SESP 2019
Digitalization Lessons Learnt for Spacecraft Engineering

Claude Cazenave [(1)], Luc Fancelli [(2)]

[(1)] Airbus Defence and Space
31 rue des Cosmonautes
Z.I. du Palays
31402 Toulouse Cedex 4
France
Email: claude.cazenave@airbus.com

[(2)] Airbus Defence and Space
31 rue des Cosmonautes
Z.I. du Palays
31402 Toulouse Cedex 4
France
Email: luc.fancelli@airbus.com

DIGITALIZATION AT AIRBUS DEFENCE AND SPACE

Digitalization at Airbus is a strategic goal to decrease cost and improve quality over all Satellite life cycle. At Airbus DS, for telecom spacecraft, as presented previously a digitalization project is in place with clear objectives and expected savings. This project is named Odyssey 4.0. It covers all disciplinary aspects (functional, electrical, mechanical, etc…) through the complete system life cycle from offering phase to in service operations including system design, detailed design, verification and manufacturing. This paper partially describes the solution put in place and operational on the new telecom spacecraft eNEO family. The focus of this paper is on the current status of the engineering activities and the relevant lessons learnt. Finally some of the steps we already foresee are presented.

SOLUTION IN PLACE

As shown in the next figure, Odyssey 4.0 is an eco-system of data oriented Interconnected Collaborative Platforms. To provide the maximum of coherency, these Platforms are overlapping to be able to provide data in the format expected by following platforms.

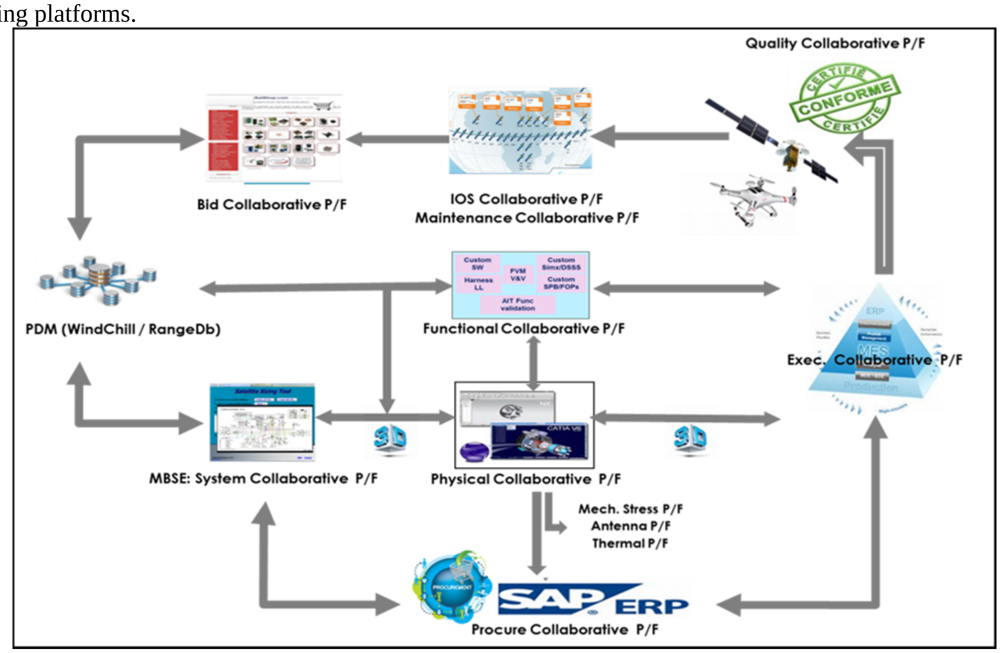

Fig. 1. Scope of Odyssey 4.

It starts with a Bid collaborative platform where engineers define a preliminary solution to answer to customer needs. This preliminary solution is used for instance, but not only, to estimate the cost of the satellite, to propose targeted Spacecraft pre-accommodations and to verify that budget are in accordance with customer expectations. When, hopefully, the contract is won, this preliminary solution is finalized using the same system collaborative platform as the one used during the offering phase. From there, thanks to the missionisation parameters provided by the System Platforms, the functional and physical/mechanical engineering activities are started in parallel:

- On the Physical Platform, detailed design activities are started allowing to start Manufacturing Engineering activities that will be executed on the Manufacturing Execution System
- On the Functional Platform, On board software customisation, functional simulator development activities are started and the Validation campaign is prepared.

