MBSE Applied to an Aerospace 'Force Fighting' Application

PLM Systems for Platform Based Systems Engineering

Bruno VUILLEMIN[1], Nicolas CROUE[2], Sandrine LOEMBE[3]

1: ALTRAN Sud-Ouest, Bat Synapse, 4 avenue Didier Daurat, 31700 Blagnac, France
2: KEONYS, 5 Avenue de l'Escadrille Normandie Niemen, 31700 Blagnac, France
3: DASSAULT SYSTEMES, 10 rue Marcel Dassault, 78140 Velizy-Villacoublay, France

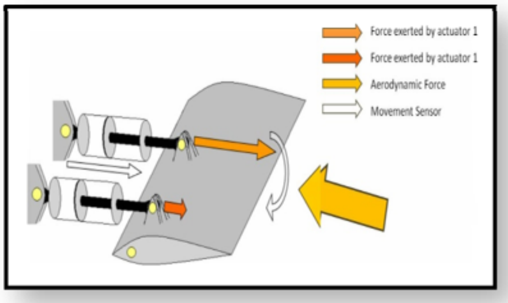
Abstract: The aerospace industry is a domain where system engineering, and in particular Model Based System Engineering (MBSE), has been adopted extensively. Based on a feedback of the on-going development of a technical use case, this paper provides a high level view of MBSE state-of-the-art and current state-of-practice as provided through the Dassault Systèmes (DS) Product Lifecycle Management (PLM) solutions.

Keywords: System Engineering, MBSE, RFLP, Modelica, Physical Modelling, Force Fighting, Dymola

1 – Introduction

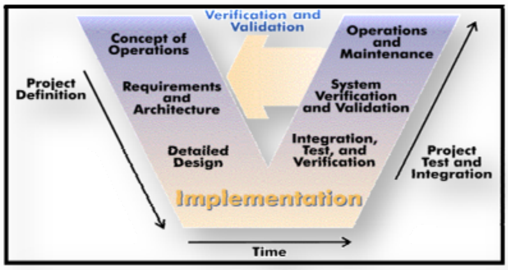
The scope of the problem studied in this paper has several layers :

- Firstly, a **concrete technical problem** concerning the force fighting consequences when two actuators are trying to move the same control surface with asynchronous or antagonist forces.



- Secondly, a **methodological problem** due to a lack of continuity in traditional system engineering approach based firstly on requirements (Requirement Based Engineering (RBE)), then on models (Model Based System Engineering), finally based 3D Digital Mockup (DMU). Those RBE, MBSE and DMU steps are often driven by separated processes, supported by parallel methods and embedded into heterogeneous tools.

  The objective is to follow steps through the V-cycle of the development of a system (from initial need … to the optimized solution).



- Finally, the performance calculation of actuation systems must take into account the flexibility of the structure. In the current process, the two environments are clearly separated; here we show a link in a **unique environment** to track all the simulations and the way to have a coupled one.

Achieving relevant capability, on demand, requires the use of integrated systems to enable synchronous, cross-functional design activities within an adaptable platform. Such an adaptable platform is one that is governed by an integrated architecture that minimizes risk, while incorporating Platform Based Engineering (PBE) and MBSE methodologies and tools.

The DS PLM solutions meet these requirements with a single platform and unified data model: conceptual design tools yield models that are elaborated to full product models, supporting direct modification and upgrades, which are used for manufacturing, as well as training and technical publications.
In this paper, we present an example of designing part of an airplanes flight control system application, specifically, the study of the 'Force Fighting' phenomena that can occur in its flap control systems. This project has been developed with the DS PLM Systems platform which will also be described. Results and perspectives of future developments will finally be exposed.

2 - Technical scope of the study

Force fighting occurs when at least two actuators are in the active mode (double pressurization), but generating forces in opposite directions to one another. This means that the two simultaneously active actuators would try to drive control the surface to different positions, if they were individually active.

Force fighting is caused by many different phenomena, like:

- Positional errors due to components,
- Adjustment tolerances,
- Delays and asynchronies errors between Electronic Control Units (ECUs),
- Signal conversion errors (e.g. Analogue to Digital (A/D), and Digital to Analogue (D/A)) conversion errors,
- Delay of the signal to activate double pressurization, etc.

