End-to-End System Engineering Portal (ESEP)

Dr. T. Stoitsev

SpaceCube GmbH

Abstract

Space system engineering is a complex activity, spanning multiple phases and involving different stakeholders using a variety of engineering approaches and tools. ESA has been investing in Model-Based System Engineering (MBSE) research for many years, developing different MBSE frameworks and tools. Most of these have been focused on specific engineering phases and domains (ground or space). Recently, a broader need has been recognized across space institutions and industry in Europe to focus the MBSE efforts on semantic interoperability and associated model integration. This has led to the Model Based for System Engineering initiative, aiming to guide the development of a common Space System Ontology (SSO) and a Model Based Engineering Hub (MBEH) to support both aspects. In this context, the End-to-End Systems Engineering Portal (ESEP) represents a downstream application, offering a federated, web-based User Interface (UI) layer on top of the MBEH infrastructure. The objective is to offer a UI environment where system engineering users with different MBSE background and skills can integrate data, identify data gaps, and transition models between different engineering phases and formats using intuitive techniques at the UI level. Due to various programmatic reasons, the ESEP activity has been completed before the MBEH technical specification has been finalized and development started. As a result, a number of MBEH aspects, especially on the data management (infrastructure) level, have been covered in the ESEP prototype, and it is intended to reuse this prototype fully as a starting point for the MBEH development in one of the MBEH consortia. On conceptual level, the ESEP prototype inherits many of the Ground Segment Engineering Framework (GSEF) capabilities. However, the ESEP uses a completely new data management approach featuring a variety of fit-for-purpose data stores and formats and the prototype has been implemented largely from scratch.

1 Background and Objectives

Space system engineering is a complex activity, spanning multiple phases and involving different stakeholders using a variety of engineering approaches and tools. ESA has been investing in Model-Based System Engineering (MBSE) research for many years, developing different MBSE frameworks and Domain Specific Tools (DSTs). Recently, a broader need has been recognized to focus the MBSE efforts on semantic interoperability and associated model integration. This has led to the Model Based for System Engineering (MB4SE) initiative, aiming to guide the development of a common Space Systems Ontology (SSO) [1] and a Model Based Engineering Hub (MBEH) [2] to support both aspects. The objective of the MBEH is to enable integration and exchange of engineering data originating from different DSTs along the space systems engineering lifecycle, based on common semantics defined through the SSO.

The main objective of the ESEP is to provide a federated, web-based User Interface (UI) layer on top of the MBEH infrastructure, where stakeholders are able to integrate and transfer data across DSTs by using the underlying hub infrastructure. The MBEH is developed in a dedicated project, which is split in two phases as described in [2]. The first phase shall elaborate a set of system engineering use cases, elicit new and consolidate existing MBEH requirements, and produce an MBEH technical specification. The MBEH detailed design and development shall take place in the second phase. Due to various programmatic reasons, the MBEH activity started after the ESEP activity, and it is expected that the first phase of the MBEH activity will end close to or after completion of the ESEP activity. As a result, no assumptions on the existence or the design of the MBEH can be made for the ESEP design and development. The approach agreed with the Agency for the ESEP activity is to consider the ESEP as a complete system by itself, without making any assumptions about the underlying MBEH infrastructure. Further, depending on the MBEH technical specification elaborated in the MBEH activity, the overall ESEP software or individual components will be reused fully or partially, and evolved further under the MBEH activity through the consortium where SpaceCube is responsible for the MBEH implementation.

2 Solution Overview and Rationale

Figure 1 provides a conceptual overview of the ESEP solution. The presented concept is closely aligned with the Ground Segment Engineering Framework (GSEF) concept [3]. Indeed, the ESEP can be seen as a major GSEF evolution, which focuses on providing a data integration and transformation environment, rather than an engineering (modelling) environment for a specific domain, such as the ground segment system engineering domain addressed through GSEF. Thereby, the integration with different DSTs and the enhanced support for dealing with different data models arise as central concerns for the ESEP.

Particularly, it is expected that each DST would support a specific (own) Domain Data Model (DDM) that will be used for elaborating the specific Domain Models (DMs), i.e. engineering models representing the system engineering products developed with the DST. It is important to note that what is implemented in a DST as well as ESEP (and later-on MBEH), is a physical data model, while the SSO is expected to provide a conceptual data model. A physical data model is the implementation of a conceptual data model, taking into account design choices and language constraints for the selected implementation platform. That is, the physical data model may have elements without semantic meaning or different from the ontology due to the tooling and/or language constraints. Thereby, it is expected that a one-to-one mapping between the conceptual and physical data models may not be possible.

