Towards Views Extraction to Ease Concurrent Review of Systems Engineering Models

Jean-Marie Gauthier[1], Julien Baclet[1], Jérôme Fasquel[1], David Brandão[2]
[1]IRT Saint Exupéry, [2]ESA - ESTEC

ABSTRACT

Within the space industry, there is an increasing interest in the use of models as a mean to support systems specification and perform architecture tradeoff. Therefore, models and architectures are becoming information items that need to be collaboratively reviewed. Especially within the ESA's Concurrent Design Facility, which consists in gathering system specialists and system engineers in the same room to perform concurrent design and review of mission concepts, system requirements and architected solutions (mostly phase 0 and A). Each of the involved engineers in a concurrent design or review session is not necessary comfortable with models produced using SysML tools or Capella. In this paper, we propose an approach that permits Review Panel Members to extract and integrate various representations of Model-Based Systems Engineering data within a review procedure. We present in this paper the view extraction specification and its implementation called EasyMOD.

Workshop Objectives: O-2; O-4; O-5

1 CONTEXT AND PROBLEM STATEMENT

The work presented in this paper is in the frame of both an OSIP initiative, funded by ESA, and a collaborative project between IRT Saint Exupéry, Airbus Defense and Space, Airbus Commercial Aircraft, and Human Design Group. This paper focuses on ESA's needs, which is to ease models reviews within Agency and, specifically within ESA's Concurrent Design Facility (CDF) [1]. Reviews are required to transition from one phase to another one, e.g. Preliminary Requirements Review to go from phase A to phase B. There are also project specific reviews that can be conducted to meet particular project needs or explore solutions for identified issues.

The Figure 1 shows a simplified view the operational activities performed by actors involved in a review project. During the review preparation phase, the Industrial Project Team is in charge of initiating the required data-package to review. Then, the ESA Project Team (ESA P-T) assesses the completeness of the data-package. In the same time, the Review Secretary drafts the review procedure including review objectives. The review objectives are the reference for the reviewers to review the data-package. Based on the success rate of these review objectives, the status of the review project is determined.

There is an increasing interest in space industry for the use of model-based approaches as a mean to support system requirements definition, perform system analysis and architecture tradeoff. Usual tools such as Cameo Systems Modeler, Capella, or Enterprise Architect, are foreseen. Therefore, in the future, the overall review board will have to provide a Review Procedure and its associated data-package that contains such models to review. However, assessing the data-package completeness, preparing the review procedure and performing the review could be challenging, especially when the actors are not Model-Based Systems Engineering (MBSE) experts. The challenges we plan to adress are summarized as follows:

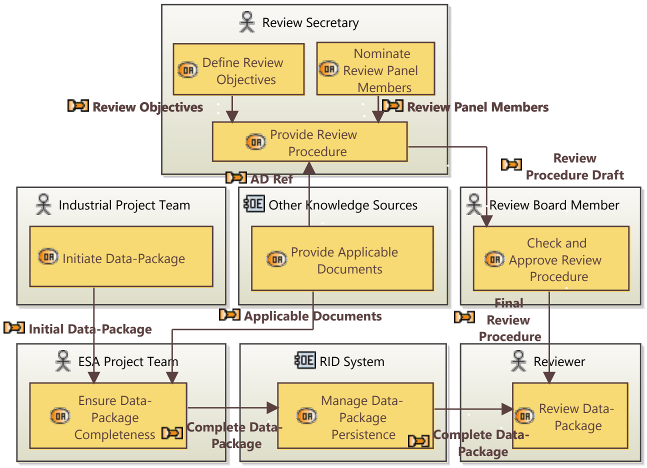


Figure 1: Simplified Operational Activities and Actors of a Review Project

- **C1 Modelling Languages Knowledge:** Modelling languages provide many concepts, which allows the detailed modelling of the system. However, those languages are not necessary known by the actors.
- **C2 Hidden Information:** Scaled system engineering models may contains lot of elements and links that are laid out within complex model structures. Navigating in a model, for which many diagrams have been created for different purposes, is difficult for other persons than the model author.
- **C3 Abstraction Level:** Abstract models are quite used to manage system complexity and to provide solution free architectures for early architecture design and trade-off. However, for non MBSE experts, abstract models are hard to review since they are not necessary represented with concrete symbols.