When spacecraft is in orbit, support activities performed with their dedicated In Orbit Support (IOS) platform are reusing data coming from the previous phases. To close the end to end loop, IOS platform will also generate in orbit data for the next Bid phases benefiting of a better knowledge on the equipment to be used and providing to the customer a well known in orbit heritage. Finally in parallel to all those activities a quality platform allows to monitor and control the quality at any stage. The current achievements, focusing on engineering activities are described here after.

Common language for engineering activities

One approach for an efficient data exchange between engineering activities is a common definition of the data to be exchanged. At a first glance it may look difficult to achieve but in practice it is manageable (see lessons learnt below). This common definition or vocabulary with all the required semantic is implemented in RangeDB [2], Airbus DS engineering database, using a so called Conceptual Data Model (CDM). This engineering database provides all the means to store under configuration control, to edit, to query, to populate, to import/export and to verify the consistency of the data. RangeDB was initially foreseen to cover functional activities like spacecraft monitoring and control. Its scope has been now extended to cover electrical, mechanical, radio frequency communication and harness activities. The result is that we now have a common language for engineering activities and a result no more translation of the data between many engineering tools used is required. When people talk about some data values they know that this is definitely the same thing they are talking about. In addition to this shared understanding the availability of the information shall be guaranteed to all the stakeholders. This is what is achieved thanks to the powerful configuration features provided by the engineering database. It allows both working in parallel by different teams for concurrent engineering and definition of well mastered baselines of the system under development

Integration with other collaborative platforms

Legacy Product Data Management (PDM) platform has been integrated with RangeDB. As a result, the consistency between on one side the documents stored in the PDM to define, for instance, the equipment used to build a system, and on the other side the engineering data used for the design of the system and stored in RangeDB is ensured. Moreover, since our design tools directly generate their data into RangeDB the configuration of the system is completely defined in the engineering database and can be used by other platforms, including the PDM, to cross check this configuration. This is in particular the case for Satellite Sizing Suite, the system Collaborative Platform in the Odyssey 4.0. This tool used to design Telecom spacecraft payloads at Airbus Defence and Space. One of the outcome of this design tool is a preliminary and optimized accommodation of payload equipment, including their connections (waveguides and RF coaxes), on the mechanical structure. So there is needs to exchange data with the mechanical design office who is using Dassault 3DX platform. For exchanging data between payload accommodation and mechanical design office we have defined digital solutions ensuring data continuity of all the relevant information such as:

- Simplified geometry of equipment starting with different level of details staring from the bounding box.
- Position of the equipment
- Parametric geometries: for mechanical structure elements like panel profiles with cutouts and for instance equipment that do not exist yet.
- Points of interest on the equipment (e.g. connector positions)

A 3D viewer is available in RangeDB to display all these information. As an example the next snaphots present the use of parametric geometries and simplified geometries. By mixing these two geometries a complete 3D view of the system is available for a better understanding of the content of the engineering database. For instance one can query the database to look for an equipment and it is then displayed in the final assembly. Furthermore one can then ask where is this connector and a zoom is applied to display it.

Fig. 2. Parametric and simplified geometries

Data continuity is interesting for engineering activity but is also very beneficial for manufacturing. On Odyssey4.0 we create a link between these two worlds so that electronic procedures for mechanical assembly are connected with the design world for aspects not limited to mechanical design. This applies to both the preparation of those procedures and their execution. For instance during preparation the manufacturing engineer accesses to the available telemetries of the system and insert a check of telemetry into the procedure. This information is provided by the engineering database and is also checked when needed to verify the consistency of the procedure if the design is changing. Similarly during execution of the procedure the right telemetry is acquired on the checkout system and the value to check is acquired on the engineering base according.

Functional Verification Manager (FVM)

FVM [4] tool is an extension of RangeDB framework to deal with functional verification activities. As an input it receives requirements. It allows the definition of verification requirements and test specifications. The test specifications are then linked with actual test procedures so that each time such a procedure is executed the status of the verification of the system is updated (e.g. to generate a Verification Control Document). In the frame of Odyssey 4.0 the use of this tool has significantly been enriched. It is not limited anymore to functional aspects and covers the Overall Test Matrix where the various test phases are described. Furthermore there is a feature to customize automatically the test specifications and its applicability across test phases according to both the design variants and the customer requirements.

