

Digital Engineering Hub Pathfinder

Extending the ECSS-E-TM-10-25 Tool Ecosystem

A. Vorobiev^{1*}, S. Gerené², N. Smiechowski¹, S. Jahnke³, J. Knippschild³, S. Weikert⁴, M. Becker⁴, S. Paquay⁵, J-P. H. Vogt⁵, I. Fontaine⁵

¹RHEA Group, Diegem, Belgium; a.vorobiev@rheagroup.com

²RHEA Group, Leiden, The Netherlands

³OHB System AG, Bremen, Germany

⁴Astos Solutions, Stuttgart, Germany

⁵Open Engineering, Angleur, Belgium

Introduction

ECSS-E-TM-10-25A is a technical memorandum published under the E-10 System Engineering branch in the European Cooperation for Space Standardization (ECSS) series of standards, handbooks, and technical memoranda. The purpose of ECSS-E-TM-10-25A is to define and promote common data definitions and exchange in the context of concurrent design, and engineering model reviews. Primarily intended to be used in Phases 0/A, it provides the facilities to create a decomposition of a system on a logical and physical architectural level with associated strongly typed parameters, supported by concepts of ownership by domain experts [1]. The protocol defined in ECSS-E-TM-10-25A was developed with concurrent Model Based System Engineering in mind from the start, where exchange of information would occur in near-real time between any arbitrary number of active engineers.

Multiple software implementations of ECSS-E-TM-10-25A exist, such as the ESA Open Concurrent Design Tool (OCDT) [4], RHEA COMET[®] (formerly known as Concurrent Design Platform (CDP4[®])) [2] and COSM from Blue Engineering [3]. All of them, however, offer very little in terms of direct interoperability with

industry standard domain specific tools (DSTs). The only practical solution for the end user to facilitate the exchange of information would be limited to either manual creation of architectures and management of parameter values, or development of connections and models in Excel, which is supported from both OCDT and COMET[®].

This shortcoming has the unfortunate side-effect of making ECSS-E-TM-10-25A look unattractive as a central data hub for users of complex simulation and analysis tools, as well as graphical architecture modeling tools as double work and potential human errors are unavoidable, yet the advantages of using those tools cannot be denied. They, in turn, almost unanimously do not address the need for concurrent engineering or interoperability with other domain specific tools, which creates a gap between the typical phase 0/A engineering activities and those following, where these DSTs become more dominant.

Thus, one of the main obstacles of full deployment of Model Based System Engineering (MBSE) in space projects is identified: a lack of a centralized, real time “data hub” that is able to seamlessly interact in a bi-directional manner with industry standard tools, bridging the gap between concurrent engineering and the

preliminary and detailed design phases that follow.

Digital Engineering Hub Pathfinder sets out to demonstrate that, while complex, this concept is entirely achievable. ECSS-E-TM-10-25 is at the heart of the DEHP project, where an Annex C.2-enabled data source (COMET® Web Services or OCDT Web Service Processor (WSP) for example) is used to provide, collect, and persist data from DSTs in an automatic, yet configurable fashion (see Figure 1).

In part supported by the work performed in previous European Space Agency (ESA) activities on explorative integration, in the frame of the DEHP project the following DSTs are integrated into the ECSS-E-TM-10-25 ecosystem via development of what we refer to as “DST Adapters”, undertaken by the identified companies:

- Simulation Applications: **Matlab (RHEA), Ecosim Pro (RHEA), ASTOS (Astos Solutions)**
- Thermo-Mechanical Engineering: **Catia v5 (RHEA), STEP-TAS/ESATAN-TMS (Open Engineering), STEP 3D CAD (Open Engineering)**
- System Engineering/Architecting: **MagicDraw SysML (RHEA), Enterprise Architect SysML (RHEA), Capella (RHEA)**
- Space Specific System Engineering/Architecting: **MARVL CIP (RHEA)**

Requirements Elicitation

In order to support the common tasks of the users for the different DSTs, a survey among them consisting of a questionnaire and follow-on interviews have been conducted. The survey is

conceived to elicit the typical tasks and interaction of the different tools, their in- & outputs down to a level of detail of parameter types and data constructs to be exchanged as well as involved stakeholders. This information is used to define the Concept of Operations for individual tools and the data hub itself, as well as

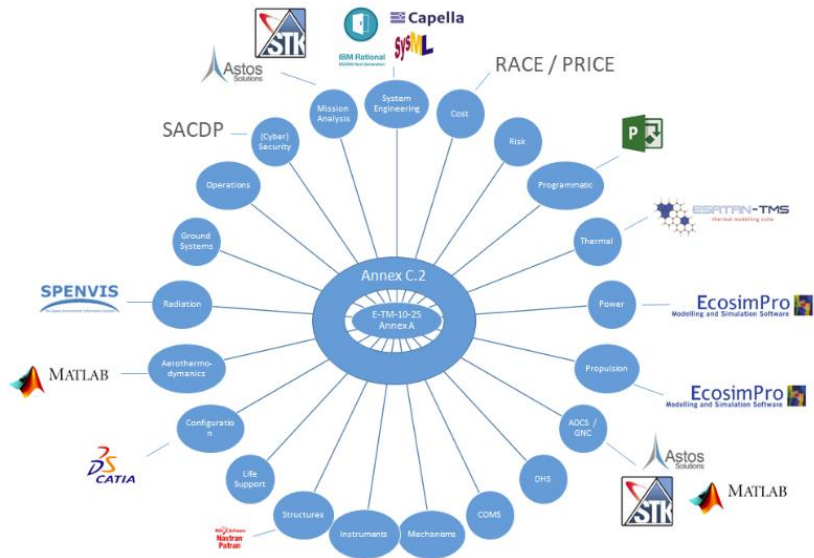


Figure 1: The spoke wheel concept of the Engineering Hub.

