# Model Based Approach for Functional Chain Engineering

Author: Alexandre Cortier[1]
alexandre.cortier@airbus.com

Co-authors: Philippe Leblond[1], Jean-Luc Marty[1], Jacques Magné[2]

philippe.leblond@airbus.com, jean-luc.marty@airbus.com, jacques.magne@airbus.com

*[1] Airbus Defence and Space SAS, 31 rue des Cosmonautes 31402 Toulouse, France*

*[2] Airbus Defence and Space SAS, 51 Route de Verneuil 78130 Les Mureaux*

***Abstract***

*O2 - Limitations of current MBSE approaches and ways to circumvent or resolve these (e.g. through customization of processes and tools)*

*This paper presents a new model based approach for functional chain engineering inherited from ATV and ESM-Orion's functional method. This approach has been implemented as an extension of the MOFLT (Mission – Operation – Functional – Logical – Technical) Airbus framework with the aim to model the different functional chains of a system at a level of granularity matching with a Design Justification File (DJF). One DJF is then generated per functional chain and can be used for internal and external reviews as well as inputs for Software Specification. The tooling supporting the Functional Chain modeling approach is currently used for the ESA PILOT project and is expected to be used in the scope of ESA CLTV project (Cis-Lunar Transfer Vehicle).*

**Index Terms:** MOFTL, Functional Chain, Design Justification File

## I. INTRODUCTION & OVERVIEW

The ATV functional method was a paper-based approach where each functional chain (FC) of the system was either a Functional Unit (FU) or a Software Unit (SU). A functional chain was described in an independent DJF (*Design Justification File*) document. As functional chains interact, this approach led to big efforts in maintaining the consistency of the overall set of documents including naming and typing of interfaces. In the scope of CLTV project, it has been decided to apply the ATV functional method but using a model-based approach to overcome this maintainability issue, improve the co-engineering between systems, functional and SW architects, and to improve internal and external review processes of the functional design.

This paper proposes an overview of this functional engineering approach that extends the Airbus MOFLT © framework (Cameo System Modeler® customization through profile extension). This extension allows to describe the functional architecture at a lower level of details.

The Figure 1 shows the expected usage of the Functional Chain models. Starting from Mission CONOPS, MSRD (Mission & System Requirement Document), MSOR (*Mission System Operational Reference*) and the system specification, the model will capture the operational and functional modes, the main processes, functional blocks and commanding sequences of the functional architecture. Note that the main objective is here to describe the Software parts of the functional chain. The SW Functions (Functional processes and blocks) are deployed onto a technical architecture bypassing the logical layer.

**Model Organization**. The global model is composed of several sub-models. The system model is defined following the standard Airbus MOFLT© method. Then for each functional chain of the system, a sub-model is defined as shown in Figure 1. One architect is responsible of one functional chain model. This organization allows a clear separation where each architect can work on a reduced MOFLT© model template: each functional chain (FC) model addresses only the operational layer, the functional and the technical one.

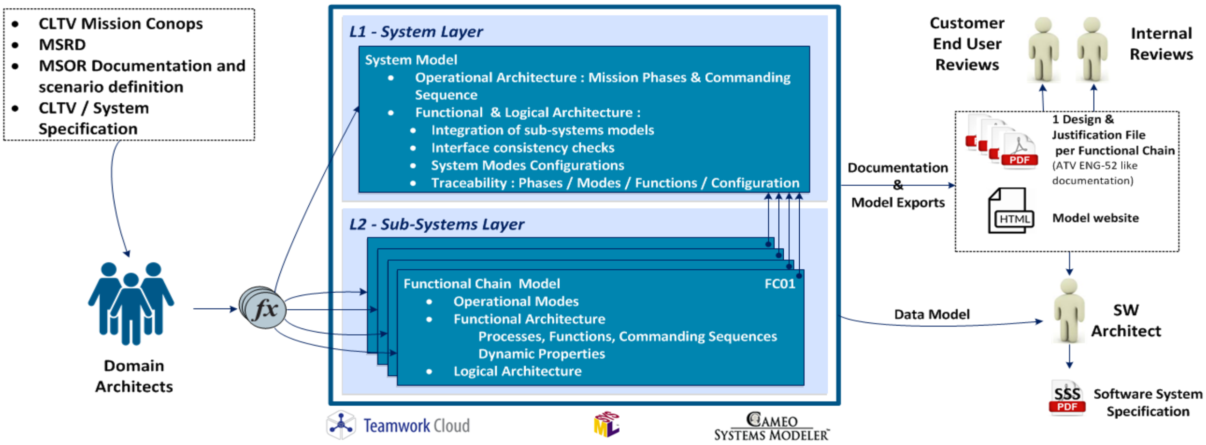


Figure 1 : Functional Chain Engineering Approach

**Design Justification File**. From each functional chain sub-model, a DJF file is automatically generated. This document can be generated either in HTML, Word or PDF format using the Airbus MBSELab add-on and starting from the same description of the document that is encoded as a model. The DJF (or directly the functional chain model) can then be used for internal and external reviews and is used as input for SW architects to define the Software System Specification (SSS).

