ENVISION MBSE STUDY

Return of experience report
Deliverable id : TN2

Written by	***Responsibility*** + handwritten signature if no electronic workflow tool
G. Garcia	GARCIA Gerald Signature numérique de GARCIA Gerald Date : 2021.10.15 17:39:28 +02'00'
Envision TAS Team	
Verified by	
Approved by	

Approval evidence is kept within the document management system

Change Records

ISSUE	DATE	§ CHANGE RECORDS	AUTHOR
1.0	29/09/2021	Initial release	

Table of contents

1. Introduction

1.1. Scope and purpose

This document is the return of experience report for the Envision MBSE study.

This study was a direct negociation between Thales Alenia Space France and ESA. This budget objective was to develop the usage of a MBSE approach into operational Envision project, not (as usual) as a side project demonstrating what could have been done but integrated into the project activity and be part of the deliverable of the study.

The top-level goals for this deployment as stated in the statement of work are to:

- Enforce the use of the physical engineering model that TAS-F already has by sub-system architects in order to have the model as a central source of truth
- Produce an interactive representation of the model allowing its consultation without having to install dedicated modelling tools
- Produce trend graphs on system budgets
- Produce document generation capability to export graphs and tables to the system and subsystem budget reports
- Provide means to automatically produce a mission performance report from a model, including requirements traceability&allocation and verification artefacts (in phase A2 these will be limited to analysis and simulation results)

The objective of this document is to present lesson lean from this study, it collect impacts on the prime process, links with Osmose and finally return of experience on the proposed solution.

1.2. Applicable documents

Internal code / DRL	Reference	Issue	Title	Location of record
AD1	ESA RFP/3-16592/20/NL/AS		RFP incl. SoW	

1.3. Reference documents

Internal code / DRL	Reference	Issue	Title	Location of record

1.4. Definitions and Acronyms

Acronym	Meaning
PM	Project meeting

2. High level view of the results

This chapter permit to have a quick overview of the realised work and achievement of this study with regard to its initial top level goals :

- Enforce the use of the physical engineering model that TAS-F already has by sub-system architects in order to have the model as a central source of truth
- Produce an interactive representation of the model allowing its consultation without having to install dedicated modelling tools
- Produce trend graphs on system budgets
- Produce document generation capability to export graphs and tables to the system and subsystem budget reports
- Provide means to automatically produce a mission performance report from a model, including requirements traceability&allocation and verification artefacts (in phase A2 these will be limited to analysis and simulation results)

<u>Enforce the use of the physical engineering model that TAS-F already has by sub-system architects in order to have the model as a central source of truth</u>

This objective consist in having a single source of truth for system "physical design". Physical design mean decomposition of the system into "real" parts (from avionics modules to structural panels), these part having physical characteristics (mass, power consumption, shape, …). This include also technical budgets related to these characteristics (mass, power, dissipation, propulsion, …) budgets.

Thales Alenia Space use IDM-CIC tool for this objective for many years systematically during 0/A/B1 phases of the observation and science projects. This is the case for Envision.

IDM-CIC is developed by the CNES and distributed freely to the space community. IDM-CIC support natively the collaboration of a whole team on the same model.

This tool permit to model (according to a dedicated data model) the whole system decomposition and characteristics (including 3D properties) and has been extensively used for Envision as a single source of truth for :

- System decomposition and parts
- Parts characteristics, general one like : mass, power consumption, … and type specific one like the ISP of a thruster.
- Definition of system, elements and equipments modes and the relation between them
- The physical accommodation of each of the elements of the system to form the complete system
- The different mechanical articulations that permit to represent several configuration for the system (launch, deployed, …)
- Mission definitions in term of sequence of manoeuvers according to scenarios (for propulsion budget)
- ….

Sub-system engineers are responsible for their own data and provide them directly in the model. The system engineers combine these data into a complete system model.

This model (and the associated tool) is the source for communication with also external partners (with import/export capabilities).

<u>Produce an interactive representation of the model allowing its consultation without having to install dedicated modelling tools</u>

This is a major achievement of this study, even if IDM-CIC tools is easy to use and easy to distribute, it is not the choice of ESA and of some other partners. Due to this, exchanging information with them was then reduced to come back to documents (may be Powerpoint, Word or Excel documents) containing (hopefully) the same information but in a non-structured and shared format.

This information was difficult to produce but also to review.