Simulation automatic configuration

The automatic configuration of the functional simulator based on SimTG [3] the Airbus Space System simulation framework and RangeDB engineering database is now achieved. This configuration has been extended to the parametrization of the thermal model used during thermal analysis activities and reused to deliver the operation simulator. This is also used to improve correlation of the thermal model with thermal results during the test phases.

Workflow management

Data exchange between platforms needs to be supported by notification mechanisms. It allows warning people something has changed and they need to act. For that purpose we have chosen JIRA product. It is already deployed everywhere in the company for other needs, thus many people already know this tool. Moreover it can be easily connected to all our platforms thanks to its interfaces available independently of the technologies used to develop the other tools. As an example, when a document is updated by an equipment provider, it goes into our Product Data Management tool and a JIRA ticket is created to notify the relevant people they have to take this into account to update the related engineering data and to check the impacts. Once the update is done, that is a commit in the database, the ticket status is updated to notify other people about the completion of this activity.

LESSONS LEARNT

Naming rules
Today data exchange between people often relies on so-called naming rules. This is convenient for human who can read these names but error prone. For instance people can wrongly understand the naming rules. Or even worse when design changes naming rules have also to change to cover new features and this prevents from reusing data. On Odyssey 4.0 the very first rule we have defined is that there is no more information in names of the aspects we deal with that software will process. In other words, names of the things are only used by humans and never by computers. Computers process data in a reliable manner thanks to the use of the Conceptual Data Model and the object oriented design of the database where there is no needs of those names to play the role of unique identifiers.

Data sheets
Unfortunately we still need to generate some Excel files to feed some very specific engineering tools. People like Excel sheets, this is really the "swiss army knife" of the engineers because it is very flexible. So during the current transition phase we still have to support it. Thanks to some improvements in terms of MMI that often consists in providing tabular views with customized columns, sorting, filtering and massive data population features we help our users to replace Excel with the engineering data base where they will get in addition collaboration means with other teams thanks to configuration features. Something even more important is the ability to check data consistency inside a given engineering activity but also with input and output of other related activities. This is again provided by the engineering database and it increases a lot the quality of used data.

Data modelling
Of course we cannot minimize the fact that having a common Conceptual Data Model (CDM) for all our engineering activity is a complex task. It is not because modelling as such is difficult. Some basic rules shall be applied like:

- Take into account the cause and the consequences. The source of the information, the cause, will not be changed while the consequences will evolve. As a result the CDM gives precedence to the link from consequences towards the source. Several containers of information will depend on a container of information that is providing the source. For instance a functional interface between two equipment is implemented using some electrical signals. Those signals will refer to the functional interface. Similarly the functional interface will refer to some requirements. This of course does not prevent from following the link using the opposite direction (source towards consequences) but the way to store and maintain under configuration control is using the "normal" direction (consequence towards source).
- Concepts shall be clearly identified using as much as possible the business vocabulary. There are many cases where people are using different names for same things. Either the name used by the data provider has to be agreed by the consumers or a new name is to be invented to work around this issue.
- Everything cannot be defined in one CDM even if the CDM design itself has to be modular. We shall provide flexibility to the users to define its own data. In addition to the CDM, RangeDB users can define a so called user data model for things not yet standardized (between projects) or things that have no chance to be standardized like specific performance of equipment. This user data model has to be considered as user data managed exactly like other data in terms of configuration control. As a result it is a very flexible solution and it increases the user buy in. Initially this user data model was applicable to parts of the system like equipment or sub systems. It is now applicable to any aspect belonging to these parts. For instance equipment contains connectors and connectors contain contacts. The CDM describes the relationship between equipment and connector and contact aspects (contact belongs to connector and connector belongs to equipment. In addition to that the user data model allows to define properties that are assigned to connectors of a given type and contact of a given type. It is important to note that the properties that are assigned by the user to a given aspect will vary according to its type ad those choices are all made by the user according to libraries of connector and contact types. At project level it is a standardization that is applied and that can be reused by next projects.

Data breakdown
RangeDB provides, as a core feature, the ability to structure your data, complying with the data model described above, into as many containers as needed. Here again having such feature is not enough because people may use it in different manners for similar things and as a result it makes the data integration difficult. So first we have defined generic rules for this breakdown of data into containers to comply with the project objectives. We store in different containers data for:

- each equipment of the catalog
- various spacecraft product configuration variants or reference architectures
- data of an actual spacecraft project.

