Covered workshop objectives: O-1, O-4, T-3, S-2

Using models for improving the efficiency of ESA mission operations

[1]Petros Pissias, [3]Cesar Guzman, [4]Miguel Rey, [2]Marcus Wallum, [5]Temesgen Gebremedhin

[1]petros.pissias@esa.int, European Space Agency, ESOC, Darmstadt, Germany
[2]marcus.wallum@esa.int, European Space Agency, ESOC, Darmstadt, Germany
[3]cesar.guzman@visionspace, VisionSpace Technologies, Darmstadt, Germany
[4]miguel.rey@visionspace, Visionspace Technologies, ESOC, Darmstadt, Germany
[5]temesgen.gebremedhin@visionspace.com, Visionspace Technologies, ESOC, Darmstadt, Germany

Abstract

In this paper, we present the combined output of two activities carried out as part of the ESA Discovery element [10] through the Open Space Innovation Platform [11], which have a similar goal: to integrate engineering models for enhancing operations at the European Space Operations Centre (ESOC) and improve the efficiency of routine operations for our space missions. The first activity, named "Enabling continuity: from design to operations"; focused on capturing and modelling information about the space system to improve mission operations in mission planning, operational procedures, and machine learning (ML). The second activity, named "MBSE and Space to Ground ICD"; focused on modelling information from the Space to Ground Interface Control Documents (SGICD) [12]. The goal is to automatically generate manually created documents and improve the efficiency of the initial configuration of next-generation mission control systems based on European Ground Systems Common Core (EGS-CC) [14].

Introduction

Model Based Systems Engineering (MBSE) is becoming a common practice in the space industry, supporting a digital engineering paradigm for design, assembly, integration, and testing of space systems. However, the reliance on MBSE often stops once the system leaves the supplier. The client/operator must then analyse the information (encoded models usually written down in text-based design documents, user manuals, and operational procedures) to configure all the needed systems (planning, data distribution, mission control, among others) and to build all the supporting knowledge and procedures required by the Mission Operations Centre (MOC) to operate the space system. Furthermore, different operational teams within the MOC typically must produce other documents or databases to include, maintain and internally share all the previously collected information needed to configure each system. Both presented activities aimed at improving this situation by defining and using system models to increase the efficiency and quality of various tasks now done manually.

The identification and selection of the use cases to be implemented with Proof of Concept (PoC) applications in the context of the activities resulted in mission preparations and routine operations that could be most positively impacted, considering:

- the effect of having MBSE tools acting as a hub to store and guarantee the consistency, integrity and latest available version of engineering data, and how it could simplify and increase the reliability of those activities depending on associated data (requirements, constraints, parameters, other).
- the possibility of having all teams on the same page in terms of i) accessing the necessary data, ii) simplifying the sharing of information, and iii) reducing the number of frequent and temporally expensive interactions between different teams.
- the possibility of automating data extraction and verification.

To aid in the elicitation of these use cases, we conducted several interviews covering different aspects of mission operations (e.g., Mission Planning and Procedure Preparation, Execution and FDIR, others). The following use cases were down-selected:

1. **Generate/update planning domain model**: configuration of a mission planning system
2. **Verify Mission Planning System (MPS) models**: verify mission planning system domain models
3. **Generate/update procedure**: generate operational procedures in the Operations Preparation ENvironment (OPEN)
4. **Verify procedure**: verify operational procedures in OPEN

In addition, we identified several potential improvements by modelling part of the information in the SGICD [z]. The main identified use cases in this area were to automatically configure part of an EGOS-CC system and automatically generate parts of documents that are now manually created. Moreover, we explored the usage of models for **supporting the development and deployment of Machine Learning (ML).**

In the next sections, the following areas are presented in more detail. The usage of models for:

- the **configuration of systems (EGOS-CC and MPS) and document generation**
- the **validation of operational artifacts**

- the **definition of operational procedures**
- aiding the **development and deployment of machine learning – use cases**

Configuration of systems (EGOS-CC and MPS) and document generation

For the **EGOS-CC configuration**, the main information sources were typical SGICD documents. Those documents are the central part for monitoring and control, data processing and distribution of monitoring and control data received from the satellite. They are typically iterated between Spacecraft (S/C) Prime and the MOC, and while some of this information is indeed unique, most of it is in fact already covered by the existing ground segment functionality and requires no development. This information includes Ground Station Link Geometry, Uplink, Downlink, Virtual Channel, MAP ID, S/C ID, etc.