The proposed solution in this study consist in providing a static web site where the reviewer can use advanced interactions (interactive plots, fold/unfold structures, …). Providing a static web site remove all the complexity of installing a tool and can be largely distributed easily.

The domains covered during this study are : mass budget, power budget, propulsion budget, specification, data-packages (documents) and simulations.

Produce trend graphs on system budgets

Obviously early phases of space project are very dynamic one in term of system baseline, everything is changing frequently and it is difficult to follow all these changes in particular if you have discrete (one per month per example) interaction with the team.

Moreover the evolution of a mass budget for example is as interesting than the absolute value at the bottom line. Some kind of asymptote on the evolution is generally a good sign of a convergence on a stable design.

This study proposed a solution to display this trend and the value history for every single engineering data. Reviewers can get immediately the evolution of value across the different PM, but also get the difference between the current value and any value in the past (for example if they skip a PM, they can review the difference between now and 2 PM before).

This required to work on the IDM-CIC model handling as the current tool does not support configuration management and versioning of models. We deploy an external configuration management system for the IDM-CIC as well as an extended capability to perform configuration management at data level in the solution.

Produce document generation capability to export graphs and tables to the system and subsystem budget reports

When document are needed (for example for design justifications), some view on the model has to be included in this documentation and the risk of divergence between the model content and the actual content of the documents is high.

During this study we study several ways of including automatically these views in the documentation but none are really fully satisfactory. We develop several “reports” from IDM-CIC model that has to be “manually” inserted in the document for the moment. Finding an efficient solution for this problem is still an open point.

By the way we develop the capability also to link from the document content toward the digital content and the other way around which is a very powerful integration between document and digital content. It permit to get “up-to-date” information when clicking on the content of a document.

Provide means to automatically produce a mission performance report from a model, including requirements traceability&allocation and verification artefacts (in phase A2 these will be limited to analysis and simulation results)

This study does not produce automatically the mission performance report for the reason invoked just before, but permit to link the mission performance report content with the requirements and the simulation underlying this performance report.

But to cover this goal, three main solutions has been implemented in this study :

- The capability to navigate in the ESA input specification and spot performances requirement
- The capability to navigate to the corresponding part of the report that justify this requirement
- The capability to exchange simulation scenarios in order to be able to review the scenarios used for sizing the system and agree on them and on underlying assumptions.

3. Impact on prime internal process

3.1. Availability of data

A lot of time is spent by engineering teams to find information and humans are quite good to do this. The problem is that compiling all the information into a single website for review is a challenge. The data are spread into many databases (DOORS, Capella, IDM-CIC, …), shared directories (documents, Excel, …).

Trying to setup a “continuous integration” of engineering data into the review web site imply that data are always at the same place and that they are all “up-to-date”. This implies that the engineering team is sufficiently “disciplined” to follow strictly this rule.

During the study we faced also many issues related to data quality. Quality here does not means that the data where not good from an engineering point of view (or any problem with the project itself) but most of the time it is related to formalism.

The level of formalism required to transmit information to a human is not the same than the level needed for an automated process. For example we found cells in the VCD referencing a document reference for results, in some cells we found “please refer to model specific tests”. Obviously it is perfectly understandable by an human and is not a project level problem… but for an automated ingestion mechanism it is a quality issue…

Even if these problems of quality in the data may be solved by “soft” constraints (trying to guess what the user means, ignoring what is not understood, …) the level of formalism needed for automated processing is quite high. That’s why it is far more easy to extract information from a Capella model than from an Excel sheet.

One possible solution is to rely on machine learning to better understand poorly structured data. Another solution would be to implement comprehensive sets of rules to be checked before data ingestion, allowing the user to correct its inputs according to the rules. Both approaches are worth investigating only in the case that such a process is deployed at a larger scale than a single project.

3.2. Versioning of data

Presenting a diff between two versions requires to be capable to perform this difference at semantic level but also to collect the different versions of the data. Presenting a baseline or a difference of two baseline during a review assumes that we are able to identify in each data source the versions composing a given baseline.

Some areas of engineering are versioned with tool support (for example DOORS), or versioned through the use of external tools (for example a git repository of a Capella model or an IDM-CIC model). Some others are either versioned by ad-hoc mechanisms (naming like file_v1, file_v2) or even not versioned at all.

When this versioning exists, it is distributed in each tool and the concept of baseline most of the time does not exist (the Orchestra environment support this concept of "distributed baseline" but on a limited set of tools integrated into the environment).