To adress those challenges, we propose to specify and develop a software solution that would allow a non MBSE expert to: ① extract model views abstracted from unnecessary concepts for a given review, ② link the extracted model views with Review Objectives and integrate them within the Review Procedure, ③ concurrently create comments on the Review Procedure or on the model views, and ④ navigate between concepts and views without loosing the context of the review operational activities. The proposed approach is based on previous results about information sharing in Extended Enterprise [2] and new human machine interactions for MBSE [3].

The rest of this paper is structured as follows: the proposed view extraction specification for Review Procedure and data-package construction is detailled in Section 2. Section 3 presents the implementation and early experiment results. We review related work in Section 4 and conclude and outline further research and development work in Section 5.

2 VIEWS EXTRACTION SPECIFICATION

The proposed approach contributes to the following activities of a Review Project: the definition of the Review Objectives and the Review Procedure, the preparation of the data-package and needed views, and the Review activity by the Reviewers. The management of Review Item Discrepancies (RIDs), tasks and actions that are created to address issues, is out of scope of the paper.

2.1 System Functional Architecture

The views extraction feature is implemented within the EasyMOD Proof-of-Concept software. This feature permits to extract (MB)SE Data and to represent them as diagrams, tables, or charts. The resulting views can be integrated within a Review Procedure. To achieve those capabilities, the EasyMOD software is upgraded with two main functions: a function that *Manage Review Procedure Edition*, and a function that *Manage (MB)SE Data View Construction*. The interfaces and functional exchanges of these two functions are specified within a Capella model. The Figure 2 shows an excerpt that focuses on high-level functions to extract and integrate views.

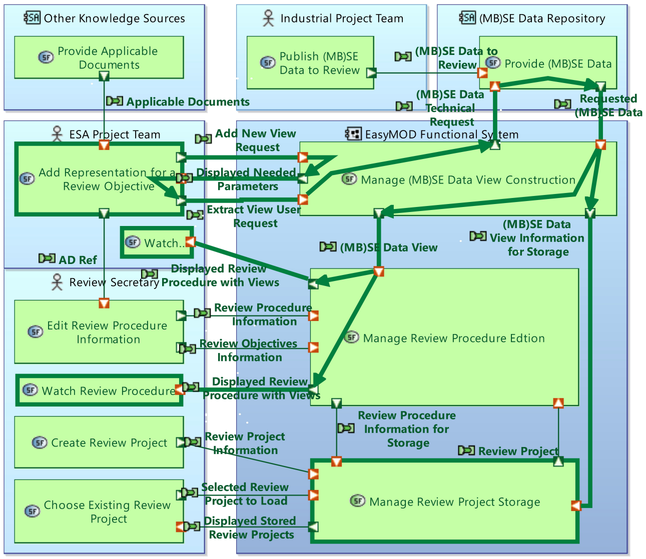

Other Knowledge Sources
Provide Applicable Documents
Industrial Project Team
Publish (MB)SE Data to Review
(MB)SE Data Repository
(MB)SE Data to Review
Provide (MB)SE Data
Applicable Documents
(MB)SE Data Technical Request
Requested (MB)SE Data
ESA Project Team
Add New View Request
EasyMOD Functional System
Add Representation for a Review Objective
Displayed Needed Parameters
Manage (MB)SE Data View Construction
Extract View User Request
Watch...
(MB)SE Data View
(MB)SE Data View Information for Storage
AD Ref
Displayed Review Procedure with Views
Review Secretary
Edit Review Procedure Information
Review Procedure Information
Review Objectives Information
Manage Review Procedure Edtion
Watch Review Procedure
Displayed Review Procedure with Views
Review Procedure Information for Storage
Create Review Project
Review Project Information
Review Project
Choose Existing Review Project
Selected Review Project to Load
Displayed Stored Review Projects
Manage Review Project Storage

Figure 2: Views Extraction Functions and Exchanges

The green functional chain (in bold green) specifies the functional path that realizes the view extraction feature. First, the ESA P-T requests to add a new view to integrate within the review procedure document. Then, the software displays required parameters to be filled by the ESA-PT before launching the view extraction request.

Then, the software retrieves the requested MBSE data that have been published by the Industrial Project Team prior the review preparation phase. A technical request is created from the parameters given by the users. After the data have been retrieved, the *Manage (MB)SE Data View Construction* function is in charge of creating the data representation and to render it for the ESA-PT and the Review Secretary. The Review Secretary and the ESA P-T hence perform a collaborative work to complete the review procedure by providing the review objectives and their associated extracted views to be reviewed. Finally, the *Manage Review Project Storage* function persists all the information related to a review procedure with its associated view. In a complete deployment, this function could also be realized by the RID System already deployed at ESA. Also, note that the Industrial Project Team would also be able to add extracted views in the review procedure.