To a degree, the effort to set up the ground segment from these documents has already been mitigated by having the satellite provider deliver the onboard S/C database (e.g., SCOS Mission Information Base), which contains the telemetry and telecommand packet definitions. Nevertheless, plenty of the available information is later mapped into the Monitoring and Control System Software Requirement Specification (MCSSRS) and only eventually into configuration variables of the control system. The goal is to greatly improve the efficiency of setting up an initial deployment of an EGOS-CC-based mission control system. Existing ICD documents that are inputs for missions served as the basis for selecting key items for a model.

For the **PoC application**, several modelling approaches have been analysed, and a combination of **Object Role Modelling (ORM)** and **Eclipse Modelling Framework (EMF)** has been selected; the first to be used as the conceptual model, and the second as the machine implementation. ORM conceptual models focus on facts of interest to capture the domains in unambiguous semantics. EMF is an Eclipse-based modelling framework and code generation facility.

With the help of EMF, a PoC tool has been developed and integrated into OPEN (See Figure 1), the EGS-CC preparation environment. With this tool the S2GICD model can be visualized and edited, and the required EGS-CC configuration can be automatically generated. This generation can be integrated into an automated process.

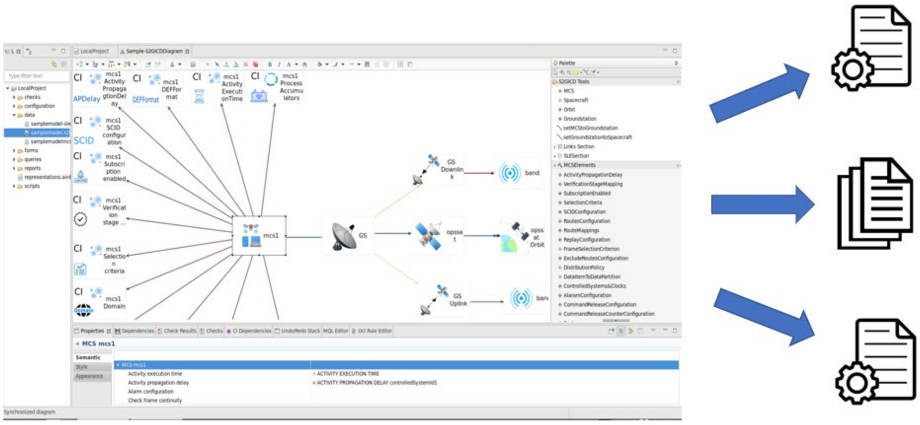

Figure 1. Prototype editor for implementing the S2GICD model

S2GICD editor generates multiple artifacts. For the **MPS configuration**, given a model of a S/C subsystem in the MBSE Hub, the planning domain model file for the subsystem (part of the configuration of the mission planning system) is automatically generated. This application demonstrates how the configuration of an MPS can rely on an MBSE infrastructure. The PoC application targeted a domain file (DDL) for ESA's APSI planning framework but could also apply to the EGOS-MPS rules file.

Validation of operational artifacts

Given a planning domain model of a S/C subsystem, we can automatically verify compliance with the subsystem model at the MBSE hub by verifying the configuration of the **MPS** with the MBSE infrastructure (e.g., state model, constraints checking, others). The PoC application targeted a domain file (DDL) for ESA's APSI planning framework but could also apply to the EGOS-MPS rules file.

For validating some aspects of the **operational procedure**, given the operational (OPEN) procedure and a formal representation of the constraints of a subsystem in the MBSE Hub, automatically check OPEN against these constraints. It is a semantic check that ensures the operational procedure respects spacecraft constraints, certain timing constraints, and others.