Force fighting can generate very high force peaks. The definition of maximum allowable values of force fighting for elevators and rudders is 60% of the maximum load capacity and for ailerons it is 100%.
The objectives of the study were to assess the effect of all command and control phenomena generating force fighting on the structure and in particular to assess transversal efforts, structure deformation / breaking limit, impact of force fighting on the actuator dynamics.

2 - Methodological RFLP approach

2.1 - Presentation of the PLM Systems Environment

The DS PLM Solution has extensive support for Model Based Systems Engineering. It has a number of domain specific, yet fully integrated modellers that encompass; Functional Architecture, 2D & 3D Logical architecture, MCAD, CAE, FEA, Kinematics, Behavioral and control system Modelling & Simulation (based on the open Modelica language), real-time simulation for Hardware in the Loop purposes for instance, systems safety modelling (based on the AltaRica language) and analysis with FMEA, and a rich systems simulation environment.

The PLM Systems environment is based on RFLP concepts. This approach provides a collaborative definition of a product across its different views from Requirements, Functional (targeted services), Logical (technology), and Physical (implementation) design. From the requirements captured in the environment, functional and logical definitions were derived. Attached to the logical components, Modelica models were developed with different levels of details in order to run simulations focusing on different activities.

For this last part, Modelica has been used to describe the plant since it is an open acausal language that allows a high fidelity level of modelling and simulation for multi physical systems. The Modelica standard library provides components in many domains like hydraulics, mechanics, fluid, thermal or control design. In order to ensure openness and global simulations involving other tools, technologies and embedded software, the DS simulations environments are compliant to Functional Mockup Interface (FMI) open standard developed within the European Modelisar project.

2.2 - RFLP approach

The RFLP approach allows the definition of Requirements, Functions, Logical architecture and Physical description.
The four layers of the RFLP approach are illustrated below with the example of actuators

and surface. In order to ensure traceability links, the objects of four layers are linked together through relationships.

From NEED ...

Requirement

Textual description

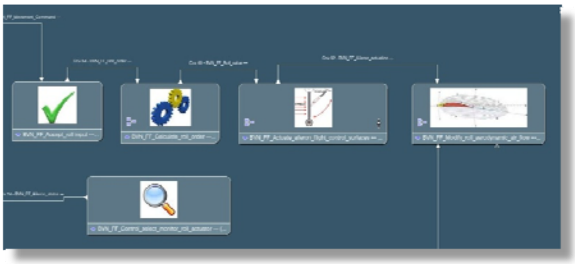
V-SRD ACT_33202
The Actuator shall not be in contact with other structural elements.

Rationale: To protect equipment and structure
Additional Information:
Rbe Status: Analysed

Functional

What the system does (functionality)

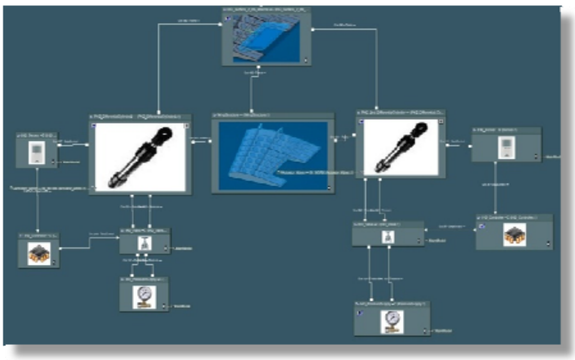
At this level, we defined the different missions of the system, the I/O interfaces of the functions, the functional breakdown, how they are connected and also how they are sequenced.



Logical

What the system is (component, interfaces, behaviour)

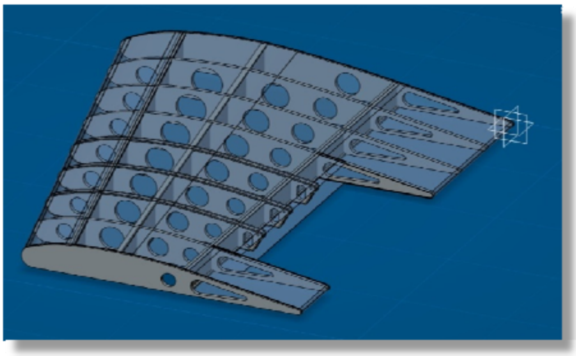
The logical architecture is defined using 2D diagrams (as shown below) where blocks represent 'real' components. To those components we may associate a 3D representation (more or less detailed to make space reservation analysis) and dynamic behaviour based on Modelica language for instance.