2.2 Logical Components and Architecture

The Figure 3 shows an excerpt of the EasyMOD Logical Architecture (functional exchanges are hidden for lisibility purpose). The EasyMOD software is divided into two main components. First, the *EasyMOD Front End* component acquires user inputs and of displays the GUI to manage Review Projects, to construct the Review Procedure and to extract and render views inside the Review Procedure. Second, the *EasyMOD Back End* is responsible for storage (*Review Project DB*), for requesting MBSE data from repositories (*(MB)SE Data Connector*), and for transforming raw MBSE data into the right format for the front-end's rendering function (*Back End Service Provider*). The *Back End Proxy* makes the front-end independent from the back-end. Indeed, the view extraction and rendering features can also be used to create any kind of dashboarding system to perform analysis in an extended-enterprise [2].

3 IMPLEMENTATION AND DISCUSSION

In this section we describe the technological choices and the implementation of the EasyMOD front-end and back-end. Then, we discuss the differences between EasyMOD and BabyMOD, and we discuss the view extraction feature.

3.1 Front-End Implementation

As for BabyMOD, the front-end of EasyMOD is being implemented using Web technologies (Angular, TypeScript, JavaScript, HTML5, CSS), standardized Web APIs, and open source third-party libraries. The front-end interface prototype is illustrated on the Figure 4. From left to right, the HMI mainly consists of: (1) a table of content that lists the review objectives, (2) the review procedure editor containing a text editor and the extracted views, (3) and the view extraction feature.

Concerning the view extraction feature, the end-user first selects the type of model from which a view is needed. Then, depending on the review objective, the end-user selects the needed viewpoint. Different representations for the same viewpoint are available. For instance, the power consumption budget by mode can be represented either with a bar chart or within a table.

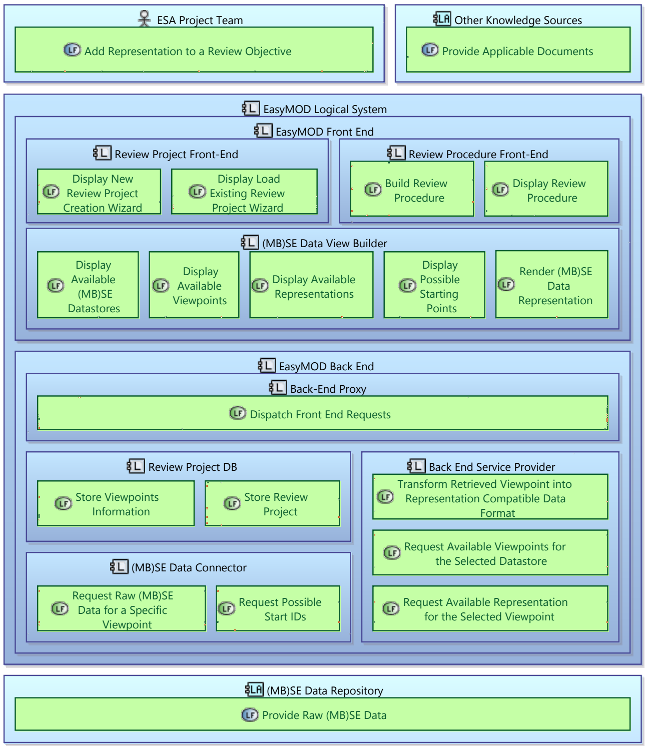

ESA Project Team
Add Representation to a Review Objective
Other Knowledge Sources
Provide Applicable Documents
EasyMOD Logical System
EasyMOD Front End
Review Project Front-End
Display New Review Project Creation Wizard
Display Load Existing Review Project Wizard
Review Procedure Front-End
Build Review Procedure
Display Review Procedure
(MB)SE Data View Builder
Display Available (MB)SE Datastores
Display Available Viewpoints
Display Available Representations
Display Possible Starting Points
Render (MB)SE Data Representation
EasyMOD Back End
Back-End Proxy
Dispatch Front End Requests
Review Project DB
Store Viewpoints Information
Store Review Project
Back End Service Provider
Transform Retrieved Viewpoint into Representation Compatible Data Format
Request Available Viewpoints for the Selected Datastore
Request Available Representation for the Selected Viewpoint
(MB)SE Data Connector
Request Raw (MB)SE Data for a Specific Viewpoint
Request Possible Start IDs
(MB)SE Data Repository
Provide Raw (MB)SE Data

Figure 3: EasyMOD Logical Components

Finally, the extracted views can be linked to review objectives. In that way, viewpoints could be suggested when creating another review procedure for another projects. Also a review procedure template could be stored and reused for further review projects.