By doing so we provide all the means for reuse of data for the next projects: as a minimum the equipment's catalog is reused, most of the time the reuse of predefined variants of a product line (or reference architectures) is promoted and in some cases the complete reuse of a spacecraft design (e.g. for constellations).

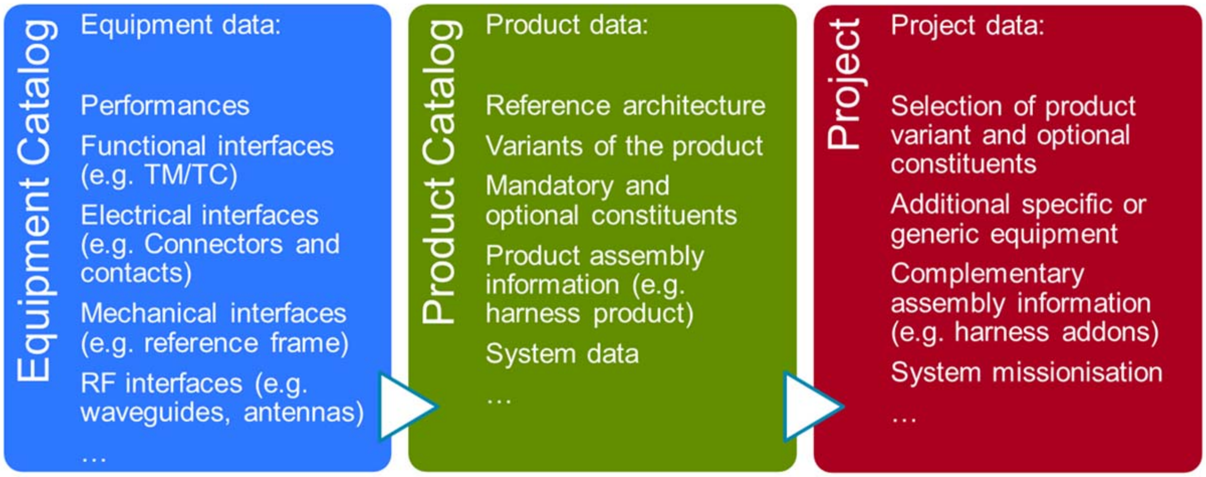

Fig. 2. Main families for containers of information

Data breakdown, also called data structure, is more complex since boundaries of those data containers can vary according to domain point of view (electrical, mechanical, functional, …) Some choices are then needed and often the breakdown has to follow the most tangible boundary of an equipment and its subparts. While from a functional point of view the boundary is defined on a case by case basis, the physical boundary is very tangible and is often the best choice to define the perimeter of the data container of an equipment. Of course this approach only fits with existing equipment and is not applicable to preliminary design phases where the physical allocation of functions is unknown.

Extended Element Definition

Element Definition, from ECSS ETM10-23, used as a starting point to define RangeDB CDM, is the container of information to store the definition of an equipment. That's typically the place where are defined the properties of an equipment like the mass and the various aspects of the equipment with regard to each engineering domain like for instance the monitoring and control parameters, the connectors and the signals. The problem is that this equipment is sometimes made of subparts that are of interest and instantiated several times in the equipment. As an example one can consider a fuse box. It contains several fuses and the user wants to characterize each of them. The solution is not to use an Element Configuration (ETM10-23) because this one is used to configure the purchased equipment (in that case the equipment integrator is not able to configure the inside of the equipment he has purchased). As explained in the next diagram our proposed solution is to use instance of definitions. Like for a configuration it allows to define how many instances are used but this time at definition level so that when the equipment will be configured all the inside will be properly instantiated as well.

In addition to that, another extension of the element definition, is the so called Variant Definition. It allows defining equipment definition as a variant of another equipment. The variant must be compatible, in terms of functional interface, but can be different in terms of physical or electrical interfaces for instance. This concept of variant can also be used to define so-called average definition of equipment, a kind of specification that a variant of it will fulfil with an actual design. This is used for instance in bid phases where typically the equipment have not been selected yet.

Both concepts, instances and variants, can be mixed as depicted here under. Adequate Man Machine Interfaces (MMI) are used to support those concepts and the related data population.