... to SOLUTION

Physical

A solution of the system



<u>Configuration management:</u> depending to the development stage, the elements of four layers can have several descriptions which are managed on configuration. These different views of the models allow the users to make simulation on more and more accurate models and save all the history.

<u>Test cases:</u> Requirement and test case attached in the same environment allows teams to make a report to check if the entire requirements are satisfied and who make the validation. Due to life cycle management, they can track who, when and with what the test case has been validated.

<u>3 – Collaboration between teams</u>

This project intends to demonstrate new methodologies in the way involved teams work together.
Nowadays, the increasing complexity of the products requires involving multi engineering domains like physical domains (mechanical, hydraulics,…) and also embedded control systems. Several companies are in charge of the development of subsystems that will finally interoperate in the final product. These aspects require considering with high attention the collaboration between actors on a same project.
The first level of collaboration is about data management and lifecycle management. The goal here is to ensure that all the data, models, results and analysis produced and used by the project members are stored following well

defined processes and can be retrieved by authorized actors of the project.
Moreover, it is frequent that these data have been created in multiple tools and so the product lifecycle environment should be able to handle these multiple sources of information.
Another level of collaboration is about the composition of 'complete' models from validated components. With the growing complexity of systems, global simulations to validate specific requirements are required. The goal is not to promote a unique high fidelity model but the ability to compose assembly of models coming from validated libraries defined by domain expert teams. A common logical architecture is used for which such or such implementation of components is chosen.
If a common standard language like Modelica is used for the modelling of the components, model reduction and symbolic manipulation algorithms are automatically applied to the Modelica part in order to get more efficient simulation in terms of time execution.
In case of several tools, cosimulation helps to run global simulations. FMI standard delivered from the Modelisar project defines API and routines for model exchange and cosimulation capabilities. As an example, from a Simulink model representing the control part, we generated a FMU (Functional Mockup Unit) integrated in the global Modelica model.
To combine different types of models (3D, FE, 0D, 1D), other techniques have been used.
To integrate deformation of the aileron in the closed loop simulation, a simplified model of the aileron has been generated in a SID format and reintroduce in the Modelica simulation thanks to Flexible Bodies library.

4 – Main added value of the approach

Architectures and simulation results will be presented in this part. During this study, the following points were highlighted.

4.1 - A unique database

The DS' MBSE approach supports Platform Based Engineering (PBE) through common reusable architectures and a single point of access to information from requirements to physical design. Decomposition and definition tools enable a relational data model that encompasses configuration, requirements, logical, physical, manufacturing process and product configuration definition.

4.2 - Traceability along the different layers

The RFLP modelling capability enables the top-down systems engineering approach (i.e. reach the real product starting from specifications) and the bottom-up approach (start from the real product to get the specifications) to mix these two engineering processes in order to establish the most effective concurrent engineering process.

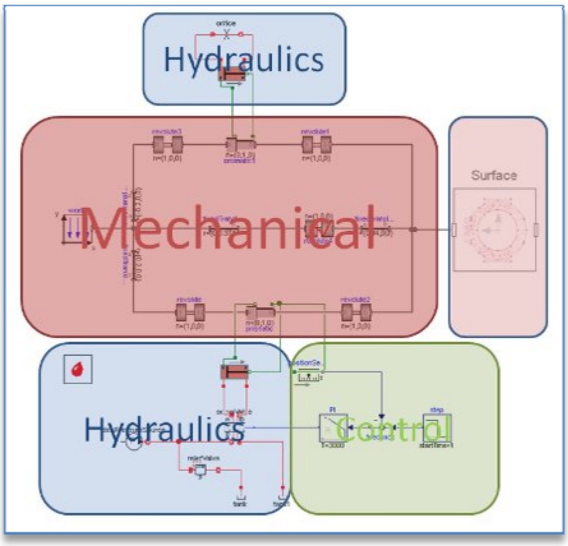
For example, following implementation relationships from Requirements objects, an impact graph is useful to analyze:

- Completeness of allocation (coverage analysis)
- Consequence of changes (impact analysis).