to identify commonly used information and interdependencies in the different analysis across domains and tools. In addition, the users are requested to define supporting features and capabilities they would need to efficiently use the DST adapters. The Use Cases, Concept of Operations and User Requirements are documented in the Software System Specification. User Requirements are categorized in General, which should be applicable for any developed DST adapter, and DST specific to cover more specialized needs and particularities.

Architecture

The principle of all adapters is the same: a piece of software that acts as a user and application

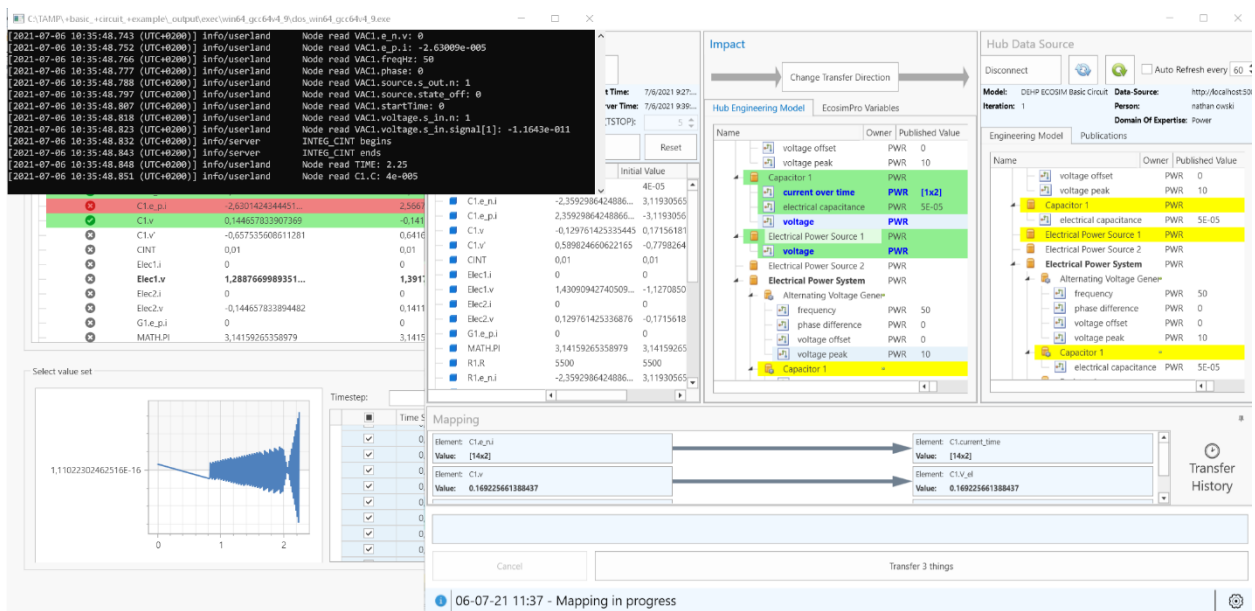


Figure 2: EcosimPro DST Adapter in Action

interface between the DST and its model and the hub. The user interface allows for inspection, configuration, and initialization of transfer. The application interface on one hand communicates to the hub via ECSS-E-TM-10-25 Annex C.2 RESTful protocol, and on the other via whatever Application Programming Interface (API) the DST allows.

Three distinct types of software architectures were identified to support the development of the adapters:

- DST Adapter Plugin: direct integration into the DST in question, utilizing the underlying infrastructure, user interface elements, programming language and access to the model. This option presents the simplest approach to regular users of the DST as transfer can be initiated directly from the tool.
- DST Adapter Bridge: a standalone application directly interfacing with a running instance of the DST on the client computer through the means of either COM Interop or some other API that the DST provides.

- DST Adapter Standalone: primarily aimed at exchange of standardized file formats rather than specific tools, such as STEP-TAS and STEP AP-242 through parsing files and extracting information inside of them.

Distinction is also made between the types of information being exchanged. Some of the adapters deal purely with mapping of arbitrary workspace variables to existing ECSS-E-TM-10-25 parameters, while others make much more complicated model transformations that influence the architecture of the system being designed. Either way, the same isolated component takes care of handling the mapping and the transfer, leaving the adapter developers focusing on writing the rules that govern the transformation from model A to model B while the mechanism remains the same.

