ECSS Requirements Management System

How Applying MBSE Can Improve the ECSS Management and its User Interactions

N. Salor Moral[1*], A. Vorobiev[1], H.P. de Koning[2]

[1]RHEA Group, Leiden, The Netherlands; n.salor@rheagroup.com , a.vorobiev@rheagroup.com

[2]DEKonsult, Amsterdam, The Netherlands; hanspeter.dekoning@dekonsult.com

Extended Abstract

Introduction

Collaborating in projects implies some rules and agreements that parties must adhere to. Depending on the applied formalism, such agreement could be called a cooperation standard. So then, what constitutes exactly a standard?

A standard represents the rules and methods agreed to be followed by different parties for ensuring interoperability. However, the world, software systems and the state or the art technology on the space domain are continuously evolving. As such, standards also need to be continuously updated in an easy, fast, and at the same time rigorous manner.

This duality is typically contradictory because formalisation and reaching consensus take time. Formal standardization must ensure information consistency and completeness while at the same time, it must correctly identify and provide solutions for possible impact any change can cause.

Considering most standard used in ECSS are defined as documents, any process carried out with them (e.g., modification, review, possible tailoring), is basically manual and thus, error prone and time consuming. Computing impact chains between several hundred-page documents is an inefficient and almost impossible task to carry out as the person responsible to detect issues will have to deal, among others, with synonyms, homonyms, implicit assumptions, contextual information, and in-depth domain knowledge.

Thus, there is a need to move from a document centric approach to a model based one. A document centric approach makes it difficult to discover inconsistencies across documents, duplications, and in general lacks harmonization.

Provisioning a software system which automates as much as possible all ECSS management processes and retrieves all the affected information (including but not exclusively standards) and raises any possible issue/inconsistency, is the main goal of the ECSS Requirement Management System (E-RMS) Project which will be described in this paper.

Although the title of the project hints that the E-RMS system will only manage requirements, its scope is larger. As the system will have to support different ECSS processes (currently described in the ECSS-D-00B standard [1]), and despite what the project title infers, not only requirements will need to be managed, but all data involved in such processes (e.g., Change Requests, Documents, New Work Items, Questionnaires, Reviews). Thus, prior to developing the software system, it is required to reach a deep understanding of:

- 'what' functions the future ECSS users will need for carrying out the different processes in which they are involved.
- 'which' information they will interact with as part of those processes.

Answers to these points will constitute the user and system specification of the E-RMS system. Ironically, defining the specification in documents, would constitute a continuation of the problem such specification tries to solve. In order to break the circle, the application of formal methods through model-based system engineering was decided.

Conceptual Data Modelling

Parallel to this project, a Space System Ontology (SSO) is being modelled. That ontology development is coordinated by the so-called "**O**verall **S**emantic **Mo**delling for **S**ystem **E**ngineering" Governance (OSMoSE) for ensuring semantic interoperability between all partners. As the Object Role Modelling Method (ORM) through the NORMA Tool is being used for the Ontology development, the conceptualization of the ECSS Management semantics, which is still running, applies the same methodology and tools to facilitate the future integration of the resulting conceptual data model will be integrated) in the Space System Ontology.

ORM is a powerful method for designing and querying information models at the conceptual level, where the application is described in terms easily understood by non-technical users. ORM data models not only represent data relationships (e.g., which maps to the traditional database columns) but often also captures business rules and are easier to validate and evolve than data models in other approach.

Within the E-RMS, the conceptual modelling exercise started by extracting the concepts from the different ECSS processes described in D-00, getting current examples of such concepts from the ECSS Standardisation Board, modelling all properties, relationships, constraints, etc. In parallel and to provide not only ECSS but normal Software System user management; users, privileges, roles, and access rights were also modelled.

An agile modelling effort has been performed; where draft models have been created and updated and delivered to the Agency for further review including those against the latest status of the SSO model. Part of the ongoing modifications refer to the split of the model diagrams in small parts, application of semantic grouping (e.g. classification of model elements in groups providing metadata), or the inclusion of the release and publish status for all elements.

The SSO will integrate different Universes of Discourse (UoD), including the conceptual model resulting of the E-RMS project. It has already been observed that the same concepts appear in multiple universes. Each UoD can be modelled in one or more ORM diagrams. However, there is no common diagram for those 'shared' concepts.

Therefore, guidelines for modelling and the correct way of persisting the models within a common repository have been required.

Process Modelling

The ECSS management processes are described in the ECSS-D-00B document. It defines the processes to be followed throughout the lifecycle of ECSS documents. In its current version the processes are defined using a quasi-UML flow diagram and tabular views describing the steps.

As the E-RMS system will have to support, as a minimum, those processes, there was a need to analyse the consistency and completeness of those processes prior to any modelling effort.

The analysis was carried out by cross-checking the diagram with the tabular information and discussing any found inconsistency with the ECSS standardisation board in dedicated meetings and change requests. Most of the found problems are related to missing explanations or criteria on how decisions are taken, handling of non-nominal situations, and misalignments between the diagram and the tabular descriptions.