4.3 – Multi-Physics modelling

The Logical architecture is composed of hydraulic sources and valves, actuators attached to wing and surface, position sensors, calculators, ... Linked to each component of the Logical architecture, bricks of Modelica models allow modelling behaviour. For the current use case, standard multi-physics libraries are used for hydraulic and mechanic aspects. A specific library "Flexible Bodies" is used for the modelling of the surface itself.

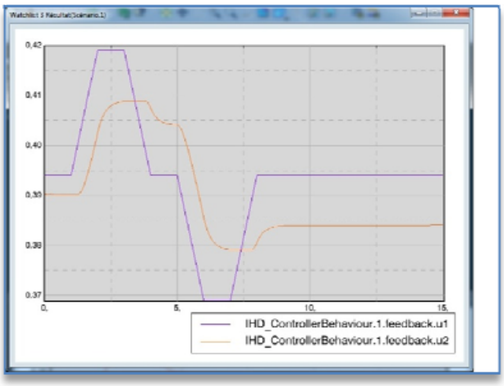
Within the Logical layer, some 3D elements are linked to logical components: actuators components are linked to wing and surface, the surface is linked to wing. The dynamic behaviour of 3D elements is modelled.



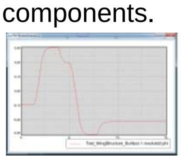
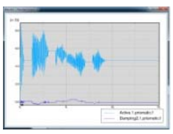
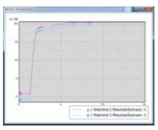
4.4 - Simulations

The multi-physic modelling associated to the components of the Logical layer can be driven by signals for simulation purpose.

Computed values can be displayed into plot screens. For example, the Calculator component requests movements of the surface, then hydraulics components are activated in order to inject hydraulics power into the actuator, and then the surface rotates according to the position of the both parts of actuators.

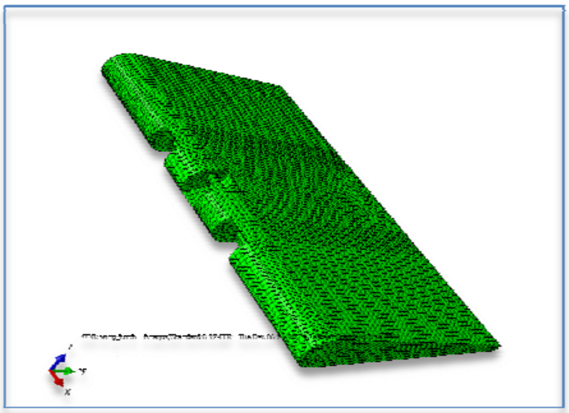


Several kinds of values can be displayed (control, pressure, position, angle, ...) to validate the model and the behaviour of the components.

4.5 - Flexible Bodies to evaluate deformations of aileron

In a first stage, a flexible beam from the Flexible Bodies library of Modelica is placed between the two actuators.
Linked to the Flexible Bodies Modelica library used in the model, the constraints on the aileron are produced during simulations. After that, to have more detailed results, we try to replace the flexible beam by a Finite Element Model (FEM) provided by Abaqus. The goal is to see detailed constraints which are analyzed through an Abaqus model.



4.6 - Validation of models

The main added value of a system engineering approach based on models is to support the Validation and Verification activities as soon as possible during the development cycle.

Using the transverse and common system referential, several levels of validation of the model are possible.

1. Requirement validation
   Based on descriptive of behaviour modelling, this approach allows identifying unclear wordings, inconsistencies and discrepancies into specifications documents produced from the common system referential. A close link between textual requirements and the several levels of models (functional, logical architecture, multi-physic, 3D) allows validation of requirements.

2. Design verification
   In the development cycle, the design of a system shall be in line with the requirements defining the system. Design verification shall answer to the question "Does design satisfy requirements ?"

3. System verification
   System verification shall answer to the question "Does system satisfy requirements ?"

Several validation means are used :

1. Coherency checks :
   Using the knowledge ware, some predefined rules are used to check traceability between four RFLP layers and coherency into the common database.
   For example : the coverage of the allocation of Requirements to Functions or components of the Logical architecture, the right allocation of inputs/outputs between several decomposition levels of the Functional and Logical layers.

2. Simulation checks :
   The analysis of curves produced during the simulations allows the validation of the Logical architecture and associated Modelica behaviour.
   For example : check if the expected rotation speed of the surface is satisfied by the Logical architecture and the associated behaviour modelled with Modelica.