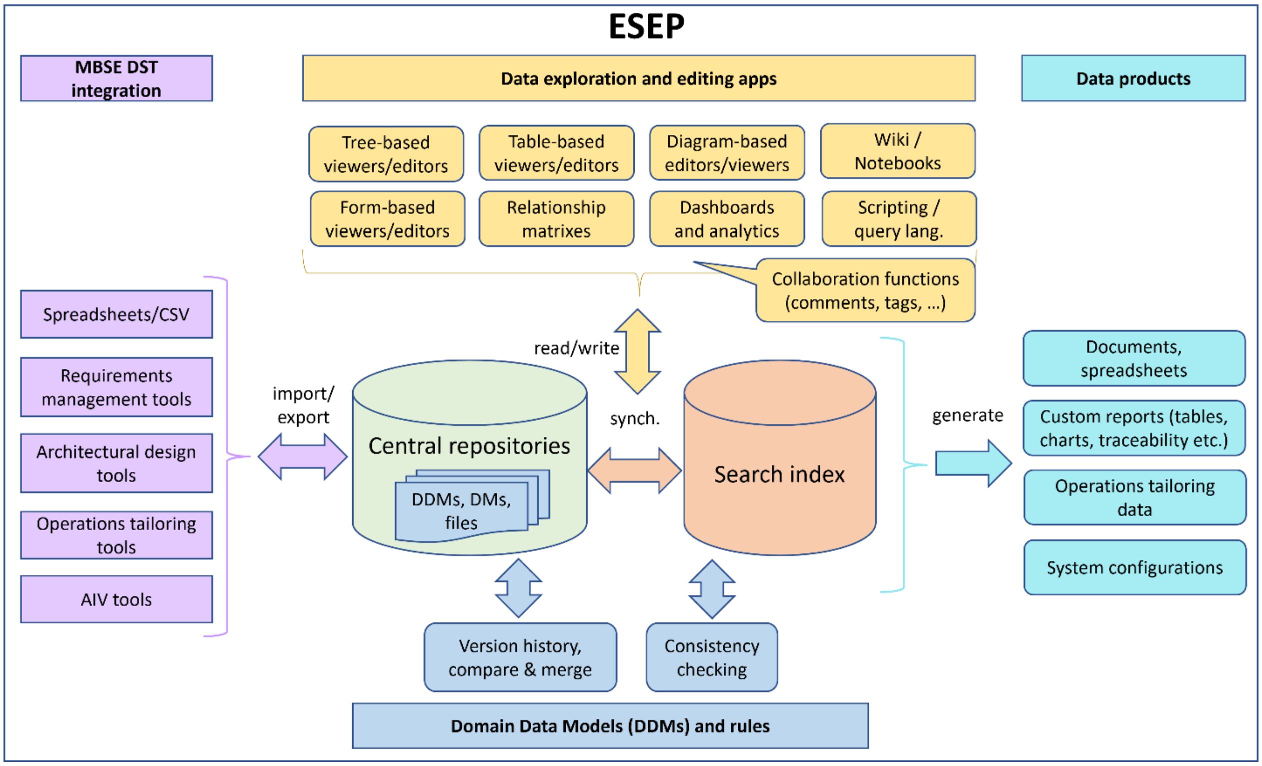

Figure 1 – ESEP High Level Concept Overview

The core architecture of the ESEP framework is derived from the GSEF [3] on conceptual level but is largely redesigned and reimplemented on a different technology platform to address the required ESEP capabilities. While the GSEF can be seen as a DST itself, focusing on elaborating engineering data (i.e. modelling) for the operations ground segment system engineering, the key focus of the ESEP is on supporting integration and transfer of engineering data across DSTs.