Special care was taken to identify reusable components that can be shared between the adapters, when possible, as well as a basis for any future integrations outside the scope of this activity. These components are then gathered into standalone libraries that can later be used to develop even more adapter than defined in

the scope of this project. Furthermore, all adapters share the general workflow of mapping, previewing, and transferring information so as to minimize the need for individual re-familiarization between the tools. The User interface is completely reused where possible, eliminating the need to reimplement on a case-by-case basis.

The adapters are able to make a sensible preview of the changes before the transfer is initiated, allow the user to make partial transfers of the full model mapping, store and display local exchange history, while the persistence of the mapping is relegated to the hub itself, enabling re-use between study participants.

Results

An example of every type of adapter is discussed below. DEHP Ecosim Pro adapter [10] (shown in Figure 2) is an example of the DST Adapter Bridge. Bidirectional exchange between experiment values and ECSS-E-TM-10-25 hub parameter values can be achieved with very little effort. The connection between the hub and the DST is established using the Open Platform Communications Unified Architecture (OPC-UA) protocol [11]. Ecosim Pro provides functionality to compile the experiments into OPC server-enabled standalone decks, which the adapter can interact with. Once connected to both the hub and the deck, the user can choose to load or create a new mapping, select the direction of transfer, preview impact on the target model based on a selected subset of the mapping and eventually initiate the data transfer itself.

The STEP AP-242 adapter is an example of a DST Standalone adapter and permits to "track" efficiently the geometric entities (from step files) attached to components of the engineering model during the concurrent design process. It gives access to a high-level representation view of an AP-242 step file. This representation exposes in a hierarchical way the different

labelled step assemblies and their relations. The mapping operation consists in referencing these assemblies within the engineering model thanks to a new "step geometry" parameter. More accurately, the STEP AP-242 adapter is unidirectional in the sense of internal data transfer, however, it allows for automatic preservation of a versioned source step file within the engineering model data.

The ASTOS adapter has been fully integrated into the ASTOS software. It allows to retrieve ASTOS input data for mission and safety analysis, trajectory optimization or vehicle design scenarios from the ECSS-E-TM-10-25 hub. It supports scalar and also vector data, latter for interpolation look-up tables as they are used for aerodynamic coefficients or performance profiles. On the other hand, the adapter allows for an export of the ASTOS results (trajectories, analysis output, optimized design parameters, etc.) into the hub. Both data exchange directions support automatic unit conversion. A wizard eases the set-up of the adapter configuration.

All adapters are utilizing either the C# or the Java versions of the COMET® SDK [5][6], which have been updated to the latest version of the COMET® meta-model.

Several noteworthy results of this activity can be mentioned aside from the adapters themselves. First, the development and release in the public domain of a C# common library (DEHP-Common [7]) that can be reused by developers in development of new DST adapters, giving access to software components and standard user interfaces to interact with the hub, perform mapping and many more auxiliary functions.

Secondly, as maintenance and evolution of ECSS-E-TM-10-25 plays a major role in this activity, the consortium implemented several improvements in the COMET® meta-model that can directly validate some of the changes proposed and still under evaluation of ECSS-E-TM-10-25 version 3.

Most note-worthy is the inclusion of the new `SampledFunctionParameterType`, which can be used for parameters whose value is a discretely sampled function where each sample consists of a given unique tuple of independent parameter values mapped to a tuple of dependent parameter values, opening the door for persistence of large variable-length sets of simulation data associated with timesteps.

Wider adaption of COMET® has further led to development of specific tools to aid in migration of data from OCDT WSPs to the COMET® Web Services, and reimplementations of legacy OCDT command line tools, such as the Syncer and the Stress Generator [8].

The DEHP project is planned to be concluded in the end of 2021, with almost all software developments being accessible as open-source with a LGPL-3.0 license located on the RHEA GitHub [9]. A demonstration of capabilities of some of the adapters is planned for the workshop.

References

[1] ECSS E-TM-10-25A, "Space engineering – Engineering design model data exchange (CDF)", ECSS Secretariat, made available 20 October 2010.

[2] COMET-IME
<https://www.rheagroup.com/cdp> ;

[3] Collaborative System Manager <http://blue-group.it/collaborative-system-manager/>

[4] Open Concurrent Design Tool – ESA Community Open Source Ready to Go!, de Koning, H.P; Gerené, S; Ferreira, I; Pickering, A.; Beyer, F; Vennekens, J; SECESA October 2014

[5] COMET-SDK-Community-Edition,
<https://github.com/RHEAGROUP/COMET-SDK-Community-Edition>

[6] COMET-SDK-Community-Edition
<https://github.com/RHEAGROUP/COMET-SDK-Community-Edition>

[7] DEHP-Common Library
<https://github.com/RHEAGROUP/DEH-Common>

[8] COMET Server Administration Tool
<https://github.com/RHEAGROUP/COMET-Server-Administration-Tool>

[9] RHEA Group GitHub
<https://github.com/RHEAGROUP>

[10] DEHP Ecosim Pro Adapter
<https://github.com/RHEAGROUP/DEH-ECOSIMPRO>

[11] OPC UA
<https://opcfoundation.org/developer-tools/specifications-unified-architecture>