3. 3D checks :
   The simulations allow the analysis of the dynamic behaviour of 3D elements.
   For example : check conflict between elements of the Physical layer (wing, actuator, surface).

5 – Perspectives

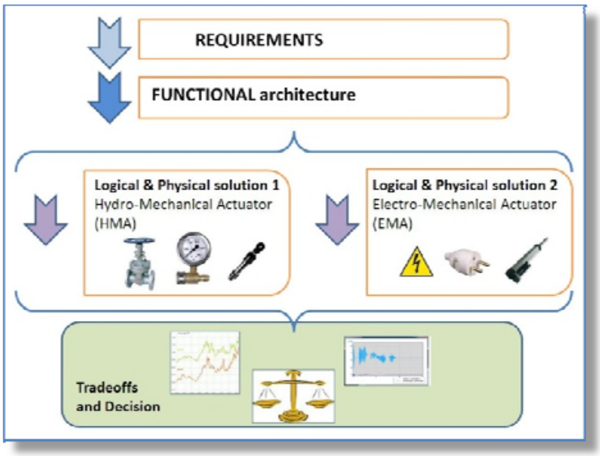
Several kinds of perspectives are identified to enhance the current methodological approach applied on the use case of "Force Fighting".

5.1 - Tradeoffs of design solutions to support decisions

In order to support a decision making approach based on tradeoffs of several design solutions, it's possible to follow a common branch for need description (mainly on Requirement and Functional layers) and parallel branches for potential solutions (mainly on Logical and Physical layers).

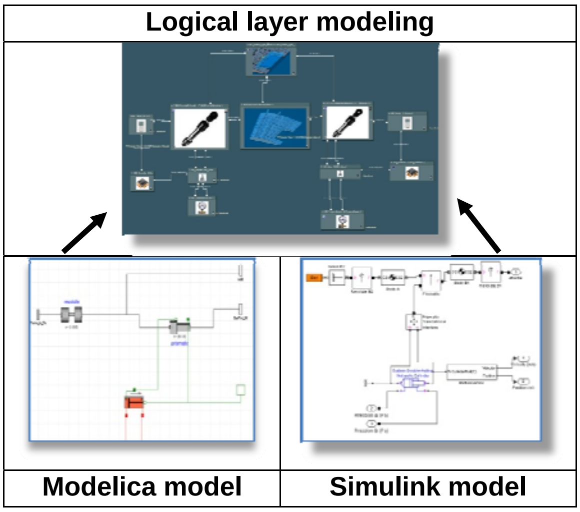
Requirement layer	
Functional layer	
Logical solution 1	Logical solution 2
Physical solution 1	Physical solution 2
Define selection criteria (performance, weight, cost, ...)	
Simulations and model checking	
Decision to select the most appropriate solution for the initial need	

The current version of the use case was modelled with a Hydro-Mechanical Actuator (HMA). The alternate solution is to use an Electro-Mechanical Actuator (EMA).



5.2 - Simulation of behaviour from multi source models

Promoted by the Modelisar project, the objective is to be able to run simulation of behaviour from multi source models. The FMI (Functional Mock-up Interface) defines an open interface to be implemented by an executable called FMU (Functional Mock-up Unit). This is the next generation of methods, standards and tools to support collaborative design, simulation and test of systems and embedded software.
For the current use case, the objective is to run simulations with several bricks of models developed with Simulink and Modelica, associated to different components of the same Logical layer.



5.3 - Other domains of investigation

Still domains to investigate like:

- **Space reservation** defining rules to avoid contact between parts or to respect segregations distances
- **Collaborative work** between several stakeholders of the model, involved in the several RFLP layers of the model
- **Model exploitation and checking** using knowledge ware rules

6 - Conclusion and way forward

The Model Based System Engineering RFLP approach embedded into the DS PLM solutions was deployed to develop the model of the use case described in this paper. The Proof of Concept is reached and main information filled the four RFLP layers.

Some additional investigations will be useful:

- to improve the continuous chain of models,
- to embed heterogeneous bricks of models,
- to study tradeoffs of several solutions,
- to support safety analysis for dysfunctional aspects
- ...

Then, the scope can be enlarged to reach a more realistic coverage for an industrial project.