[4] describes a set of use cases and associated user requirements, which have been used to derive a set of functional capabilities to be supported by the ESEP framework. The elicited ESEP functional capabilities can be summarized as follows:

- Standardized representation of different DST DDMs and DMs through appropriate data provider abstraction. The Open Services for Lifecycle Collaboration (OSLC) [5] can be used as the conceptual framework for the standardized DST data representation and integration approach.
- DST DDM translation to an internal ESEP DDM, aligned with the SSO, for supporting integration and transfer of associated DMs originating from different DSTs over the ESEP, by enforcing the SSO semantics.
- DDM import in supported formats such as the Eclipse Modeling Framework (EMF) Ecore metamodel format [6], allowing usage of the overall ESEP system with the given metamodel as the internal ESEP DDM.
- Import of DST DMs into an internal DM representation, aligned with an associated DDM within the ESEP.
- Import and integration of multiple DST DMs into a single, internal DM aligned with a given DDM within the ESEP. Such a DM represents a Global System Model, which can be used for generating added-value engineering products (technical budgets) or for analysis results, analogously to individual DMs captured in the ESEP, i.e. no explicit distinction is made, whether a DM represents a Global System Model or a single model, imported from one DST.

- Central repository for all DDMs and DMs managed in the ESEP, with a data structure that is defined according to a formal, abstract Generic Data Model (GDM). The GDM is in fact a meta-metamodel, i.e. a metamodel, which is used to define metamodels for system engineering models.
- A search index storing derived models in a way that is as close as possible to specific user and/or application needs, allowing enhanced data queries and full-text search.
- Version control of all data, including DDMs, DMs, as well as configurations for data integration and transfer through the ESEP.
- Web based frontend implemented as an Angular 12 Single Page Application (SPA), featuring intuitive capabilities for data exploration and editing using standard mechanisms such as trees, tables, forms, and diagrams.
    - Integration of DSTs and/or associated capabilities through dedicated Micro-Frontend Components (MFCs).
    - Support for dashboards for data analysis, using the underlying search index storage.
- Collaboration covering:
    - Ad-hoc discussions on DDM or DM data.
    - Review workflows for updates on DDM or DM data.
    - Push notifications at various levels – discussions, reviews, DDM or DM data updates.
- Consistency checking of DMs within the ESEP according to the associated DDMs.
- Artifact (e.g. document) generation from DMs captured in the internal ESEP representation.
- Security through OpenID Connect (OIDC) [7] on top of OAuth2 [8] based on the Keycloak Single-Sign-On implementation. The latter also allows federation of different security realms and integration e.g. of Lightweight Directory Access Protocol (LDAP) for the user authentication. Security can be configured at different levels – user roles, OAuth2 scopes, OIDC audiences, which provides comprehensive support for multi-tenancy and federative system usage as well as for integration of different system assets and services e.g. in the context of automated data ingestion workflows.

The ESEP prototype has a state-of-the-art microservices architecture based on the Spring Cloud stack, featuring Spring Cloud Gateway, Netflix Eureka service discovery and Netflix Ribbon client-side load balancing for the microservices. The communication between the different microservices as well as the front-end and the backend goes through service discovery and client-side load balancer(s). This also enables a distributed, cloud-based deployment with enhanced availability and fail-over support.

With respect to the MBEH, it is expected that the following major areas outlined above will be evolved further in the context of the MBEH development:

1. DST data import/export and associated adapter infrastructure
2. Central repositories and data management, especially with respect to branching, comparison and merge, and consistency checking.
3. Data products generation based on integrated data from the central repositories and search index

3 Main Achievemets

<u>Data Model Levels</u>

The ESEP utilizes a data management approach that supports the following data model levels:

1. A highly abstract Generic Data Model (GDM) based on the Essential Meta Object Facility (EMOF) [9] metamodel and particularly its implementation into the EMF Ecore. The GDM is used as meta-metamodel to specify other data models (i.e. metamodels) for specific engineering domains such as requirements engineering, architectural design, verification and validation.

2. A set of data models (metamodels) for the different engineering domains mentioned e.g. in the point above. In the following such a data model is referred to as a Domain Data Model (DDM). A model that represents an instance of a DDM is referred to as a Domain Model (DM) and an object within a DM is referred to as a Domain Model Object (DMO). DMs are generally expected to result from ingesting DST data into the ESEP.
3. Support for having multiple versions of a DDM with multiple associated DM versions for each DDM version at the same time in the ESEP.

This approach facilitates the mapping and transformation of engineering data produced by a given MBSE DST in line with its internal DDM to an equivalent DDM representation in the ESEP, especially for DST DDMs compatible with the GDM (i.e. EMOF). Ultimately, this approach allows importing e.g. requirements, design elements, operations tailoring data or any custom DST data from different DSTs into the ESEP and exploring and managing the data in parallel in the ESEP environment by using different DDMs corresponding to the associated DST DDMs. For example, a concurrent design model can co-exist in the ESEP with a Capella, GSEF or operations tailoring model based on the European Ground Systems Common Core (EGS-CC) Conceptual Data Model (CDM) [10]. Thereby a system engineer would be able to explore at the same time a requirement originating from a requirements management DST, the associated design elements such as components or interfaces that the requirement is traced to, and the operations tailoring data used e.g. for the operational validation of the components and interfaces.

