to be delivered at Detailed Design Review (DDR). They can be easily extracted from the *Work Products Table*.

*Table 3: 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 |

Figure 2 shows part of the model-based development process, using for the WPs a colour code that directly maps to the software reviews. Moreover, specific areas are drawn in the diagram to map the WPs with the standard ECSS-E-ST-40C processes. This helps to visually relate the WPs to the standard. There are several advantages on the adoption of models and their connection to reviews. Firstly, this implementation makes it possible to perform the project and technical reviews more efficiently, with the support of tooling. It is an opportunity to automate the review process, and check quickly if all the information is provided, avoiding the process inconsistences. Secondly, this paradigm allows a

continuous monitoring of the development. For example, model-based metrics may provide a global and objective figure of the evolution and pending work to reach each milestone.

Finally, the efforts are allocated on the development of the models. Once a milestone is reached, a model might replace a set of documentation.

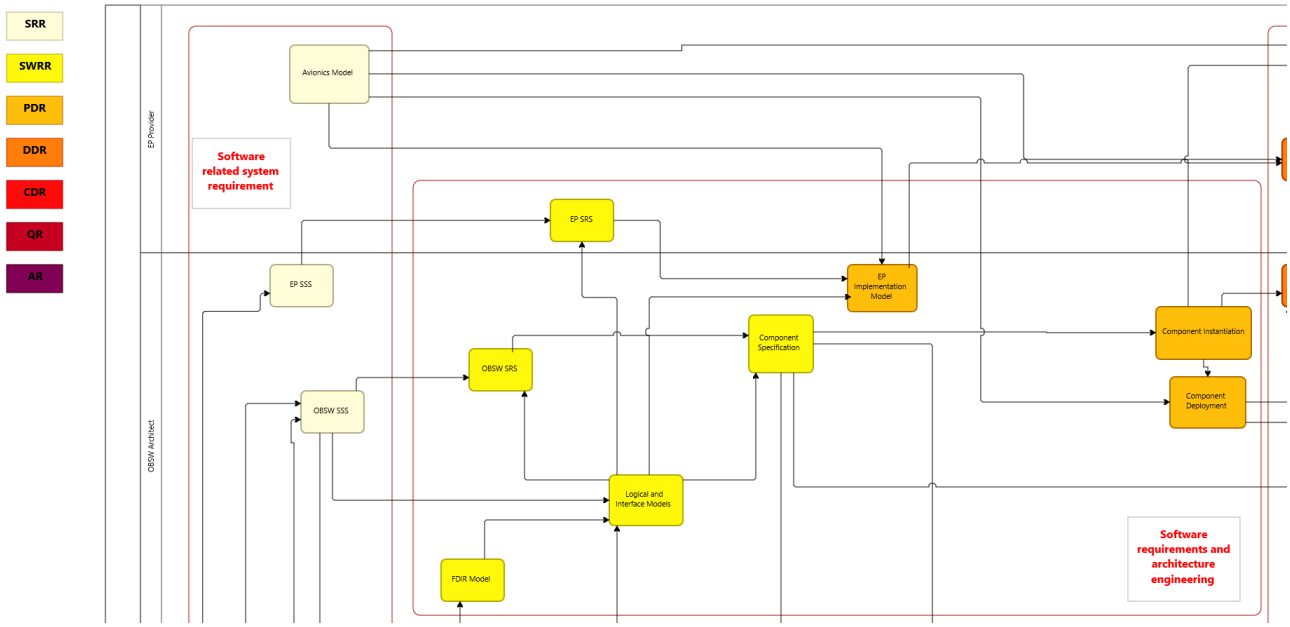


*Figure 2: ECSS Reviews and processes (excerpt)*

## 4. CONCLUSIONS

Compared to ECSS-E-ST-40C, MODEX shows how SW related exchanges can be 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. Moreover, the 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 opportunities for automation. In particular, the formalization of process artefact properties and relationships enables an advanced analysis on the process and supports the deployment of digital continuity and traceability. In addition, the identification of roles, ownership and exchanges should enable an efficient and guided process deployment. Last but not least, the specification links with the Software Factory and Data Hub, becoming by construction compatible with such advanced MBSE infrastructures.

## REFERENCES


[1] "*Handbook of Model Exchange for Software Engineering*", GMV 20767/20 V4/20.

[2] "*Space engineering – Software*", ECSS-E-ST-40C, 6 March 2009.

[3] "*Space product assurance – Software product assurance*", ECSS-Q-ST-80C Rev.1, 15 February 2017.

[4] "Space engineering - Software engineering handbook", ECSS-E-HB-40A, 11 Dec. 2013.

[5] ReqIf: https://www.omg.org/spec/ReqIF/1.2/About-ReqIF/

[6] OSMoSE: https://mb4se.esa.int/OSMOSE_Main.html


## 5. ACKNOWLEDEMENTS


This work was supported by the MODEX project, ESA contract 4000125540/18/NL/FE. The partners involved in this study are GMV as prime contractor and Thales Alenia Space France, Terma and CGI as subcontractors.