3.2 Back-End Implementation

The back-end of EasyMOD is implemented with Spring-Boot. This framework permits the configuration and the connection with the data model, implemented as an H2 Java database. An excerpt of the data model is illustrated by the Figure 5. The database persists all the data related to review procedure, the extracted views, the comments, and the user role and credentials.

The communications between the front-end and the back-end are done using http requests. Then the back-end is able to dispatch the requests to the needed services: saving, retrieving MBSE data, user connection, comments persistence, etc.

3.3 Discussion

The EasyMOD software integrates some of the features inherited from BabyMOD. BabyMOD [3] is a web-based multi-modal model editor that targeted three main objectives: visualizing models, editing models, and reviewing them through model annotations. It integrates real-time sketch recognition assistant that allows multiple users to edit or annotate models in a free-form modelling way. In addition to sketch recognition, it integrates text and voice recoginition.

EasyMOD distinguishes from BabyMOD on various points. First, EasyMOD permits to edit review objectives and to integrate them within a review procedure document in a web application. Then, it goes one level further in maturity for industrial usages. Indeed, whereas BabyMOD can extract data from Capella models only, the end-user can now import several types of model: Capella models, COMET models, SysML models (stored in TeamWork Cloud). Then, the commenting capabilities are fully implemented: a user can comment a view, and the comments are persisted in database. Finally, whereas BabyMOD supports one type of graphical representation, EasyMO permits to aggregate in the same page, several views rendered with diagram based or chart based representations. In that sense, all the viewpoints that were developped for the OSIP TeePee4Space [2] can be reused within EasyMOD for review purposes. The final EasyMOD software targets a Technology Readiness Level 4.

The views extraction and their layout within a web page is not only usefull for reviews. Indeed, this feature establishes the basis for leveraging MBSE data visualization. It would be possible to use this feature so as to provide a dashboard system (like Kibana [4]), or a decision cockpit. In addition, we would like to investigate the exploration of MBSE data by dynamically creating views from other views on user request.

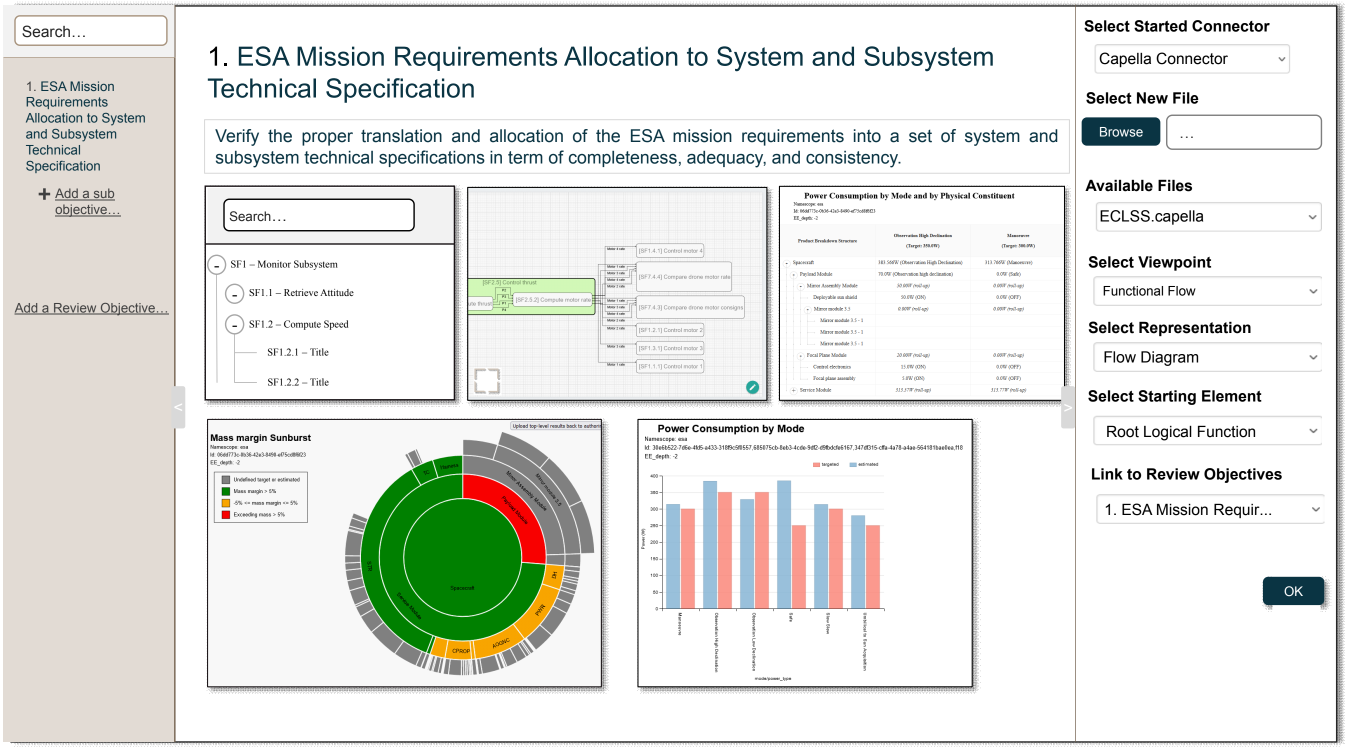
Finally, the view extraction feature coudl be used as an opportunity to investigate research questions about the MBSE data representations itself. For instances, what are the best representations for the logical data flow viewpoint? What kind of dynamic filter could be applied for a specific concern? How to visualize several layers at a time without beeing lost in the complexity of the diagrams, or without losing the context of the current task?