It is important to note that the main use case for the ESEP/MBEH would be to perform the DST engineering data mappings and transformation based on a central ESEP/MBEH DDM that represents the common Space Systems Ontology (SSO). However, even while the SSO is still under development, the ESEP can be used as a relaxed digital engineering integration and exploration environment that favors different DST DDMs in parallel. The described ESEP approach has been validated based on realistic ground segment system engineering use cases and associated data sets from the GSEF development context, i.e. utilizing the GSEF DDM.

Hybrid Data Management Approach & View-Based Interfaces

The ESEP embraces the idea that there is no ‘one-size-fits-all’ MBSE solution, not only at the data model level but also at the level of individual system engineering use cases and application contexts. Particularly, it is recognized that in some cases, enhanced branching and merging capabilities may need to be supported, similarly to Git based solutions used for software development. In other cases, fine-grained change tracking at the level of individual attribute or reference changes for an engineering data object may be needed, e.g. similarly to change tracking in JIRA or Confluence. In some cases, consistency checks based on formal languages such as the Object Constraint Language (OCL) may be needed. In other cases, it may be required to support consistency check queries based on languages such as SQL or even custom search index DSL queries, which go beyond what OCL can offer (e.g. fuzzy search queries, geospatial queries etc.). In some cases, collaborative engineering data editing with simultaneous access by multiple users to the same engineering data and direct visibility of changes made by other users may be needed. In other cases, a distributed engineering approach similar to software development based on Git may be needed, where every user has their own version of the engineering data and synchronization (merge) is performed when and as needed. In many of these areas the associated requirements are conflicting at the level of the supporting technical frameworks and approaches. While GSEF covers the Git-based approach, the ESEP prototype has proven the feasibility of the alternative approach featuring fine-grained change tracking and direct data updates by different users. Thereby, ESEP foresees the possibility to support both approaches in parallel by using enhanced synchronization across the different data stores and formats.

A further important achievement of the ESEP is the utilization of view-based interfaces from the beginning, utilizing configurations related to a concrete DDM and defined on top of the GDM. These configurations are the key for supporting multiple different DDMs and versions thereof with multiple associated DMs and versions thereof in the ESEP. The configurations are also the key enablers for the view-based interfaces, where only the data relevant for a given view (tree, table, details form) of a DM or DMO is retrieved from the server, leading to enhanced performance also in case of limited network capabilities.

The ESEP hybrid data management approach goes even further, by using generation of derived models, capturing specific DM representations tailored to specific use cases and information needs. Therewith, efficient data queries and engineering data analysis can be performed, which can be applied for any DM based on the underlying DDM. The ability to tailor the ESEP for a given DDM from the UI over the view-based interfaces, down to the persistence layer (with respect to derived models) offers enhanced flexibility for addressing the MBEH use cases and for accommodating on-going SSO developments.

Data Mapping and Import

One of the key capabilities of the ESEP is the mapping and transformation of DMs originating from DSTs for their import into the ESEP. Import/export of engineering data in Comma Separated Values (CSV) format is common for many engineering tools and used for a number of engineering data exchanges along the space system engineering lifecycle. Indeed, in many engineering areas system engineers use extensively standard software tools such as e.g. Microsoft (MS) Office Excel. CSV import and export into MS Excel were addressed as key use cases in GSEF and exercised with real mission data. The CSV import in the GSEF is limited to specifics of the underlying GSEF DDM and the Eclipse Modeling Framework (EMF) data management libraries and only specific types of constructs can be mapped and imported. On the other hand, a more generic CSV import capability has been prototyped in the ESEP, allowing mappings across multiple DMO reference levels. Although such mappings are not adequate for a fully EMF based approach where an object needs to be contained in another object or in an EMF resource to be persisted in an EMF resource in the first place, these mappings and transformations are relevant for a more generic solution that can depart from the pure EMF mechanisms and persistence approach.