Generation of operational procedures

Given a subsystem's model in MBSE Hub, we can automatically generate a goal-oriented controller (EGS-CC Activity implemented in OPEN) to manage the subsystem's state. The controller can be subsequently validated against a simulator. The controller is built around Weighted Finite State Automata (WFSA) and encodes the monitoring and control logic needed to move the system between states. Given the target state (the goal), the controller can find the best path to move from the current state to the target and maintain the state in the presence of perturbations' (e.g., exogenous events cause the subsystem to move from the target state). We simulated the OPEN Activity application using an EGS-CC-based mission configuration.

Development and deployment of machine learning - use cases

For environments with very low risk threshold, such as S/C operations, ML models must be well defined and well tested such that they are deemed safe for deployment. The design and operation of ML models can be seen through the same lens as that of an MBSE infrastructure, particularly for learning assurance [6]. Quality measures of ML systems affected by low robustness are like those affected by a low-quality requirements specification in MBSE [7]. On the other hand, some techniques in ML can improve the development of MBSE models. In particular, the causal inference, where the causal effect of a specific data variable within a huger system can be understood [8] and helps to create the relationship in the MBSE model.

We evaluated MBSE for ML pipelines in S/C mission operations by addressing a more autonomous process to support business understanding, design, verification, and validation of the ML models used in S/C operations by formalising the knowledge of Space Systems and ML domains in an MBSE model (ORM, OWL, RDF, SysML, among others), that then helps the data science experts to automate feature extraction for ML model building and deployment. Figure 1 shows the overall Architecture inspired upon the CRISP-ML [5], and MLOps [1]. For the PoC validation purposes, a mock datahub containing an RDF/OWL model of the Mars Express mission was used. The Quality Gates (QG) ensures the data quality [3], representativeness, robustness, and trustworthiness of the ML models. Note that quality is not only defined by the product's fitness for its purpose, but the quality of the task executions in any phase of the ML application.

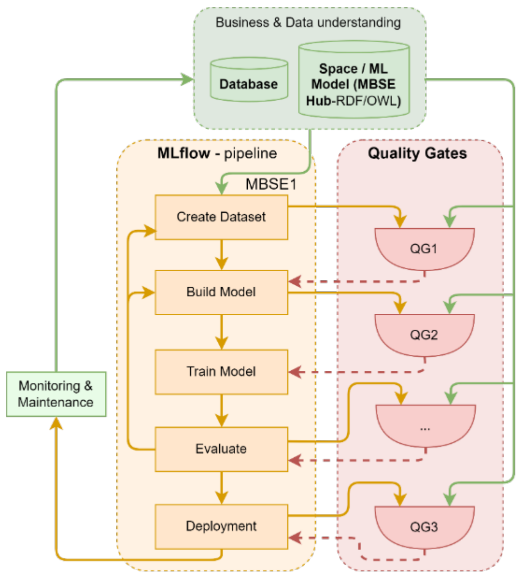

Figure 2. Machine Learning pipeline with Quality Gates

QG1 - The data is representative: ensure training data's quality and representativeness to improve the ML models robustness.

QG2 - The algorithm is appropriate: gather all relevant information about the problem to select the appropriate ML Algorithm.

QG3 - Final output model passes constraints: ensure the ML application provides valid outputs and they are within some operational constraints. So, the model can safely deploy in operations.

Each QG has one or a set of Quality Checks (QC), e.g., QG1 has a QC that through the great expectation library checks that all the relevant features encompass some given constraints, another QC verifies the expected distribution of the data.

ML Engineers could automatically identify relevant features that they can include in the training dataset for building ML models to deploy (happens during the step create dataset of the pipeline). Let us consider the example application of predicting the value of subsystem's electric power consumption, NPWD2372. The ML ontology can aid in the identification of relevant features located in the dataset (explanatory features) thanks to the MBSE ontology. E.g., the solar aspect angle, solar angle of the X axis of satellite. The data scientists do not have to be experts in the domain since all the information is in the ontology. Figure 3 depicts the mentioned idea.