On the document side, there are baseline that are ensured by the PDM tool (versioning of data package delivered to the customer at each review).

The practiced versioning principles are not well established in the system engineering community and when versioning exists, it is not really formal and not connected between the different configuration management repositories.

System engineering data would benefit from a formal configuration management, just like the other disciplines. This requires putting in place a clean configuration management framework and train the engineers to use it.

3.3. Change justification / changelog management

The versioning is a key capability for being able to trace the history of the different data. Nevertheless there is a difference between versioning and change management. The reviewer are only interested by the difference between two reviews (that may be not consecutives) that between each of the intermediate versions that lead to the version submitted to review. One example is a reviewer participating to each major reviews (PDR, CDR, …) does not want to see the modifications that occurred at each progress meeting. On the opposite the customer project team is interested to view the differences between each PM and the TAS internal team the differences at even finer grain.

This difference in need lead to the capability to display diff between today (the review) and any point in time.

Once this capability is acquired, this is not sufficient as the reviewer will be interested on this differences, but also on the reason of this differences (taking into account a input change, a design change, …) that we call the changelog capacity.

3.3.1. Difficulties

The changelog capacity is very tricky, in particular due to the fact that all users do not want to see the same granularity. If we obtain a very fine grained change log useful for day to day team activities, it will not be adapted to a reviewer participating only in key reviews. The solution of containing the different "micro" changelog will not be satisfactory because there will be tenths of changes from where it is difficult to extract a precise picture of what has changed in the project.

Getting the changelog from the engineering team is not also a trivial task. Indeed, the change management "culture" is not as strict as in some software communities (where all changes on the code base are documented, tested, reviewed and approved). This is due to the complexity and the heterogeneity of the data manipulated by the team, tooling capabilities but also habits. This prevent us for collecting changelog easily. As with data versioning, this change management has to be trained and ideally supported by tooling.

3.3.2. Choices made

We studied several solutions to collect and produce automatically changelog that are "useful" for the reviewer but many of them failed to produce correct results. We finally choose to rely on the team to provide meaningful change logs.

The established process is the following :

- A first "editable" version of the review site is provided to the team
- On this version, the team is capable to annotate the relevant differences between the reference review and this one directly on the web site.
- This changelog information is stored in the knowledge graph (as new links with annotation on the graph).
- This information is used to produce the final version of the review site where the changelog is no more editable

This allows to add relevant changelog on each diff identification. During the review, if a diff is not explained, the reviewer can ask for a changelog in one click (it will create automatically a RID for that).

This implementation is a first step of a long journey on this topic. The manual annotation is time consuming and due to the nature of the data, it may lead to the duplication of the comments on several data (for example on the sub-system and on the system it-self or on the same mass for different system configurations). This has to be enhanced, several ideas have to be explored, the most promising one is to use machine learning algorithms to help the team to fill these changelogs (for example by learning implicit relation in the data).

3.4. Reliability of the infrastructure

The prototype solution in this study rely on many different elements forming a chain from raw engineering data sourced in engineering tools to the final review web site to be delivered to the customer for reviews.

This chain is quite complex by it-self but also by the security measures in place on the Thales Alenia Space network (proxies, lan connections, …).

We have used as much as possible devops technics (continuous integration, continuous deployment, …) to ensure that the is a continuous path between the users and the final web site but also the developers and the deployed solution.

As demonstrated several times during the study, this infrastructure required a high level of availability of quite a number of elements of the chain. This is quite new as before except in some of the engineering domains (the CAD for example), many activities could be performed even in a (very) degraded environment (using user laptops, exchange by usb keys, mail, …) in case of major problem. This is no more the case, the user will not be able to deliver its review data-package if there is a failure in one of the chain elements.

This implies measures on the prime side on how to secure and to provide a high level of availability of this kind of solution. There is no technical blocking point (after all these are the technologies used by big players of the internet with an extremely high level of availability) but more cultural and architectural changes in development teams, the IT teams and finally the final engineering users.

To make a parallel, ESA talks about engineering factory, the parallel with the tooling is relevant and the tool chain setup to produce digital version of the engineering data is very similar to a production line of a factory. When one of the elements has a problem, it has repercussions on the whole chain if not managed properly.