Data Comparison

Comparison between different versions of the same DM/DMO or comparison between different DMOs/DMs are highly relevant in the context of the ESEP/MBEH use cases. Comparison of object-oriented data structures is a complex topic, which has not been addressed in the GSEF. On the other hand, the ESEP prototype includes features allowing comparison between different versions of the same DMO, but also between different DMOs, even such that are based on different DDMs. This approach offers a lot of flexibility, which is especially relevant in the context of the enhanced data management framework. Particularly, while being able to import and manage data from different MBSE DSTs, conforming to different DDMs (or different versions of the same DDM), the ESEP comparison capabilities allow a user to compare not only the values of various DMO attributes and references, but also to see the differences between the attributes and references defined for the given type of DMO in the associated DDMs on both sides. This allows implicitly also comparison of the underlying DDMs when DMOs are being compared.

4 Conclusions

ESEP has been designed and developed with the MBEH use cases in mind. Although GSEF has been used a conceptual starting point for the ESEP prototyping, the resulting ESEP prototype has been implemented largely from scratch, based on a completely different data management approach, addressing more adequately the ESEP/MBEH use cases and known limitations of the GSEF approach. The ESEP prototype has demonstrated that the selected approach is feasible and can be successfully employed to build comprehensive engineering data integration and management environments, using production-grade technical frameworks and libraries, which diverge from classical MBSE approaches based e.g. on EMF.

ESEP embraces the concept that there is no ‘one-size-fits-all’ solution not only at a level of a Domain Data Model (DDM), but also at the level of engineering use cases in general. All these different use cases may need to be supported to a different extent in different engineering (organizational) contexts and they raise different, often conflicting requirements in terms of technical implementation and available software libraries. The ESEP recognizes this reality and addresses it through a combination of different methods, techniques, and software modules that can handle efficiently the various use cases and contribute individually and as a whole to the overall ESEP solution.

5 Next Steps

It is envisaged to use the ESEP prototype as a starting point for the development of the MBEH in the consortium led by Airbus, where SpaceCube is responsible for the MBEH technical implementation. In the context of the ESEP activity, SpaceCube has communicated to ESA that in order to be efficient and to maximize return on investment for the Agency, efforts in various areas, including MBSE, shall be streamlined along a product portfolio. Thereby, various studies shall be organized in such a way, that they are not self-standing and producing ‘throw away’ prototypes, but instead contribute to a single product line and ideally to a comprehensive product/solution that produces added value for ESA and European Space industry at large. SpaceCube has adopted this strategy from their side and are in fact combining the outcomes and experiences from the GSEF development and the ESEP development towards a comprehensive solution that can be reused and further evolved under the MBEH activity. Known areas from GSEF that could not be realistically covered in the scope of the ESEP activity but could be complemented in the context of the MBEH study include the following:

1. GSEF: Configurable export to Microsoft Excel
2. GSEF: Consistency checks
3. GSEF: Document generation
4. GSEF: Collaboration support (change reviews and formal reviews, discussions, push notifications, labels)

While some of these areas are not reflected through formal requirements in [2], real-life applications of GSEF have clearly exemplified that these capabilities are highly relevant for space systems engineering at large, not only in the ground segment engineering domain. The specific areas to further focus on and elaborate during the MBEH activity will be clarified in the context of the MBEH study.

References

1. ESA, Space System Ontology Development, Statement of Work, ESA-TECSWM-SOW-015604, v1.0, 19 Oct 2019

2. ESA, Model-Based Engineering Hub, Statement of Work, ESA-TRP-TECSYE-SOW-014909, Issue 2, Revision 1, 1 Feb 2021

3. T. Stoitsev, Paperless End-to-End Ground Segment Engineering (PLGSE) final delivery

4. T. Stoitsev, S. Jahnke, End-to-End System Engineering Portal (ESEP) - User Requirements, ESEP.TN.URQ, Issue 1.0, 20 Oct 2021

5. Open Services for Lifecycle Collaboration, Core Specification, https://docs.oasis-open-projects.org/oslc-op/core/v3.0/ps02/oslc-core.html

6. Eclipse Modeling Framework, Ecore, https://download.eclipse.org/modeling/emf/emf/javadoc/2.11/org/eclipse/emf/ecore/package-summary.html

7. Open ID Connect (OIDC), https://openid.net/connect/

8. OAuth2 Authorization Framework, https://oauth.net/2/

9. Object Management Group, Meta Object Facility (MOF) Core Specification, formal/2019-10-01, Version 2.5.1, October 2019

10. EGS-CC System Engineering Team, Conceptual Data Model, EGSCC-SYS-TN-1004, Issue 1.15a, 26 Apr 2019