During the clarification meetings, it came as no surprise that the people involved in the described processes did not check the D-00 document from time to time or make sure that adaptations which take place are correctly reflected in the standard. This lack of maintenance happens often because of not having an automatic end to end validation of the process while it is being performed against the standard itself.

These issues would automatically disappear if the processes were modelled instead of described and the E-RMS would automatically adapt the software that allows users to perform the steps while validating them in the defined order. Business Process Modelling (BPM) and its notation (BPMN) provide such capability and thus, it has been used for re-engineering the ECSS processes. The resulting BPMN diagrams will be included in the future version of ECSS-D-00.

BPM tools provide organizations with a systematic approach for managing and optimizing their business processes by helping organizations design, model, implement, and measure workflows and business rules. They frequently allow non-IT specialists to build business workflows and connect disparate systems. Core capabilities include:

- Workflow management: for designing, testing, and executing the interactions between employees, systems, and data.
- Business rules engine: for creating business rules and conditions.
- Form generator: for building web-forms.
- Collaboration: tools for discussion, decision management, and idea management.
- Analytics: for defining metrics and KPIs and generating reports.
- Integrations: for using data across systems and via interfaces.

Starting from the existing flow diagrams (which are a total of 24 high level processes) and after the analysis described, the new BPM diagrams were modelled using the Camunda Modeler [2]. The new diagrams are not simple ‘translations’ from the old ones, but a complete re-engineering effort. For example, common tasks (or steps) which were repeated across processes or which functionality was very similar were generalized and parametrized so they could be reused as BPMN SubProcesses. This will mean that the reusable subprocess will produce a dedicated method with parameters called from the different processes instead of having duplicated code blocks on code generation.

However, the E-RMS system will also need to support the creation of new workflows in the future or update existing ones. Modelling the original 24 D-00 processes has been done manually using the Camunda tool which generates a *.bpmn* file (which then is displayed as a diagram). However, the updates or dynamic creation should be simplified as much as possible. As BPMN's file format is standardized through an XML Schema, it was decided to conceptualize a sufficient subset of BPMN elements in a data model (using the elements in the already modelled processes as discriminator). The BPMN conceptual data model was created in ORM as well and demonstrated by populating the obtained data model with one of the most complex processes already modelled (i.e. Drafting Process of D-00).

Therefore, the way in which users can manage the workflows is foreseen to be done by the population of the conceptual data model, so that users do not need to know BPMN diagram tools or technical specifications. Consequently, it is still missing, but on the pipeline to be done, the automatic generation of a compliant BPMN file from the conceptual data model of the BPMN subset.

Orthogonal Model Consistency

The resulting software system requires that data semantics and behaviours be modelled, as already stated. However, behaviours involve data elements and data element constraints can trigger behaviours. These inter-dependencies also need to be reflected in the models. But how can we link these different modelling standards/methodologies while creating relationship links between both models?

This orthogonal consistency and integration is achieved by extending the conceptual data model with representation of the defining BPM workflow elements.

Definition of a behavioural task (e.g. Create Change Request) which handles a concept (e.g. Change Request) in a workflow (e.g. Change Request Process) will be represented through the following entities:

- A Process instance for the workflow identified by the name value “Change Request Process”.
- A Change Request instance for the specific change request named “D-00 Process 7.1 missing Input in Task 4” and identified with a unique UUID.
- A User Task instance for the task identified by the name value “Create Change Request” and which will require:
    - creation of an instance of an HMI Widget with name “CR Creation Form” and its content (HTML, etc.)
    - reference to the Change Request instance to establish the relationship with the User Task that will handle the concept.

Lastly, and to ensure consistency across models, it is important to persist in the orthogonal model not just the names, but the identifiers used in each of the referred models.

Exchange Interfaces

In order to ensure that E-RMS can be integrated into operational environments (at agencies, in industry) a comprehensive collection of data exchange interfaces will be supported, including ReqIF, DOORS, MS Excel, MS Word.

Conclusions

In the E-RMS Phase 1.2 activity, in progress, an integrated set of software requirements, conceptual data model and business process modelling capabilities have been developed, using precise data semantics defined in ORM and selecting a subset of BPMN that covers the needs of the ECSS development and usage processes. The next phase will be the implementation of this solid basis into a fully functional, operational software system, most likely on the basis of an existing solution. The objective is that ECSS itself, but also all uses of ECSS in projects in the European space sector, can evolve efficiently and effectively, using an integrated model-based engineering approach.

References

[1] ECSS-D-00B – ECSS organization and processes (12 September 2018)
https://ecss.nl/active-policy-document/ecss-d-00b-ecss-ecss-organization-and-processes-12-september-2018/

[2] Camunda Modeler
https://camunda.com/download/modeler/

[3] Object Role Modeling (ORM),
http://www.orm.net/