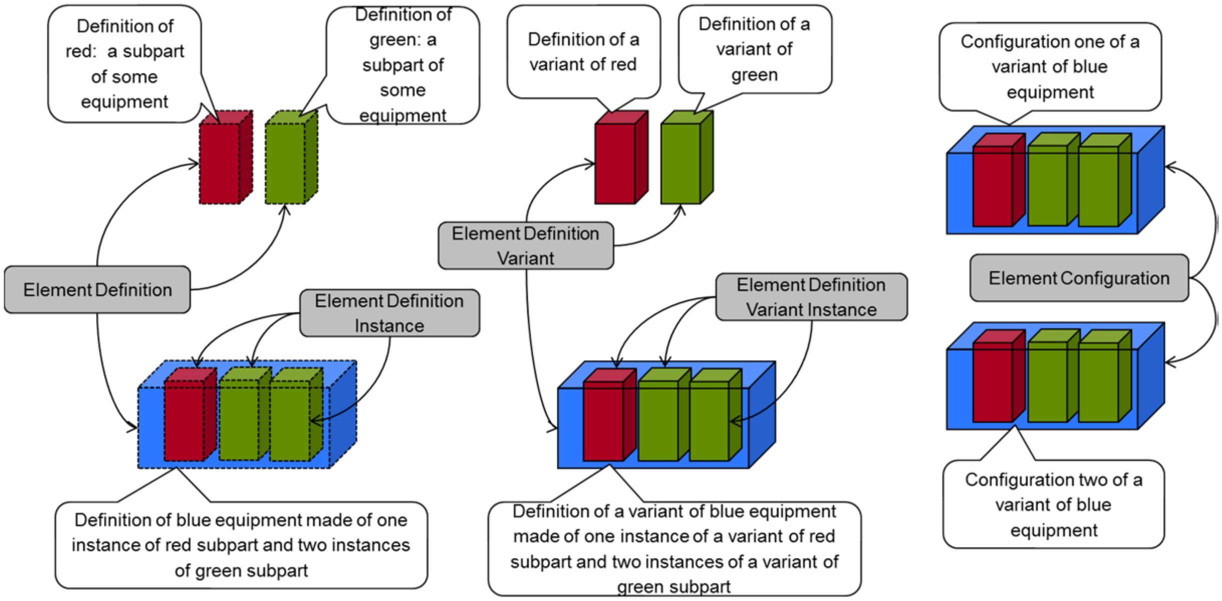

Fig. 3. Definitions, instances, variants and configurations

User view

As described in the previous chapters, the data structure gives priority to the provider of the information as there is one source of information, the provider, but several consumers of the information. However, from system end user point of view there is no need to know all those details. For instance let's imagine a system made of an on board computer, the related flight software, a generic purpose acquisition unit (the one that transform analog signals into numerical values on a data handling bus) and some equipment connected to this unit. The end user does not care about the detailed design of the spacecraft or the way the data is structured in the database (i.e. according to this design). The end user wants to acquire a measure (or send a command) on the equipment connected to this generic purpose unit but without knowing the path to be used for this acquisition (or this command). In other worlds the end user wants to "talk" directly to this equipment but unfortunately the data available at equipment level are not sufficient for that (the equipment does not know how it has been integrated in a given system configuration). To fix this issue a new element has been defined in the CDM to allow users to restructure data according to their needs. This data restructuring is made automatically consistent with the other sources by avoiding any duplication of the information. One can consider it is a view of some data/aspects spread in different containers (following the logic of the provider of the information) all stored in a dedicated end user container to be convenient for the end user i.e. the consumer. See an example of it in the next picture.