## II. FUNCTIONAL CHAIN MODELS : CONCEPTS & FEATURES

**Operational Phases**. In the operational layer, the FC architect defines the operational phases with a specific focus on the functional chain he managed. These operational phases, formalized as *SysML::State*, can be linked to the description of the mission and/or operational concept defined at System Level to ensure an end-to-end traceability.

**Functional Modes**. Then the functional modes of the FC are defined as states of a state machine. Modes are traced upward to the life profile (operational phases) and downward to the SW functions.

**SW Functions: Functional Processes & Blocks**. Two kinds of SW functions are identified: the processes and the functional blocks. These two kinds of functions are represented in the model with the same concept of *Function Behavior* identified in Airbus MOFLT© method (*SysML::Activity*). While processes are composite functions invocating sub-functions, the functional blocks are leaf behaviors. In addition to the definition of the input and output flows, the proposed extension allows: (1) to specify in more details the expected dynamic of the processes; (2) to specify the engineering unit as well as the encoding format of the inputs and outputs flows (i.e. functions parameters); (3) to specify the expected dynamic of flows reception and emission;

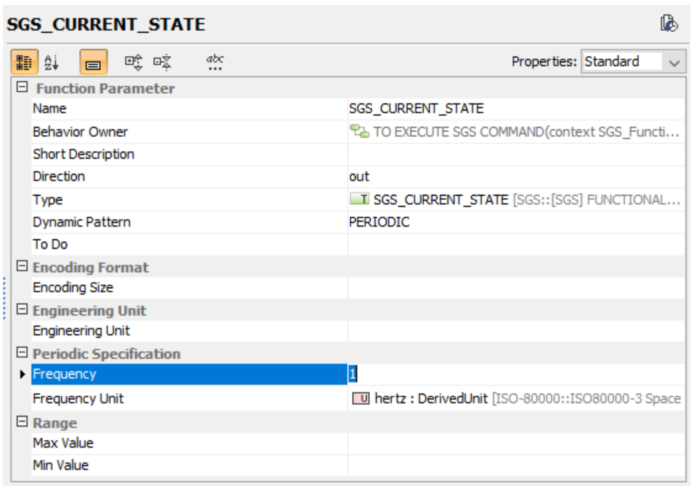

Figure 2 : Specification view of function's parameters

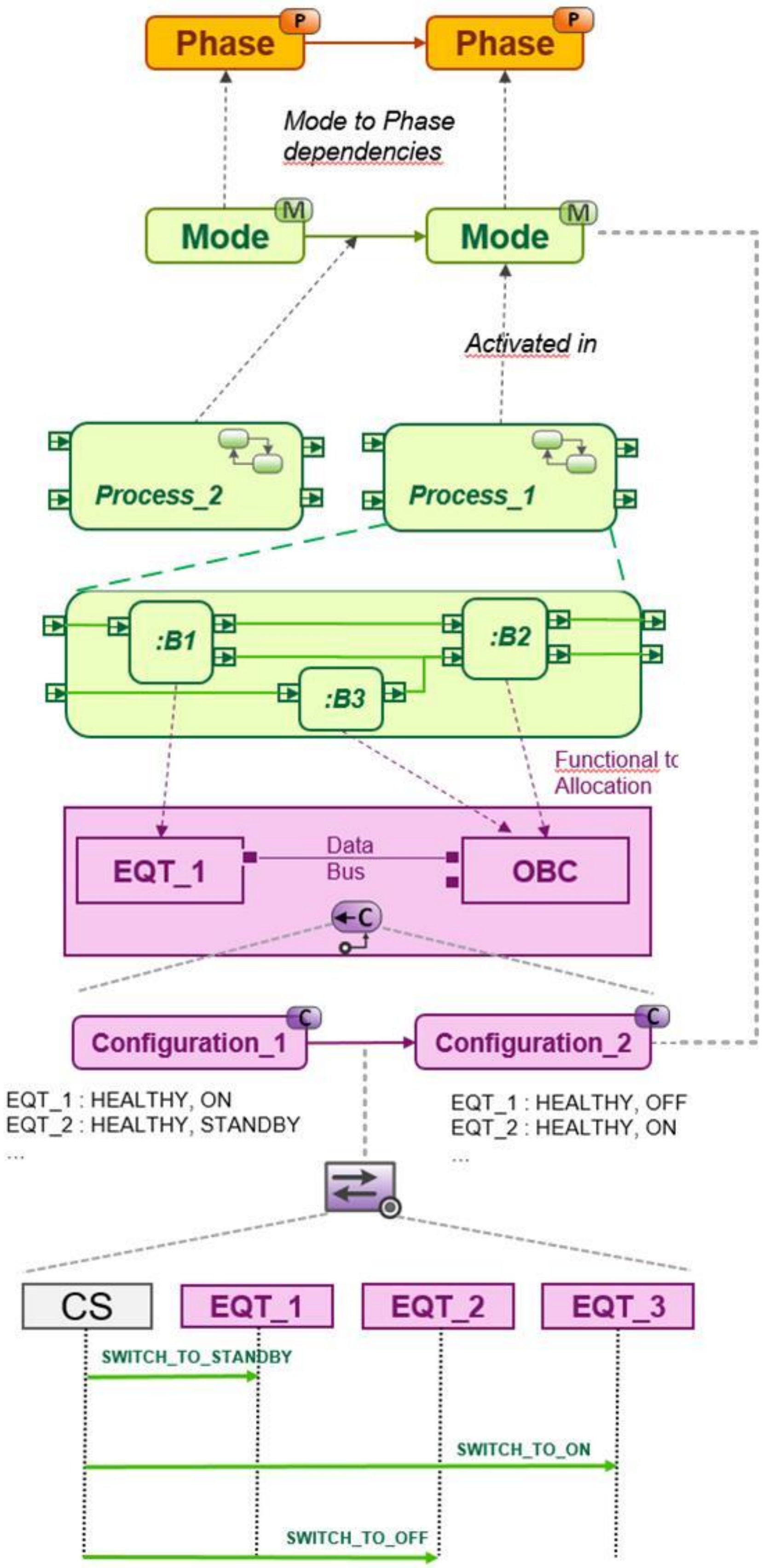

Figure 3 : Concepts of Functional Chain Engineering Approach

**Dynamic Specification**. The way to specify the dynamic of the processes and function parameters relies on the same approach. The architect can specify in which mode the element is activated and the expected dynamic pattern (CYCLIC or ASYNCHRONOUS) of the element. For cyclic pattern he has to specify the frequency of activation and for asynchronous one he can define a command or an event triggering the function execution (see Figure 4).

| Id | Label | Name | Activation Scheme | Is Leaf Behavior | Dynamic Pattern | Activable By Mode |
|---|---|---|---|---|---|---|
| 1 | SGS_B1 | TO EXECUTE SGS COMMAND | ASYNC [ON_EVT: SGS_MC_CMD ] | true | ASYNCHRONOUS | SGS_M_DEPLOYMENT<br>SGS_M_FIXED<br>SGS_M_ROTATING<br>SGS_M_STOWED |
| 2 | SGS_B2 | TO MANAGE SOLAR WINGS DEPLOYMENT | CYCLIC [1.0 Hz] | true | PERIODIC | SGS_M_DEPLOYMENT |

Figure 4 : Dynamic Specification of Functional Processes

**Observability Specification**. The Functional Chain extension allows the architect to specify Observability Constraints- on function's parameter. An observability constraint identify: the constrained parameters, the observers of the parameter, the observability category (HK or RECOREDED) and the dynamic pattern of observability (ASYNCHRONOUS or CYCLIC)

| # | Constraint Name | Constrained Element | Observers | Observability TM_ Category | Dynamic Pattern |