4 RELATED WORK

From the state-of-the-art, Bucchiarone et al. [5] and Chami et al. [6], have identified challenges related to MBSE and Model-Driven Engineering adoption. In particular, they relate the challenges about model edition through views, layout of model views, or linking views and navigating between them amongst others. Those challenges are true for advanced MBSE practicionners (47,62% of the participant have more than 5 years of MBSE practice [6]).

The JPL is working since several years in the development of a plateform called Open Model Based Engineering Environment (OpenMBEE) [7]. An interesting feature of OpenMBEE is the View Editor [8]. This feature proposes a web-based and document oriented editor that allows agile virtual reviews and real-time collaboration. However, to the best of our knowledge the View Editor can only extract diagrams or views that already presents in the models. Our approach is different since we propose to reconstruct views on demand with the representation that better fits the user's needs for a specific review objective.

Open CAESAR [9] (Computer Aided Engineering for systems ARchitecture) is also a NASA JPL initiative. It aims at providing an Integrated Model Centric Engineering approach (IMCE) that supports continuous integration, change management and agile systems development process. The IMCE approach relies on ontologies, viewpoints creation and requests to federate systems engineering information, verify consistency and perform analyzes.



1. ESA Mission Requirements Allocation to System and Subsystem Technical Specification

Verify the proper translation and allocation of the ESA mission requirements into a set of system and subsystem technical specifications in term of completeness, adequacy, and consistency.