4. Use of Osmose as an input ontology

The MB4SE community, under the ESA umbrella, has acknowledged the importance of digital exchange of data in the engineering process (interoperability even at extended entreprise scale) and decided to invest into the development of a space engineering ontology called Osmose focusing on the semantic of the exchanges that occur between project partners. This initiative is supported by different studies focusing on the definition of this exchanges (SaSYF led by GMV) and Osmose development studies. The goal of Osmose is to agree on an ontology expressed in ORM.

Obviously there is a link between this initiative and the structure of the knowledge graph. Even if the modeling language seleted for Osmose (ORM) and the one used on the prototype (ecore) are not the same, there is a strong mapping between the two.

In order to simplify interoperability (which is the main goal of Osmose) in particular from eco-system to eco-system (where the data exchanged are not authored in a single tool) it will be very interesting to go through the knowledge graph and the closest it will be from Osmose ontology the easier it will be.

Unfortunately the domains covered for the Envision needs are not in line with current domain of discourse currently explored by Osmose, in particular the physical description and the associated system budgets. This study constitutes a good input for these domains.

Finally the dynamic (time dependent, used for exchanging scenarios) attributes and links are not covered by Osmose up to now.

5. Return on experience on proposed solution

5.1. Providing a static site

In traditional web applications hybrid solutions part of the application is running on the browser (mostly the user interface part) and part is running on a remote infrastructure (on cloud or on premises). This remote architecture is composed of several servers (applications servers, databases, caches, …) that perform all the heavy duty of the application to offload the work of the browser.

The bet of this study was to produce a static side where all the application is executed on the browser without any dependency to a server (except for collaborative features) and to produce a user experience as close as possible to state of the art of “traditional” web applications.

The proposed static site permit to achieve very good navigation performance even on large dataset (and perhaps better than what we could obtain from a hybrid application). We achieve this performance at the cost of increase of the delivery size (there is here a trade-of). Current deliveries for Envision have a size around 100Mb but larger sizes are expected.

Even complex and data intensive displays (for example displaying a timeline of a scenario or a plot with a lot of points or even 3D display) are achievable thanks to the increase of computing power on the browser side but also to the improvement of browser execution engines. Due mainly to network latencies and intermittent connectivity this trend to have more and more complex applications on the browser will continue in the coming years (in particular with the edge computing move).

Nevertheless obviously this has some limitation once again linked to the computing power and the storage capability of the browser. In particular smart searching (more advanced than what is currently delivered by the solution) has to be implemented at server side, but also advance data manipulation, filtering and visualization (millions of data points of a simulation, …) will be difficult to achieve on the browser side.

The optimal solution is certainly to move some heavy processing to dedicated servers (eventually deployed on the reviewer premises) and have a graceful degradation when these services are not available. For example advanced exploration of simulation results could be possible on “standardized” IT infrastructure on the reviewer side.

5.2. Relying on a knowledge graph

From a first look, this was not really strictly needed given the objective of the study to introduce this intermediate data hub to produce digitalized content. Indeed as we already have Capella html exporter that permit to provide a web version of a Capella model, we could have a IDM-CIC web viewer that produce an HTML version of the content of the IDM-CIC model.

During this study we discovered that in fact it is a key enabler for being able to provide a seamless navigation into the engineering data (it is more simple, from a technical point of view but also from a user point of view, than trying to glue together different exports from different tools with different UX). It is also a key enabler for a unified handling of the history of data (i.e. proposing versions at global baseline that cross the configuration management silo of each tools). Finally it permit bidirectional navigation in the data even if the link is unidirectional in the authoring tool which would be very difficult to handle if we directly export from tools.

5.3. “Duplication” of authoring tools capabilities

For sure the proposed solution permits to display the engineering data, most of them are already available in the different authoring tools. For example the table for power consumption budget proposed for Envision is almost identical to the one one can find in the IDM-CIC tool.

On one side it oblige to duplicate the work in the review solution and in the authoring tool (and duplicating the risk of bugs and efforts to perform the maintenance), on the other side it permit to give a homogeneous user experience whatever the authoring tool is.

For tables and simple plots, it is quite easy to develop these views and for the Envision use-case it was sufficient, but for more complex visualization of data (A Capella diagram, a 3D model, …) it may not be the case and even in some cases it may not be possible (for example a UML or Capella diagram where the layout has been done by hand). Even if we start to have automated diagramming solutions with automated layout, the results may be quite different than an hand-crafted diagram.

It may be the case that depending on the phase of the project or type of data exchanged, the proposed solution reaches its limits and both the supplier and the customer realize that they would rather share a single model than a static website that only allows reporting but not collaborative work.