|---|---|---|---|---|---|
| 1 | <HK> [{1.0 Hz}] | out SGS_CURRENT_STATE : SGS_CURRENT_STATE | GROUND_Technical_SoI | HK | PERIODIC |

Figure 5 : Observability Specification

**Technical Layer and Allocation**. Each architect is responsible of defining the technical architecture of its functional chain. The objective of this step is to model the avionic architecture of the functional chain by identifying the involved components (i.e. equipment) and their interconnections supporting the exchanges of data between functions. Once the avionic architecture is described, the deployment of functions on component is specified.

**Technical Configurations**. The Functional Chain extension of MOFLT© introduce the new concept of Configuration at technical layer. A technical configuration identifies states of each equipment of the FC. For doing that, generic state machines are defined for each equipment: (1) the power state machine identifying the power status of the equipment (ON/OFF/STANDBY by default); (2) And the Healthy State machine (HEALTHY, FAILED).

Technical configurations are encoded in the SysML model using a configuration state-machine, each state representing a specific and stable configuration of the functional chain. Concrete description of a configuration is managed using a *SysML::InstanceSpecification*. This instance identifies the power and healthy states of each equipment for the considered configuration and is attached to a configuration state.

**Commanding Sequences**. The new concept of commanding sequence identifies the sequence of commands to be executed on-board to switch from one configuration to another. Commanding sequences are modeled as *SysML::SequenceDiagram* and are linked to the transitions of the configuration state machine. Commanding sequences can be seen as a first specification of the Flight Control Procedures (FCP) at functional level.

## III. CONCLUSION

Relying on the proven ATV functional paper-based method, the extension of Airbus MOFLT© for functional chain engineering allows to bridge the gap between System and Software specifications using a model-based approach. This new approach allows to model the different functional chains of the system in a collaborative and concurrent way while ensuring consistency of the functional definitions and interfaces. The preliminary tooling extension is actually used for the ESA PILOT project and ESA CLTV would intend to rely on it for its functional design definition. This extension intends to be improved in the future by providing specific budget analysis based on the specification of dynamic scheme for functions and function's parameter. Coarse grain budget analysis on TM/TC bandwidth and bus load is expected in the next.