Select Started Connector: Capella Connector

Select New File: …

Available Files: ECLSS.capella

Select Viewpoint: Functional Flow

Select Representation: Flow Diagram

Select Starting Element: Root Logical Function

Link to Review Objectives: 1. ESA Mission Requir…

Figure 4: Graphical User Interface Mockup

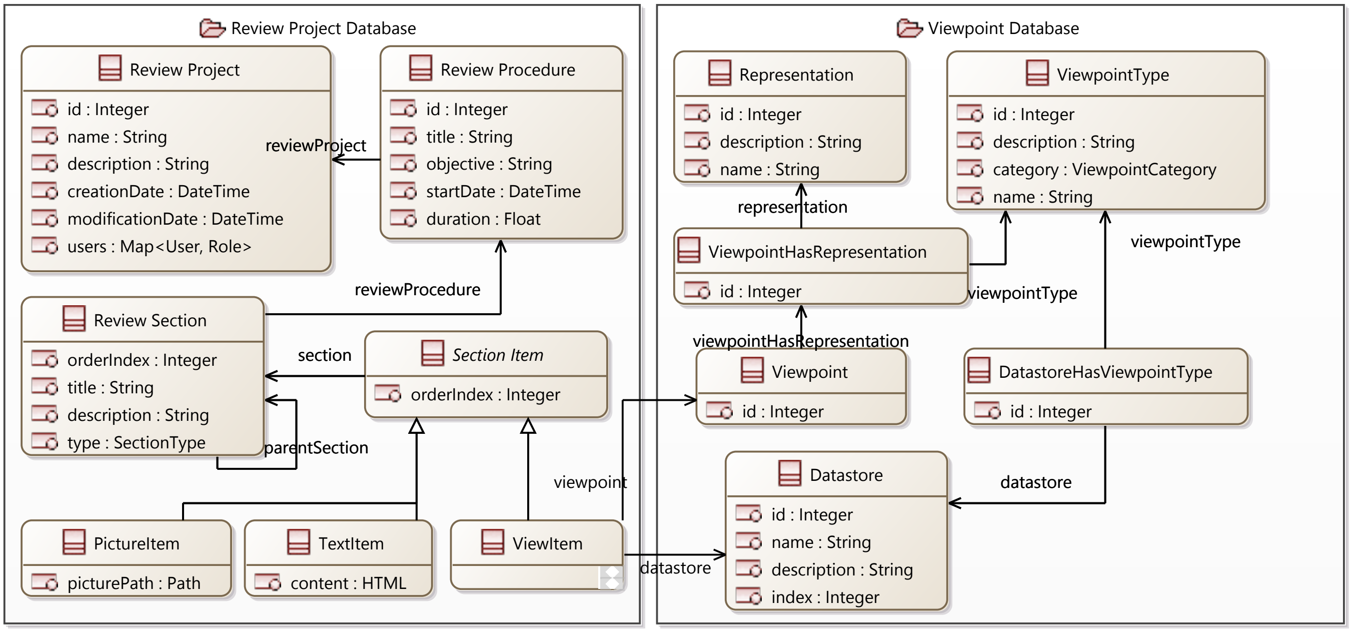


Figure 5: Excerpt of the back-end database; focus on review procedures and viewpoints

The Cameo Collaborator tool [10] enables the review of SysML models. It permits to export models as a review document with integrated views. Textual and graphical comments are possible directly on the diagrams. In addition, it provides features to edit the model directly in the review document. However, all the views that are integrated within the review document are in SysML formats.

This may lead to difficulties when the reviewers are not comfortable with the SysML language and symbology.

5 CONCLUSION

In this paper we presented EasyMOD, an on-going project at the IRT saint Exupéry. We first recalled the context and the challenges to be addressed in order to achieve views extraction for collaborative reviews of systems engineering models. Then, we described the specification of the proposed views extraction feature within the EasyMOD software. Finally, we described the ongoing implementation of EasyMOD and its HMI. We explained how to extract views from models, and we showed the capacity of linking the views to the review procedure and review objectives. As the software is still under development, definitive results are not presented in this abstract but more concrete results are expected to be available at the time of the conference.

REFERENCES

[1] Massimo Bandecchi et al. “The ESA/ESTEC concurrent design facility”. In: *2nd European Systems Engineering Conference (EUSEC)*. 2000.

[2] Julien Baclet et al. “TeePee4Space : a Practical Application of Information Sharing in Extended Entreprise to the Space Industry”. In: *Model Based Space Systems and Software Engineering - MBSE2021*. Sept. 2021.

[3] Nicolas Hili and Patrick Farail. “BabyMOD, a Collaborative Model Editor for Mastering Model Complexity in MBSE”. In: *Model Based Space Systems and Software Engineering - MBSE2021*. Sept. 2020.

[4] Yuvraj Gupta. *Kibana essentials*. Packt Publishing Ltd, 2015.

[5] Antonio Bucchiarone et al. “Grand challenges in model-driven engineering: an analysis of the state of the research”. In: *Software and Systems Modeling* 19 (Jan. 2020).

[6] Mohammad Chami and Jean-Michel Bruel. “A Survey on MBSE Adoption Challenges”. In: *The Systems Engineering Conference of the Europe, Middle-East and Africa (EMEA) Sector of INCOSE (EMEASEC 2018)*. Nov. 2018.

[7] Benjamin Kruse and Mark Blackburn. “Collaborating with OpenMBEE as an authoritative source of truth environment”. In: *Procedia Computer Science* 153 (2019).

[8] Thomas Boyer Chammard et al. “Assisted Authoring of Model-Based Systems Engineering Documents”. In: *23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. 2020.

[9] Maged Elaasar et al. “The Case for Integrated Model Centric Engineering”. In: *10th Model-Based Enterprise Summit (MBE 2019)*. 2019.

[10] NoMagic, Inc. *Cameo Collaborator*. [Accessed 2022 06 07]. URL: https://www.3ds.com/products-services/catia/products/no-magic/cameo-collaborator-for-teamwork-cloud.