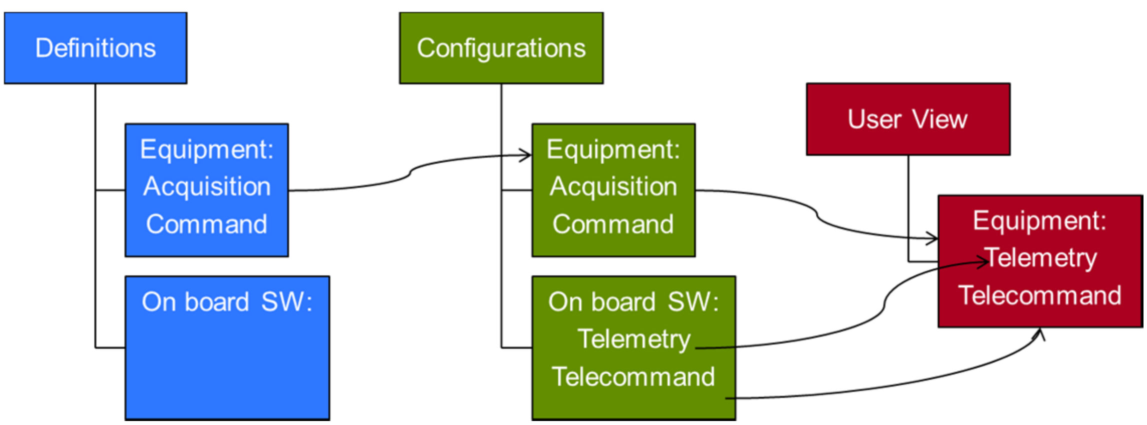

Fig. 4. User View

Collaborative platforms and deployment

As described in the quick presentation of Odyssey4.0 digitalization project, the approach is to perform data exchange between pillars of engineering and manufacturing activities. The so called “collaborative platforms” may look like to be very easy to integrate. This is often not at all the case for various reasons:

- Sometime the platforms are collaborative internally but not at all with rest of the world. Different tool vendors strategy exists. Some are providing data exchange interfaces for instance with XML schema while others have difficulties to describe APIs they may provide. This can be even worse when the openness of the platform is reduced for the new versions.
- According to deployment strategy of the platform there is a risk of lack of data exchange features at network level or for security reasons. This may also compromise the data exchange as such or significantly reduce its performance.

More generally, deployment of an engineering database used by all the disciplinary along the complete life cycle of the system is a sensible topic. There is a significant increase of regulation rules on data access and this creates many security constraints that are to be managed with care.

Finally adequate training of people to use such a generic purpose solution efficiently on their day to day work is a key aspect. In response to the growth of RangeDB training courses in UK, Germany and France we have now starting the development of eLearning modules. The first ones have been deployed with very good feedbacks.

NEXT STEPS

Design to cost

The next update of RangeDB data model will be related to cost management. Some prototype already exists and the mechanisms to ensure the restricted access to this level of information are in place. This is a very promising improvement to support design to cost.

Data analytics

Data continuity is in place for engineering activities. However the connection between test data and live data with engineering data is not completely achieved. This is a great area of improvement to take benefit of all these data and improve the design or for example to automatize some complex testing phases.

RangeDB NG

The number of users but even more the significant increase of the scope of the data to be managed is progressively showing some problems of performance on the database. This is for the time being manageable but we can easily foresee that the increase of the scope will continue. To solve these issues new technologies will have to be selected to deliver a new version of RangeDB framework. This will be the perfect timing to take into account data analytics needs and to solve the limitations described here above. The development will start this year and the update of the tool will be as much as possible transparent to the end users.

CONCLUSION

Thanks to the data continuity that is in place we are now ready to take efficiently benefit of data analytics. Indeed data analytics with its unlimited data storage capacity is a very promising approach. However big data systems often suffer from the lack of link between the data values stored in data lakes and what is the definition of these data values. Because of technology changes the link was broken. Tool vendors already propose solutions to recover this but it is again a new semantic data layer while all the semantics already exist. Here again we intend to keep data continuity without any duplication of information so that data analytics can be applied not on a case by case basis but to the widest scope very efficiently.

A lot of software development, in close collaboration with end users, have already been achieved. It is now time to start using this digitalization approach on the first spacecraft. This will for sure bring additional lessons learnt and new ideas will come to enhance again our processes keeping always in mind that the main problems to be solved are not really technical, or at least not at the level of software programming. It is rather the complexity of the understanding of the needs and the ability to provide solutions that fits with this understanding. This, as other engineering domains activities requires a close collaboration with the various disciplines.

References

[1] Claude Cazenave, Luc Fancelli, Harald Eisenmann, "Benefiting of Digitalization for Spacecraft Engineering", SESP 2017

[2] Harald Eisenmann, Claude Cazenave, Thierry Noblet, "RangeDB the product to meet the challenges of nowadays System Database", SESP 2015

[3] Claude Cazenave, William Arrouy, "Implementing SMP2 Standard within SimTG simulation infrastructure", SESP 2012

[4] Jean-Francois Vasse, "The functional Verification Manager to support end to end the satellite validation process", SESP 2017