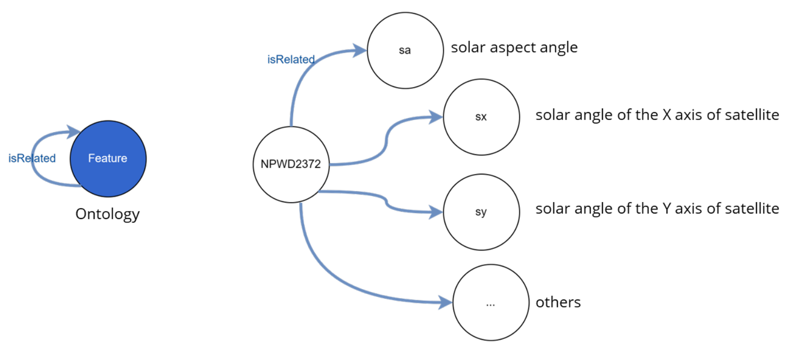

Figure 3. Search for the explanatory features

Conclusion

At the end of these activities, it is demonstrated that there is a clear benefit of using model-based approaches in mission operations. The activities and proof of concept implementations demonstrated that we can use engineering models to

- Improve the efficiency of the configuration of EGS-CC based systems and Mission Planning Systems
- Generate parts of manually generated documents in the context of mission operations
- Validate Mission planning systems configuration and operational procedures
- Generate operational procedures (or at least skeletons of operational procedures)
- Aid the development and deployment of machine learning use cases

As next steps and future work, the work presented in this paper, besides potential enhancements with the tools themselves in order to be more efficiently used in the ESOC operational environment, will aid future tool and infrastructure developments such as the Model Based Engineering Hub and others under the MB4SE advisory group roadmap.

References

[1] Sridhar Alla and Suman Kalyan Adari. What is mlops? In Beginning MLOps with MLFlow, pages 79–124. Springer, 2021.

[2] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. The crisp-dm user guide. In 4th CRISP-DM SIG Workshop in Brussels in March, volume 1999. sn, 1999.

[3] IEEE. Standard for software quality assurance plans. IEEE Std 730-1998, pages 1–24, 1998.

[4] Sara C Spangelo, David Kaslow, Chris Delp, Bjorn Cole, Louise Anderson, Elyse Fosse, Brett Sam Gilbert, Leo Hartman, Theodore Kahn, and James Cutler. Applying model based systems engineering (mbse) to a standard cubesat. In 2012 IEEE aerospace conference, pages 1–20. IEEE, 2012.

[5] Stefan Studer, Thanh Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. Towards crisp-ml (q): a machine learning process model with quality assurance methodology. Machine Learning and Knowledge Extraction, 3(2):392–413, 2021.

[6] P. Wilson and H. A. Mantooth, "Chapter 6 - Block Diagram Modeling and System Analysis," inModel-Based Engineering for Complex Electronic Systems, P. Wilson and H. A. Mantooth, Eds. Oxford: Newnes, 2013, pp. 169–196.

[7] R. Marinescu, C. Seceleanu, H. Le Guen, and P. Pettersson, "Chapter Three - A Research Overview of Tool-Supported Model-based Testing of Requirements-based Designs," vol. 98, A. R. Hurson, Ed. Elsevier, 2015, pp. 89–140.

[8] C. E. Perez, "The Meta Model and Meta Meta-Model of Deep Learning," 2016. https://medium.com/intuitionmachine/the-meta-model-and-meta-meta-model-of-deep-learning-10062f0bf74c.

[9] Fratini, S., & Cesta, A. (2012). The APSI framework: a platform for timeline synthesis. Proc. The 4th International Competition on Knowledge Engineering for Planning and Scheduling, 8-15.

[10] ESA discovery and preparation elements
https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Discovery_and_Preparation

[11] Open Space Innovation Platform (OSIP)
https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Discovery_and_Preparation/The_Open_Space_Innovation_Platform_OSIP

[12] ECSS ECSS-E-ST-70C Space engineering, Ground systems and operations.

[13] http://www.egscc.esa.int/

[14] The EGS-CC based Mission Control Infrastructure at ESOC, Simulation and EGSE for Space Programmes SESP 2017 conference, https://indico.esa.int/event/180/timetable/?view=standard